DivvyAnalaysis

Sajith

2022-11-21

STEP 1: Importing libraries

```
library(lubridate)
```

```
## Loading required package: timechange

##
## Attaching package: 'lubridate'

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

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(ggplot2)
library("anytime")
```

STEP 2: Collecting Data

```
q2_2019 <- read.csv("Divvy_Trips_2019_Q2.csv")
q3_2019 <- read.csv("Divvy_Trips_2019_Q3.csv")
q4_2019 <- read.csv("Divvy_Trips_2019_Q4.csv")
q1_2020 <- read.csv("Divvy_Trips_2020_Q1.csv")
```

STEP 3: Wrangle Data AND COMBINE INTO A SINGLE FILE

Comparing cols of each file

the names are not in same order so we change them into same col heads throughout

then we combine all of these files into a single new csv file

```
colnames(q1_2020)
```

```
##  [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"
```

```
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"
```

```
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(q2_2019)
```

```
##  [1] "X01...Rental.Details.Rental.ID"
##  [2] "X01...Rental.Details.Local.Start.Time"
##  [3] "X01...Rental.Details.Local.End.Time"
##  [4] "X01...Rental.Details.Bike.ID"
##  [5] "X01...Rental.Details.Duration.In.Seconds.Uncapped"
##  [6] "X03...Rental.Start.Station.ID"
##  [7] "X03...Rental.Start.Station.Name"
##  [8] "X02...Rental.End.Station.ID"
##  [9] "X02...Rental.End.Station.Name"
## [10] "User.Type"
## [11] "Member.Gender"
## [12] "X05...Member.Details.Member.Birthday.Year"
```

STEP 4: Renaming,Mutating, and Transformation of data

Renaming other files using the same col heads as this q1__2020 dataset

Inspecting the dataframes for incongruencies

```
str(q1_2020)
```

```
## 'data.frame':    426887 obs. of  13 variables:
##  $ ride_id           : chr  "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A388DAC6ABF3
##  $ rideable_type     : chr  "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : chr  "2020-01-21 20:06:59" "2020-01-30 14:22:39" "2020-01-09 19:29:26" "2020-0
##  $ ended_at          : chr  "2020-01-21 20:14:30" "2020-01-30 14:26:22" "2020-01-09 19:32:17" "2020-0
##  $ start_station_name: chr  "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway & Belmont
##  $ start_station_id  : int  239 234 296 51 66 212 96 96 212 38 ...
##  $ end_station_name  : chr  "Clark St & Leland Ave" "Southport Ave & Irving Park Rd" "Wilton Ave & Be
##  $ end_station_id    : int  326 318 117 24 212 96 212 212 96 100 ...
##  $ start_lat         : num  42 42 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  42 42 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

```
str(q4_2019)
```

```
## 'data.frame':    704054 obs. of  12 variables:
##  $ ride_id           : int  25223640 25223641 25223642 25223643 25223644 25223645 25223646 25223647 2
##  $ started_at        : chr  "2019-10-01 00:01:39" "2019-10-01 00:02:16" "2019-10-01 00:04:32" "2019-1
##  $ ended_at          : chr  "2019-10-01 00:17:20" "2019-10-01 00:06:34" "2019-10-01 00:18:43" "2019-1
##  $ rideable_type     : int  2215 6328 3003 3275 5294 1891 1061 1274 6011 2957 ...
##  $ tripduration      : chr  "940.0" "258.0" "850.0" "2,350.0" ...
##  $ start_station_id  : int  20 19 84 313 210 156 84 156 156 336 ...
##  $ start_station_name: chr  "Sheffield Ave & Kingsbury St" "Throop (Loomis) St & Taylor St" "Milwauke
##  $ end_station_id    : int  309 241 199 290 382 226 142 463 463 336 ...
##  $ end_station_name  : chr  "Leavitt St & Armitage Ave" "Morgan St & Polk St" "Wabash Ave & Grand Ave
##  $ member_casual     : chr  "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ gender            : chr  "Male" "Male" "Female" "Male" ...
##  $ birthyear         : int  1987 1998 1991 1990 1987 1994 1991 1995 1993 NA ...
```

