Cyclistic - Capstone project 1

Zacharie Ndoumga

2022-03-15

This is document is produced for the capstone project at the end of the Google Professional Data Analyst Certificate program. on Coursera. Data cleaning and processing will be carried out here, then the cleaned document will be exported for analysis on other programs, precisely Tabeau, and a final report/presentation will also be done (Look into the github repository).

Libraries

For this project, we will need a handful of libraries. These will be installed and executed

```r
#install.packages("tidyverse")
#install.packages("here")
#install.packages("purr")
#install.packages("skimr")
#install.packages("janitor")
#install.packages("lubridate")

library(tidyverse)
library(here)
library(skimr)
library(janitor)
library(lubridate)
library(ggplot2)
library(dplyr)
library(scales)
```

Importing the data

Now that our environment is set up, the csv files with the data are uploaded.

```r
setwd("~/Documents/Data Analytics/Portfolio Projects/Track 1 - Bicycle/Dataset/12 last months/CSV")

y2021_03 <- read.csv("202103-divvy-tripdata.csv")
y2021_04 <- read.csv("202104-divvy-tripdata.csv")
y2021_05 <- read.csv("202105-divvy-tripdata.csv")
y2021_06 <- read.csv("202106-divvy-tripdata.csv")
y2021_07 <- read.csv("202107-divvy-tripdata.csv")
y2021_08 <- read.csv("202108-divvy-tripdata.csv")
y2021_09 <- read.csv("202109-divvy-tripdata.csv")
y2021_10 <- read.csv("202110-divvy-tripdata.csv")
y2021_11 <- read.csv("202111-divvy-tripdata.csv")
y2021_12 <- read.csv("202112-divvy-tripdata.csv")
y2022_01 <- read.csv("202201-divvy-tripdata.csv")
y2022_02 <- read.csv("202202-divvy-tripdata.csv")
```

The next step is to combine the all the files into one file to form our data frame. (PS: The data was checked prior to determine the framework on all the files were similar and could thus be combined)

Data cleaning and processing

The data frame is checked to see iff all the data types are conguent with what they should be.

```
## 'data.frame':    5667986 obs. of  13 variables:
##  $ ride_id           : chr  "CFA86D4455AA1030" "30D9DC61227D1AF3" "846D87A15682A284" "994D05AA75A168F2" ...
##  $ rideable_type     : chr  "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2021-03-16 08:32:30" "2021-03-28 01:26:28" "2021-03-11 21:17:29" "2021-03-11 13:2
6:42" ...
##  $ ended_at          : chr  "2021-03-16 08:36:34" "2021-03-28 01:36:55" "2021-03-11 21:33:53" "2021-03-11 13:5
5:41" ...
##  $ start_station_name: chr  "Humboldt Blvd & Armitage Ave" "Humboldt Blvd & Armitage Ave" "Shields Ave & 28th
Pl" "Winthrop Ave & Lawrence Ave" ...
##  $ start_station_id  : chr  "15651" "15651" "15443" "TA1308000021" ...
##  $ end_station_name  : chr  "Stave St & Armitage Ave" "Central Park Ave & Bloomingdale Ave" "Halsted St & 35th
St" "Broadway & Sheridan Rd" ...
##  $ end_station_id    : chr  "13266" "18017" "TA1308000043" "13323" ...
##  $ start_lat         : num  41.9 41.9 41.8 42 42 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.8 42 42.1 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
```

It is noticed that the start and end times are in string types. They will be change to time formats. A few column names also will be changed to be a bit more intuitive and making the rest of the process easier.

```r
## renaming a few columns
full_year <- rename(full_year, bike_type = rideable_type, start_time = started_at, end_time = ended_at, user_type
= member_casual)

## converting start and end time to time formats
full_year <- mutate(full_year, start_time = strptime(start_time, "%Y-%m-%d %H:%M:%S"), end_time = strptime(end_ti
me, "%Y-%m-%d %H:%M:%S"))
```

