Cyclistic Divvy Bikes Case Study

Sean Mattison

2023-10-13

Cyclistic_Divvy_Trips_2019_Analysis

Using the Divvy_Trips_2019 dataset for the case study. The purpose of this script is to consolidate downloaded Divvy data into a single dataframe and then conduct

simple analysis to help answer the key question: "In what ways do members and casual riders use Divvy bikes differently?" The data is publicly available and has been provided by Motivate International Inc. under an appropriate license agreement.

Check and set working directory to simplify calls to data

```
getwd()
```

```
## [1] "/Users/seanmattison1/Desktop/divvy_bike_data"
```

```
setwd("/Users/seanmattison1/Desktop/divvy_bike_data")
```

Install required packages

```
options(repos = structure(c(CRAN = "https://cran.r-project.org")))
install.packages("tidyverse")
```

```
##
## The downloaded binary packages are in
##  /var/folders/pj/7ff7wgg501zc31s5np081fn40000gp/T//Rtmpd003qk/downloaded_packages
```

```
install.packages("lubridate")
```

```
##
## The downloaded binary packages are in
##  /var/folders/pj/7ff7wgg501zc31s5np081fn40000gp/T//Rtmpd003qk/downloaded_packages
```

```
install.packages("ggplot2")
```

```
##
## The downloaded binary packages are in
##  /var/folders/pj/7ff7wgg501zc31s5np081fn40000gp/T//Rtmpd003qk/downloaded_packages
```

```
install.packages("tinytex")
```

```
##
## The downloaded binary packages are in
##  /var/folders/pj/7ff7wgg501zc31s5np081fn40000gp/T//Rtmpd003qk/downloaded_packages
```

```
library(tidyverse) # helps wrangle data
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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 errors
```

```
library(lubridate) # helps wrangle date attributes
library(ggplot2) # helps visualize data
library(dplyr) # manipulation and transformation
library(tinytex)
```

#====================== #STEP 1: COLLECT DATA #======================
# Collect data, upload Divvy dataset

```
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
## num  (1): tripduration
## dttm (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...
## num  (1): 01 - Rental Details Duration In Seconds Uncapped
## dttm (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
## num  (1): tripduration
## dttm (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
## num  (1): tripduration
## dttm (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.
```

#=== #STEP 2: WRANGLE DATA AND COMBINE INTO A SINGLE FILE #================================
# Compare column names in each of the files

```
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"
```

```r
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"
```

```r
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 columns to make them consistent with q1__2019, q2__2019, and q3__2019

```r
(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"))
```

```
## # A tibble: 1,108,163 x 12
##     trip_id start_time          end_time             bikeid tripduration
##       <dbl> <dttm>              <dttm>                <dbl>        <dbl>
##  1 22178529 2019-04-01 00:02:22 2019-04-01 00:09:48   6251          446
##  2 22178530 2019-04-01 00:03:02 2019-04-01 00:20:30   6226         1048
##  3 22178531 2019-04-01 00:11:07 2019-04-01 00:15:19   5649          252
##  4 22178532 2019-04-01 00:13:01 2019-04-01 00:18:58   4151          357
##  5 22178533 2019-04-01 00:19:26 2019-04-01 00:36:13   3270         1007
##  6 22178534 2019-04-01 00:19:39 2019-04-01 00:23:56   3123          257
##  7 22178535 2019-04-01 00:26:33 2019-04-01 00:35:41   6418          548
##  8 22178536 2019-04-01 00:29:48 2019-04-01 00:36:11   4513          383
##  9 22178537 2019-04-01 00:32:07 2019-04-01 01:07:44   3280         2137
## 10 22178538 2019-04-01 00:32:19 2019-04-01 01:07:39   5534         2120
## # i 1,108,153 more rows
## # i 7 more variables: from_station_id <dbl>, from_station_name <chr>,
## #   to_station_id <dbl>, to_station_name <chr>, usertype <chr>, gender <chr>,
## #   birthyear <dbl>
```

Inspect the data frames and look for incongruencies

```
str(q1_2019)
```

```
## spc_tbl_ [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)
```

```
## spc_tbl_ [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)
```

```
## spc_tbl_ [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
##  $ 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)
```

```
## spc_tbl_ [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>
```

Stack indavidual quarter's data frames into one big data frame

```
all_trips <- bind_rows(q1_2019,q2_2019,q3_2019,q4_2019)
```

Remove birthyear and gender fields as this data is not relevant to our analysis