```
str(q3_2019)
```

```
## 'data.frame':    1640718 obs. of  12 variables:
##  $ ride_id           : int  23479388 23479389 23479390 23479391 23479392 23479393 23479394 23479395 2
##  $ started_at        : chr  "2019-07-01 00:00:27" "2019-07-01 00:01:16" "2019-07-01 00:01:48" "2019-0
##  $ ended_at          : chr  "2019-07-01 00:20:41" "2019-07-01 00:18:44" "2019-07-01 00:27:42" "2019-0
##  $ rideable_type     : int  3591 5353 6180 5540 6014 4941 3770 5442 2957 6091 ...
##  $ tripduration      : chr  "1,214.0" "1,048.0" "1,554.0" "1,503.0" ...
##  $ start_station_id  : int  117 381 313 313 168 300 168 313 43 43 ...
##  $ start_station_name: chr  "Wilton Ave & Belmont Ave" "Western Ave & Monroe St" "Lakeview Ave & Ful
##  $ end_station_id    : int  497 203 144 144 62 232 62 144 195 195 ...
##  $ end_station_name  : chr  "Kimball Ave & Belmont Ave" "Western Ave & 21st St" "Larrabee St & Webst
##  $ member_casual     : chr  "Subscriber" "Customer" "Customer" "Customer" ...
##  $ gender            : chr  "Male" "" "" "" ...
##  $ birthyear         : int  1992 NA NA NA NA 1990 NA NA NA NA ...
```

```
str(q2_2019)
```

```
## 'data.frame':    1108163 obs. of  12 variables:
##  $ ride_id                                       : int  22178529 22178530 22178531 22178532 221785
##  $ started_at                                    : chr  "2019-04-01 00:02:22" "2019-04-01 00:03:0
##  $ ended_at                                      : chr  "2019-04-01 00:09:48" "2019-04-01 00:20:3
##  $ rideable_type                                 : int  6251 6226 5649 4151 3270 3123 6418 4513 32
##  $ X01...Rental.Details.Duration.In.Seconds.Uncapped: chr  "446.0" "1,048.0" "252.0" "357.0" ...
##  $ start_station_id                              : int  81 317 283 26 202 420 503 260 211 211 ...
##  $ start_station_name                            : chr  "Daley Center Plaza" "Wood St & Taylor St
##  $ end_station_id                                : int  56 59 174 133 129 426 500 499 211 211 ...
##  $ end_station_name                              : chr  "Desplaines St & Kinzie St" "Wabash Ave &
##  $ member_casual                                 : chr  "Subscriber" "Subscriber" "Subscriber" "Su
##  $ Member.Gender                                 : chr  "Male" "Female" "Male" "Male" ...
##  $ X05...Member.Details.Member.Birthday.Year        : int  1975 1984 1990 1993 1992 1999 1969 1991 N
```

Convert ride_id and rideable_type to character so that they can stack correctly

```
q4_2019 <-  mutate(q4_2019, ride_id = as.character(ride_id)
                   ,rideable_type = as.character(rideable_type))

q3_2019 <-  mutate(q3_2019, ride_id = as.character(ride_id)
                   ,rideable_type = as.character(rideable_type))

q2_2019 <-  mutate(q2_2019, ride_id = as.character(ride_id)
                   ,rideable_type = as.character(rideable_type))
```

Stacking individual dataframes into one big dataframe

```
all_trips <- bind_rows(q1_2020,q2_2019,q3_2019,q4_2019)
```

Remove lat, long, birthyear, and gender fields as this data was dropped beginning in 2020

```
all_trips <- all_trips %>%
  select(-c(start_lat,start_lng,end_lat,end_lng,gender,"X01...Rental.Details.Duration.In.Seconds.Uncapp
            Member.Gender,tripduration,"X05...Member.Details.Member.Birthday.Year"))

all_trips <- all_trips %>%
  select(-c(birthyear))
```

STEP 5: CLEANING UP DATA AND AND ADD DATA TO PREPARE FOR ANALYSIS

Inspecting new data fram created for further analysis

```
nrow(all_trips) #How many rows are in data frame?
```

```
## [1] 3879822
```

```
colnames(all_trips) #List of column names
```

```
## [1] "ride_id"            "rideable_type"      "started_at"
## [4] "ended_at"           "start_station_name" "start_station_id"
## [7] "end_station_name"   "end_station_id"     "member_casual"
```

