000_notes

Michael-Philipp Stiebing

2023-02-22

2022-12-05

- Download source data from **https://divvy-tripdata.s3.amazonaws.com/index.html**
- License for using the data from **https://ride.divvybikes.com/data-license-agreement**
- Create project root folder - **/20221205-capstone_datascience/** - as **../** in this document
- Move raw, zipped data to **../099_original_raw_data/001_raw_compressed/**
- Unzip data into **../099_original_raw_data/002_raw_csv/**
- Download copy of the assignment to **../098_documentation/20221205-DAC8_Case_Study_1.pdf**
- Download copy of the license to **../098_documentation/20221205-Data_License_Agreement_Divvy_Bikes.pdf**
- Download copy of the R Script to **../098_documentation/20221205-Copy_of_Divvy_Exercise_R_Script.txt**
- Create **../001_deliverables/**
- Copy data to **../002_data/001_csv/** as a working copy
- Rename file **202209-divvy-publictripdata.csv** in **../002_data/001_csv/** to **202209-divvy-tripdata.csv** in order to match filename schema
- Download quarterly formatted data from **https://divvy-tripdata.s3.amazonaws.com/index.html** to compare with monthly formatted data
- Copy dataset **Divvy_Trips_2020_Q1.csv** to **../099_original_raw_data/002_raw_csv/** and working copy to **../002_data/001_csv/**
- Comparing data in **Divvy_Trips_2020_Q1.csv** to **202210-divvy-tripdata.csv** as to similarity.
  - Using OpenOffice Calc V 4.1.7 for comparison
  - Column Names match
  - The rideable_type column contains more information in the **202210-divvy-tripdata.csv** as opposed to the **Divvy_Trips_2020_Q1.csv**
    - In **Divvy_Trips_2020_Q1.csv** only one type: dockable_bike
    - In **202210-divvy-tripdata.csv** either classic_bike or electric_bike
  - The columns start_station_id and end_station_id have different values
    - In **Divvy_Trips_2020_Q1.csv** the values are numerical
    - In **202210-divvy-tripdata.csv** some values are numerical, others are a mix of uppercase letters and numbers a few also include a dash '-'

  - The columns start_lat, start_lng, end_lat, end_lng are formatted differently
    * In **Divvy_Trips_2020_Q1.csv** the values are truncated to 4 decimal places
    * In **202210-divvy-tripdata.csv** the values contain up to 10 decimal places
  - Some records in **202210-divvy-tripdata.csv** do not contain start or end station data, leading me to believe it includes data from bikes that were rented not from a station, but standing around in the city.

- Checking **202110-divvy-tripdata.csv**, the oldest of the monthly datasets, to confirm that the differences persist

- I will be using the monthly datasets for the analysis, because:

  - The data is more current
  - It includes additional data as to the type of bike used
  - The data that is missing (station names) or not formatted nicely (gps coordinates) doesn't seem to impact the analysis

2022-12-06

- Creating R Studio Project in **../003_Rstudio_project/20221206-RStudio_Project01/**
- Create R Studio Script **001_Capstone01.R**
- Work through **20221205-Copy_of_Divvy_Exercise_R_Script** and adapt some commands to the new dataset

```
library(tidyverse)
library(lubridate)
library(ggplot2)
library(readr)
```

```
setwd("/home/mikiR/remote_transfer/")
X202110_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202110-divvy-tripdata.csv")
X202111_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202111-divvy-tripdata.csv")
X202112_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202112-divvy-tripdata.csv")
X202201_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202201-divvy-tripdata.csv")
X202202_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202202-divvy-tripdata.csv")
X202203_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202203-divvy-tripdata.csv")
X202204_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202204-divvy-tripdata.csv")
X202205_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202205-divvy-tripdata.csv")
X202206_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202206-divvy-tripdata.csv")
X202207_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202207-divvy-tripdata.csv")
X202208_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202208-divvy-tripdata.csv")
X202209_divvy_tripdata <-
```

```
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202209-divvy-tripdata.csv")
X202210_divvy_tripdata <-
  read_csv("./20221205-capstone_datascience-01/002_data/001_csv/202210-divvy-tripdata.csv")

all_trips <- bind_rows(X202110_divvy_tripdata, X202111_divvy_tripdata, X202112_divvy_tripdata,
                       X202201_divvy_tripdata, X202202_divvy_tripdata, X202203_divvy_tripdata,
                       X202204_divvy_tripdata, X202205_divvy_tripdata, X202206_divvy_tripdata,
                       X202207_divvy_tripdata, X202208_divvy_tripdata, X202209_divvy_tripdata,
                       X202210_divvy_tripdata)
```

```
all_trips$date <- as.Date(all_trips$started_at)
all_trips$month <- format(as.Date(all_trips$date), "%m")
all_trips$day <- format(as.Date(all_trips$date), "%d")
all_trips$year <- format(as.Date(all_trips$date), "%Y")
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A")
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$started_at)

all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
```

