Chicago Divvy Case Study

Chris Fair

2022-02-28

Divvy Customer Usage analysis

This analysis is based on the Divvy case study "'Sophisticated, Clear, and Polished': Divvy and Data Visualization" written by Kevin Hartman (found here: https://artscience.blog/home/divvy-dataviz-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?"

Including required packages.

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

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)  #helps wrangle date attributes
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(ggplot2)  #helps visualize data
library(readxl) #importing excel files
library(tinytex) #knitting to PDF
```

Step 1: Importing Data

The data for this study was made publicly available by Divvy via: https://divvy-tripdata.s3.amazonaws.com/index.html

Throughout this report, *membership riders* will refer to riders who have a membership, while *casual riders* will refer to riders who pay by the ride.

```
jan_2021 <- read_excel("202101-divvy-tripdata.xlsx")
feb_2021 <- read_excel("202102-divvy-tripdata.xlsx")
march_2021 <- read_excel("202103-divvy-tripdata.xlsx")
april_2021 <- read_excel("202104-divvy-tripdata.xlsx")
may_2021 <- read_excel("202105-divvy-tripdata.xlsx")
june_2021 <- read_excel("202106-divvy-tripdata.xlsx")
july_2021 <- read_excel("202107-divvy-tripdata.xlsx")
aug_2021 <- read_excel("202108-divvy-tripdata.xlsx")
sept_2021 <- read_excel("202109-divvy-tripdata.xlsx")
oct_2021 <- read_excel("202110-divvy-tripdata.xlsx")
nov_2021 <- read_excel("202111-divvy-tripdata.xlsx")
dec_2021 <- read_excel("202112-divvy-tripdata.xlsx")
jan_2022 <- read_excel("202201-divvy-tripdata.xlsx")
```

Step 2: Wrangling Data

Now that the data has been imported, it needs to be combined into one large dataframe. To do this, first manually check each dataframes column names to make sure they are the same, then append all the rows together.

```
colnames(jan_2021)
```

```
##  [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"
```

```
colnames(feb_2021)
```

```
##  [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"
```

```
colnames(march_2021)
```

```
##  [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"
```

```
colnames(april_2021)
```

```
##  [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"
```

```
colnames(may_2021)
```

```
##  [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"
```

```
colnames(june_2021)
```

```
##  [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"
```

```
colnames(july_2021)
```

```
##  [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"
```

```
colnames(aug_2021)
```

```
##  [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"
```

```
colnames(sept_2021)
```

```
##  [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"
```

```r
colnames(oct_2021)
```

```
##  [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"
```

```r
colnames(nov_2021)
```

```
##  [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"
```

```r
colnames(dec_2021)
```

```
##  [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"
```

```r
colnames(jan_2022)
```

```
##  [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"
```

```r
all_trips <- bind_rows(jan_2021, feb_2021, march_2021, april_2021,may_2021,june_2021,july_2021,aug_2021
```

Step 3: Cleaning Up Data and Preparing for Analysis

Now that all of the data is in one dataframe, it's time to inspect that data frame.

```r
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"     "start_lat"         
## [10] "start_lng"          "end_lat"            "end_lng"           
## [13] "member_casual"      "ride_length"        "day_of_week"
```

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

```
## [1] 5698833
```

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

```
## [1] 5698833      15
```

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

```
## # A tibble: 6 x 15
##   ride_id rideable_type started_at          ended_at            start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 E19E6F~ electric_bike 2021-01-23 16:14:19 2021-01-23 16:24:44 California Ave ~
## 2 DC88F2~ electric_bike 2021-01-27 18:43:08 2021-01-27 18:47:12 California Ave ~
## 3 EC45C9~ electric_bike 2021-01-21 22:35:54 2021-01-21 22:37:14 California Ave ~
## 4 4FA453~ electric_bike 2021-01-07 13:31:13 2021-01-07 13:42:55 California Ave ~
## 5 BE5E8E~ electric_bike 2021-01-23 02:24:02 2021-01-23 02:24:45 California Ave ~
## 6 5D8969~ electric_bike 2021-01-09 14:24:07 2021-01-09 15:17:54 California Ave ~
## # ... with 10 more variables: start_station_id <chr>, end_station_name <chr>,
## #   end_station_id <chr>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>,
## #   end_lng <dbl>, member_casual <chr>, ride_length <dttm>, day_of_week <dbl>
```

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

```
## # A tibble: 6 x 15
##   ride_id rideable_type started_at          ended_at            start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 9C80CD~ electric_bike 2022-01-09 18:56:50 2022-01-09 19:02:50 Broadway & Wave~
## 2 8788DA~ electric_bike 2022-01-18 12:36:48 2022-01-18 12:46:19 Clinton St & Wa~
## 3 C6C3B6~ electric_bike 2022-01-27 11:00:06 2022-01-27 11:02:40 Racine Ave & Ra~
## 4 CA281A~ electric_bike 2022-01-10 16:14:51 2022-01-10 16:20:58 Broadway & Wave~
## 5 44E348~ electric_bike 2022-01-19 13:22:11 2022-01-19 13:24:27 Racine Ave & Ra~
## 6 E477C5~ electric_bike 2022-01-13 17:24:43 2022-01-13 17:28:14 Clinton St & Wa~
## # ... with 10 more variables: start_station_id <chr>, end_station_name <chr>,
## #   end_station_id <chr>, start_lat <dbl>, start_lng <dbl>, end_lat <dbl>,
## #   end_lng <dbl>, member_casual <chr>, ride_length <dttm>, day_of_week <dbl>
```

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