Now the data type is fixed. To provide a higher level of granularity when analyzing the data, the start time will be broken down into time of the day, week day and month. The ride length will also be calculated by taking the difference of the end and start time. Since the data will be altered, a new variable will be created to house the modified version.

```r
full_year_v1 <- mutate(full_year, ride_length = as.numeric(difftime(end_time, start_time, units = "mins")), hours
= format(start_time, "%H"), week_day = weekdays(start_time), month_day = format(start_time, "%d"), month = month(
start_time, label = TRUE))

str(full_year_v1)
```

```
## 'data.frame':    5667986 obs. of  18 variables:
##  $ ride_id           : chr  "CFA86D4455AA1030" "30D9DC61227D1AF3" "846D87A15682A284" "994D05AA75A168F2" ...
##  $ bike_type         : chr  "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ start_time        : POSIXlt, format: "2021-03-16 08:32:30" "2021-03-28 01:26:28" ...
##  $ end_time          : POSIXlt, format: "2021-03-16 08:36:34" "2021-03-28 01:36:55" ...
##  $ start_station_name: chr  "Humboldt Blvd & Armitage Ave" "Humboldt Blvd & Armitage Ave" "Shields Ave & 28th
Pl" "Winthrop Ave & Lawrence Ave" ...
##  $ start_station_id  : chr  "15651" "15651" "15443" "TA1308000021" ...
##  $ end_station_name  : chr  "Stave St & Armitage Ave" "Central Park Ave & Bloomingdale Ave" "Halsted St & 35th
St" "Broadway & Sheridan Rd" ...
##  $ end_station_id    : chr  "13266" "18017" "TA1308000043" "13323" ...
##  $ start_lat         : num  41.9 41.9 41.8 42 42 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.8 42 42.1 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ user_type         : chr  "casual" "casual" "casual" "casual" ...
##  $ ride_length       : num  4.07 10.45 16.4 28.98 17.93 ...
##  $ hours             : chr  "08" "01" "21" "13" ...
##  $ week_day          : chr  "Tuesday" "Sunday" "Thursday" "Thursday" ...
##  $ month_day         : chr  "16" "28" "11" "11" ...
##  $ month             : Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<..: 3 3 3 3 3 3 3 3 3 3 ...
```

The days of the week are not listed in a chronological order. This has to be fixed

Now the data is checked for NA rows.

```r
summary(full_year_v1)
```

```
##    ride_id           bike_type           start_time
##  Length:5667986     Length:5667986     Min.   :2021-03-01 00:01:09
##  Class :character   Class :character   1st Qu.:2021-06-13 11:43:00
##  Mode  :character   Mode  :character   Median :2021-08-07 19:13:29
##                                        Mean   :2021-08-10 07:46:57
##                                        3rd Qu.:2021-10-02 14:16:29
##                                        Max.   :2022-02-28 23:58:44
##                                        NA's   :59
##     end_time                     start_station_name start_station_id
##  Min.   :2021-03-01 00:06:28   Length:5667986     Length:5667986
##  1st Qu.:2021-06-13 12:11:14   Class :character   Class :character
##  Median :2021-08-07 19:36:42   Mode  :character   Mode  :character
##  Mean   :2021-08-10 08:10:11
##  3rd Qu.:2021-10-02 14:39:36
##  Max.   :2022-03-01 08:55:17
##  NA's   :102
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:5667986     Length:5667986     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        user_type          ride_length
##  Min.   :41.39   Min.   :-88.97   Length:5667986     Min.   :  -58.03
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character   1st Qu.:    6.67
##  Median :41.90   Median :-87.64   Mode  :character   Median :   11.87
##  Mean   :41.90   Mean   :-87.65                      Mean   :   21.75
##  3rd Qu.:41.93   3rd Qu.:-87.63                      3rd Qu.:   21.57
##  Max.   :42.17   Max.   :-87.49                      Max.   :55944.15
##  NA's   :4617    NA's   :4617                        NA's   :114
##     hours              week_day        month_day               month
##  Length:5667986     Monday   :723358   Length:5667986     Jul    : 822410
##  Class :character   Tuesday  :755914   Class :character   Aug    : 804352
##  Mode  :character   Wednesday:767510   Mode  :character   Sep    : 756147
##                     Thursday :746267                      Jun    : 729595
##                     Friday   :815404                      Oct    : 631226
##                     Saturday :991967                      May    : 531633
##                     Sunday   :867566                      (Other):1392623
```

As expected, there are quite a few NA rows. As the longitudes and latitudes are not of importance in our analysis, we are not going to drop those. However, the rows with NA start time, end time and trip duration will be dropped. The data is agained modified, so a new version variable of the data is created.

```r
full_year_v2 <- drop_na(full_year_v1, start_time, end_time, ride_length)
```

