ride.R

elkhateebnaser

2022-10-30

```
# Project  Case Study: How Does a Bike-Share Navigate Speedy Success/ ----

# In this case study, I will follow the common data analysis process steps
# ASK, PREPARE, PROCESS, ANALYZE, SHARE, ACT
# to support the marketing manager to make a data driven decision.
# A bike-share company based in Chicago. The director of marketing need to
# Test a hypothesis that the company should pay attention to annual
# membership.
# I will participate with Compare between casual and annual membership.
# to help the marketing manager to develop marketing strategy to convert
# the casual to annual membership.

## Preparing The R Environment ----

### INSTALL AND LOAD PACKAGES ----

#### Install pacman package manager ----
if (!require("pacman")) {install.packages("pacman")}
```

```
## Loading required package: pacman
```

```
#### load the required packages using pacman ----
# pacman package manager
# rio to use import
# tidyverse collective packages
# vctrs to append all tables
# lubridate to calculate date
# hydroTSM to calculate season
# chron for time covnersion
pacman::p_load(pacman, rio, tidyverse, vctrs, lubridate, hydroTSM, chron)

### disable scientific number formatting ----
options(scipen = 999)  # turn off scientific notation like 1e+06

## Import the 12 paste month files ----

# source https://divvy-tripdata.s3.amazonaws.com/index.html

### set the working directory where the downloaded files ----
setwd("~/Documents/dsProject/ride bike")
getwd()
```

```
## [1] "/Users/elkhateebnaser/Documents/dsProject/ride bike"
```

```
### read import the files ----
df01 <- import("data/202109-divvy-tripdata.csv")
df02 <- import("data/202110-divvy-tripdata.csv")
df03 <- import("data/202111-divvy-tripdata.csv")
df04 <- import("data/202112-divvy-tripdata.csv")
df05 <- import("data/202201-divvy-tripdata.csv")
df06 <- import("data/202202-divvy-tripdata.csv")
df07 <- import("data/202203-divvy-tripdata.csv")
df08 <- import("data/202204-divvy-tripdata.csv")
df09 <- import("data/202205-divvy-tripdata.csv")
df10 <- import("data/202206-divvy-tripdata.csv")
df11 <- import("data/202207-divvy-tripdata.csv")
df12 <- import("data/202208-divvy-tripdata.csv")

### combine the 12 data sets into one data frame ----
df <- vec_c(df01,
            df02,
            df03,
            df04,
            df05,
            df06,
            df07,
            df08,
            df09,
            df10,
            df11,
            df12) %>%
  data.frame()

### remove unused dataframes ----
rm(df01,
   df02,
   df03,
   df04,
   df05,
   df06,
   df07,
   df08,
   df09,
   df10,
   df11,
   df12)

## Do some necessary data wrangling ####

### Change Rideable_type And Member_casual To Factor Data Type ----
df <- df %>%
  mutate(rideable_type = as_factor(rideable_type)) %>%
  mutate(start_station_name = as_factor(start_station_name)) %>%
  mutate(member_casual = as_factor(member_casual)) %>%
  rename(cyclistic_casual = member_casual)

### Add Time Difference Column ----
df <- df %>%
  mutate(ridding_minutes = as.integer(round(
```

```
    difftime(ended_at, started_at) / 60, digits = 0
  )))

df$ride_id %>% length()
```

```
## [1] 5883043
```

```
df %>% str()
```

```
## 'data.frame':    5883043 obs. of  14 variables:
##  $ ride_id           : chr  "9DC7B962304CBFD8" "F930E2C6872D6B32" "6EF72137900BB91
0" "78D1DE133B3DBF55" ...
##  $ rideable_type     : Factor w/ 3 levels "electric_bike",..: 1 1 1 1 1 1 1 1 1 1
...
##  $ started_at        : POSIXct, format: "2021-09-28 16:07:10" "2021-09-28 14:24:5
1" ...
##  $ ended_at          : POSIXct, format: "2021-09-28 16:09:54" "2021-09-28 14:40:0
5" ...
##  $ start_station_name: Factor w/ 1439 levels "","Clark St & Grace St",..: 1 1 1 1
1 1 1 1 1 1 ...
##  $ start_station_id  : chr  "" "" "" "" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.9 41.9 41.8 41.8 41.9 ...
##  $ start_lng         : num  -87.7 -87.6 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 42 41.8 41.8 41.9 ...
##  $ end_lng           : num  -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ cyclistic_casual  : Factor w/ 2 levels "casual","member": 1 1 1 1 1 1 1 1 1 1
...