```
## tibble [5,698,833 x 15] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5698833] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA4
##  $ rideable_type     : chr [1:5698833] "electric_bike" "electric_bike" "electric_bike" "electric_bike
##  $ started_at        : POSIXct[1:5698833], format: "2021-01-23 16:14:19" "2021-01-27 18:43:08" ...
##  $ ended_at          : POSIXct[1:5698833], format: "2021-01-23 16:24:44" "2021-01-27 18:47:12" ...
##  $ start_station_name: chr [1:5698833] "California Ave & Cortez St" "California Ave & Cortez St" "Cal
##  $ start_station_id  : chr [1:5698833] "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr [1:5698833] NA NA NA NA ...
##  $ end_station_id    : chr [1:5698833] NA NA NA NA ...
##  $ start_lat         : num [1:5698833] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:5698833] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:5698833] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:5698833] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:5698833] "member" "member" "member" "member" ...
##  $ ride_length       : POSIXct[1:5698833], format: "1899-12-31 00:10:25" "1899-12-31 00:04:04" ...
##  $ day_of_week       : num [1:5698833] 7 4 5 5 7 7 2 5 7 1 ...
```

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

```
##    ride_id          rideable_type        started_at
##  Length:5698833     Length:5698833     Min.   :2021-01-01 00:02:05
##  Class :character   Class :character   1st Qu.:2021-06-08 14:51:48
##  Mode  :character   Mode  :character   Median :2021-08-03 01:18:38
##                                        Mean   :2021-08-01 10:30:05
##                                        3rd Qu.:2021-09-27 15:07:19
##                                        Max.   :2022-01-31 23:58:37
##
##     ended_at                   start_station_name start_station_id
##  Min.   :2021-01-01 00:08:39   Length:5698833     Length:5698833
##  1st Qu.:2021-06-08 15:16:41   Class :character   Class :character
##  Median :2021-08-03 02:12:44   Mode  :character   Mode  :character
##  Mean   :2021-08-01 10:51:53
##  3rd Qu.:2021-09-27 15:27:45
##  Max.   :2022-02-01 01:46:16
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:5698833     Length:5698833     Min.   :41.64   Min.   :-87.84
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64
##                                        Mean   :41.90   Mean   :-87.65
##                                        3rd Qu.:41.93   3rd Qu.:-87.63
##                                        Max.   :45.64   Max.   :-73.80
##
##     end_lat          end_lng       member_casual
##  Min.   :41.39   Min.   :-88.97   Length:5698833
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character
##  Median :41.90   Median :-87.64   Mode  :character
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.17   Max.   :-87.49
##  NA’s   :4857    NA’s   :4857
##   ride_length                   day_of_week
##  Min.   :1899-12-30 23:01:58   Min.   :1.000
##  1st Qu.:1899-12-31 00:06:41   1st Qu.:2.000
##  Median :1899-12-31 00:11:54   Median :4.000
##  Mean   :1899-12-31 00:21:49   Mean   :4.109
##  3rd Qu.:1899-12-31 00:21:37   3rd Qu.:6.000
##  Max.   :1900-02-07 20:24:09   Max.   :7.000
##
```

Adding additional columns to aggregate the data by.

```
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$hour <- format(all_trips$started_at,"%H")
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A")
```

Adding a ride_length calculation and reformatting it to use in later calculations.

```
# https://stat.ethz.ch/R-manual/R-devel/library/base/html/difftime.html
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$started_at)
is.factor(all_trips$ride_length)
```

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

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

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

```
str(all_trips)
```

```
## tibble [5,698,833 x 20] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5698833] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA
##  $ rideable_type     : chr [1:5698833] "electric_bike" "electric_bike" "electric_bike" "electric_bik
##  $ started_at        : POSIXct[1:5698833], format: "2021-01-23 16:14:19" "2021-01-27 18:43:08" ...
##  $ ended_at          : POSIXct[1:5698833], format: "2021-01-23 16:24:44" "2021-01-27 18:47:12" ...
##  $ start_station_name: chr [1:5698833] "California Ave & Cortez St" "California Ave & Cortez St" "Ca
##  $ start_station_id  : chr [1:5698833] "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr [1:5698833] NA NA NA NA ...
##  $ end_station_id    : chr [1:5698833] NA NA NA NA ...
##  $ start_lat         : num [1:5698833] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:5698833] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:5698833] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:5698833] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:5698833] "member" "member" "member" "member" ...
##  $ ride_length       : num [1:5698833] 625 244 80 702 43 ...
##  $ day_of_week       : chr [1:5698833] "Saturday" "Wednesday" "Thursday" "Thursday" ...
##  $ date              : Date[1:5698833], format: "2021-01-23" "2021-01-27" ...
##  $ month             : chr [1:5698833] "01" "01" "01" "01" ...
##  $ day               : chr [1:5698833] "23" "27" "21" "07" ...
##  $ year              : chr [1:5698833] "2021" "2021" "2021" "2021" ...
##  $ hour              : chr [1:5698833] "16" "18" "22" "13" ...
```

The dataframe includes entries where ride_length was negative that need to be removed, as well as rides lasting longer than a day, which are likely errors due to missing bikes or locking/unlocking issues. Creating a new dataframe (v2) to remove these data.

```
all_trips_v2 <- all_trips[!(all_trips$ride_length<=0 |
                              all_trips$ride_length>86400),]
#checking to make sure new numbers make sense
mean(all_trips$ride_length)
```

```
## [1] 1308.833
```

```
mean(all_trips_v2$ride_length)
```

```
## [1] 1161.638
```

Step 4: Descriptive Analysis

A quick check of the summary statistics to get a feel for the data and see if everything is making sense.