The data is made up of some rows with negative ride times. This is the case for when the bikes are taken out for maintenance. These rows will be dropped as well.

```r
full_year_v3 <- subset(full_year_v2, ride_length > 0)
```

Now the data is ready for the analyzing.

```r
df <- full_year_v3
```

Analyzing the data

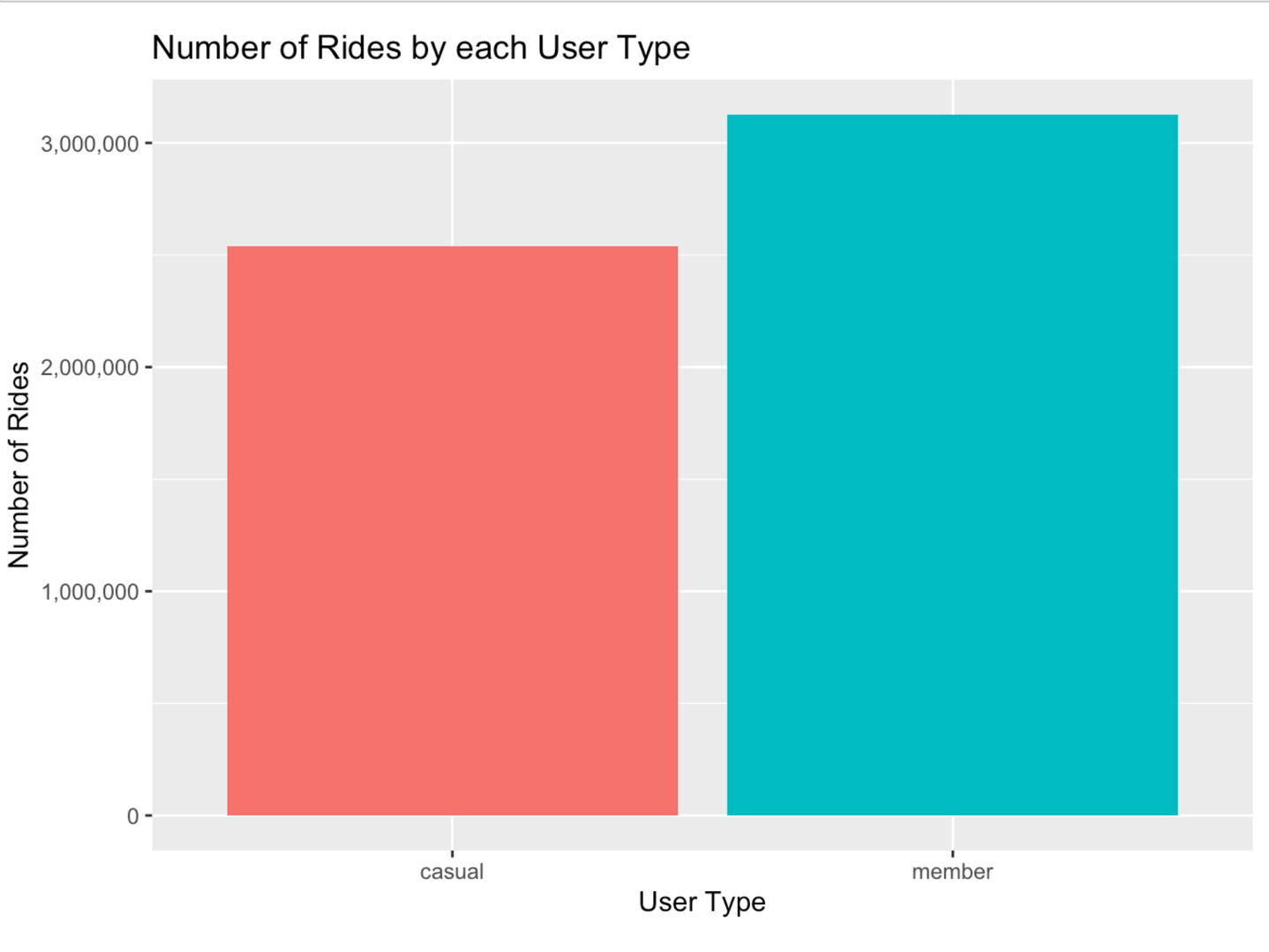
Total number of rides

```r
summarise(df, number_of_rides = n_distinct(ride_id))
```

```
##   number_of_rides
## 1         5667219
```