```
dim(all_trips)#Dimensions of the data frame?
```

```
## [1] 3879822       9
```

```
head(all_trips)#See the first 6 rows of data frame.
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 EACB19130B0CDA4A   docked_bike 2020-01-21 20:06:59 2020-01-21 20:14:30
## 2 8FED874C809DC021   docked_bike 2020-01-30 14:22:39 2020-01-30 14:26:22
## 3 789F3C21E472CA96   docked_bike 2020-01-09 19:29:26 2020-01-09 19:32:17
## 4 C9A388DAC6ABF313   docked_bike 2020-01-06 16:17:07 2020-01-06 16:25:56
## 5 943BC3CBECCFD662   docked_bike 2020-01-30 08:37:16 2020-01-30 08:42:48
## 6 6D9C8A6938165C11   docked_bike 2020-01-10 12:33:05 2020-01-10 12:37:54
##         start_station_name start_station_id              end_station_name
## 1 Western Ave & Leland Ave              239          Clark St & Leland Ave
## 2  Clark St & Montrose Ave              234 Southport Ave & Irving Park Rd
## 3   Broadway & Belmont Ave              296       Wilton Ave & Belmont Ave
## 4   Clark St & Randolph St               51       Fairbanks Ct & Grand Ave
## 5     Clinton St & Lake St               66          Wells St & Hubbard St
## 6    Wells St & Hubbard St              212    Desplaines St & Randolph St
##   end_station_id member_casual
## 1            326        member
## 2            318        member
## 3            117        member
## 4             24        member
## 5            212        member
## 6             96        member
```

```
tail(all_trips)#See the last 6 rows of data frame.
```

```
##          ride_id rideable_type          started_at            ended_at
## 3879817 25962899          5996 2019-12-31 23:54:54 2020-01-01 00:22:02
## 3879818 25962900          2196 2019-12-31 23:56:13 2020-01-01 00:15:45
## 3879819 25962901          4877 2019-12-31 23:56:34 2020-01-01 00:22:08
## 3879820 25962902           863 2019-12-31 23:57:05 2020-01-01 00:05:46
## 3879821 25962903          2637 2019-12-31 23:57:11 2020-01-01 00:05:45
## 3879822 25962904          5930 2019-12-31 23:57:17 2019-12-31 23:59:18
##                          start_station_name start_station_id
## 3879817 Mies van der Rohe Way & Chestnut St              145
## 3879818              Green St & Randolph St              112
## 3879819                     Millennium Park               90
## 3879820               Michigan Ave & 8th St              623
## 3879821               Michigan Ave & 8th St              623
## 3879822              Broadway & Sheridan Rd              256
##                     end_station_name end_station_id member_casual
## 3879817    Michigan Ave & Pearson St             25    Subscriber
## 3879818     Halsted St & Dickens Ave            225    Subscriber
## 3879819              Millennium Park             90    Subscriber
## 3879820       Michigan Ave & Lake St             52    Subscriber
## 3879821       Michigan Ave & Lake St             52    Subscriber
## 3879822 Sheridan Rd & Irving Park Rd            240    Subscriber
```

```
str(all_trips)#See list of columns and data types (numeric, character, etc)
```

```
## ’data.frame’:    3879822 obs. of  9 variables:
##  $ ride_id           : chr  "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A388DAC6ABF3
##  $ rideable_type     : chr  "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : chr  "2020-01-21 20:06:59" "2020-01-30 14:22:39" "2020-01-09 19:29:26" "2020-0
##  $ ended_at          : chr  "2020-01-21 20:14:30" "2020-01-30 14:26:22" "2020-01-09 19:32:17" "2020-0
##  $ start_station_name: chr  "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway & Belmont
##  $ start_station_id  : int  239 234 296 51 66 212 96 96 212 38 ...
##  $ end_station_name  : chr  "Clark St & Leland Ave" "Southport Ave & Irving Park Rd" "Wilton Ave & Be
##  $ end_station_id    : int  326 318 117 24 212 96 212 212 96 100 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

```
summary(all_trips)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:3879822     Length:3879822     Length:3879822     Length:3879822
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                            
##                                                                            
##                                                                            
##                                                                            
##  start_station_name start_station_id end_station_name   end_station_id 
##  Length:3879822     Min.   :  1.0    Length:3879822     Min.   :  1.0  
##  Class :character   1st Qu.: 77.0    Class :character   1st Qu.: 77.0  
##  Mode  :character   Median :174.0    Mode  :character   Median :174.0  
##                     Mean   :202.9                       Mean   :203.8  
##                     3rd Qu.:291.0                       3rd Qu.:291.0  
##                     Max.   :675.0                       Max.   :675.0  
##                                                         NA’s   :1      
##  member_casual     
##  Length:3879822    
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
##                    
```

