Bike-sharing Case Study

Chinonso

2024-02-15

#Setting up my environment #Setting up my R environment by loading: tidyverse, ggplot2, dplyr, skimr, janitor, here

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

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

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("skimr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("janitor")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("here")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("rmarkdown")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## 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(ggplot2)
library(dplyr)
library(skimr)
library(janitor)
```

```
##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(here)
```

```
## here() starts at /cloud/project
```

```
library(conflicted)
```

#STEP I: DATA COLLECTION #Step1: Data Collection

```
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.
```

```
head(q1_2019)
```

```
## # A tibble: 6 x 12
##    trip_id start_time          end_time            bikeid tripduration
##      <dbl> <dttm>              <dttm>               <dbl>        <dbl>
## 1 21742443 2019-01-01 00:04:37 2019-01-01 00:11:07   2167          390
## 2 21742444 2019-01-01 00:08:13 2019-01-01 00:15:34   4386          441
## 3 21742445 2019-01-01 00:13:23 2019-01-01 00:27:12   1524          829
## 4 21742446 2019-01-01 00:13:45 2019-01-01 00:43:28    252         1783
## 5 21742447 2019-01-01 00:14:52 2019-01-01 00:20:56   1170          364
## 6 21742448 2019-01-01 00:15:33 2019-01-01 00:19:09   2437          216
## # 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>
```

```
q1_2020<- read_csv("Divvy_Trips_2020_Q1.csv")
```

```
## Rows: 426887 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
```

```
## dttm (2): started_at, ended_at
##
## 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.
```

```
head(q1_2020)
```

```
## # A tibble: 6 x 13
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 EACB19130B0CDA4A docked_bike   2020-01-21 20:06:59 2020-01-21 20:14:30
## 2 8FED874C809DC021 docked_bike   2020-01-30 14:22:39 2020-01-30 14:26:22
## 3 789F3C21E472CA96 docked_bike   2020-01-09 19:29:26 2020-01-09 19:32:17
## 4 C9A388DAC6ABF313 docked_bike   2020-01-06 16:17:07 2020-01-06 16:25:56
## 5 943BC3CBECCFD662 docked_bike   2020-01-30 08:37:16 2020-01-30 08:42:48
## 6 6D9C8A6938165C11 docked_bike   2020-01-10 12:33:05 2020-01-10 12:37:54
## # i 9 more variables: start_station_name <chr>, start_station_id <dbl>,
## #   end_station_name <chr>, end_station_id <dbl>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>
```

#Step2: comparing column names

```
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(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"
```

#STEP II: WRANGLE DATA & COMBINE INTO A SINGLE FILE #Step1: Wrangling data & combine into a single file #After comparing column names, we need to make them consistent (rename to make them consistent)

```
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
                )
q1_2019
```

```
## # A tibble: 365,069 x 12
##     ride_id started_at          ended_at            rideable_type tripduration
##       <dbl> <dttm>              <dttm>                      <dbl>        <dbl>
##  1 21742443 2019-01-01 00:04:37 2019-01-01 00:11:07          2167          390
##  2 21742444 2019-01-01 00:08:13 2019-01-01 00:15:34          4386          441
```

```
##  3 21742445 2019-01-01 00:13:23 2019-01-01 00:27:12          1524          829
##  4 21742446 2019-01-01 00:13:45 2019-01-01 00:43:28           252         1783
##  5 21742447 2019-01-01 00:14:52 2019-01-01 00:20:56          1170          364
##  6 21742448 2019-01-01 00:15:33 2019-01-01 00:19:09          2437          216
##  7 21742449 2019-01-01 00:16:06 2019-01-01 00:19:03          2708          177
##  8 21742450 2019-01-01 00:18:41 2019-01-01 00:20:21          2796          100
##  9 21742451 2019-01-01 00:18:43 2019-01-01 00:47:30          6205         1727
## 10 21742452 2019-01-01 00:19:18 2019-01-01 00:24:54          3939          336
## # i 365,059 more rows
## # i 7 more variables: start_station_id <dbl>, start_station_name <chr>,
## #   end_station_id <dbl>, end_station_name <chr>, member_casual <chr>,
## #   gender <chr>, birthyear <dbl>
```

#Step2: Inspecting the dataframes and checks for incongruencies