2023-01-09

- Continue preparing the dataset
- Remove bad data

```
all_trips_v2 <- all_trips[!(all_trips$ride_length<0),]
```

- move bad data into a dataframe to doublecheck

```
all_trips_errors <- all_trips[(all_trips$ride_length<0),]
```

- Conduct descriptive analysis as per script

```
# Descriptive analysis on ride_length (all figures in seconds)
mean(all_trips_v2$ride_length) #straight average (total ride length / rides)
```

```
## [1] 1164.48
```

```
median(all_trips_v2$ride_length) #midpoint number in the ascending array of ride lengths
```

```
## [1] 621
```

```
max(all_trips_v2$ride_length) #longest ride
```

```
## [1] 2483235
```

```
min(all_trips_v2$ride_length) #shortest ride
```

```
## [1] 0
```

```
# You can condense the four lines above to one line using summary() on the specific attribute
summary(all_trips_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     352     621    1164    1115 2483235
```

```
# Compare members and casual users
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = mean)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                1747.3620
## 2                     member                 761.6182
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = median)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                       793
## 2                     member                       529
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = max)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                  2483235
## 2                     member                    93594
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = min)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                         0
## 2                     member                         0
```

```
# See the average ride time by each day for members vs casual users
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                   Friday                1669.6638
## 2                      member                   Friday                 747.5315
## 3                      casual                   Monday                1743.7891
## 4                      member                   Monday                 733.6668
## 5                      casual                 Saturday                1948.5742
## 6                      member                 Saturday                 850.5721
## 7                      casual                   Sunday                2050.0862
## 8                      member                   Sunday                 846.8236
## 9                      casual                 Thursday                1515.9128
```

```
## 10                     member                Thursday                730.8069
## 11                     casual                 Tuesday               1540.5572
## 12                     member                 Tuesday                725.7491
## 13                     casual               Wednesday               1481.9751
## 14                     member               Wednesday                723.0776
```

```
# Notice that the days of the week are out of order. Let's fix that.
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "

# Now, let's run the average ride time by each day for members vs casual users
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                   Sunday                2050.0862
## 2                      member                   Sunday                 846.8236
## 3                      casual                   Monday                1743.7891
## 4                      member                   Monday                 733.6668
## 5                      casual                  Tuesday                1540.5572
## 6                      member                  Tuesday                 725.7491
## 7                      casual                Wednesday                1481.9751
## 8                      member                Wednesday                 723.0776
## 9                      casual                 Thursday                1515.9128
## 10                     member                 Thursday                 730.8069
## 11                     casual                   Friday                1669.6638
## 12                     member                   Friday                 747.5315
## 13                     casual                 Saturday                1948.5742
## 14                     member                 Saturday                 850.5721
```

```
# analyze ridership data by type and weekday
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(member_casual, weekday) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n()                                                    #calculates th
  ,average_duration = mean(ride_length)) %>%                 # calculates the average duration
  arrange(member_casual, weekday)                                                          # sort
```

