Case study: How does a bike-share navigate speedy success?

Bobby Aguirre

2024-09-24

Install and Load the packages

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

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

STEP 1: COLLECT DATA

```
# # Upload Divvy datasets (csv files) here
q1_2019 <- read.csv("C:/Users/ASUS VIVOBOOK/Downloads/Google Data Analytics_Projects/Case-Study-1_How-a-bike-shar
e-navigate-speedy-success/2019-2020 Q1/Divvy_Trips_2019_Q1.csv")
q2_2019 <- read.csv("C:/Users/ASUS VIVOBOOK/Downloads/Google Data Analytics_Projects/Case-Study-1_How-a-bike-shar
e-navigate-speedy-success/2019-2020 Q1/Divvy_Trips_2019_Q2.csv")
q3_2019 <- read.csv("C:/Users/ASUS VIVOBOOK/Downloads/Google Data Analytics_Projects/Case-Study-1_How-a-bike-shar
e-navigate-speedy-success/2019-2020 Q1/Divvy_Trips_2019_Q3.csv")
q4_2019 <- read.csv("C:/Users/ASUS VIVOBOOK/Downloads/Google Data Analytics_Projects/Case-Study-1_How-a-bike-shar
e-navigate-speedy-success/2019-2020 Q1/Divvy_Trips_2019_Q4.csv")
q1_2020 <- read.csv("C:/Users/ASUS VIVOBOOK/Downloads/Google Data Analytics_Projects/Case-Study-1_How-a-bike-shar
e-navigate-speedy-success/2019-2020 Q1/Divvy_Trips_2020_Q1.csv")
```

STEP 2: WRANGLE DATA AND COMBINE INTO A SINGLE FILE

```
# Compare column names each of the files
# While the names don't have to be in the same order, they DO need to match perfectly before
#we can use a command to join them into one file
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"
## [13] "ride_length"       "day_of_week"
```

```
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"
```

```
colnames(q3_2019)
```