##  $ ridding_minutes   : int  3 15 4 9 11 7 22 23 12 22 ...
```

```
### Remove Rows That Have Ridding_length <= 0 ----
df <- df %>%
  filter(!ridding_minutes <= 0)

### Remove The Ridding Time (Minutes) Outliers ----
qs <- df$ridding_minutes %>%
  quantile(probs = c(.25, .75))

iqr <- df$ridding_minutes %>% IQR()

Lower <- qs[1] - 1.5 * iqr
Upper <- qs[2] + 1.5 * iqr

df <- subset(df, df$ridding_minutes > Lower &
               df$ridding_minutes < Upper)

rm(iqr, Lower, Upper, qs)

df$ride_id %>% length()
```

```
## [1] 5410183
```

```
### Add ride_week_day_name Of Ride ----
df <- df %>%
  mutate(day = as_factor(weekdays(started_at)))

### Add ride_month_name ----
df <- df %>%
  mutate(month = as_factor(month(started_at, label = T)))

### Add ride_season ----
df <- df %>%
  mutate(season = as_factor(time2season(started_at, out.fmt = "seasons")))

## some statistics ----

### Print Columns Names ----
names(df)
```

```
##  [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] "cyclistic_casual"   "ridding_minutes"    "day"
## [16] "month"              "season"
```

```
### Fix "Autumn" Season Name ----
df$season <-
  as_factor(gsub("autumm", "autumn", df$season))

### Renaming And Removing The Noise ----
df <- df %>%
  rename(
    id = ride_id,
    type = rideable_type,
    station = start_station_name,
    slat = start_lat,
    slng = start_lng,
    elat = end_lat,
    elng = end_lng,
    member = cyclistic_casual,
    minutes = ridding_minutes
  ) %>%
  select(id,
         member,