```
# Descriptive analysis on ride_length (all figures in seconds)
summary(all_trips_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##       1     401     714    1162    1295   86397
```

```
# 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                1608.3574
## 2                     member                 800.9731
```

```
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                      954
## 2                     member                      572
```

```
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                    86395
## 2                     member                    86397
```

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

```
# See how members vs casual rides vary by month
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$month, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$month all_trips_v2$ride_length
## 1                      casual                 01                1163.3250
## 2                      member                 01                 727.7236
## 3                      casual                 02                1820.1485
## 4                      member                 02                 948.5144
## 5                      casual                 03                1885.5581
## 6                      member                 03                 832.0296
## 7                      casual                 04                1866.3993
## 8                      member                 04                 870.8224
## 9                      casual                 05                1897.9637
## 10                     member                 05                 867.4332
## 11                     casual                 06                1761.9078
## 12                    member                    06               864.8334
## 13                    casual                    07              1651.1498
## 14                    member                    07               843.6786
## 15                    casual                    08              1579.9378
## 16                    member                    08               832.2806
## 17                    casual                    09              1514.6118
## 18                    member                    09               811.2417
## 19                    casual                    10              1371.5517
## 20                    member                    10               734.4471
## 21                    casual                    11              1123.7972
## 22                    member                    11               665.9314
## 23                    casual                    12              1091.4501
## 24                    member                    12               649.2855
```

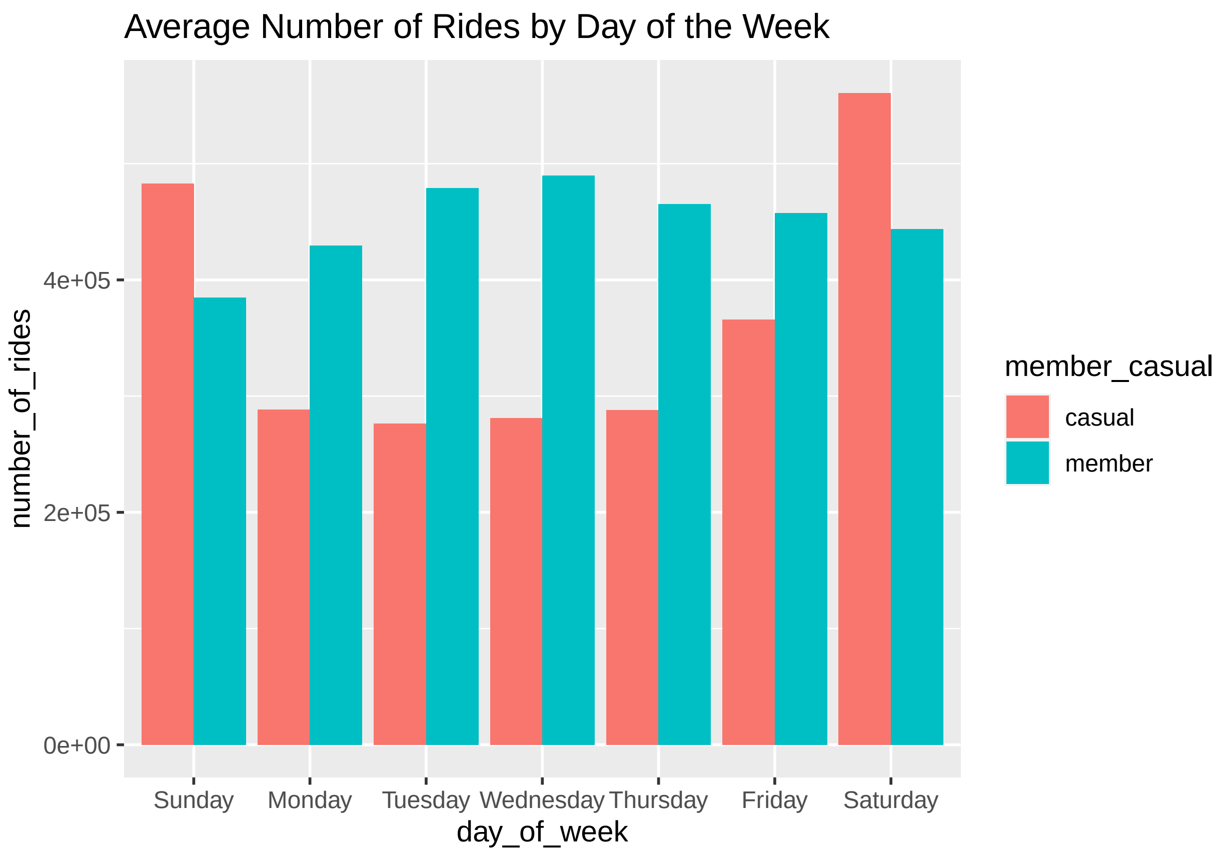
```
# Setting order for the days of week when sorting
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "W