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

```
colnames(q4_2019)
```

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

```
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"      "ride_length"        "day_of_week"
```

```
# Rename columns to make them consistent with q1_2020 (as this will be the supposed going-forward table design fo
r Divvy)
(q1_2019 <- rename(q1_2019
,ride_id = trip_id
,rideable_type = bikeid
,started_at = start_time
,ended_at = end_time
,start_station_name = from_station_name
,start_station_id = from_station_id
,end_station_name = to_station_name
,end_station_id = to_station_id
,member_casual = usertype
))
```

```
(q2_2019 <- rename(q2_2019
,ride_id = "X01...Rental.Details.Rental.ID"
,rideable_type = "X01...Rental.Details.Bike.ID"
,started_at = "X01...Rental.Details.Local.Start.Time"
,ended_at = "X01...Rental.Details.Local.End.Time"
,start_station_name = "X03...Rental.Start.Station.Name"
,start_station_id = "X03...Rental.Start.Station.ID"
,end_station_name = "X02...Rental.End.Station.Name"
,end_station_id = "X02...Rental.End.Station.ID"
,member_casual = "User.Type"
))
```

```
(q3_2019 <- rename(q3_2019
,ride_id = trip_id
,rideable_type = bikeid
,started_at = start_time
,ended_at = end_time
,start_station_name = from_station_name
,start_station_id = from_station_id
,end_station_name = to_station_name
,end_station_id = to_station_id
,member_casual = usertype
))
```

```
(q4_2019 <- rename(q4_2019
,ride_id = trip_id
,rideable_type = bikeid
,started_at = start_time
,ended_at = end_time
,start_station_name = from_station_name
,start_station_id = from_station_id
,end_station_name = to_station_name
,end_station_id = to_station_id
,member_casual = usertype
))
```

```
# Inspect the dataframes and look for incongruencies
str(q1_2019)
```

```
## 'data.frame':    365069 obs. of  14 variables:
##  $ ride_id           : int  21742443 21742444 21742445 21742446 21742447 21742448 21742449 21742450 21742451 2
1742452 ...
##  $ started_at        : chr  "01/01/2019 0:04" "01/01/2019 0:08" "01/01/2019 0:13" "01/01/2019 0:13" ...
##  $ ended_at          : chr  "01/01/2019 0:11" "01/01/2019 0:15" "01/01/2019 0:27" "01/01/2019 0:43" ...
##  $ rideable_type     : int  2167 4386 1524 252 1170 2437 2708 2796 6205 3939 ...
##  $ tripduration      : chr  "390" "441" "829" "1,783.00" ...
##  $ start_station_id  : int  199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr  "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave & 18th St" "Californ
ia Ave & Milwaukee Ave" ...
##  $ end_station_id    : int  84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr  "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "Western Ave & Fillmo
re St (*)" "Clark St & Elm St" ...
##  $ member_casual     : chr  "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
##  $ gender            : chr  "Male" "Female" "Female" "Male" ...
##  $ birthyear         : int  1989 1990 1994 1993 1994 1983 1984 1990 1995 1996 ...
##  $ ride_length       : chr  "12:06:30 AM" "12:07:21 AM" "12:13:49 AM" "12:29:43 AM" ...
##  $ day_of_week       : int  3 3 3 3 3 3 3 3 3 ...
```

```
str(q2_2019)
```

```
## 'data.frame':    1108163 obs. of  12 variables:
##  $ ride_id                                           : int  22178529 22178530 22178531 22178532 22178533 221785
34 22178535 22178536 22178537 22178538 ...
##  $ started_at                                        : chr  "2019-04-01 00:02:22" "2019-04-01 00:03:02" "2019-0
4-01 00:11:07" "2019-04-01 00:13:01" ...
##  $ ended_at                                          : chr  "2019-04-01 00:09:48" "2019-04-01 00:20:30" "2019-0
4-01 00:15:19" "2019-04-01 00:18:58" ...
##  $ rideable_type                                     : int  6251 6226 5649 4151 3270 3123 6418 4513 3280 5534 .
..
##  $ 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" "LaSalle
St & Jackson Blvd" "McClurg Ct & Illinois 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 & Roosevelt
Rd" "Canal St & Madison St" "Kingsbury St & Kinzie St" ...
##  $ member_casual                                     : chr  "Subscriber" "Subscriber" "Subscriber" "Subscriber"
...