         type,
         started_at,
         station,
         slat,
         slng,
         elat,
         elng,
         minutes,
         day,
         month,
         season)

## find out how our data look like ----

### Headers Names ----
names(df)
```

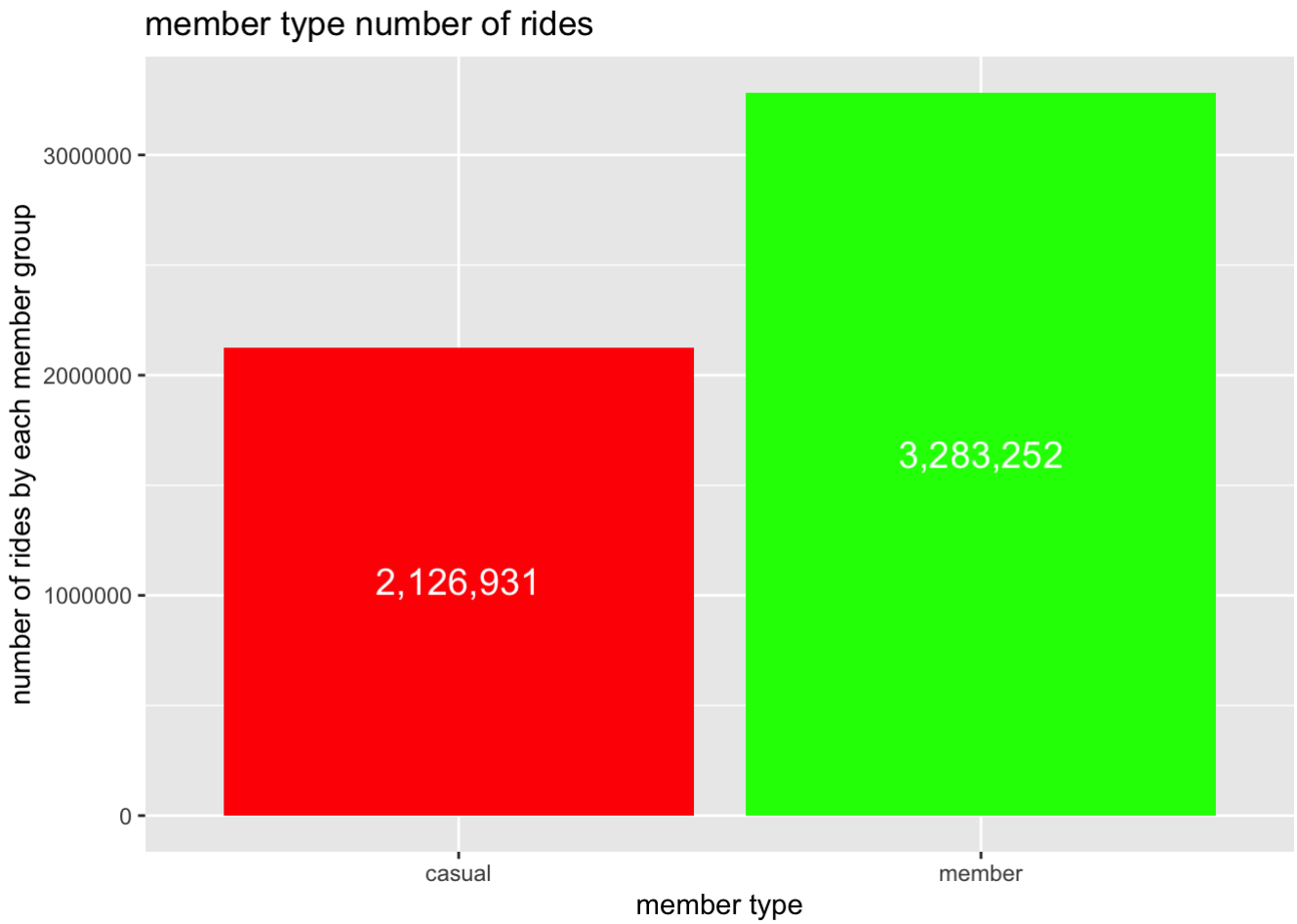
```
##  [1] "id"         "member"     "type"       "started_at" "station"
##  [6] "slat"       "slng"       "elat"       "elng"       "minutes"
## [11] "day"        "month"      "season"
```

```
### summary ----
summary(df)
```

```
##       id                member                    type
##  Length:5410183     casual:2126931   electric_bike:2602849
##  Class :character   member:3283252   classic_bike :2671231
##  Mode  :character                    docked_bike  : 136103
##
##
##
##
##    started_at                                              station
##  Min.   :2021-09-01 00:00:06.00                                 : 828376
##  1st Qu.:2021-11-06 14:43:16.00   Streeter Dr & Grand Ave       :  59225
##  Median :2022-05-04 12:44:33.00   Wells St & Concord Ln         :  39161
##  Mean   :2022-03-21 10:03:27.07   DuSable Lake Shore Dr & North Blvd:  37263
##  3rd Qu.:2022-07-06 09:08:27.00   Clark St & Elm St             :  35921
##  Max.   :2022-08-31 23:59:39.00   Kingsbury St & Kinzie St      :  34519
##                                   (Other)                       :4375718
##       slat            slng             elat            elng
##  Min.   :41.64   Min.   :-87.84   Min.   :41.60   Min.   :-87.88
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64