```
str(q1_2019)
```

```
## spc_tbl_ [365,069 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : num [1:365069] 21742443 21742444 21742445 21742446 21742447 ...
##  $ started_at        : POSIXct[1:365069], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ ended_at          : POSIXct[1:365069], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
##  $ rideable_type     : num [1:365069] 2167 4386 1524 252 1170 ...
##  $ tripduration      : num [1:365069] 390 441 829 1783 364 ...
##  $ start_station_id  : num [1:365069] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:365069] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ end_station_id    : num [1:365069] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:365069] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "W
##  $ member_casual     : 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(q1_2020)
```

```
## spc_tbl_ [426,887 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:426887] "EACB19130B0CDA4A" "8FED874C809DC021" "789F3C21E472CA96" "C9A38
##  $ rideable_type     : chr [1:426887] "docked_bike" "docked_bike" "docked_bike" "docked_bike" ...
##  $ started_at        : POSIXct[1:426887], format: "2020-01-21 20:06:59" "2020-01-30 14:22:39" ...
##  $ ended_at          : POSIXct[1:426887], format: "2020-01-21 20:14:30" "2020-01-30 14:26:22" ...
##  $ start_station_name: chr [1:426887] "Western Ave & Leland Ave" "Clark St & Montrose Ave" "Broadway
##  $ start_station_id  : num [1:426887] 239 234 296 51 66 212 96 96 212 38 ...
```

```
##  $ end_station_name  : chr [1:426887] "Clark St & Leland Ave" "Southport Ave & Irving Park Rd" "Wilt
##  $ end_station_id    : num [1:426887] 326 318 117 24 212 96 212 212 96 100 ...
##  $ start_lat         : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:426887] -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:426887] 42 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:426887] -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:426887] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_double(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_double(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

#Step3: Convert ride_id and rideable_type to character, so they can stack correctly

```
q1_2019<- mutate(q1_2019, ride_id= as.character(ride_id),
                 rideable_type= as.character(rideable_type))
```

#Step4: Stack individual quarters data frames into one big #data frames

```
all_trips <- bind_rows(q1_2019, q1_2020)
head(all_trips)
```

```
## # A tibble: 6 x 16
##   ride_id  started_at          ended_at            rideable_type tripduration
##   <chr>    <dttm>              <dttm>              <chr>                <dbl>
## 1 21742443 2019-01-01 00:04:37 2019-01-01 00:11:07 2167                   390
## 2 21742444 2019-01-01 00:08:13 2019-01-01 00:15:34 4386                   441
## 3 21742445 2019-01-01 00:13:23 2019-01-01 00:27:12 1524                   829
## 4 21742446 2019-01-01 00:13:45 2019-01-01 00:43:28 252                   1783
## 5 21742447 2019-01-01 00:14:52 2019-01-01 00:20:56 1170                   364
## 6 21742448 2019-01-01 00:15:33 2019-01-01 00:19:09 2437                   216
## # i 11 more variables: start_station_id <dbl>, start_station_name <chr>,
## #   end_station_id <dbl>, end_station_name <chr>, member_casual <chr>,
## #   gender <chr>, birthyear <dbl>, start_lat <dbl>, start_lng <dbl>,
## #   end_lat <dbl>, end_lng <dbl>
```

#Step5: Remove lat, long, birthday and gender fields as the fields were dropped beginning 2020

```
all_trips <-all_trips %>%
  select(-c(start_lat, start_lng, end_lat, end_lng,
            birthyear, gender, "tripduration"))
```

#STEP III: CLEAN UP AND ADD DATA TO PREPARE FOR ANALYSIS #Step1:Inspect the new table that has just been crreated.

```r
colnames(all_trips)
```

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

```r
nrow(all_trips)
```

```
## [1] 791956
```

```r
ncol(all_trips)
```