# 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                1860.4478
## 2                      member                   Sunday                 913.8312
## 3                      casual                   Monday                1629.6212
## 4                      member                   Monday                 776.8454
## 5                      casual                  Tuesday                1465.4033
## 6                      member                  Tuesday                 755.0585
## 7                      casual                Wednesday                1393.3015
## 8                      member                Wednesday                 757.5224
## 9                      casual                 Thursday                1386.4415
## 10                     member                 Thursday                 751.5171
## 11                     casual                   Friday                1493.2978
## 12                     member                   Friday                 783.8712
## 13                     casual                 Saturday                1747.7280
## 14                     member                 Saturday                 893.4363
```

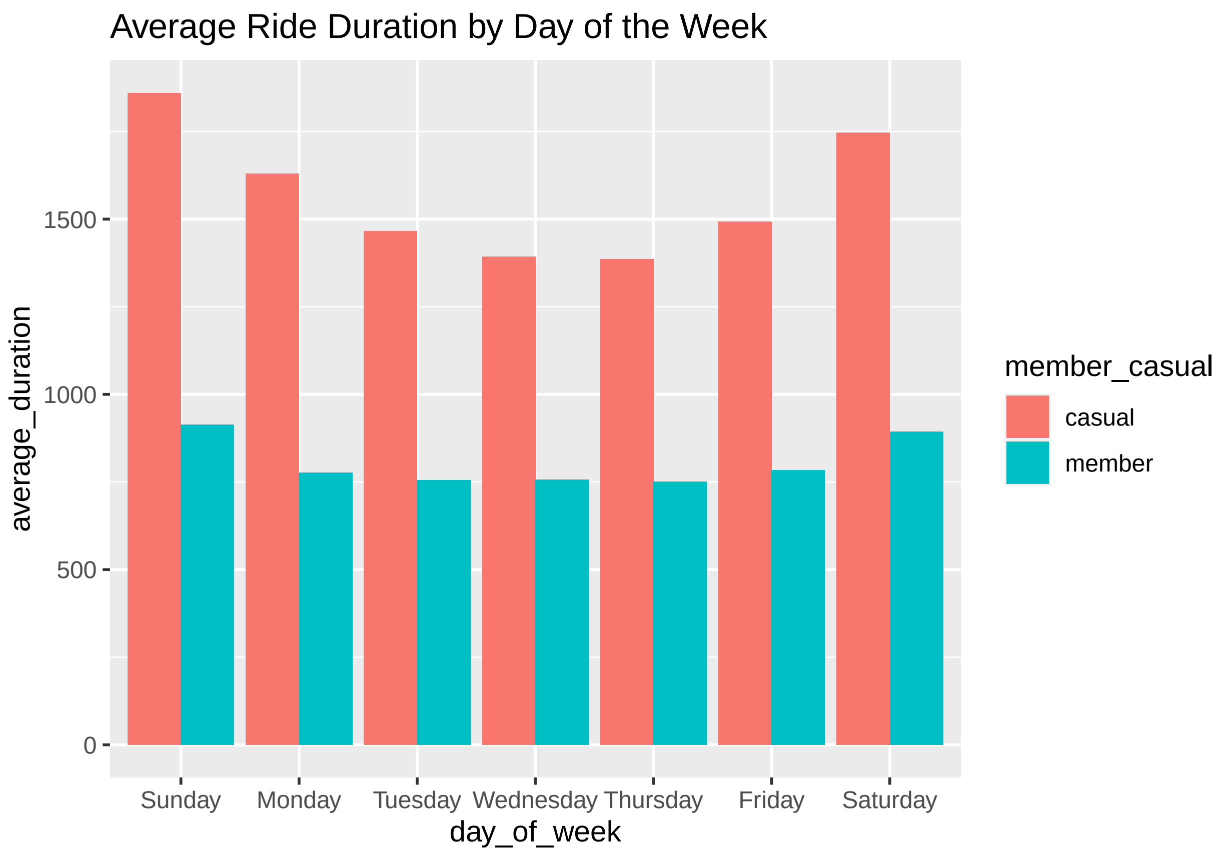
Step 5: Visualizations