##  Mean   :41.90   Mean   :-87.65   Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :45.64   Max.   :-73.80   Max.   :42.12   Max.   :-87.50
##                                   NA's   :278     NA's   :278
##     minutes              day             month            season
##  Min.   : 1.00   Tuesday  :766124   Jul    : 745958   autumn:1614266
##  1st Qu.: 6.00   Monday   :706426   Aug    : 721995   winter: 446926
##  Median :10.00   Wednesday:791098   Jun    : 699386   spring:1181652
##  Mean   :12.53   Saturday :857989   Sep    : 685674   summer:2167339
##  3rd Qu.:17.00   Friday   :759283   Oct    : 585594
##  Max.   :40.00   Thursday :784680   May    : 572786
##                  Sunday   :744583   (Other):1398790
```

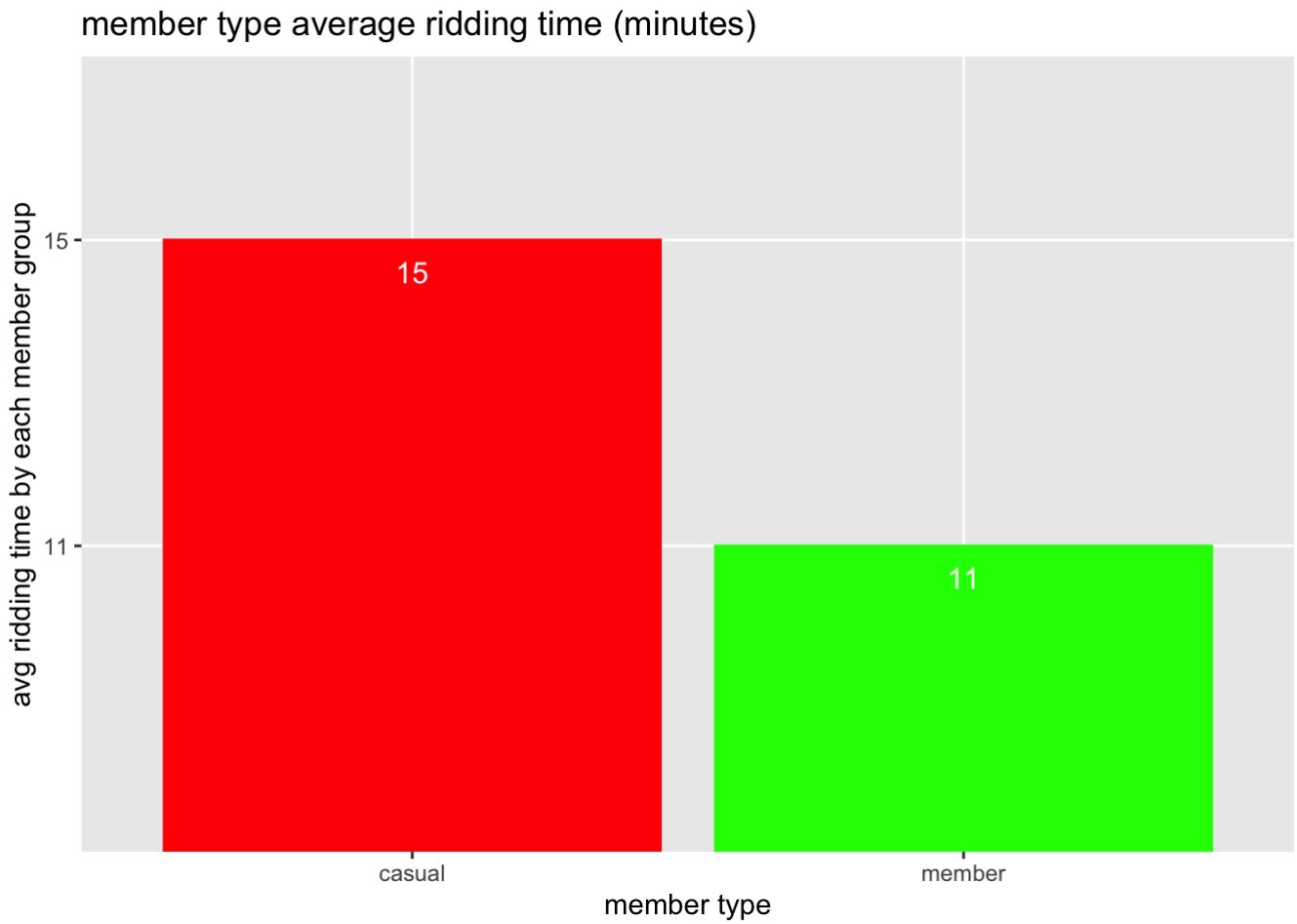
```
## Draw some plots to make better data understanding ----

### How many rides each membership type have done? ----
ggplot(data = df, aes(x = member),) +
  geom_bar(fill = c("red", "green")) +
  stat_count(
    geom = "text",
    size = 5,
    aes(label = prettyNum(
      stat(count),
      big.mark = ",",
      scientific = FALSE
    )),
    position = position_stack(vjust = .5),
    colour = "white"
  ) +
  ggtitle("member type number of rides") +
  labs(x = "member type",
       y = "number of rides by each member group")
```



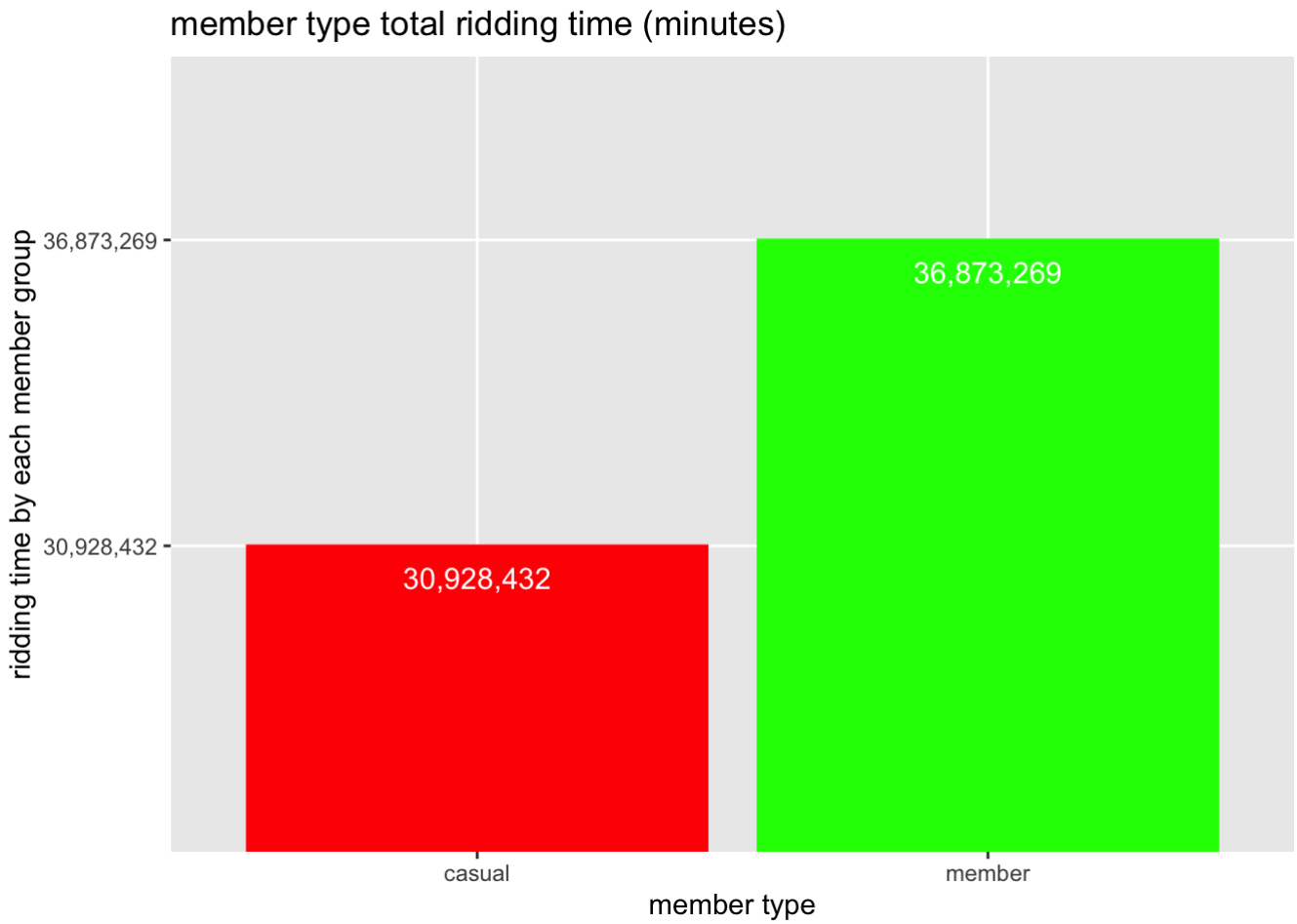

```