##  $ Member.Gender                                     : chr  "Male" "Female" "Male" "Male" ...
##  $ X05...Member.Details.Member.Birthday.Year         : int  1975 1984 1990 1993 1992 1999 1969 1991 NA NA ...
```

```
str(q3_2019)
```

```
## 'data.frame':    1640718 obs. of  12 variables:
##  $ ride_id           : int  23479388 23479389 23479390 23479391 23479392 23479393 23479394 23479395 23479396 2
3479397 ...
##  $ started_at        : chr  "2019-07-01 00:00:27" "2019-07-01 00:01:16" "2019-07-01 00:01:48" "2019-07-01 00:0
2:07" ...
##  $ ended_at          : chr  "2019-07-01 00:20:41" "2019-07-01 00:18:44" "2019-07-01 00:27:42" "2019-07-01 00:2
7:10" ...
##  $ 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 & Fullerton Pkw
y" "Lakeview Ave & Fullerton Pkwy" ...
##  $ 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 & Webster Ave" "L
arrabee St & Webster Ave" ...
##  $ member_casual     : chr  "Subscriber" "Customer" "Customer" "Customer" ...
##  $ gender            : chr  "Male" "" "" "" ...
##  $ birthyear         : int  1992 NA NA NA NA 1990 NA NA NA NA ...
```

```
str(q4_2019)
```

```
## 'data.frame':    704054 obs. of  12 variables:
##  $ ride_id           : int  25223640 25223641 25223642 25223643 25223644 25223645 25223646 25223647 25223648 2
5223649 ...
##  $ started_at        : chr  "2019-10-01 00:01:39" "2019-10-01 00:02:16" "2019-10-01 00:04:32" "2019-10-01 00:0
4:32" ...
##  $ ended_at          : chr  "2019-10-01 00:17:20" "2019-10-01 00:06:34" "2019-10-01 00:18:43" "2019-10-01 00:4
3:43" ...
##  $ 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" "Milwaukee Ave & G
rand Ave" "Lakeview Ave & Fullerton Pkwy" ...
##  $ 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" "Kedzie
Ave & Palmer Ct" ...
##  $ 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(q1_2020)
```

```
## 'data.frame':    426887 obs. of  15 variables:
##  $ ride_id           : chr  "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A388DAC6ABF313" ...
##  $ rideable_type     : chr  "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : chr  "21/01/2020 20:06" "30/01/2020 14:22" "09/01/2020 19:29" "06/01/2020 16:17" ...
##  $ ended_at          : chr  "21/01/2020 20:14" "30/01/2020 14:26" "09/01/2020 19:32" "06/01/2020 16:25" ...
##  $ start_station_name: chr  "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway & Belmont Ave" "Cla
rk St & Randolph St" ...
##  $ 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 & Belmont Ave
" "Fairbanks Ct & Grand Ave" ...
##  $ 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" ...
##  $ ride_length       : chr  "12:07:31 AM" "12:03:43 AM" "12:02:51 AM" "12:08:49 AM" ...
##  $ day_of_week       : int  3 5 5 2 5 6 6 6 6 6 ...
```

```
# Convert ride_id and rideable_type to character so that they can stack correctly
q1_2019 <- mutate(q1_2019, ride_id = as.character(ride_id),rideable_type = as.character(rideable_type), started_a
t = as_datetime(started_at), ended_at = as_datetime(ended_at))
# Convert ride_id and rideable_type to character so that they can stack correctly
q2_2019 <- mutate(q2_2019, ride_id = as.character(ride_id)
,rideable_type = as.character(rideable_type), started_at = as_datetime(started_at), ended_at = as_datetime(ended_
at))
# Convert ride_id and rideable_type to character so that they can stack correctly
q3_2019 <- mutate(q3_2019, ride_id = as.character(ride_id)
,rideable_type = as.character(rideable_type), started_at = as_datetime(started_at), ended_at = as_datetime(ended_
at))
# 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), started_at = as_datetime(started_at), ended_at = as_datetime(ended_
at))
q1_2020 <- mutate(q1_2020, ride_id = as.character(ride_id)
,rideable_type = as.character(rideable_type), started_at = as_datetime(started_at), ended_at = as_datetime(ended_
at))
```

```
# Stack individual quarter's data frames into one big data frame
all_trips <- bind_rows(q1_2019, q2_2019, q3_2019, q4_2019, q1_2020)
```

```
# 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, birthyear, gender, "tripduration"))
```