There are a few problems we will need to fix:

(1) In the “member_casual” column, there are two names for members (“member” and “Subscriber”) and two names for casual riders (“Customer” and “casual”). We will need to consolidate that from four to two labels.

(2) The data can only be aggregated at the ride-level, which is too granular. We will want to add some additional columns of data – such as day, month, year – that provide additional opportunities to aggregate the data.

(3) We will want to add a calculated field for length of ride since the 2020 data did not have the “tripduration” column. We will add “ride_length” to the entire dataframe for consistency.

(4) There are some rides where tripduration shows up as negative, including several hundred rides where Divvy took bikes out of circulation for Quality Control reasons. We will want to delete these rides.

In the “member_casual” column, replace “Subscriber” with “member” and “Customer” with “casual”

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

Before 2020, Divvy used different labels for these two types of riders ... we will want to make our dataframe consistent with this format

```
table(all_trips$member_casual)
```

checking if it changed

```
##
##  casual  member
##  905954 2973868
```

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

```
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")
```

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

```
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$started_at)
```

Add a "ride_length" calculation to all_trips (in seconds)

```
str(all_trips)
```

Inspecting the structure of the columns

```
## 'data.frame':    3879822 obs. of  15 variables:
##  $ ride_id          : chr  "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A388DAC6ABF3
##  $ rideable_type     : chr  "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : chr  "2020-01-21 20:06:59" "2020-01-30 14:22:39" "2020-01-09 19:29:26" "2020-
##  $ ended_at          : chr  "2020-01-21 20:14:30" "2020-01-30 14:26:22" "2020-01-09 19:32:17" "2020-
##  $ start_station_name: chr  "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway & Belmont
##  $ start_station_id  : int  239 234 296 51 66 212 96 96 212 38 ...
##  $ end_station_name  : chr  "Clark St & Leland Ave" "Southport Ave & Irving Park Rd" "Wilton Ave & B
##  $ end_station_id    : int  326 318 117 24 212 96 212 212 96 100 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
##  $ date              : Date, format: "2020-01-21" "2020-01-30" ...
##  $ month             : chr  "01" "01" "01" "01" ...
##  $ day               : chr  "21" "30" "09" "06" ...
##  $ year              : chr  "2020" "2020" "2020" "2020" ...
##  $ day_of_week       : chr  "Tuesday" "Thursday" "Thursday" "Monday" ...
##  $ ride_length       : 'difftime' num  451 223 171 529 ...
##   ..- attr(*, "units")= chr "secs"
```

```
head(all_trips)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 EACB19130B0CDA4A   docked_bike 2020-01-21 20:06:59 2020-01-21 20:14:30
## 2 8FED874C809DC021   docked_bike 2020-01-30 14:22:39 2020-01-30 14:26:22
## 3 789F3C21E472CA96   docked_bike 2020-01-09 19:29:26 2020-01-09 19:32:17
## 4 C9A388DAC6ABF313   docked_bike 2020-01-06 16:17:07 2020-01-06 16:25:56
## 5 943BC3CBECCFD662   docked_bike 2020-01-30 08:37:16 2020-01-30 08:42:48
## 6 6D9C8A6938165C11   docked_bike 2020-01-10 12:33:05 2020-01-10 12:37:54
##         start_station_name start_station_id               end_station_name
## 1 Western Ave & Leland Ave              239          Clark St & Leland Ave
## 2  Clark St & Montrose Ave              234 Southport Ave & Irving Park Rd
## 3   Broadway & Belmont Ave              296       Wilton Ave & Belmont Ave
## 4   Clark St & Randolph St               51       Fairbanks Ct & Grand Ave
## 5     Clinton St & Lake St               66          Wells St & Hubbard St
## 6    Wells St & Hubbard St              212    Desplaines St & Randolph St
##   end_station_id member_casual       date month day year day_of_week
## 1            326        member 2020-01-21    01  21 2020     Tuesday
## 2            318        member 2020-01-30    01  30 2020    Thursday
## 3            117        member 2020-01-09    01  09 2020    Thursday
## 4             24        member 2020-01-06    01  06 2020      Monday
## 5            212        member 2020-01-30    01  30 2020    Thursday
## 6             96        member 2020-01-10    01  10 2020      Friday
##   ride_length
## 1    451 secs
## 2    223 secs
## 3    171 secs
## 4    529 secs
## 5    332 secs
## 6    289 secs
```

```
is.factor(all_trips$ride_length)
```