```r
all_trips <- all_trips %>%
    select(-c(gender, birthyear, tripduration))
```

#== #STEP 3: CLEAN UP AND ADD DATA TO PREPARE FOR ANALYSIS #===========================
# Inspect the new table that has been created

```r
colnames(all_trips)
```

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

```r
nrow(all_trips)
```

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

```r
dim(all_trips)
```

```
## [1] 3818004       9
```

```r
head(all_trips)
```

```
## # A tibble: 6 x 9
##    trip_id start_time          end_time            bikeid from_station_id
##      <dbl> <dttm>              <dttm>               <dbl>           <dbl>
## 1 21742443 2019-01-01 00:04:37 2019-01-01 00:11:07   2167             199
## 2 21742444 2019-01-01 00:08:13 2019-01-01 00:15:34   4386              44
## 3 21742445 2019-01-01 00:13:23 2019-01-01 00:27:12   1524              15
## 4 21742446 2019-01-01 00:13:45 2019-01-01 00:43:28    252             123
## 5 21742447 2019-01-01 00:14:52 2019-01-01 00:20:56   1170             173
## 6 21742448 2019-01-01 00:15:33 2019-01-01 00:19:09   2437              98
## # i 4 more variables: from_station_name <chr>, to_station_id <dbl>,
## #   to_station_name <chr>, usertype <chr>
```

```r
tail(all_trips)
```

```
## # A tibble: 6 x 9
##    trip_id start_time          end_time            bikeid from_station_id
##      <dbl> <dttm>              <dttm>               <dbl>           <dbl>
## 1 25962899 2019-12-31 23:54:54 2020-01-01 00:22:02   5996             145
## 2 25962900 2019-12-31 23:56:13 2020-01-01 00:15:45   2196             112
## 3 25962901 2019-12-31 23:56:34 2020-01-01 00:22:08   4877              90
## 4 25962902 2019-12-31 23:57:05 2020-01-01 00:05:46    863             623
## 5 25962903 2019-12-31 23:57:11 2020-01-01 00:05:45   2637             623
## 6 25962904 2019-12-31 23:57:17 2019-12-31 23:59:18   5930             256
## # i 4 more variables: from_station_name <chr>, to_station_id <dbl>,
## #   to_station_name <chr>, usertype <chr>
```

```r
str(all_trips)
```

```
## tibble [3,818,004 x 9] (S3: tbl_df/tbl/data.frame)
##  $ trip_id          : num [1:3818004] 21742443 21742444 21742445 21742446 21742447 ...
##  $ start_time       : POSIXct[1:3818004], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ end_time         : POSIXct[1:3818004], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
##  $ bikeid           : num [1:3818004] 2167 4386 1524 252 1170 ...
##  $ from_station_id  : num [1:3818004] 199 44 15 123 173 98 98 211 150 268 ...
##  $ from_station_name: chr [1:3818004] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ to_station_id    : num [1:3818004] 84 624 644 176 35 49 49 142 148 141 ...
##  $ to_station_name  : chr [1:3818004] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "W
##  $ usertype         : chr [1:3818004] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
```

```r
summary(all_trips)
```

```
##     trip_id           start_time                 
##  Min.   :21742443   Min.   :2019-01-01 00:04:37.00  
##  1st Qu.:22873787   1st Qu.:2019-05-29 15:49:26.50  
##  Median :23962320   Median :2019-07-25 17:50:54.00  
##  Mean   :23915629   Mean   :2019-07-19 21:47:37.11  
##  3rd Qu.:24963703   3rd Qu.:2019-09-15 06:48:05.75  
##  Max.   :25962904   Max.   :2019-12-31 23:57:17.00  
##     end_time                          bikeid     from_station_id
##  Min.   :2019-01-01 00:11:07.00   Min.   :   1   Min.   :  1.0  
##  1st Qu.:2019-05-29 16:09:28.25   1st Qu.:1727   1st Qu.: 77.0  
##  Median :2019-07-25 18:12:23.00   Median :3451   Median :174.0  
##  Mean   :2019-07-19 22:11:47.56   Mean   :3380   Mean   :201.7  
##  3rd Qu.:2019-09-15 08:30:13.25   3rd Qu.:5046   3rd Qu.:289.0  
##  Max.   :2020-01-21 13:54:35.00   Max.   :6946   Max.   :673.0  
##  from_station_name  to_station_id   to_station_name      usertype        
##  Length:3818004     Min.   :  1.0   Length:3818004     Length:3818004    
##  Class :character   1st Qu.: 77.0   Class :character   Class :character  
##  Mode  :character   Median :174.0   Mode  :character   Mode  :character  
##                     Mean   :202.6                                        
##                     3rd Qu.:291.0                                        
##                     Max.   :673.0                                        
```

Remove rows where the ‘end_time’ or ‘start_time’ column contains the year 2020