Creating visuals to get a clearer picture of the data and see what suspected trends might or might not exist.

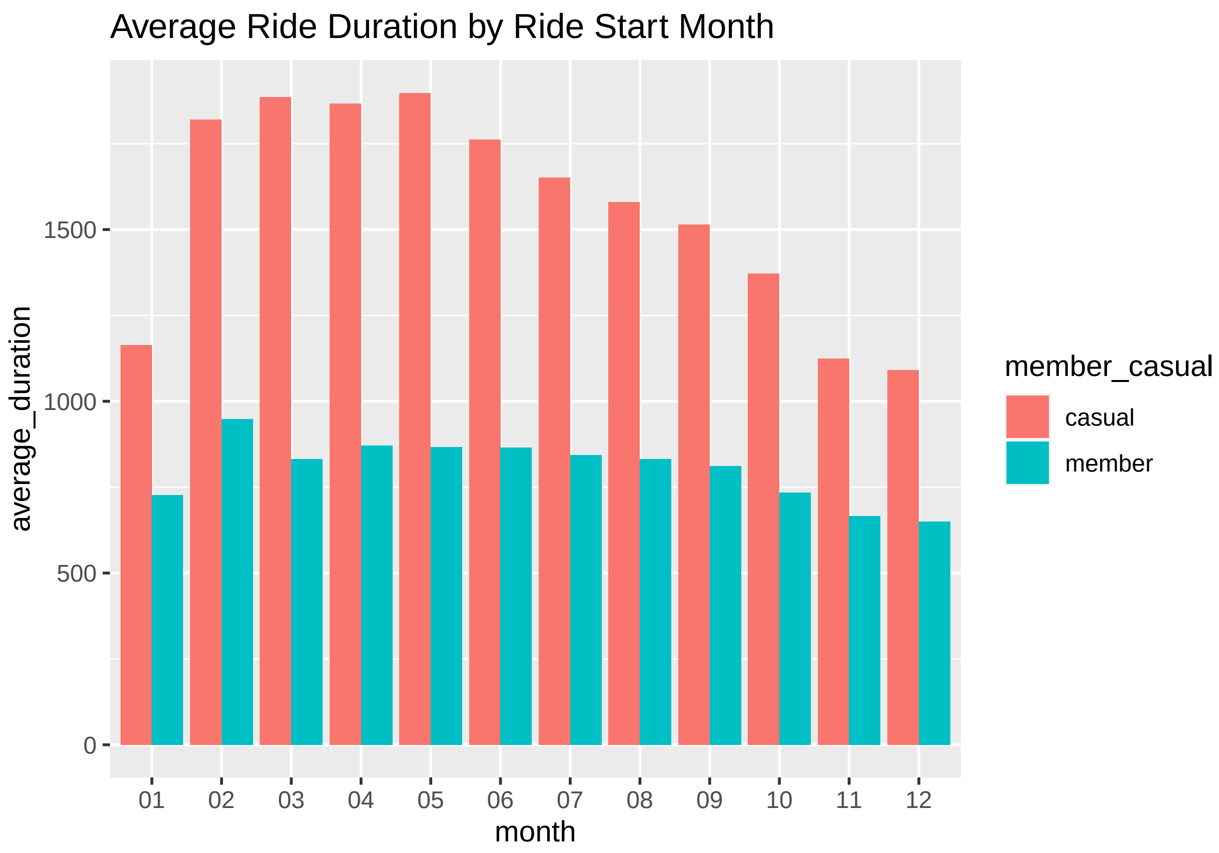
```
## ‘summarise()‘ has grouped output by ’member_casual’. You can override using the
## ‘.groups‘ argument.
```



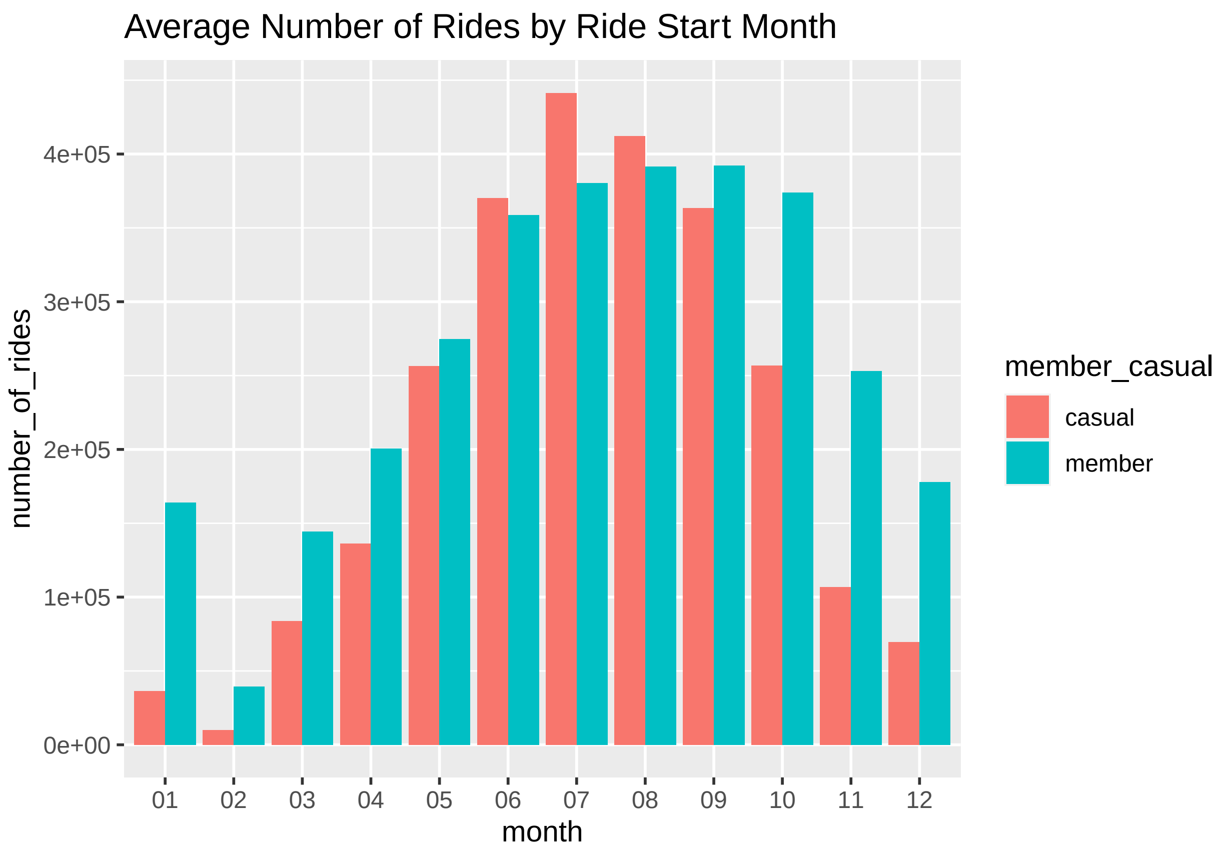
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



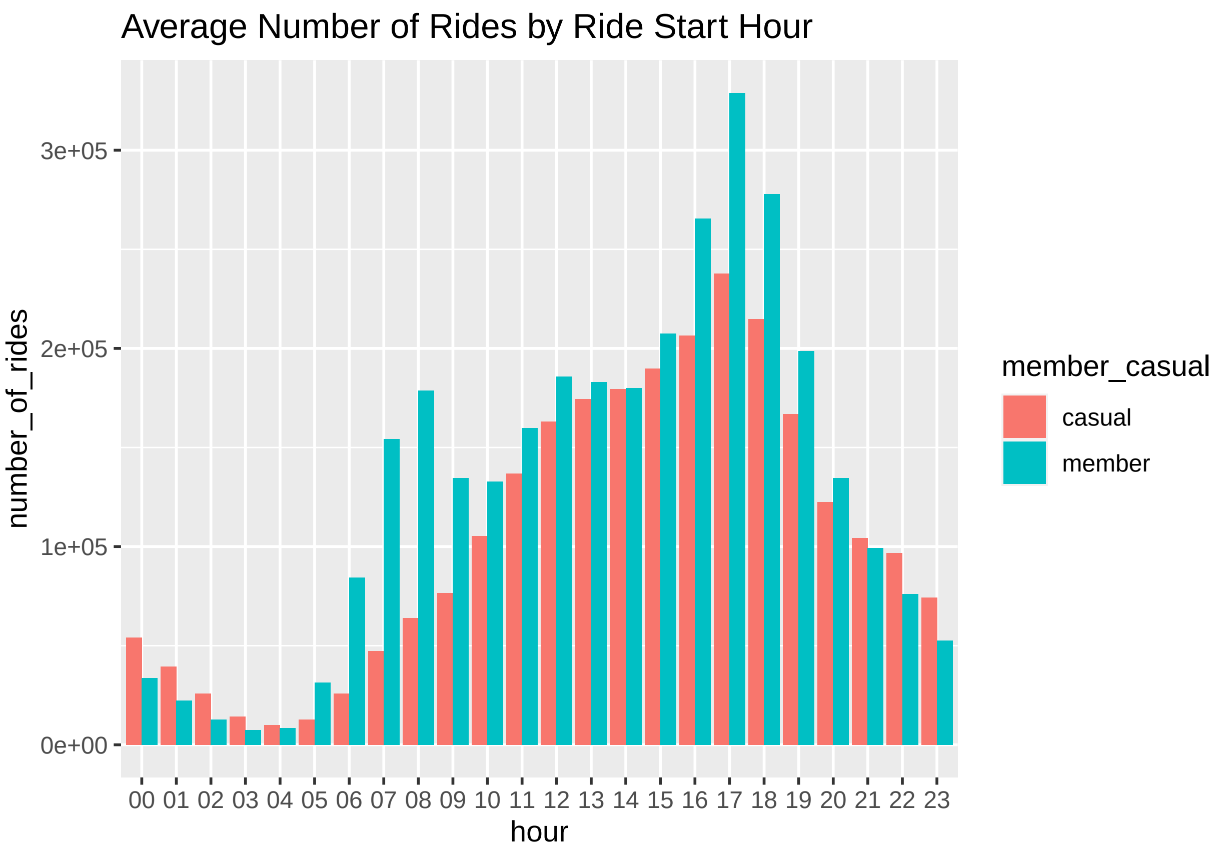
```
## ‘summarise()‘ has grouped output by ’member_casual’. You can override using the
## ‘.groups‘ argument.
```



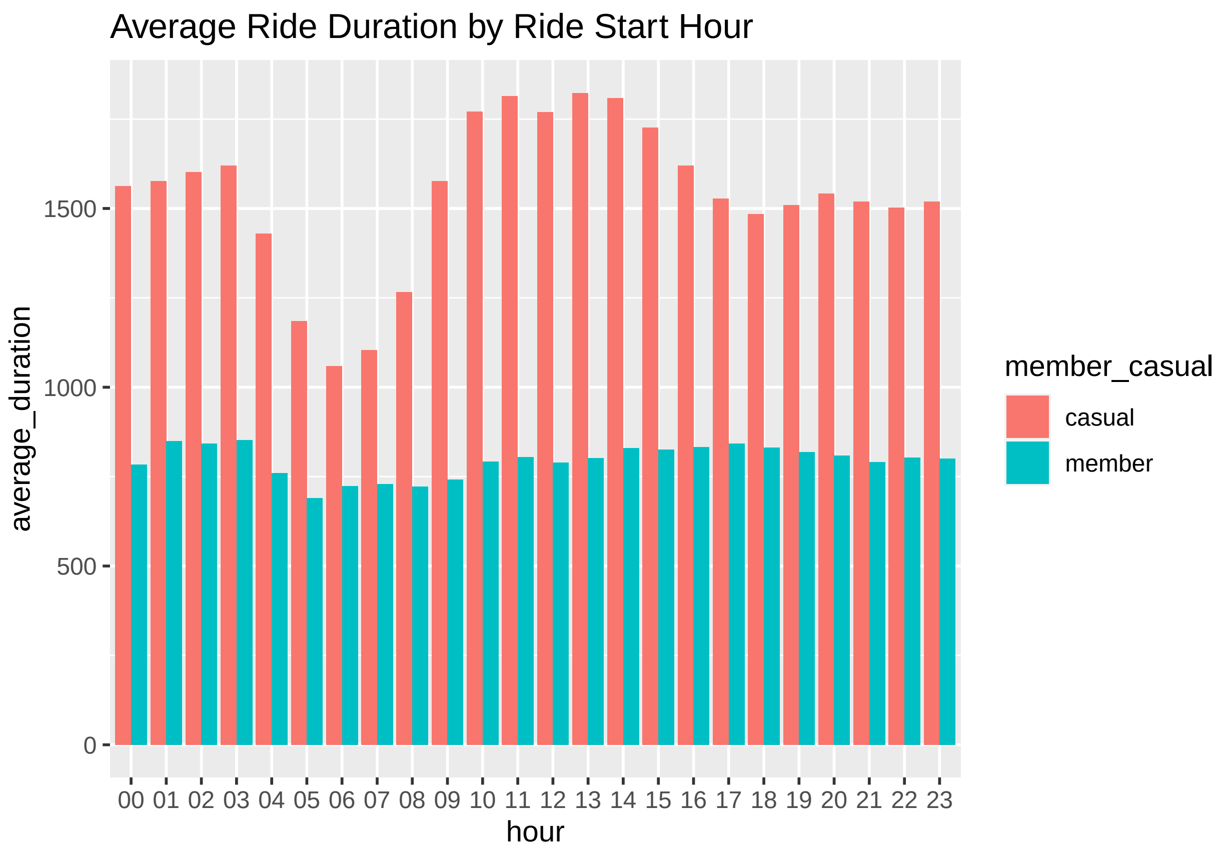
```