Convert "ride_length" from Factor to numeric so we can run calculations on the data

```
## [1] FALSE
```

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

```
## [1] TRUE
```

```
str(all_trips)
```

```
## 'data.frame':    3879822 obs. of  15 variables:
##  $ ride_id           : chr  "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A388DAC6ABF3
##  $ rideable_type     : chr  "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : chr  "2020-01-21 20:06:59" "2020-01-30 14:22:39" "2020-01-09 19:29:26" "2020-
##  $ ended_at          : chr  "2020-01-21 20:14:30" "2020-01-30 14:26:22" "2020-01-09 19:32:17" "2020-
##  $ start_station_name: chr  "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway & Belmont
##  $ start_station_id  : int  239 234 296 51 66 212 96 96 212 38 ...
##  $ end_station_name  : chr  "Clark St & Leland Ave" "Southport Ave & Irving Park Rd" "Wilton Ave & Be
##  $ end_station_id    : int  326 318 117 24 212 96 212 212 96 100 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
##  $ date              : Date, format: "2020-01-21" "2020-01-30" ...
##  $ month             : chr  "01" "01" "01" "01" ...
##  $ day               : chr  "21" "30" "09" "06" ...
##  $ year              : chr  "2020" "2020" "2020" "2020" ...
##  $ day_of_week       : chr  "Tuesday" "Thursday" "Thursday" "Monday" ...
##  $ ride_length       : num  451 223 171 529 332 289 289 297 295 203 ...
```

Removing Bad Data

The dataframe includes a few hundred entries when bikes were taken out of docks and checked for quality by Divvy or ride_length was negative

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

We will create a new cleaned dataframe

STEP 6: CONDUCT DESCRIPTIVE ANALYSIS

```
mean(all_trips$ride_length) #straight average (total ride length / rides)
```

Descriptive analysis on ride_length (all figures in seconds)

```
## [1] 1477.691
```

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

```
## [1] 712
```

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

```
## [1] 9387024
```

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

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

```
summary(all_trips_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1     412     712    1479    1289 9387024
```

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

Compare members and casual users

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                3552.7502
## 2                     member                 850.0662
```

```
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                     1546
## 2                     member                      589
```

```
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                  9387024
## 2                     member                  9056634
```

```
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                        2
## 2                     member                        1
```

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

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

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                   Friday                3773.8351
## 2                      member                   Friday                 824.5305
## 3                      casual                   Monday                3372.2869
## 4                      member                   Monday                 842.5726
## 5                      casual                 Saturday                3331.9138
## 6                      member                 Saturday                 968.9337
## 7                      casual                   Sunday                3581.4054
## 8                      member                   Sunday                 919.9746
## 9                      casual                 Thursday                3682.9847
## 10                     member                 Thursday                 823.9278
## 11                     casual                  Tuesday                3596.3599
## 12                     member                  Tuesday                 826.1427
## 13                     casual                Wednesday                3718.6619
## 14                     member                Wednesday                 823.9996
```

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

Notice that the days of the week are out of order. Let's fix that.