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

```
# Inspect the new table that has been created
colnames(all_trips) #List of column names
```

```
##  [1] "ride_id"
##  [2] "started_at"
##  [3] "ended_at"
##  [4] "rideable_type"
##  [5] "start_station_id"
##  [6] "start_station_name"
##  [7] "end_station_id"
##  [8] "end_station_name"
##  [9] "member_casual"
## [10] "ride_length"
## [11] "day_of_week"
## [12] "X01...Rental.Details.Duration.In.Seconds.Uncapped"
## [13] "Member.Gender"
## [14] "X05...Member.Details.Member.Birthday.Year"
```

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

```
## [1] 4244891
```

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

```
## [1] 4244891      14
```

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

```
##    ride_id          started_at            ended_at rideable_type
## 1 21742443 2001-01-20 19:00:04 2001-01-20 19:00:11          2167
## 2 21742444 2001-01-20 19:00:08 2001-01-20 19:00:15          4386
## 3 21742445 2001-01-20 19:00:13 2001-01-20 19:00:27          1524
## 4 21742446 2001-01-20 19:00:13 2001-01-20 19:00:43           252
## 5 21742447 2001-01-20 19:00:14 2001-01-20 19:00:20          1170
## 6 21742448 2001-01-20 19:00:15 2001-01-20 19:00:19          2437
##   start_station_id                  start_station_name end_station_id
## 1              199              Wabash Ave & Grand Ave             84
## 2               44              State St & Randolph St            624
## 3               15                Racine Ave & 18th St            644
## 4              123      California Ave & Milwaukee Ave            176
## 5              173 Mies van der Rohe Way & Chicago Ave             35
## 6               98          LaSalle St & Washington St             49
##                  end_station_name member_casual ride_length day_of_week
## 1      Milwaukee Ave & Grand Ave    Subscriber 12:06:30 AM           3
## 2 Dearborn St & Van Buren St (*)    Subscriber 12:07:21 AM           3
## 3  Western Ave & Fillmore St (*)    Subscriber 12:13:49 AM           3
## 4              Clark St & Elm St    Subscriber 12:29:43 AM           3
## 5        Streeter Dr & Grand Ave    Subscriber 12:06:04 AM           3
## 6        Dearborn St & Monroe St    Subscriber 12:03:36 AM           3
##   X01...Rental.Details.Duration.In.Seconds.Uncapped Member.Gender
## 1                                              <NA>          <NA>
## 2                                              <NA>          <NA>
## 3                                              <NA>          <NA>
## 4                                              <NA>          <NA>
## 5                                              <NA>          <NA>
## 6                                              <NA>          <NA>
##   X05...Member.Details.Member.Birthday.Year
## 1                                        NA
## 2                                        NA
## 3                                        NA
## 4                                        NA
## 5                                        NA
## 6                                        NA
```

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

```
## 'data.frame':    4244891 obs. of  14 variables:
##  $ ride_id                                          : chr  "21742443" "21742444" "21742445" "21742446" ...
##  $ started_at                                       : POSIXct, format: "2001-01-20 19:00:04" "2001-01-20 19:00
:08" ...
##  $ ended_at                                         : POSIXct, format: "2001-01-20 19:00:11" "2001-01-20 19:00
:15" ...
##  $ rideable_type                                    : chr  "2167" "4386" "1524" "252" ...
##  $ start_station_id                                 : int  199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name                               : chr  "Wabash Ave & Grand Ave" "State St & Randolph St" "
Racine Ave & 18th St" "California Ave & Milwaukee Ave" ...
##  $ end_station_id                                   : int  84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name                                 : chr  "Milwaukee Ave & Grand Ave" "Dearborn St & Van Bure
n St (*)" "Western Ave & Fillmore St (*)" "Clark St & Elm St" ...
##  $ member_casual                                    : chr  "Subscriber" "Subscriber" "Subscriber" "Subscriber"
...
##  $ ride_length                                      : chr  "12:06:30 AM" "12:07:21 AM" "12:13:49 AM" "12:29:43
AM" ...
##  $ day_of_week                                      : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ X01...Rental.Details.Duration.In.Seconds.Uncapped: chr  NA NA NA NA ...
##  $ Member.Gender                                    : chr  NA NA NA NA ...
##  $ X05...Member.Details.Member.Birthday.Year        : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
summary(all_trips) #Statistical summary of data. Mainly for numerics
```

```
##    ride_id            started_at
##  Length:4244891     Min.   :2001-01-20 19:00:04.00
##  Class :character   1st Qu.:2019-05-24 16:36:19.00
##  Mode  :character   Median :2019-07-28 06:46:12.00
##                     Mean   :2018-09-18 10:18:35.01
##                     3rd Qu.:2019-09-24 11:41:29.00
##                     Max.   :2031-03-20 20:23:51.00
##
##     ended_at                      rideable_type      start_station_id
##  Min.   :2001-01-20 19:00:11.00   Length:4244891     Min.   :  1.0
##  1st Qu.:2019-05-24 16:54:42.50   Class :character   1st Qu.: 77.0
##  Median :2019-07-28 07:43:18.00   Mode  :character   Median :174.0
##  Mean   :2018-09-18 11:12:46.98                      Mean   :202.5
##  3rd Qu.:2019-09-24 12:08:20.00                      3rd Qu.:289.0
##  Max.   :2031-03-20 20:23:58.00                      Max.   :675.0
##
##  start_station_name end_station_id  end_station_name   member_casual
##  Length:4244891     Min.   :  1.0   Length:4244891     Length:4244891
##  Class :character   1st Qu.: 77.0   Class :character   Class :character
##  Mode  :character   Median :174.0   Mode  :character   Mode  :character
##                     Mean   :203.3
##                     3rd Qu.:291.0
##                     Max.   :675.0
##                     NA's   :1
##  ride_length         day_of_week
##  Length:4244891     Min.   :1
##  Class :character   1st Qu.:3
##  Mode  :character   Median :4
##                     Mean   :4
##                     3rd Qu.:5
##                     Max.   :7
##                     NA's   :3452935
##  X01...Rental.Details.Duration.In.Seconds.Uncapped Member.Gender
##  Length:4244891                                    Length:4244891
##  Class :character                                  Class :character
##  Mode  :character                                  Mode  :character
##
##
##
##
##  X05...Member.Details.Member.Birthday.Year
##  Min.   :1759
##  1st Qu.:1979
##  Median :1987
##  Mean   :1984
##  3rd Qu.:1992
##  Max.   :2014
##  NA's   :3317681
```