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

```r
dim(all_trips)
```

```
## [1] 791956      9
```

```r
head(all_trips)
```

```
## # A tibble: 6 x 9
##   ride_id started_at          ended_at            rideable_type start_station_id
##   <chr>   <dttm>              <dttm>              <chr>                     <dbl>
## 1 217424~ 2019-01-01 00:04:37 2019-01-01 00:11:07 2167                        199
## 2 217424~ 2019-01-01 00:08:13 2019-01-01 00:15:34 4386                         44
## 3 217424~ 2019-01-01 00:13:23 2019-01-01 00:27:12 1524                         15
## 4 217424~ 2019-01-01 00:13:45 2019-01-01 00:43:28 252                         123
## 5 217424~ 2019-01-01 00:14:52 2019-01-01 00:20:56 1170                        173
## 6 217424~ 2019-01-01 00:15:33 2019-01-01 00:19:09 2437                         98
## # i 4 more variables: start_station_name <chr>, end_station_id <dbl>,
## #   end_station_name <chr>, member_casual <chr>
```

```r
str(all_trips)
```

```
## tibble [791,956 x 9] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:791956] "21742443" "21742444" "21742445" "21742446" ...
##  $ started_at        : POSIXct[1:791956], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ ended_at          : POSIXct[1:791956], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
##  $ rideable_type     : chr [1:791956] "2167" "4386" "1524" "252" ...
##  $ start_station_id  : num [1:791956] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:791956] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ end_station_id    : num [1:791956] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:791956] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "
##  $ member_casual     : chr [1:791956] "Subscriber" "Subscriber" "Subscriber" "Subscriber" ...
```

```r
summary(all_trips)
```

```
##    ride_id            started_at
##  Length:791956      Min.   :2019-01-01 00:04:37.00
##  Class :character   1st Qu.:2019-02-28 17:04:04.75
##  Mode  :character   Median :2020-01-07 12:48:50.50
##                     Mean   :2019-09-01 11:58:08.35
##                     3rd Qu.:2020-02-19 19:31:54.75
##                     Max.   :2020-03-31 23:51:34.00
##
##     ended_at                      rideable_type      start_station_id
##  Min.   :2019-01-01 00:11:07.00   Length:791956      Min.   :  2.0
##  1st Qu.:2019-02-28 17:15:58.75   Class :character   1st Qu.: 77.0
```

```
##  Median :2020-01-07 13:02:50.00   Mode  :character   Median :174.0
##  Mean   :2019-09-01 12:17:52.17                      Mean   :204.4
##  3rd Qu.:2020-02-19 19:51:54.50                      3rd Qu.:291.0
##  Max.   :2020-05-19 20:10:34.00                      Max.   :675.0
##
##  start_station_name end_station_id  end_station_name   member_casual
##  Length:791956      Min.   :  2.0   Length:791956      Length:791956
##  Class :character   1st Qu.: 77.0   Class :character   Class :character
##  Mode  :character   Median :174.0   Mode  :character   Mode  :character
##                     Mean   :204.4
##                     3rd Qu.:291.0
##                     Max.   :675.0
##                     NA's   :1
```

Step2: In the "member_casual" column, there are two names for members (

#"member and"Subscriber") and two names for casual riders("Customer" and "casual").

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

```
##
##     casual   Customer     member Subscriber
##      48480      23163     378407     341906
```

#Step3: Replace Subscriber with member and Customer" with "casual

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

```
##
## casual member
##  71643 720313
```

#Step4: Add columns that list the date, month, day and year of each ride

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

#Step5: Add ride_length calculation to all_trips(in seconds)

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

#Step6: Inspect the structure of the columns

```
str(all_trips)
```

```
## tibble [791,956 x 15] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:791956] "21742443" "21742444" "21742445" "21742446" ...
##  $ started_at        : POSIXct[1:791956], format: "2019-01-01 00:04:37" "2019-01-01 00:08:13" ...
##  $ ended_at          : POSIXct[1:791956], format: "2019-01-01 00:11:07" "2019-01-01 00:15:34" ...
```

```
##  $ rideable_type     : chr [1:791956] "2167" "4386" "1524" "252" ...
##  $ start_station_id  : num [1:791956] 199 44 15 123 173 98 98 211 150 268 ...
##  $ start_station_name: chr [1:791956] "Wabash Ave & Grand Ave" "State St & Randolph St" "Racine Ave &
##  $ end_station_id    : num [1:791956] 84 624 644 176 35 49 49 142 148 141 ...
##  $ end_station_name  : chr [1:791956] "Milwaukee Ave & Grand Ave" "Dearborn St & Van Buren St (*)" "W
##  $ member_casual     : chr [1:791956] "member" "member" "member" "member" ...
##  $ date              : Date[1:791956], format: "2019-01-01" "2019-01-01" ...
##  $ month             : chr [1:791956] "01" "01" "01" "01" ...
##  $ day               : chr [1:791956] "01" "01" "01" "01" ...
##  $ year              : chr [1:791956] "2019" "2019" "2019" "2019" ...
##  $ day_of_week       : chr [1:791956] "Tuesday" "Tuesday" "Tuesday" "Tuesday" ...
##  $ ride_length       : 'difftime' num [1:791956] 390 441 829 1783 ...
##   ..- attr(*, "units")= chr "secs"
```