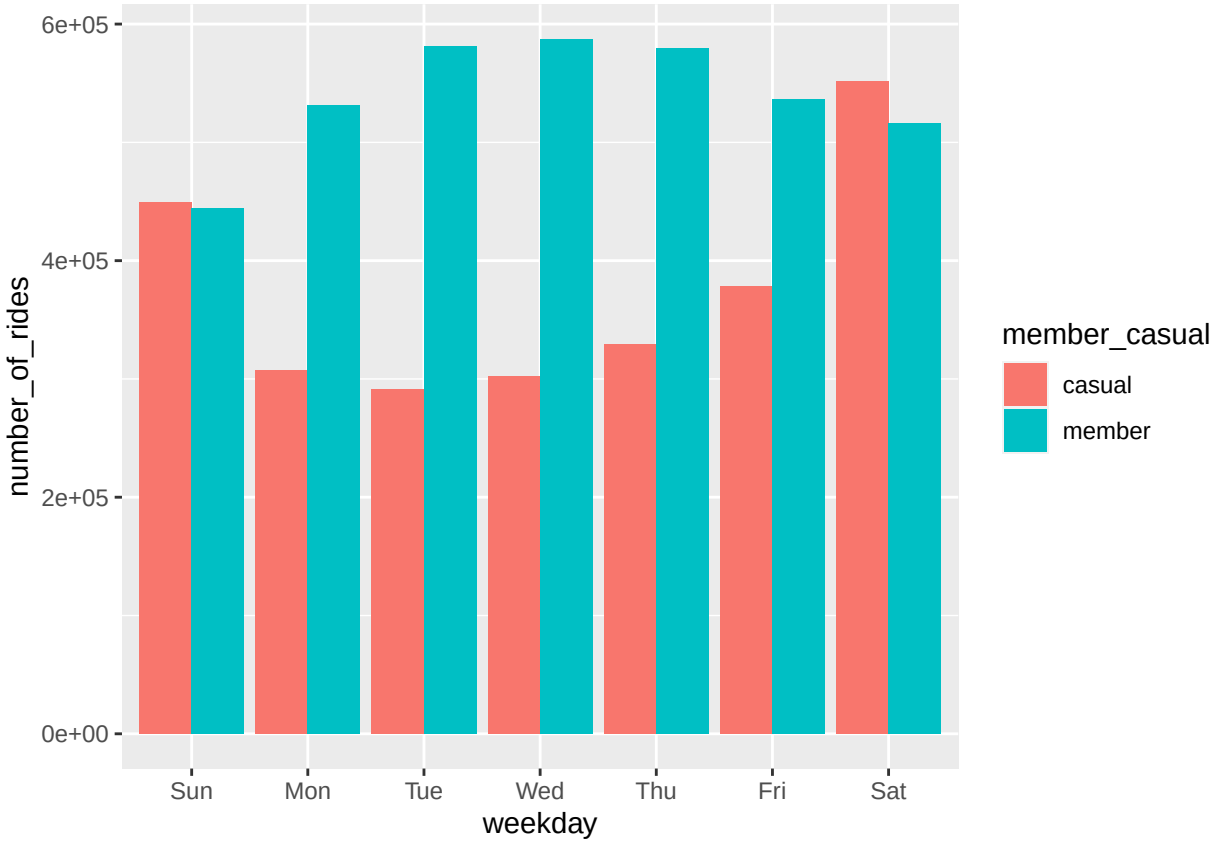
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              449690            2050.
##  2 casual        Mon              307019            1744.
##  3 casual        Tue              291466            1541.
##  4 casual        Wed              302253            1482.
##  5 casual        Thu              329276            1516.
##  6 casual        Fri              378487            1670.
##  7 casual        Sat              552023            1949.
##  8 member        Sun              443967             847.
```

```
##  9 member        Mon              531346            734.
## 10 member        Tue              581267            726.
## 11 member        Wed              587381            723.
## 12 member        Thu              579785            731.
## 13 member        Fri              536489            748.
## 14 member        Sat              516359            851.
```

```r
# Let's visualize the number of rides by rider type
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