```r
all_trips <- all_trips %>%
  filter(year(start_time) != 2020 & year(end_time) != 2020)
```

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

Before 2020, Divvy used different labels for these two types of riders... we will want to make our data frame consistent with their currennt nomenclature, including all column names aswell

Reanme all column names for better understanding and consistency with future data sets

```
all_trips <- rename(all_trips
            ,ride_id=trip_id
            ,started_at=start_time
            ,ended_at=end_time
            ,rideable_type=bikeid
            ,start_station_id=from_station_id
            ,start_station_name=from_station_name
            ,end_station_id=to_station_id
            ,end_station_name=to_station_name
            ,member_casual=usertype)
```

Begin by seeing how many observations fall under each usertype

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

```
##
##   Customer Subscriber
##     880619    2937356
```

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"))
```

Check to make sure the proper number of observations were reassigned

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

```
##
##  casual  member
##  880619 2937356
```

Add columns that list the date, month, day, and year of each ride, this will allow aggregated ride data for each month, day, or year…

The default format is yyyy-mm-dd

```
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")
```

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

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

Inspect the structure of the columns

```
str(all_trips)
```

```
## tibble [3,817,975 x 15] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : num [1:3817975] 21742443 21742444 21742445 21742446 21742447 ...
##  $ started_at        : POSIXct[1:3817975], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ ended_at          : POSIXct[1:3817975], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
##  $ rideable_type     : num [1:3817975] 2167 4386 1524 252 1170 ...
##  $ start_station_id  : num [1:3817975] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:3817975] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave
##  $ end_station_id    : num [1:3817975] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:3817975] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)"
##  $ member_casual     : chr [1:3817975] "member" "member" "member" "member" ...
##  $ date              : Date[1:3817975], format: "2019-01-01" "2019-01-01" ...
##  $ month             : chr [1:3817975] "01" "01" "01" "01" ...
##  $ day               : chr [1:3817975] "01" "01" "01" "01" ...
##  $ year               : chr [1:3817975] "2019" "2019" "2019" "2019" ...
##  $ day_of_week        : chr [1:3817975] "Tuesday" "Tuesday" "Tuesday" "Tuesday" ...
##  $ ride_length        : 'difftime' num [1:3817975] 6.5 7.35 13.8166666666667 29.7166666666667 ...
##   ..- attr(*, "units")= chr "mins"
```

Convert “ride_length” from Factor to numeric so calculations can be performed on the data

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

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

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

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

Remove “bad” data

The data frame includes a few hundred entries when bikes were taken out of docks nd checked for quality by Divvy or ride_length was negative.

Since data is being removed, a new version of the data frame (v2) will be created

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

Calculate IQR and define upper bound for outliers

```r
Q1 <- quantile(all_trips$ride_length, 0.25)
Q3 <- quantile(all_trips$ride_length, 0.75)
IQR <- Q3 - Q1
upper_bound <- Q3 + 1.5 * IQR
```

Identify outliers

```r
outliers <- all_trips$ride_length > upper_bound
```

Remove outliers

```r
all_clean <- all_trips[!outliers, ]
```

#======================================= #STEP 4: CONDUCT DESCRIPTIVE ANALYSIS #======================================= # Descriptive analysis on ride_length (all figures in seconds)

```r
mean(all_clean$ride_length)
```

```
## [1] 13.63319
```

```r
median(all_clean$ride_length)
```

```
## [1] 10.98333
```

```r
max(all_clean$ride_length)
```

```
## [1] 43.21667
```

```r
min(all_clean$ride_length)
```

```
## [1] 1.016667
```

```r
summary(all_clean$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.017   6.567  10.983  13.633  18.600  43.217
```

Compare members and casual users

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

```
##   all_clean$member_casual all_clean$ride_length
## 1                  casual              20.80183
## 2                  member              12.03463
```

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

```
##   all_clean$member_casual all_clean$ride_length
## 1                  casual              20.09167
## 2                  member               9.70000
```

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

```
##   all_clean$member_casual all_clean$ride_length
## 1                  casual              43.21667
## 2                  member              43.21667
```

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

```
##   all_clean$member_casual all_clean$ride_length
## 1                  casual              1.016667
## 2                  member              1.016667
```