#Step7: Convert ride_length from factor to numeric to 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
```

#Step8: remove where ride_length is zero(0). We'll create a new dataframe since we're removing data and assign it to all_trips_v2

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

```
## [1] 788189
```

#STEP IV: CONDUCT DESCRIPTIVE ANALYSIS #Step1: Descriptive analysis on ride_length(all figures in seconds) #(a) straight average (total ride length / rides)

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

```
## [1] 1189.459
```

#(b) mid point no in the ascending array of ride length

```
median(all_trips_v2$ride_length)
```

```
## [1] 539
```

#(c) longest ride

```
max(all_trips_v2$ride_length)
```

```
## [1] 10632022
```

#(d) shortest ride

```
min(all_trips_v2$ride_length)
```

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

#all four lines above can be joined with summary

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

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##        1      331      539     1189      912 10632022
```

#Step2: Comparing members and casual users

```r
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                5372.7839
## 2                     member                 795.2523
```

```r
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                      1393
## 2                     member                       508
```

```r
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                10632022
## 2                     member                 6096428
```

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

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

#Step 3: Average ride time by each day for members & casual users

```r
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                6090.7373
## 2                      member                   Friday                 796.7338
## 3                      casual                   Monday                4752.0504
## 4                      member                   Monday                 822.3112
## 5                      casual                 Saturday                4950.7708
## 6                      member                 Saturday                 974.0730
## 7                      casual                   Sunday                5061.3044
## 8                      member                   Sunday                 972.9383
## 9                      casual                 Thursday                8451.6669
## 10                     member                 Thursday                 707.2093
## 11                     casual                  Tuesday                4561.8039
## 12                     member                  Tuesday                 769.4416
## 13                     casual                Wednesday                4480.3724
## 14                     member                Wednesday                 711.9838
```

#Step 4: Days of the week are out of order, let's fix it

```
all_trips_v2$day_of_week<- ordered(all_trips_v2$day_of_week,
                                   levels= c("Sunday", "Monday", "Tuesday",
                                             "Wednesday", "Thurday","Friday","Saturday"))
```

#We can rerun average ride time by each day of the week

```
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                5061.3044
## 2                      member                   Sunday                 972.9383
## 3                      casual                   Monday                4752.0504
## 4                      member                   Monday                 822.3112
## 5                      casual                  Tuesday                4561.8039
## 6                      member                  Tuesday                 769.4416
## 7                      casual                Wednesday                4480.3724
## 8                      member                Wednesday                 711.9838
## 9                      casual                   Friday                6090.7373
## 10                     member                   Friday                 796.7338
## 11                     casual                 Saturday                4950.7708
## 12                     member                 Saturday                 974.0730
```