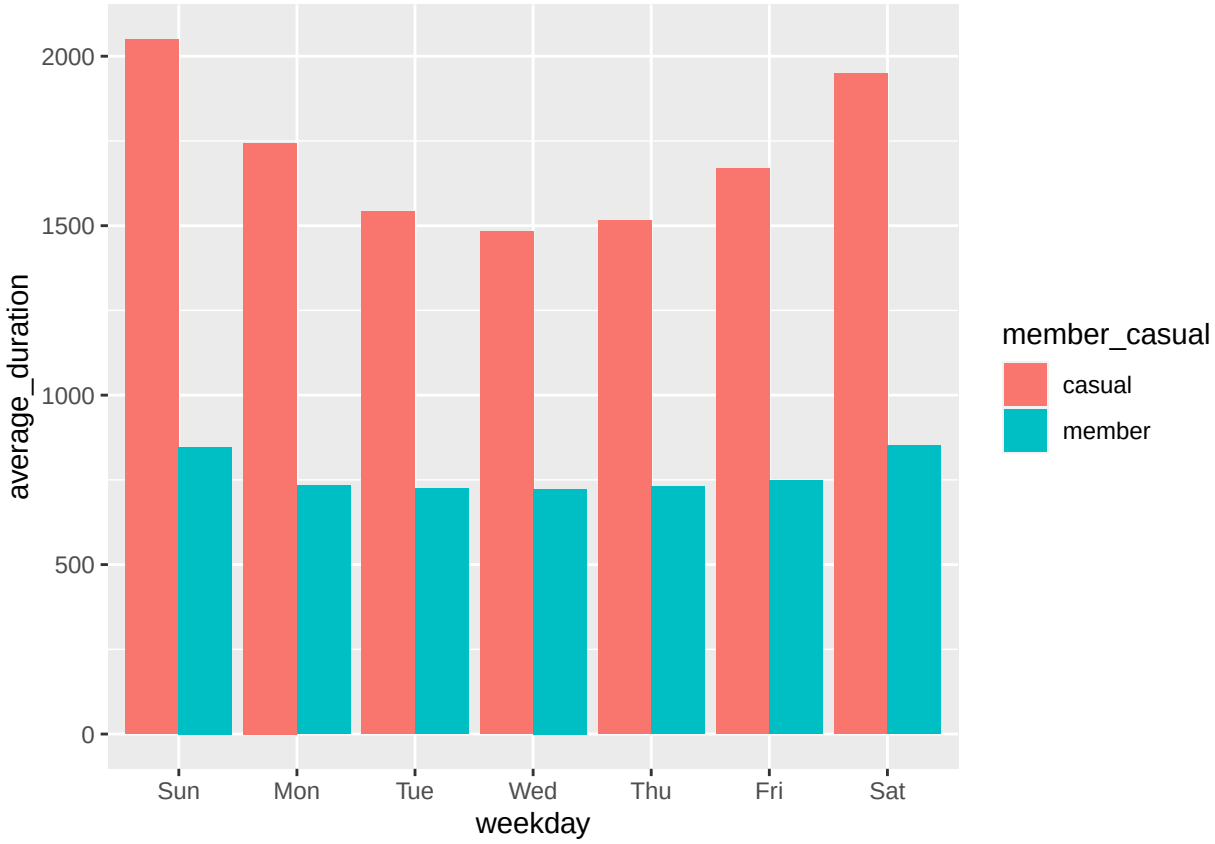
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```




```r
# Let's create a visualization for average duration
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
```

```
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```

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



2023-01-10

- Trying to calculate the distance between gps coordinates strart / end

```
library(geosphere)
```

```
all_trips_v3 <- all_trips_v2 %>%  mutate(geodist = distHaversine(cbind(all_trips_v2$start_lng,all_trips
```

- To fix performance issues, I had to setup a RStudio server

```
CentOS 9 Stream
RStudio 2022.12.0+353 "Elsbeth Geranium" Release
R version 4.2.2
```

2023-01-17

- filter all trips with distance = 0 into a dataframe called round_trips, the assumtion being that when the trip ends where it started

```
round_trips = filter(all_trips_v3,geodist == 0)
```

```
# Descriptive analysis on ride_length (all figures in seconds)
mean(round_trips$ride_length) #straight average (total ride length / rides)
```

```
## [1] 1379.296
```

```
median(round_trips$ride_length) #midpoint number in the ascending array of ride lengths
```

```
## [1] 360
```

```
max(round_trips$ride_length) #longest ride
```

```
## [1] 1336784
```

```
min(round_trips$ride_length) #shortest ride
```

```
## [1] 0
```

```
# You can condense the four lines above to one line using summary() on the specific attribute
summary(round_trips$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0      81     360    1379    1734 1336784
```