## 'summarise()' has grouped output by 'member_casual'. You can override using the
## '.groups' argument.
```



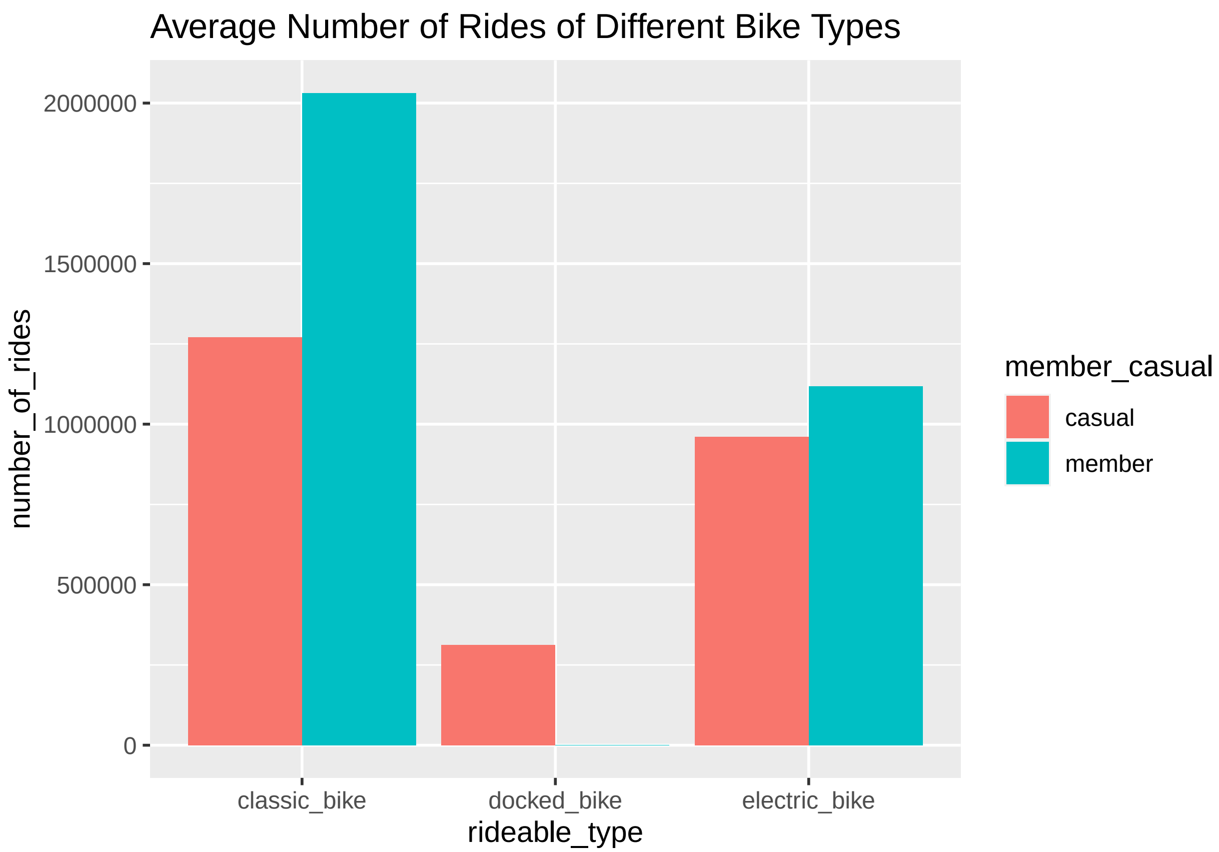
```
## 'summarise()' has grouped output by 'member_casual'. You can override using the
## '.groups' argument.
```



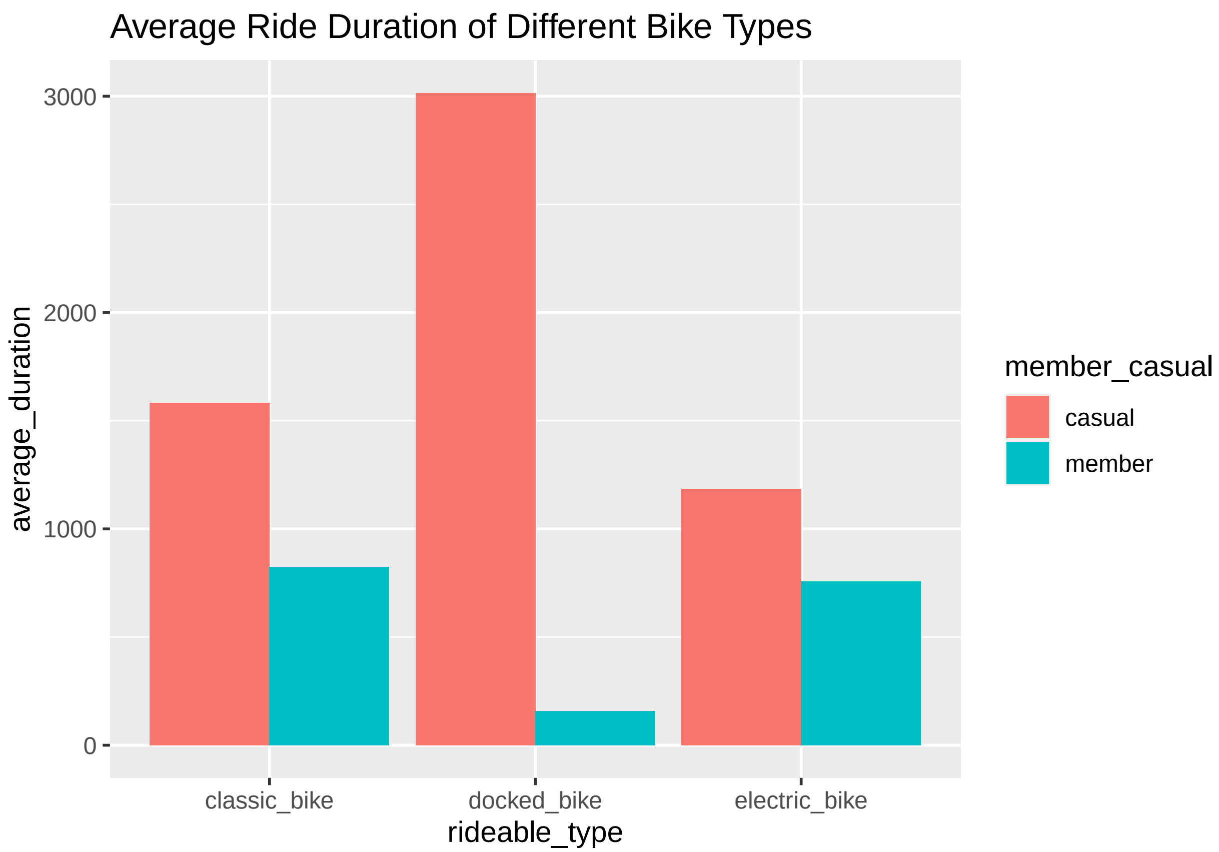
```
## ‘summarise()‘ has grouped output by ’member_casual’. You can override using the
## ‘.groups‘ argument.
```



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



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



Step 6: Exporting Results

Below was used to export summary files for further analysis in Tableau.