#Step 5: Analyse ridership data by type and weekday #(a) create weekday field using wday() #(b) Group the usertype and weekday #(c) Calculate the no of rides and 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)
```

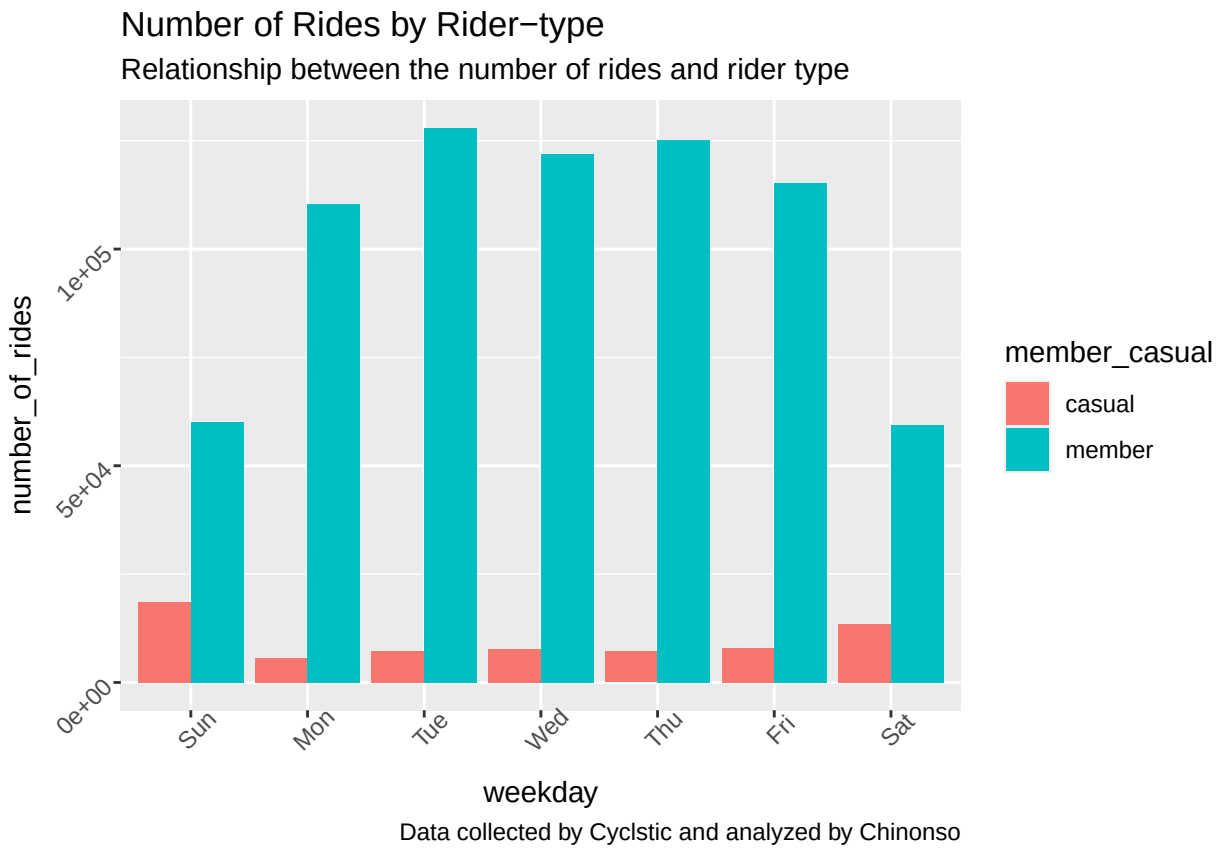
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun               18652            5061.
##  2 casual        Mon                5591            4752.
##  3 casual        Tue                7311            4562.
##  4 casual        Wed                7690            4480.
##  5 casual        Thu                7147            8452.
##  6 casual        Fri                8013            6091.
##  7 casual        Sat               13473            4951.
##  8 member        Sun               60197             973.
##  9 member        Mon              110430             822.
## 10 member        Tue              127974             769.
## 11 member        Wed              121902             712.
## 12 member        Thu              125228             707.
## 13 member        Fri              115168             797.
## 14 member        Sat               59413             974.
```