```
# Compare members and casual users
aggregate(round_trips$ride_length ~ round_trips$member_casual, FUN = mean)
```

```
##   round_trips$member_casual round_trips$ride_length
## 1                    casual               1958.8456
## 2                    member                695.7375
```

```
aggregate(round_trips$ride_length ~ round_trips$member_casual, FUN = median)
```

```
##   round_trips$member_casual round_trips$ride_length
## 1                    casual                     811
## 2                    member                     205
```

```
aggregate(round_trips$ride_length ~ round_trips$member_casual, FUN = max)
```

```
##   round_trips$member_casual round_trips$ride_length
## 1                    casual                1336784
## 2                    member                  86721
```

```
aggregate(round_trips$ride_length ~ round_trips$member_casual, FUN = min)
```

```
##   round_trips$member_casual round_trips$ride_length
## 1                    casual                       0
## 2                    member                       0
```

```
# See the average ride time by each day for members vs casual users
aggregate(round_trips$ride_length ~ round_trips$member_casual + round_trips$day_of_week, FUN = mean)
```

```
##    round_trips$member_casual round_trips$day_of_week round_trips$ride_length
## 1                    casual                   Sunday                2220.1281
## 2                    member                   Sunday                 790.8003
## 3                    casual                   Monday                2135.8462
## 4                    member                   Monday                 711.7544
## 5                    casual                  Tuesday                1828.5223
## 6                    member                  Tuesday                 671.0404
## 7                    casual                Wednesday                1791.0633
## 8                    member                Wednesday                 651.3714
## 9                    casual                 Thursday                1759.3558
## 10                   member                 Thursday                 655.4954
## 11                   casual                   Friday                1747.3120
## 12                   member                   Friday                 665.9085
## 13                   casual                 Saturday                2003.3756
## 14                   member                 Saturday                 730.6534
```

```
# Notice that the days of the week are out of order. Let's fix that.
round_trips$day_of_week <- ordered(round_trips$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "We

# Now, let's run the average ride time by each day for members vs casual users
aggregate(round_trips$ride_length ~ round_trips$member_casual + round_trips$day_of_week, FUN = mean)
```

```
##    round_trips$member_casual round_trips$day_of_week round_trips$ride_length
## 1                    casual                   Sunday                2220.1281
## 2                    member                   Sunday                 790.8003
## 3                    casual                   Monday                2135.8462
## 4                    member                   Monday                 711.7544
## 5                    casual                  Tuesday                1828.5223
## 6                    member                  Tuesday                 671.0404
## 7                    casual                Wednesday                1791.0633
## 8                    member                Wednesday                 651.3714
## 9                    casual                 Thursday                1759.3558
## 10                   member                 Thursday                 655.4954
## 11                   casual                   Friday                1747.3120
## 12                   member                   Friday                 665.9085
## 13                   casual                 Saturday                2003.3756
## 14                   member                 Saturday                 730.6534
```