#### conclusion 1 ----
# The members have annual membership do 30% more rides than the casual ones
# The more rides can convert to more business and more income

### Does longer average ridding time make the difference ----
df %>%
  select(member, minutes) %>%
  group_by(member) %>%
  summarise(member_ride_time = prettyNum(round(mean(minutes), digits = 0),
                                         big.mark = ",")) %>%
  data.frame() %>%
  ggplot(aes(x = member,
             y = member_ride_time)) +
  geom_col(color = c("red", "green"),
           fill = c("red", "green")) +
  geom_text(
    aes(label = member_ride_time),
    angle = 0,
    color = "white",
    vjust = 2,
    hjust = .5,
    size = 4
  ) +
  ggtitle("member type average ridding time (minutes)") +
  labs(x = "member type",
       y = "avg ridding time by each member group")
```



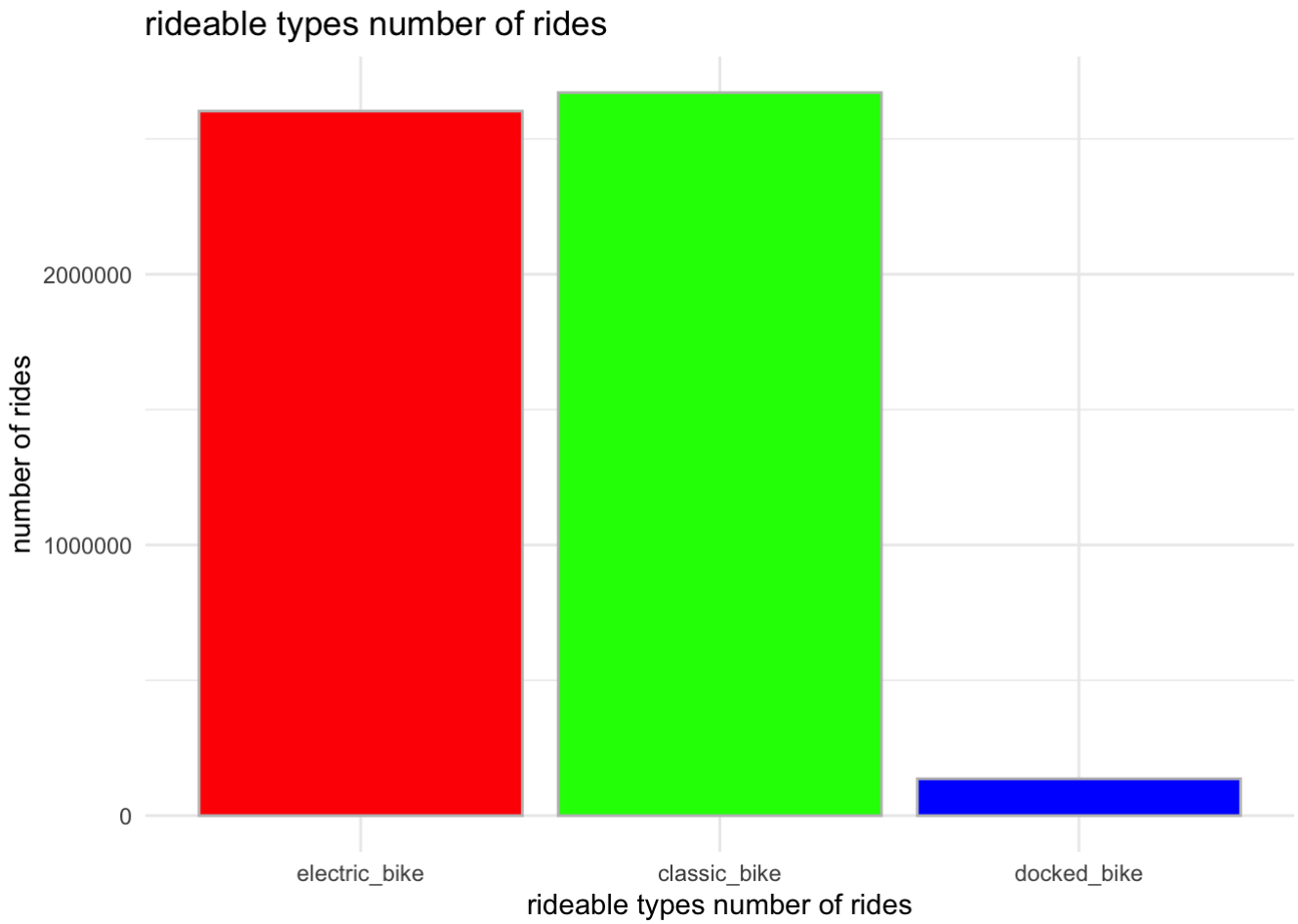

```
#### Concussion 2 ----
# We can see the average ridding time for casual riders is 30% more than the
# annual members. but it does not mean more income. because the trip fees is
# 45 mins which include wide range of ridding lenghts for the same fees

### Who really ride more Cyclistics or Casual Riders ----
df %>%
  select(member, minutes) %>%
  group_by(member) %>%
  summarise(member_ride_time = prettyNum(sum(minutes), big.mark = ",")) %>%
  data.frame() %>%
  ggplot(aes(x = member,
             y = member_ride_time)) +
  geom_col(color = c("red", "green"),
           fill = c("red", "green")) +
  geom_text(
    aes(label = member_ride_time),
    angle = 0,
    color = "white",
    vjust = 2,