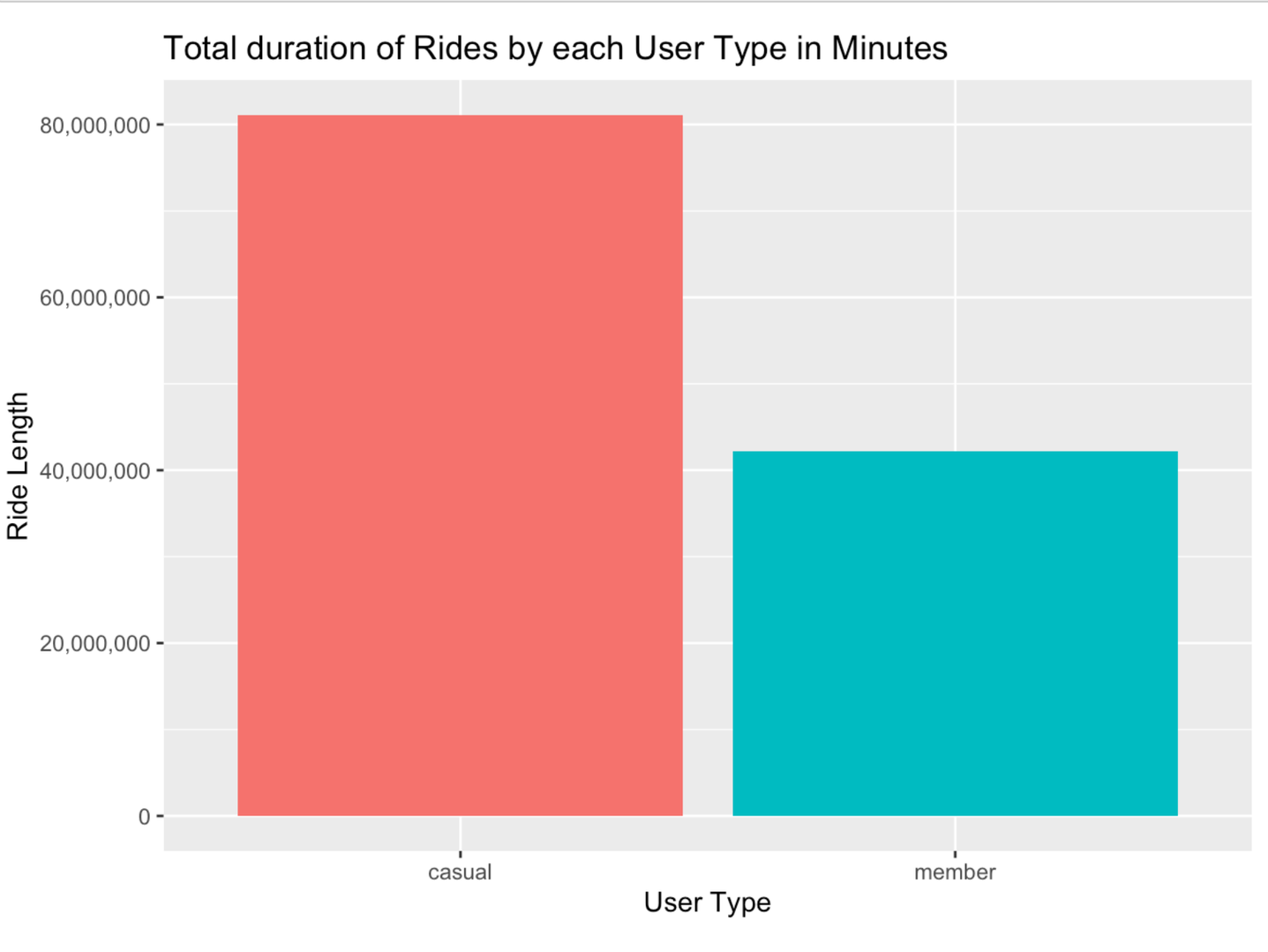
Number of rides by user type

```r
df %>% group_by(user_type) %>%
  summarise(number_of_rides = n()) %>%
  ggplot( aes(x = user_type, y=number_of_rides, fill=user_type)) +
  geom_col() + labs(title ="Number of Rides by each User Type", x = "User Type" , y="Number of Rides")  + theme(l
egend.position ="none") + scale_y_continuous(labels = comma)
```