```
# analyze ridership data by type and weekday
round_trips %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday field using wday()
```

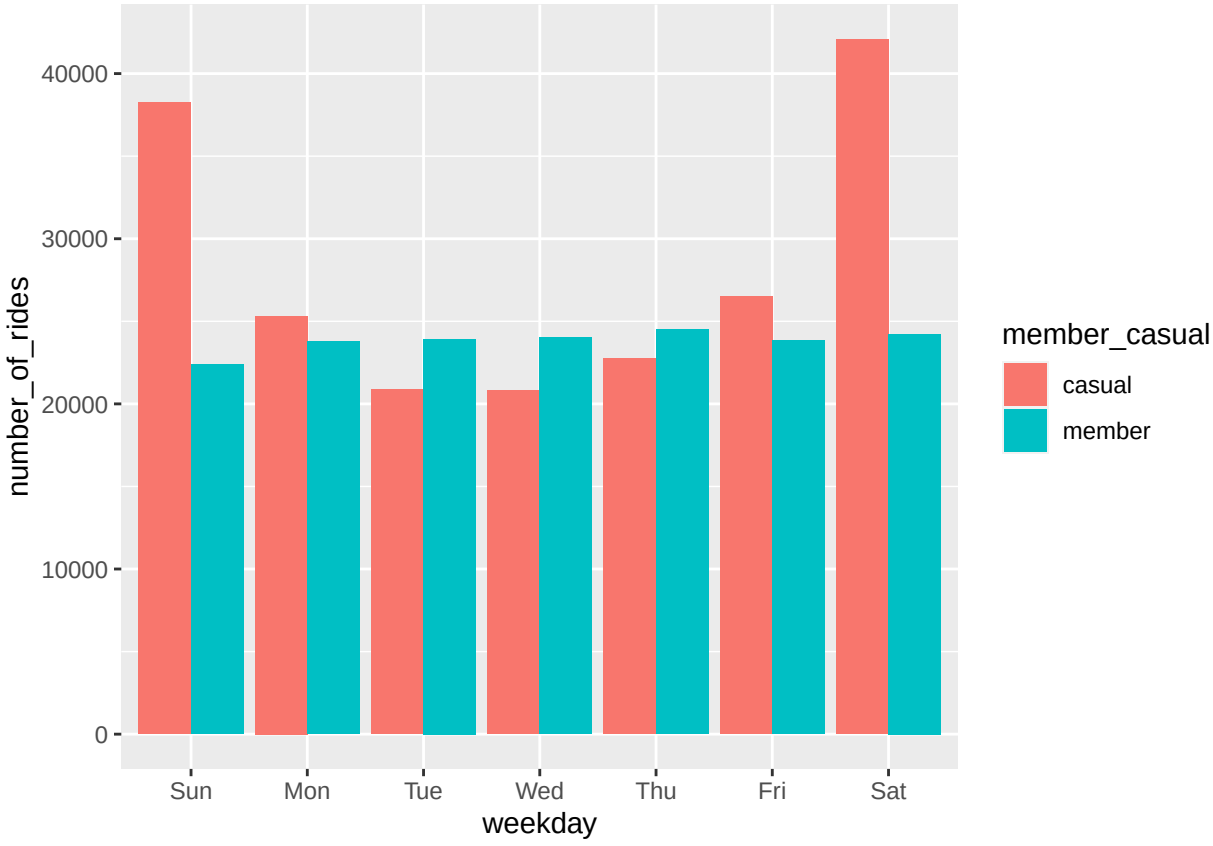
```
  group_by(member_casual, weekday) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n()                                          #calculates th
  ,average_duration = mean(ride_length)) %>%                 # calculates the average duration
  arrange(member_casual, weekday)                                                  # sort
```

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

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun               38269            2220.
##  2 casual        Mon               25315            2136.
##  3 casual        Tue               20886            1829.
##  4 casual        Wed               20810            1791.
##  5 casual        Thu               22745            1759.
##  6 casual        Fri               26485            1747.
##  7 casual        Sat               42074            2003.
##  8 member        Sun               22367             791.
##  9 member        Mon               23779             712.
## 10 member        Tue               23933             671.
## 11 member        Wed               24011             651.
## 12 member        Thu               24491             655.
## 13 member        Fri               23853             666.
## 14 member        Sat               24238             731.
```