    hjust = .5,
    size = 4
  ) +
  ggtitle("member type total ridding time (minutes)") +
  labs(x = "member type",
       y = "ridding time by each member group")
```



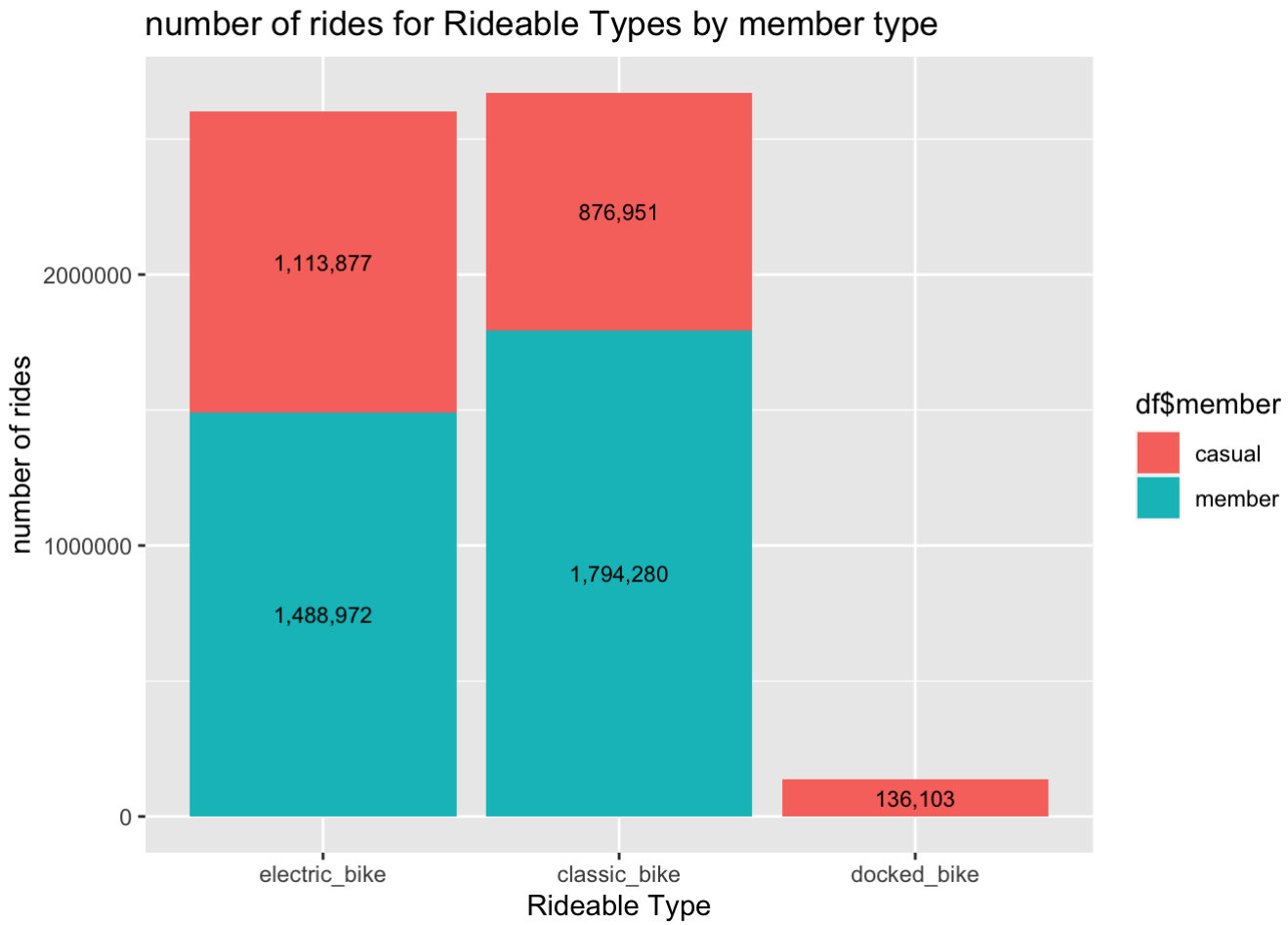

```
#### Concussion 3 ----
# Again the annual members rides around 6 million minutes logner than the
# casual members which is a lot of income.

### Rideabe Types: Do They Have The Same Demand ----
df %>%
  ggplot() +
  geom_bar(
    mapping = aes(x = type),
    color = "grey",
    fill = c("red", "green", "blue")
  ) +
  theme_minimal () +
  ggtitle("rideable types number of rides") +
  labs (x = "rideable types number of rides",
        y = "number of rides")
```



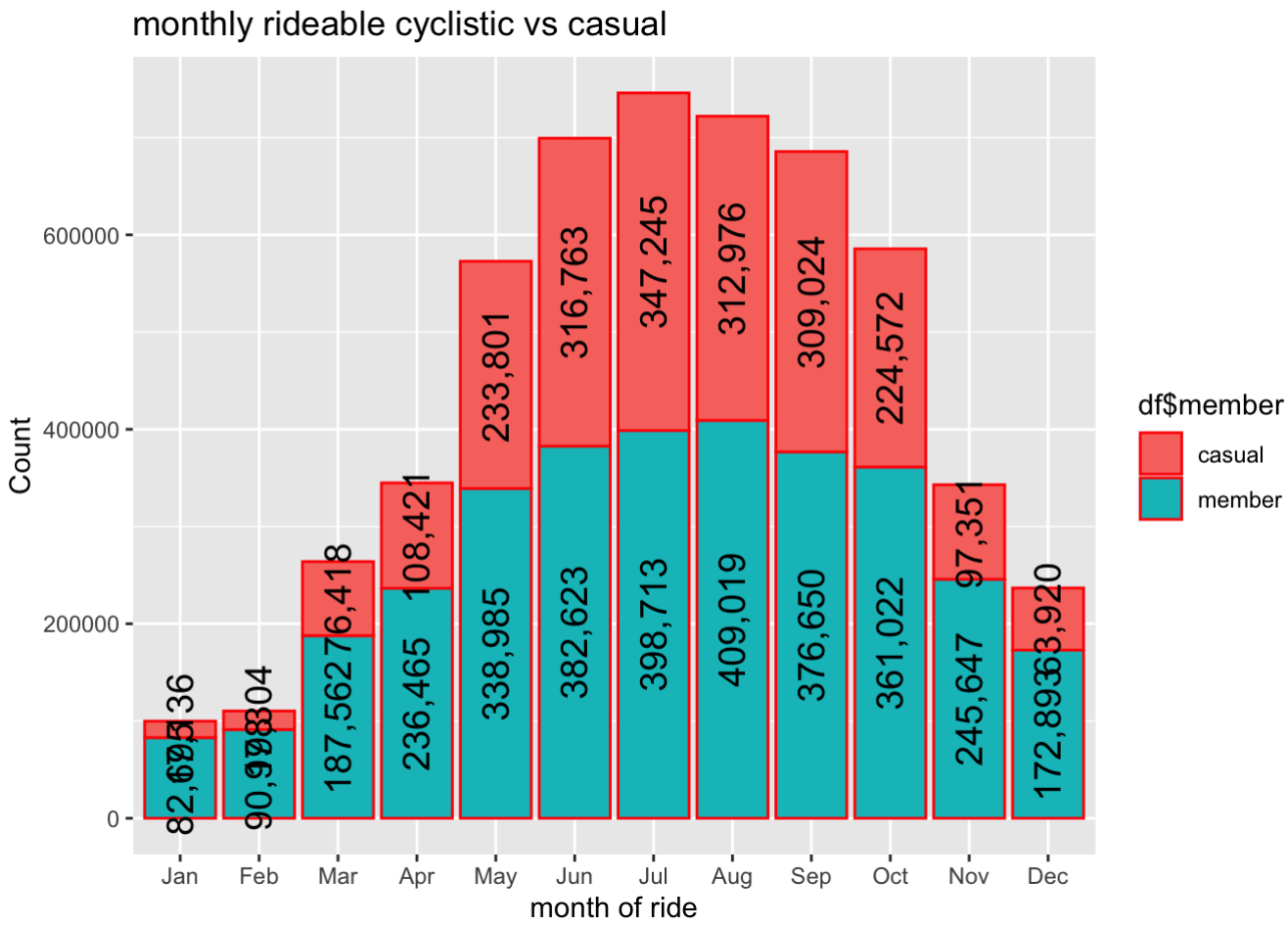

```
#### Concussion 4 ----
# Looking at the rideable types. most of riders (annual and casual) prefer
# electric and classic bikes over docked. does it because there is shortage
# of docked bikes availability! let's see who use docked bikes more.

### What Rideable Type each Membership group prefer? ----
qplot(
  df$type,
  geom = "bar",
  fill = df$member,
  # alpha = .5,
  # color = I("red"),
  xlab = "Rideable Type",
  ylab = "number of rides",
  main = "number of rides for Rideable Types by member type"
) +
  stat_count(
    geom = "text",
    size = 3,
    aes(label = prettyNum (
      stat(count),
      big.mark = ",",
      scientific = FALSE
    )),
    position = position_stack(vjust = .5),
    colour = "black")
```



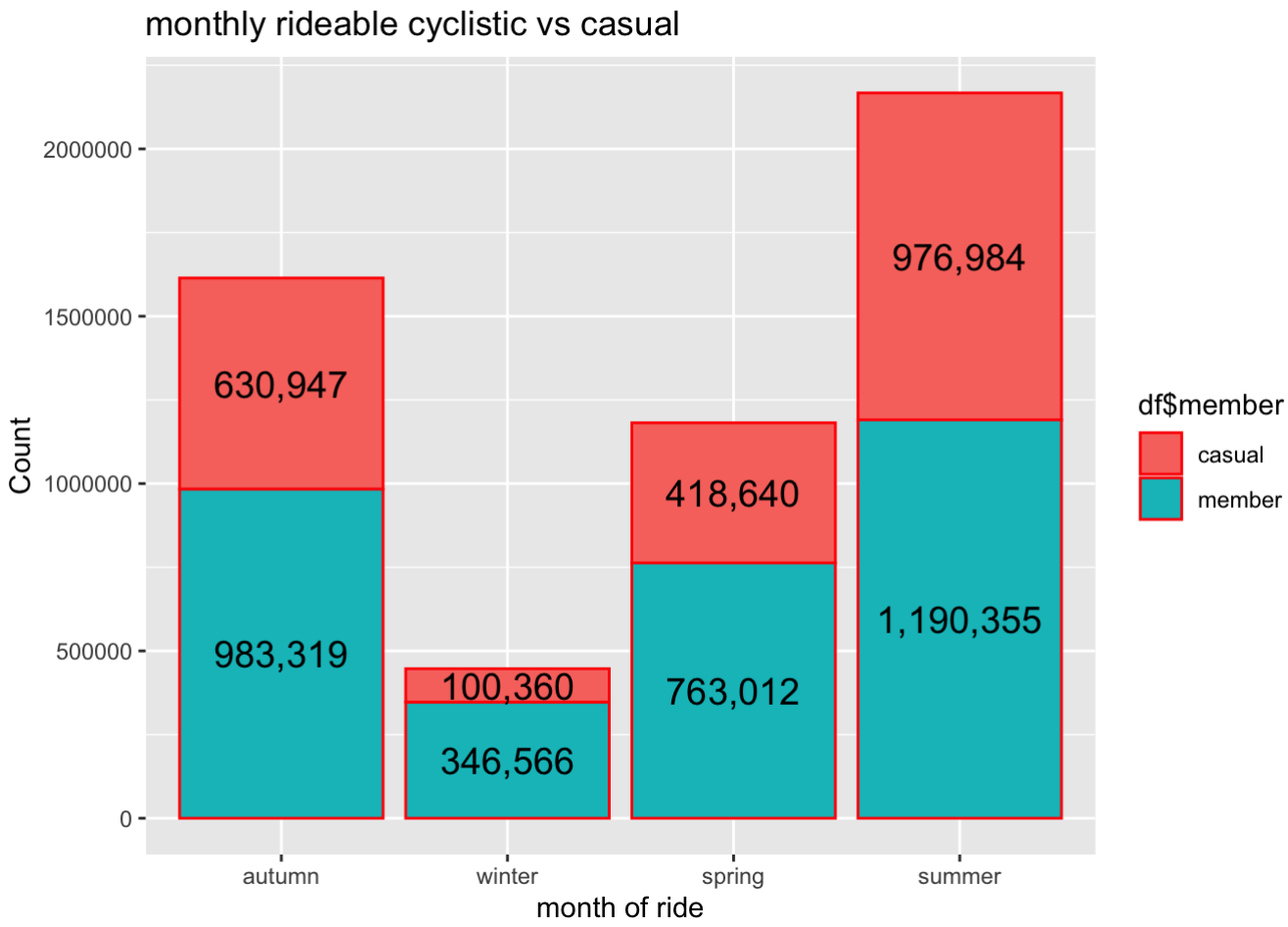