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

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

```
##    all_clean$member_casual all_clean$day_of_week all_clean$ride_length
## 1                   casual                Friday              20.45114
## 2                   member                Friday              11.73499
## 3                   casual                Monday              20.64095
## 4                   member                Monday              11.87639
## 5                   casual              Saturday              21.84645
## 6                   member              Saturday              12.97707
## 7                   casual                Sunday              21.42252
## 8                   member                Sunday              12.84820
## 9                   casual              Thursday              20.01271
## 10                  member              Thursday              11.83005
## 11                  casual               Tuesday              19.93088
## 12                  member               Tuesday              11.84825
## 13                  casual             Wednesday              19.84644
## 14                  member             Wednesday              11.88958
```

Fix the order of the days of the week

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

Analyze ridership data by type and weekday

```
all_clean %>%
    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 ridesand average duration
    ,average_duration=mean(ride_length)) %>% # calculates the average duration
    arrange(member_casual, weekday) # sorts
```

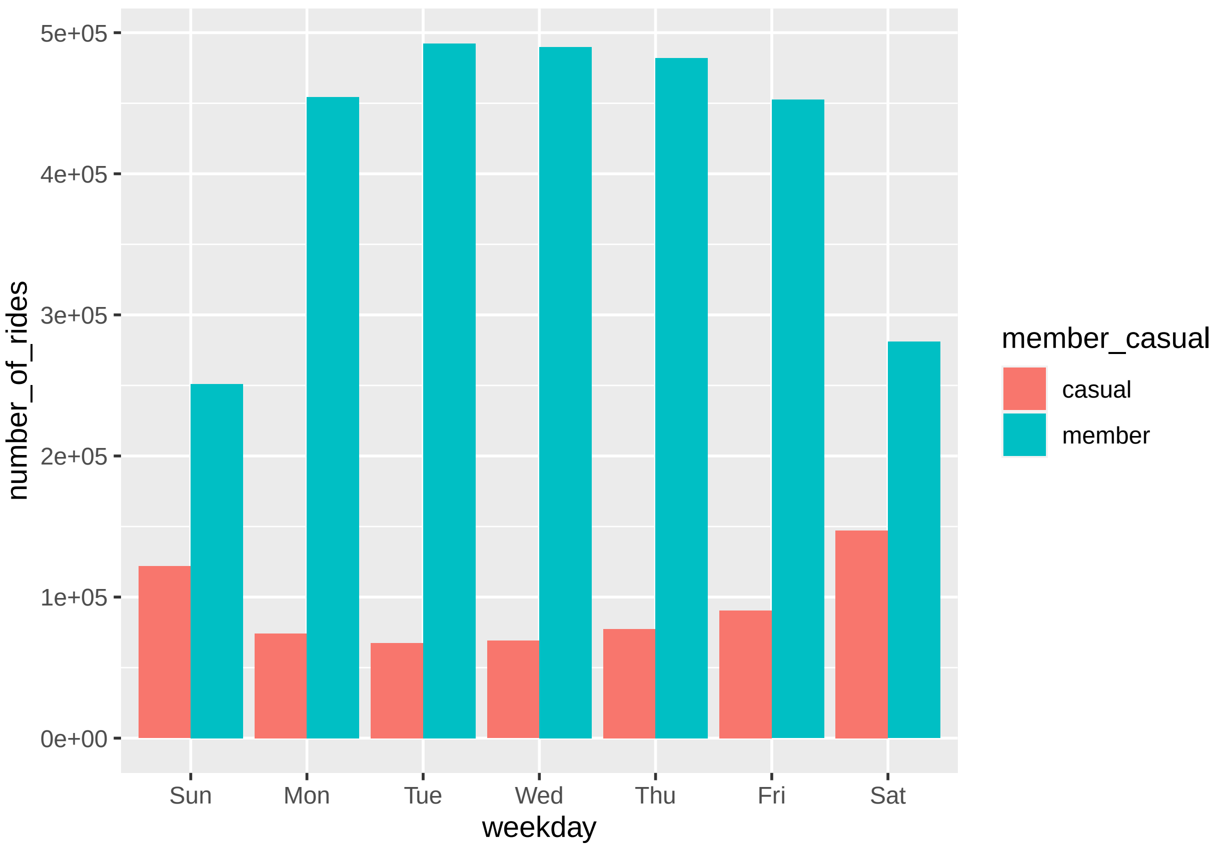
```
## ‘summarise()‘ has grouped output by ’member_casual’. You can override using the
## ‘.groups‘ argument.
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              121839             21.4
##  2 casual        Mon               74126             20.6
##  3 casual        Tue               67373             19.9
##  4 casual        Wed               69035             19.8
##  5 casual        Thu               77431             20.0
##  6 casual        Fri               90382             20.5
##  7 casual        Sat              147166             21.8
##  8 member        Sun              250913             12.8
##  9 member        Mon              454483             11.9
## 10 member        Tue              492446             11.8
## 11 member        Wed              489792             11.9
## 12 member        Thu              482020             11.8
## 13 member        Fri              452405             11.7
## 14 member        Sat              280966             13.0
```