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

Now, let's run the average ride time by each day for members vs casual users

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                   Sunday                3581.4054
## 2                      member                   Sunday                 919.9746
## 3                      casual                   Monday                3372.2869
## 4                      member                   Monday                 842.5726
## 5                      casual                  Tuesday                3596.3599
## 6                      member                  Tuesday                 826.1427
## 7                      casual                Wednesday                3718.6619
## 8                      member                Wednesday                 823.9996
## 9                      casual                 Thursday                3682.9847
## 10                     member                 Thursday                 823.9278
## 11                     casual                   Friday                3773.8351
## 12                     member                   Friday                 824.5305
## 13                     casual                 Saturday                3331.9138
## 14                     member                 Saturday                 968.9337
```

```
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 the number of rides and average dur
            ,average_duration = mean(ride_length)) %>%  # calculates the average duration
  arrange(member_casual, weekday)                       # sorts
```

analyze ridership data by type and weekday

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

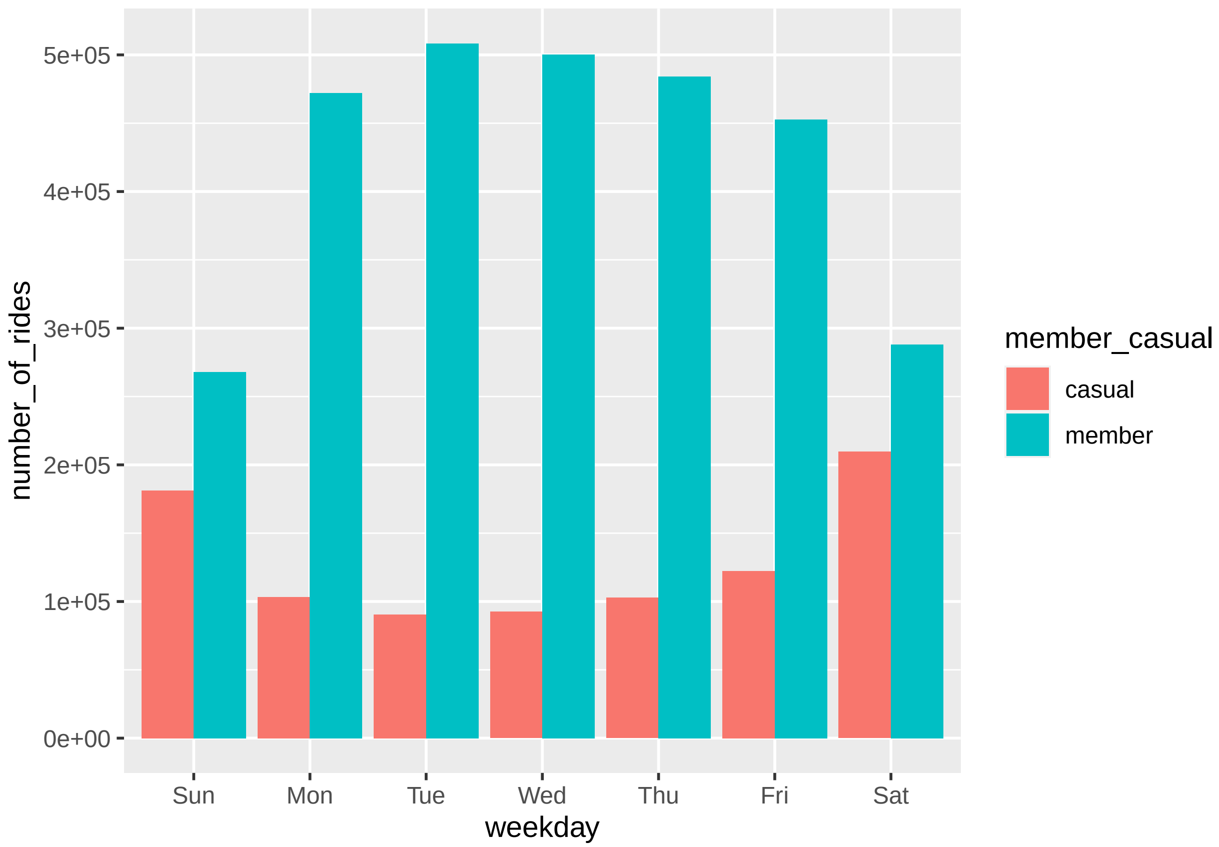
```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              181293            3581.
##  2 casual        Mon              103296            3372.
##  3 casual        Tue               90510            3596.
##  4 casual        Wed               92457            3719.
##  5 casual        Thu              102679            3683.
##  6 casual        Fri              122404            3774.
##  7 casual        Sat              209543            3332.
##  8 member        Sun              267965             920.
##  9 member        Mon              472196             843.
## 10 member        Tue              508445             826.
## 11 member        Wed              500329             824.
## 12 member        Thu              484177             824.
## 13 member        Fri              452790             825.
## 14 member        Sat              287958             969.
```

STEP 7: VISUALIZATION