```
#### Concussion 5 ----
# Casual riders use all bikes types and annual members use only electric and
# classic bikes. again is it because there is shortage in docked bikes.
# anyway looks like the annual member does not prefer docked bikes because
# they are note represented in docked bike column

### Casual Cyclistic: Who Will Disappear In The Winter Tough Times! ----
qplot(
  x = df$month,
  geom = "bar",
  fill = df$member,
  # alpha = .5,
  color = I("red"),
  xlab = "month of ride",
  ylab = "Count",
  main = "monthly rideable cyclistic vs casual"
) +
  stat_count(
    geom = "text",
    size = 5,
    aes(label = prettyNum (
      stat(count),
      big.mark = ",",
      scientific = FALSE
    )),
    position = position_stack(vjust = .5),
    colour = "black",
    angle = 90)
```



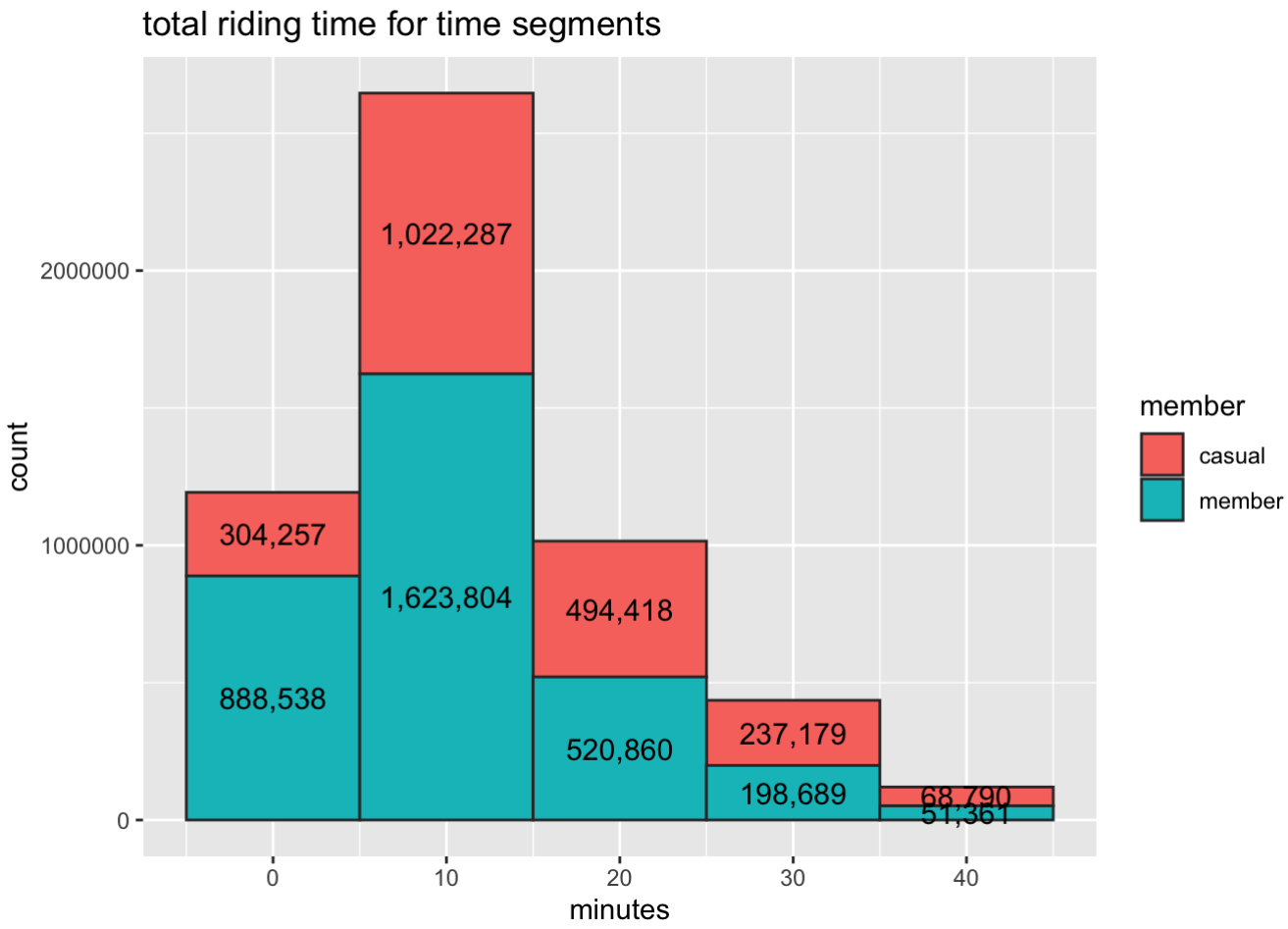

```
#### Concussion 6 ----
# Through the year month, the annual members are fairly represented. even
# in the worst season (Winter) they almost 50% of maximum. on the other hand
# the casual riders disappear almost a 3 month with is horrible disturbance \
# for income. even the other 9 month they are less than annual members.

### Bar Plot Membership Type ----
qplot(
  x = df$season,
  geom = "bar",
  fill = df$member,
  # alpha = .5,
  color = I("red"),
  xlab = "month of ride",
  ylab = "Count",
  main = "monthly rideable cyclistic vs casual"
) +
  stat_count(
    geom = "text",
    size = 5,
    aes(label = prettyNum (
      stat(count),
      big.mark = ",",
      scientific = FALSE
    )),
    position = position_stack(vjust = .5),
    colour = "black",
    angle = 0)
```




```
#### Concussion 7 ----
# Looking to the seasons of the year. annual member always represented good.
# casual presence on the other hand is not stable and they disappear in
# Winter.

### Who Really Rides Longer in Segments? ----
ggplot(aes(x = minutes),
       data = df) +
  geom_histogram(aes(fill = member),
                 binwidth = 10,
                 colour = "grey20") +
  stat_bin(
    binwidth = 10,
    geom = "text",
    colour = "black",
    size = 4,
    aes(label = prettyNum(..count.., big.mark = ","), group = member),
    position = position_stack(vjust = 0.5),
    angle = 0
  ) +
  ggtitle("total riding time for time segments")
```


total riding time for time segments
count
2000000
1000000
0
1,022,287
304,257
1,623,804
494,418
888,538
520,860
237,179
198,689
68,790
51,361
0
10
20
30
40
minutes
member
casual
member