Visualize the number of rides by rider type

```
all_clean %>%
    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")
```

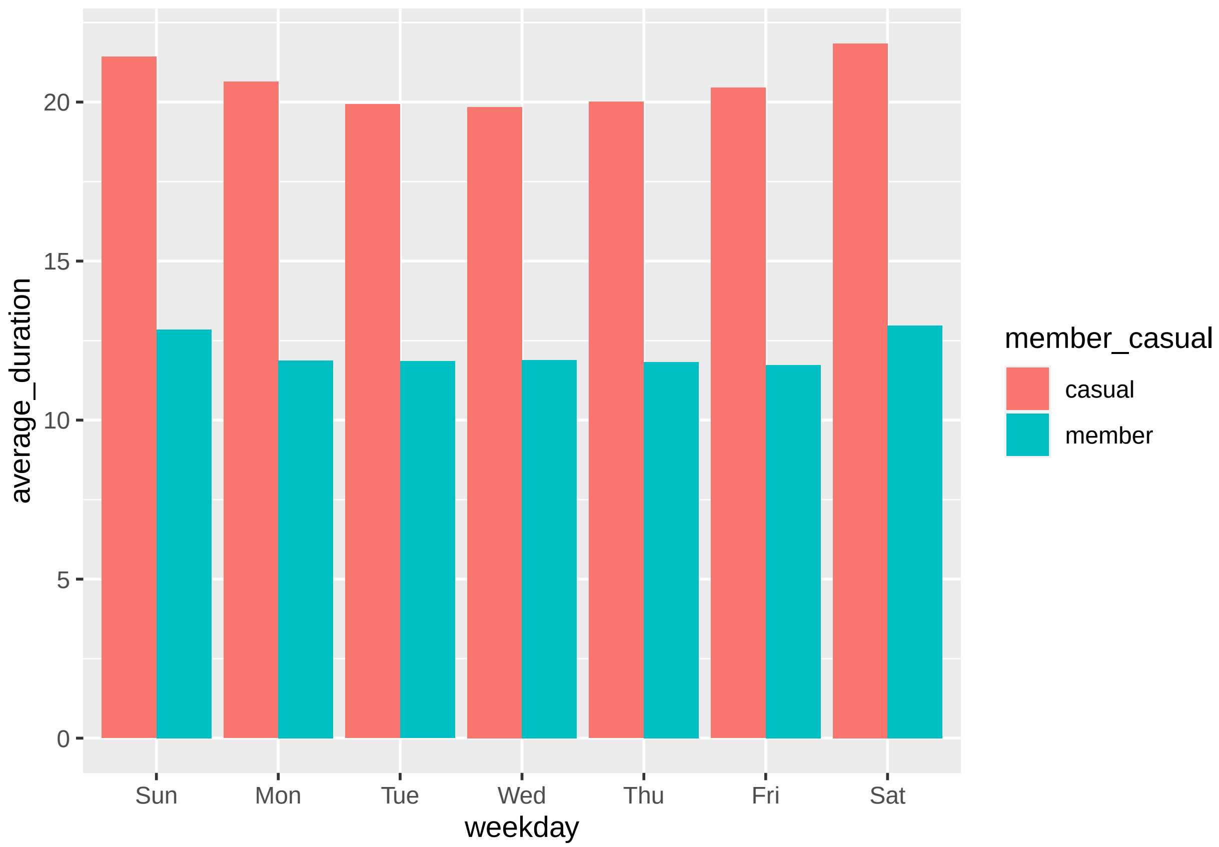
```
## ‘summarise()‘ has grouped output by ’member_casual’. You can override using the
## ‘.groups‘ argument.
```



Visualization for average duration

```
all_clean %>%
    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.
```



#== #STEP 5: EXPORT SUMMARY FILE FOR FURTHER ANALYSIS #======================================
# Create a csv file that will be visualized in Tableau

```
counts <- aggregate(all_clean$ride_length~all_clean$member_casual+
all_clean$day_of_week,FUN=mean)
write.csv(counts,file='~/Desktop/divvy_bike_data/avg_ride_length.csv')
write.csv(all_clean, file = '~/Desktop/divvy_bike_data/all_trips_clean.csv')
```