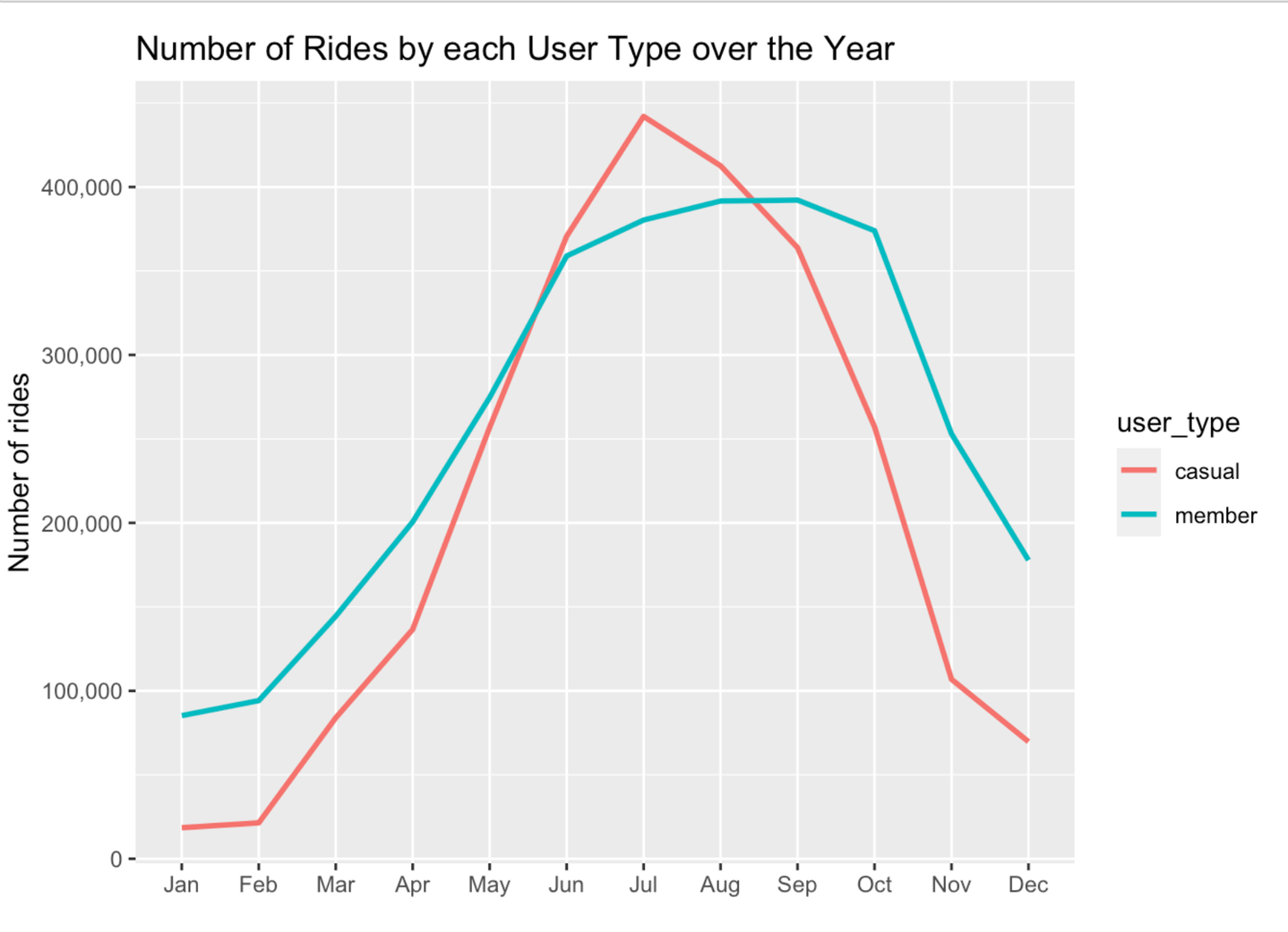
Total ride time by user type over the last 12 months

```r
df %>% group_by(user_type) %>%
  summarise(total_ride_length = sum(ride_length)) %>%
  ggplot( aes(x = user_type, y=total_ride_length, fill=user_type)) +
  geom_col() + labs(title ="Total duration of Rides by each User Type in Minutes", x = "User Type" , y="Ride Leng
th") + theme(legend.position ="none") + scale_y_continuous(labels = comma)
```