```
#### Concussion 8 ----
# if we look to the segments of ridding most of riders are ridding between
# 10 and 20 minutes. however most of the shorter rides are by annual members
# that mean more bikes availability and less supporting Divvy staff needed to
# handle the used bikes.

## export summaries to use in in Tableau ----

### Export Average Ridding Time For Each Rideable Type ----
df %>%
  select(type, minutes) %>%
  group_by(type) %>%
  summarise("avg ridding time" = round(mean(minutes))) %>%
  data.frame() %>%
  rename("rideable type" = type,
         "avg ridding time (minutes)" = avg.ridding.time) %>%
  export("exports/avg ridding time by rideable type.csv", row.names = FALSE)

### Export Count Of Rides For Each Rideable Type ----
df %>%
  select(type, id) %>%
  group_by(type) %>%
  summarise("count ridding time" = length(id)) %>%
  data.frame() %>%
  rename("rideable type" = type,
         "count of rides" = count.ridding.time) %>%
  export("exports/count of rides by rideable type.csv", row.names = FALSE)

### Export Rideabe Types Ridding Time Count, Average, and Median ----
df %>%
  select(day, minutes, started_at, member) %>%
  mutate(weekday_number = as.integer(wday(started_at))) %>%
  group_by(day, weekday_number, member) %>%
  summarise(
    count_of_rides = length(minutes),
    mean_of_ridding_minutes = as.integer(mean(minutes)),
    median_of_ridding_minutes = as.integer(median(minutes))
  ) %>%
  arrange(as.integer(weekday_number)) %>%
  within(rm(weekday_number)) %>%
  rename(
    "day name" =  day,
    "nu. of rides" = count_of_rides,
    "avg ridding (mins)" = mean_of_ridding_minutes,
    "median ridding (mins)" = median_of_ridding_minutes
  ) %>%
  export("exports/rideabel types.csv", row.names = FALSE)
```

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

```
### Export Membership Type Summary ----
df %>%
  select(member) %>%
  table() %>%
  data.frame() %>%
  rename("nu. observations" = Freq,
         "member type" = "member") %>%
  export("exports/member_casual.csv", row.names = FALSE)

### Export Membership Type Average Ridding Time ----
df %>%
  select(member, minutes) %>%
  group_by(member) %>%
  summarise(member_type_avg_ridding_minutes = round(mean(minutes), digits = 0)) %>%
  data.frame() %>%
  export("exports/member_type_avg_ridding_minutes.csv", row.names = FALSE)

### Export Membership Type Average Ridding Time By Rideable Type ----
df %>%
  select(member, type) %>%
  table() %>%
  data.frame() %>%
  rename(
    "member type" = member,
    "rideable type" = type,
    "nu. observations" = Freq
  ) %>%
  export("exports/riding_time_by_member_and_rideable_type.csv",
         row.names = FALSE)

### Export Membership Type Yearly Monthly Average Ridding Time ----
df %>%
  mutate(month_num = month(started_at, label = F)) %>%
  select(member,
         season,
         month,
         minutes,
         month_num)  %>%
  group_by(season, month, month_num, member) %>%
  summarise(mean_riding_time = mean(minutes)) %>%
  arrange(month_num) %>%
  mutate(mean_riding_time = round(mean_riding_time, digits = 0)) %>%
  rename("avg ridding time (mins)" = mean_riding_time) %>%
  within(rm(month_num)) %>%
  export("exports/member_type_riding_time_yearly_monthly.csv",
         row.names = FALSE)
```

```
## `summarise()` has grouped output by 'season', 'month', 'month_num'. You can
## override using the `.groups` argument.
```

```
### Export Membership Type Yearly Monthly Average Ridding Time ----
df %>%
  select(member,
         season,
         month,
         minutes,
         started_at) %>%
  mutate(month_num = month(started_at, label = F)) %>%
  group_by(member, season, month, month_num) %>%
  summarise(avg_ridding_time_minutes = round(mean(minutes))) %>%
  arrange(month_num) %>%
  mutate(month_num = NULL) %>%
  export("exports/member_type_avg_rididng_time_yearly_monthly.csv",
         row.names = FALSE)
```

```
## `summarise()` has grouped output by 'member', 'season', 'month'. You can
## override using the `.groups` argument.
```

```
### Export Membership Type Yearly Monthly Count of Rids ----
df %>%
  mutate(month_num = month(started_at, label = F)) %>%
  select(id,
         member,
         season,
         month,
         minutes,
         month_num)  %>%
  group_by(season, month, month_num, member) %>%
  summarise(count_of_rides = length(id)) %>%
  arrange(month_num) %>%
  within(rm(month_num)) %>%
  export("exports/member_type_count_of_rides_yearly_monthly.csv",
         row.names = FALSE)
```

```
## `summarise()` has grouped output by 'season', 'month', 'month_num'. You can
## override using the `.groups` argument.
```

```
### Export Membership Type Yearly Monthly Count of Rids ----
df %>%
  select(member,
         season,
         month,
         minutes,
         started_at) %>%
  mutate(month_num = month(started_at, label = F)) %>%
  group_by(member, season, month, month_num) %>%
  summarise(count_of_rides = length(member)) %>%
  arrange(month_num) %>%
  mutate(month_num = NULL) %>%
  export("exports/member_type_rides_count_yearly_monthly.csv",
         row.names = FALSE)
```

```
## `summarise()` has grouped output by 'member', 'season', 'month'. You can
## override using the `.groups` argument.
```

```
### Export Membership Type Geographical Distribution ----
# member_type_riding_lat_long
# this huge goe points will be difficult to show on tableau
# so I will use a sample instead of the total population
# for population 5410183, Confidence Level 95%, Margin of Error 5%
# we can use sample of 385
# https://www.surveymonkey.com/mp/sample-size-calculator/
# according to survey monkey sample calc
df %>%
  select(id, member, slat, elat, slng, elng, minutes) %>%
  rename(
    "member type" = member,
    "start latitude" = slat,
    "start longtude" = slng,
    "end latitude" = elat,
    "end longtude" = elng
  ) %>%
  sample_n(385) %>%
  export("exports/member_type_riding_lat_long.csv", row.names = FALSE)

## CLEAN UP ----

### Clear environment ----
rm(list = ls())

### Clear packages ----
p_unload(all)  # Remove all add-ons
```

```
## The following packages have been unloaded:
## chron, hydroTSM, xts, zoo, lubridate, vctrs, forcats, stringr, dplyr, purrr, read
r, tidyr, tibble, ggplot2, tidyverse, rio, pacman
```

```
### Clear console ----
cat("\014")  # ctrl+L
```