#Step6: Visualizing 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') +
  theme(axis.text = element_text(angle = 45)) +
  labs(title = "Number of Rides by Rider-type",
       subtitle = "Relationship between the number of rides and rider type",
       caption = "Data collected by Cyclstic and analyzed by Chinonso")
```

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

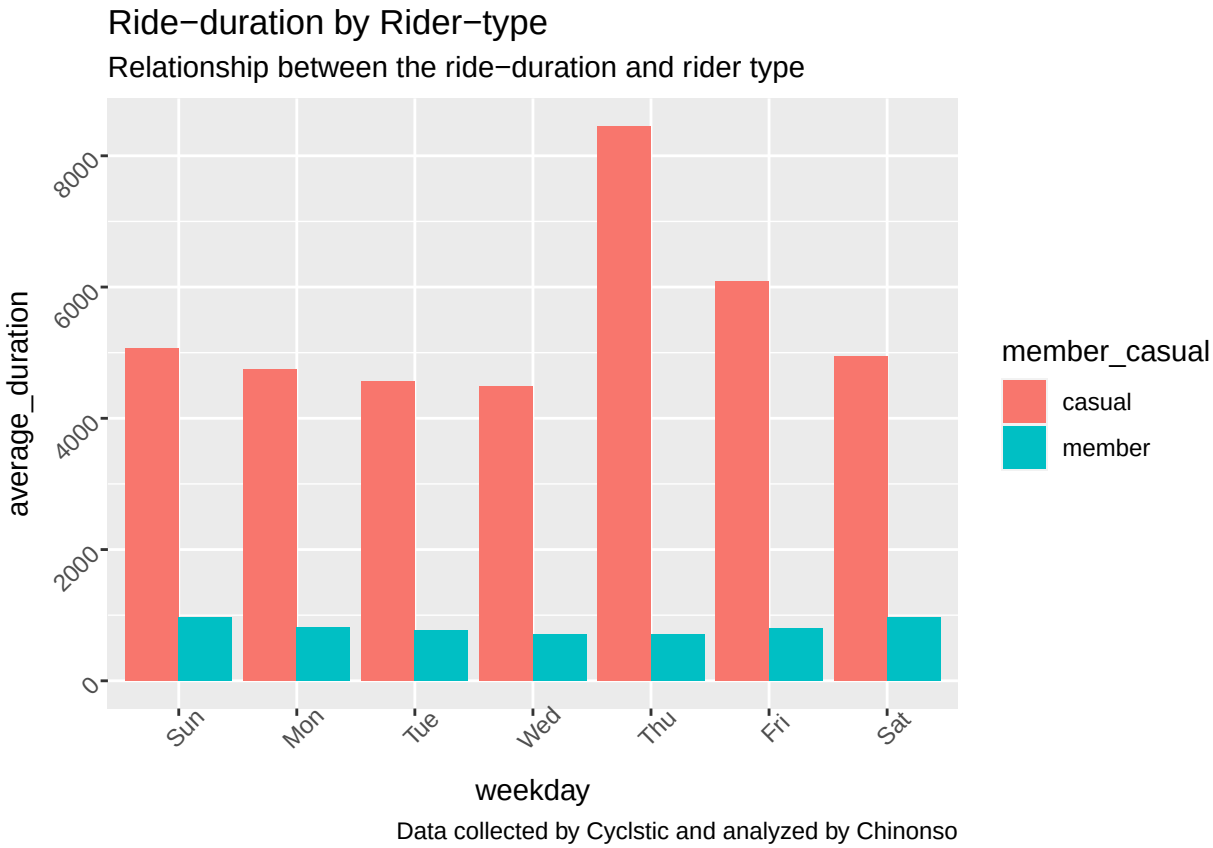


#Step7: Visualizing the ride duration by weekdays

```
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') +
  theme(axis.text = element_text(angle = 45)) +
```

```
  labs(title = "Ride-duration by Rider-type",
       subtitle = "Relationship between the ride-duration and rider type",
       caption = "Data collected by Cyclstic and analyzed by Chinonso")
```

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



#CONCLUSIONS AND RECOMMENDATIONS

#Conclusions 1. The graph of the number of rides and rider-type by weekdays shows that member riders engage averagely in more rides than casual riders. Members had more average rides on Monday to Friday with the peak (highest) on Tuesdays and Thursdays while weekends (Saturday and Sunday) had the least patronage. The casual rides had a fewer number of rides compared to the member riders. The casual riders had their most rides on average on weekends (Saturdays and Sundays) with the highest rides on Sundays. The casual riders had their lowest rides on weekdays which directly opposite of member riders.

2. The graph of average duration and rider-type by weekdays shows casual riders had longer rides averagely compared to the member riders. Casual riders had the longest rides on Thursday while member riders had the shortest average ride-duration on the same day (Thursday).

#Recommendations 1. Casual riders can be made to see the benefits of being members and the lots of discounts associated with longer rides for members. 2. Since casual riders engaged in longer rides on Thursday, they could be given smooth transmission to members on Thursday after covering a specific duration with lots of benefits. 3. Creation of loyalty program that rewards casual riders when they take more rides, this will increase their number of rides. 4. Creation of a seamless upgrade process. We could make it easy for casual riders to upgrade to members without hassle.