```
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")
```

Let's visualize the number of rides by rider type

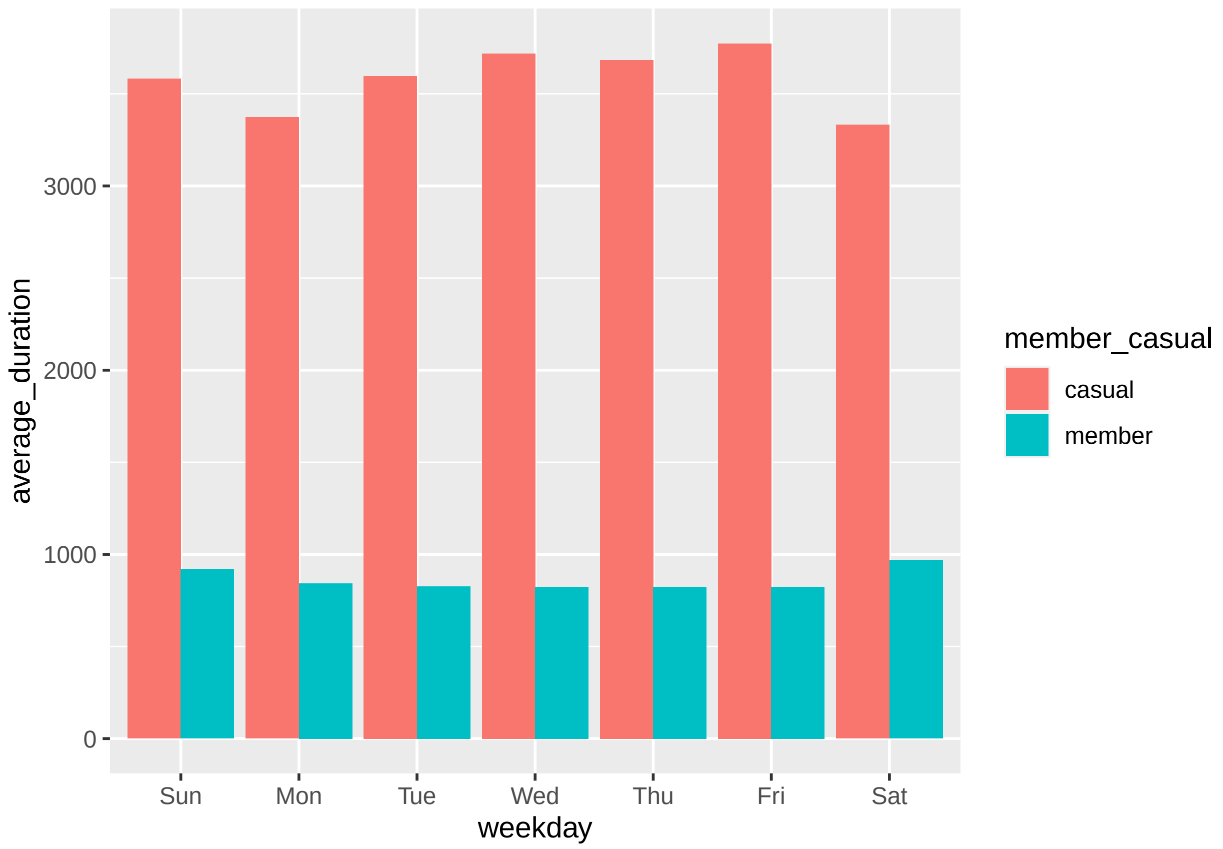
```
## 'summarise()' has grouped output by 'member_casual'. You can override using the
## '.groups' argument.
```



```
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")
```

Let's create a visualization for average duration

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


average_duration
3000
2000
1000
0
Sun
Mon
Tue
Wed
Thu
Fri
Sat
weekday
member_casual
casual
member