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 2020Q1 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”

Before 2020, Divvy used different labels for these two types of riders … we will want to make our dataframe consistent with their current nomenclature

N.B.: “Level” is a special property of a column that is retained even if a subset does not contain any values from a specific level

Begin by seeing how many observations fall under each usertype table(all_trips$member_casual)

```
# Reassign to the desired values (we will go with the current 2020 labels)
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
##  929117 3315774
```

```
# Add columns that list the date, month, day, and year of each ride
# This will allow us to aggregate ride data for each month, day, or year ... before completing these operations w
e could only aggregate at the ride level
# https://www.statmethods.net/input/dates.html more on date formats in R found at that link
all_trips$date <- as.Date(all_trips$started_at) #The default format is yyyy-mm-dd
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)
# https://stat.ethz.ch/R-manual/R-devel/library/base/html/difftime.html
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$started_at)
```

```
# Inspect the structure of the columns
str(all_trips)
```

```
## 'data.frame':    4244891 obs. of  18 variables:
##  $ ride_id                                          : chr  "21742443" "21742444" "21742445" "21742446" ...
##  $ started_at                                       : POSIXct, format: "2001-01-20 19:00:04" "2001-01-20 19:00
:08" ...
##  $ ended_at                                         : POSIXct, format: "2001-01-20 19:00:11" "2001-01-20 19:00
:15" ...
##  $ rideable_type                                    : chr  "2167" "4386" "1524" "252" ...
##  $ start_station_id                                 : int  199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name                               : chr  "Wabash Ave & Grand Ave" "State St & Randolph St" "
Racine Ave & 18th St" "California Ave & Milwaukee Ave" ...
##  $ end_station_id                                   : int  84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name                                 : chr  "Milwaukee Ave & Grand Ave" "Dearborn St & Van Bure
n St (*)" "Western Ave & Fillmore St (*)" "Clark St & Elm St" ...
##  $ member_casual                                    : chr  "member" "member" "member" "member" ...
##  $ ride_length                                      : 'difftime' num  7 7 14 30 ...
##   ..- attr(*, "units")= chr "secs"
##  $ day_of_week                                      : chr  "Saturday" "Saturday" "Saturday" "Saturday" ...
##  $ X01...Rental.Details.Duration.In.Seconds.Uncapped: chr  NA NA NA NA ...
##  $ Member.Gender                                    : chr  NA NA NA NA ...
##  $ X05...Member.Details.Member.Birthday.Year        : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ date                                             : Date, format: "2001-01-20" "2001-01-20" ...
##  $ month                                            : chr  "01" "01" "01" "01" ...
##  $ day                                              : chr  "20" "20" "20" "20" ...
##  $ year                                             : chr  "2001" "2001" "2001" "2001" ...
```

```
# Convert "ride_length" from Factor to numeric so we can run calculations on the data
is.factor(all_trips$ride_length)
```