```
# counts <- aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual +
#                       all_trips_v2$day_of_week + all_trips_v2$month +
#                       all_trips_v2$hour, FUN = mean)
#
# write.csv(counts, file = '~/Documents/counts.csv')
#
# write.csv(all_trips_v2, file = '~/Documents/all_trips_vs.csv')
# #Above file was too large for Tableau, so making a smaller one.
# tableau_data <- all_trips_v2 %>%
#   select(-c(ended_at, end_station_name, end_station_id, end_lat, end_lng))
#
# write.csv(tableau_data, file='~/Documents/tableau_data.csv')
```

Conclusions

Overall, membership riders take more rides than casual riders, but casual riders have a longer average ride time. This is likely caused by the pricing structure (fee to undock a bike every ride) incentivizing casual riders to take fewer, but longer trips, whereas membership riders have no price incentive since undocking is free for them.

Additionally, membership riders are likely to use Divvy to commute to work, while casual riders appear to use it more for leisure purposes, which could be contributing to the decreased ride duration for members compared to casual riders. This is supported by the weekday data, which shows membership riders having their highest number of rides during the week, while casual riders most popular days of use are on the weekends. Further support of this comes from inspecting the hourly number of rides, which shows two peaks for membership riders, one at 8am and one at 5pm, which would be representative of typical commute hours.

Both casual and membership ridership peak during the summer months and trail off as winter approaches, with the lowest ridership occuring during the month of February. This is primarily due to the weather in Chicago during the winter months and the large amount of snow accumulation that can make riding both uncomfortable and dangerous. It is worth noting that casual riders have a more extreme difference in rides taken between summer and winter months compared to membership riders. This is likely due to a combination of membership riders still relying on Divvy bikes to commute in the winter, as well as the sunk cost of an annual membership incentivizing membership riders to continue riding, whereas casual riders have no sunk cost.

Finally, when it comes to bike preference, both casual riders and membership riders seem to prefer classic bikes over electric bikes. This preference appears to be stronger amongst membership riders and, for both types of riders, appears to be driven by the increased cost of the electric bike compared to a classic bike.

Recommendations

The following are just a few available options that could help convert more riders from casual riders to membership riders:

- Make a marketing pushing in the early summer to convert casual riders to membership riders as this is when Divvy is most in demand.
- Explore varying price structure on weekdays vs weekends to see if casual riders can be incentivized to convert to memberships.
- Explore allowing members to reserve bikes during certain weekend hours when bikes would be most in-demand.

Additional work could be done in the following areas to provide a clearer picture of what motivates Divvy users:

- Explore the relationship between start/end ride locations and rider type (member vs casual).
- Investigate more precisely how weather relates to members/casual riders willingness to ride (in both summer and winter months).
- Investigate how large city events (Lollapalooze, Chicago Marathon, etc.) affect ridership.
- Investigate what affect bike supply (electric vs classic), weather, time of day, etc. has on rider selection.