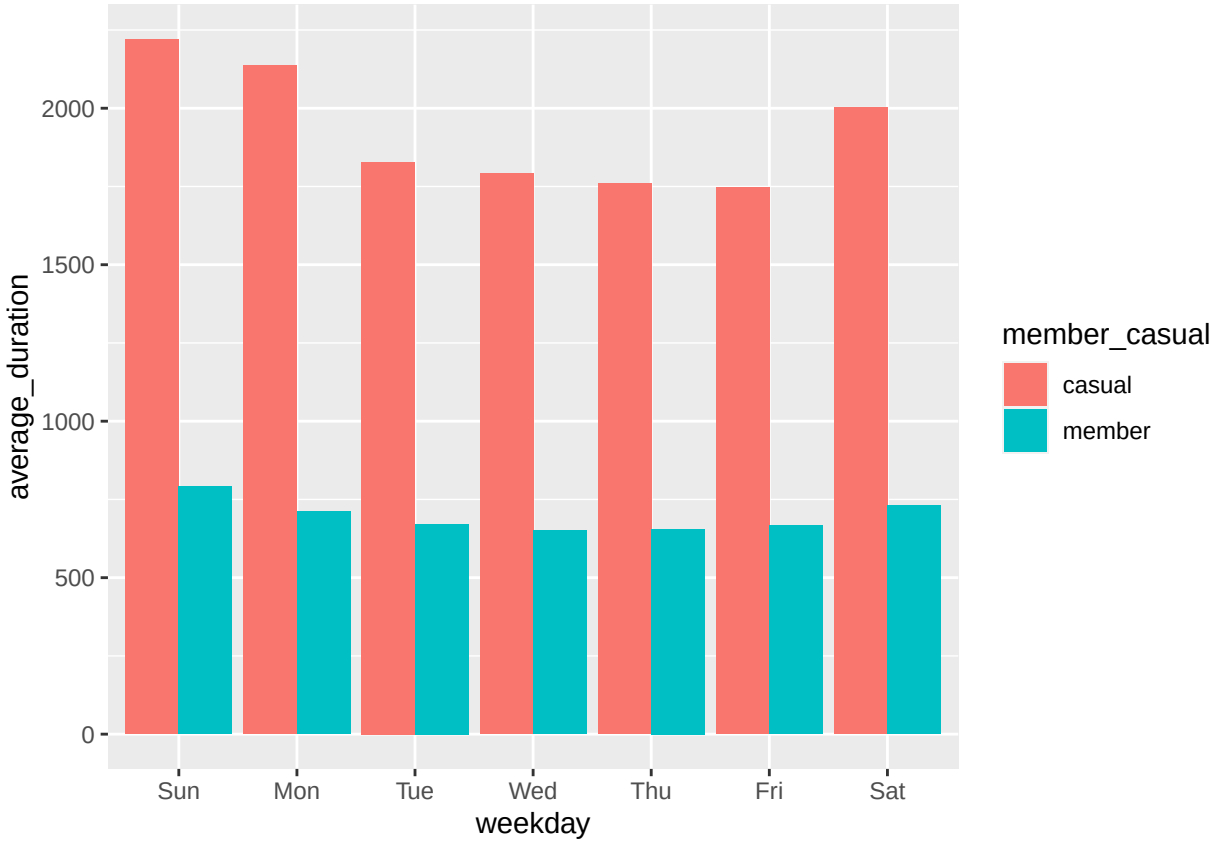
```
# Let's visualize the number of rides by rider type
round_trips %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

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




```
# Let's create a visualization for average duration
round_trips %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```

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




```
aggregate(round_trips$ride_length ~ round_trips$member_casual, FUN = mean)
```

```
##   round_trips$member_casual round_trips$ride_length
## 1                    casual               1958.8456
## 2                    member                695.7375
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = mean)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                1747.3620
## 2                     member                 761.6182
```

```
aggregate(all_trips_v3$ride_length ~ all_trips_v3$member_casual, FUN = mean)
```

```
##   all_trips_v3$member_casual all_trips_v3$ride_length
## 1                     casual                1747.3620
## 2                     member                 761.6182
```

```
aggregate(round_trips$ride_length ~ round_trips$member_casual, FUN = mean)
```

```
##   round_trips$member_casual round_trips$ride_length
## 1                    casual               1958.8456
## 2                    member                695.7375
```

```
aggregate(round_trips$ride_length ~ round_trips$member_casual + round_trips$day_of_week, FUN = mean)
```

```
##    round_trips$member_casual round_trips$day_of_week round_trips$ride_length
## 1                     casual                  Sunday               2220.1281
## 2                     member                  Sunday                790.8003
## 3                     casual                  Monday               2135.8462
## 4                     member                  Monday                711.7544
## 5                     casual                 Tuesday               1828.5223
## 6                     member                 Tuesday                671.0404
## 7                     casual               Wednesday               1791.0633
## 8                     member               Wednesday                651.3714
## 9                     casual                Thursday               1759.3558
## 10                    member                Thursday                655.4954
## 11                    casual                  Friday               1747.3120
## 12                    member                  Friday                665.9085
## 13                    casual                Saturday               2003.3756
## 14                    member                Saturday                730.6534
```