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

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

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

```
# Remove "bad" data
# The dataframe includes a few hundred entries when bikes were taken out of docks and checked for quality by Divv
y or ride_length was negative
# We will create a new version of the dataframe (v2) since data is being removed
# https://www.datasciencemadesimple.com/delete-or-drop-rows-in-r-with-conditions-2/
all_trips_v2 <- all_trips[!(all_trips$start_station_name == "HQ QR" | all_trips$ride_length<0),]
```

STEP 4: CONDUCT DESCRIPTIVE ANALYSIS

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

```
## [1] 22435.9
```

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

```
## [1] 578
```

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

```
## [1] 946684785
```

```
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       246       578     22436      1147 946684785
```

```
# 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                 54170.58
## 2                     member                 13580.14
```

```
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                     1452
## 2                     member                      465
```

```
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                946684785
## 2                     member                820454515
```

```
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                37014.87
## 2                      member                   Friday                12801.98
## 3                      casual                   Monday                53219.72
## 4                      member                   Monday                10533.52
## 5                      casual                 Saturday                31501.47
## 6                      member                 Saturday                16450.44
## 7                      casual                   Sunday                48996.56
## 8                      member                   Sunday                15676.10
## 9                      casual                 Thursday                70679.61
## 10                     member                 Thursday                11897.48
## 11                     casual                  Tuesday               103025.64
## 12                     member                  Tuesday                12493.09
## 13                     casual                Wednesday                66413.53
## 14                     member                Wednesday                16627.89
```

```
# 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", "Wednesday", "Thursday", "Friday", "Saturday"))
```

```
# 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                48996.56
## 2                      member                   Sunday                15676.10
## 3                      casual                   Monday                53219.72
## 4                      member                   Monday                10533.52
## 5                      casual                  Tuesday               103025.64
## 6                      member                  Tuesday                12493.09
## 7                      casual                Wednesday                66413.53
## 8                      member                Wednesday                16627.89
## 9                      casual                 Thursday                70679.61
## 10                     member                 Thursday                11897.48
## 11                     casual                   Friday                37014.87
## 12                     member                   Friday                12801.98
## 13                     casual                 Saturday                31501.47
## 14                     member                 Saturday                16450.44
```

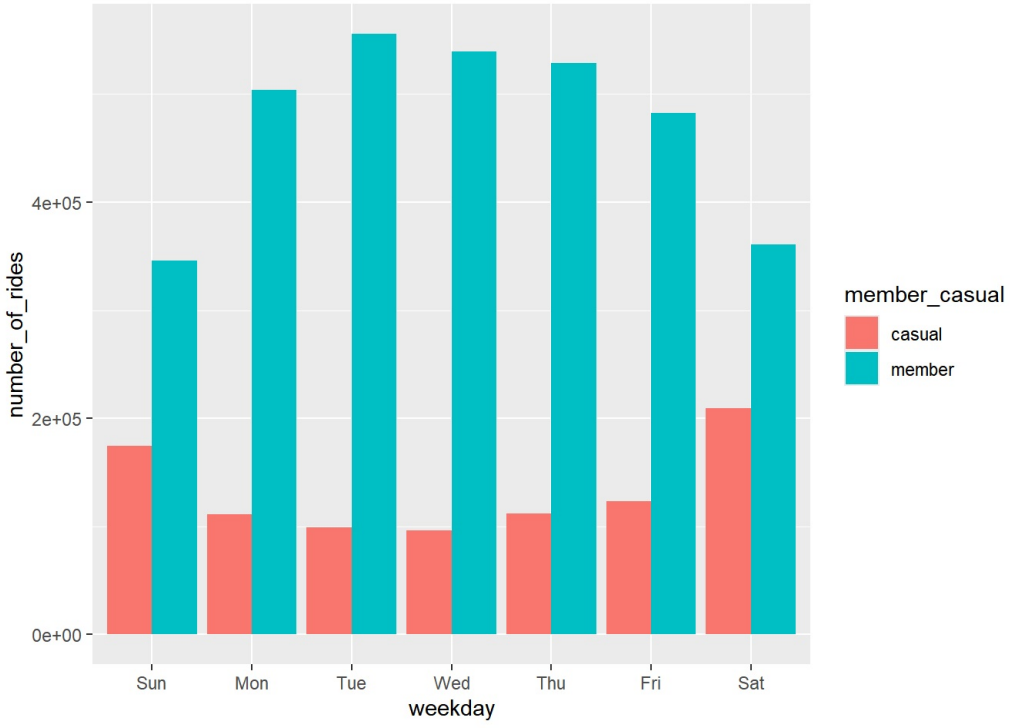
```
# analyze ridership data by type and weekday
all_trips_v2 %>%
mutate(weekday = wday(started_at, label = TRUE)) %>% #creates weekday field using wday()
  group_by(member_casual, weekday) %>% #groups by usertype and weekday
    summarise(number_of_rides = n() #calculates the number of rides and average 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 × 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              174526           48997.
##  2 casual        Mon              111107           53220.
##  3 casual        Tue               98968          103026.
##  4 casual        Wed               96169           66414.
##  5 casual        Thu              112120           70680.
##  6 casual        Fri              123340           37015.
##  7 casual        Sat              209044           31501.
##  8 member        Sun              345888           15676.
##  9 member        Mon              503598           10534.
## 10 member        Tue              555671           12493.
## 11 member        Wed              539099           16628.
## 12 member        Thu              528655           11897.
## 13 member        Fri              482198           12802.
## 14 member        Sat              360614           16450.
```

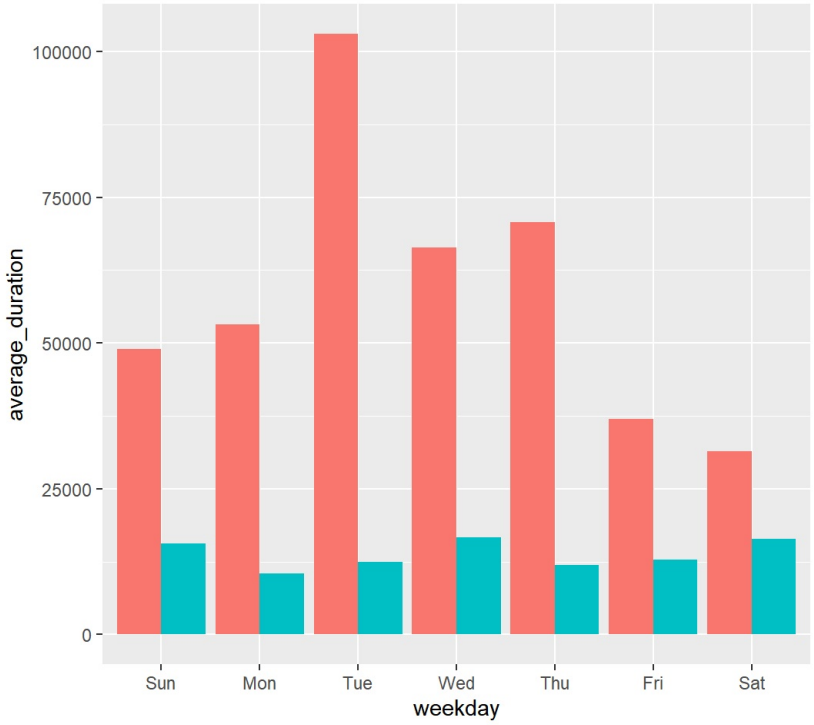
```
# Let's visualize the number of rides by rider type
all_trips_v2 %>%
mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%

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.
```




```
# 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.
```



STEP 5: EXPORT SUMMARY FILE

FOR FURTHER ANALYSIS

```
# Create a csv file that we will visualize in Excel, Tableau, or my presentation software
# N.B.: This file location is for a Mac. If you are working on a PC, change the file location accordingly (most l
ikely "C:\Users\YOUR_USERNAME\Desktop\...") to export the data. You can read more here: https://datatofish.com/ex
port-dataframe-to-csv-in-r/
counts <- aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual +
all_trips_v2$day_of_week, FUN = mean)
write.csv(counts, file = 'avg_ride_length.csv')
```