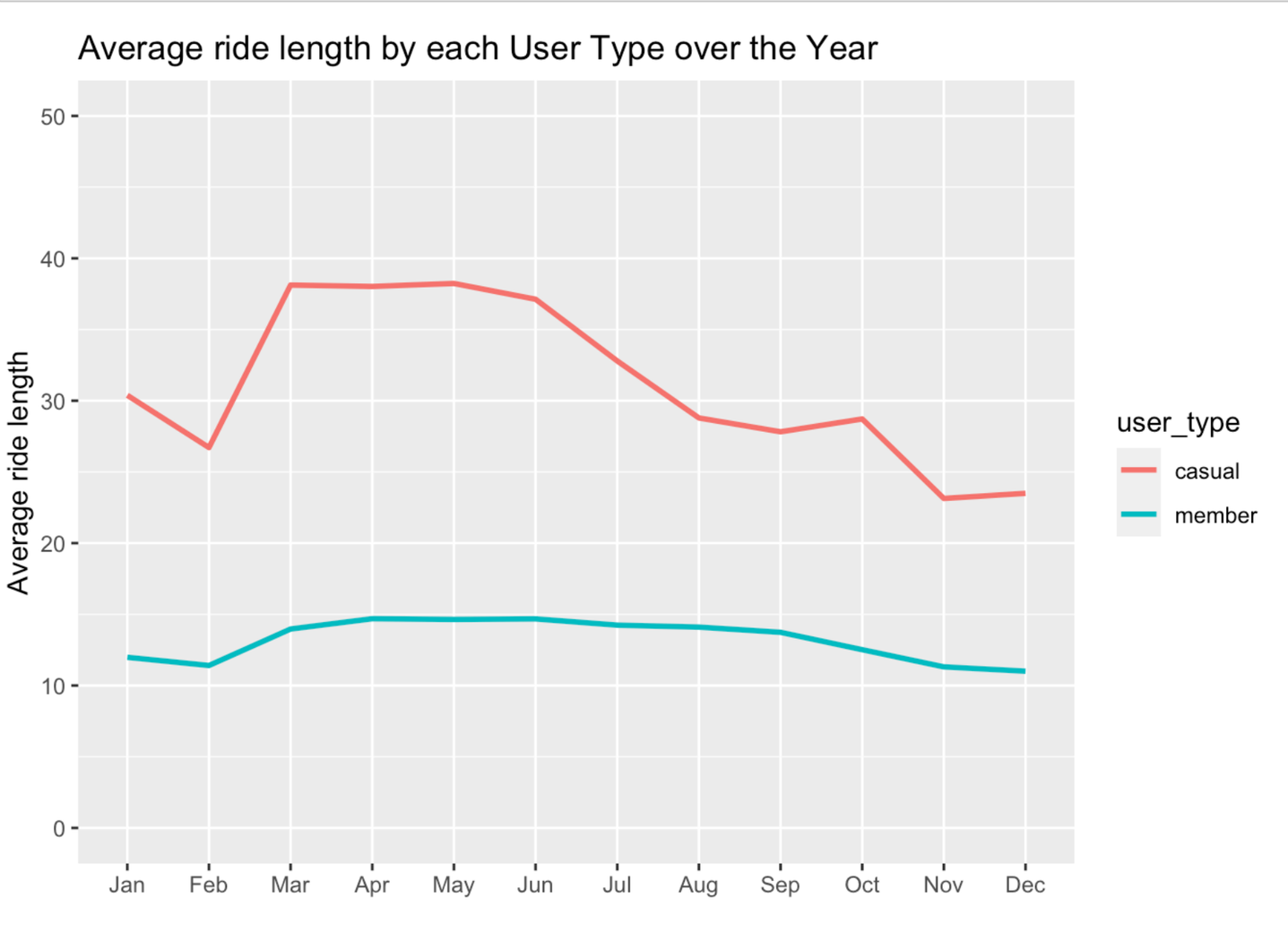
Rides by user type over the past 12 months

```r
df %>% group_by(month,user_type) %>%
  summarise(number_of_rides = n()) %>%
  ggplot( aes(x = month, y=number_of_rides, group= user_type, color=user_type)) +
  geom_line(size=1) + labs(title ="Number of Rides by each User Type over the Year", x = "" , y="Number of rides"
) + scale_y_continuous(labels = comma)
```



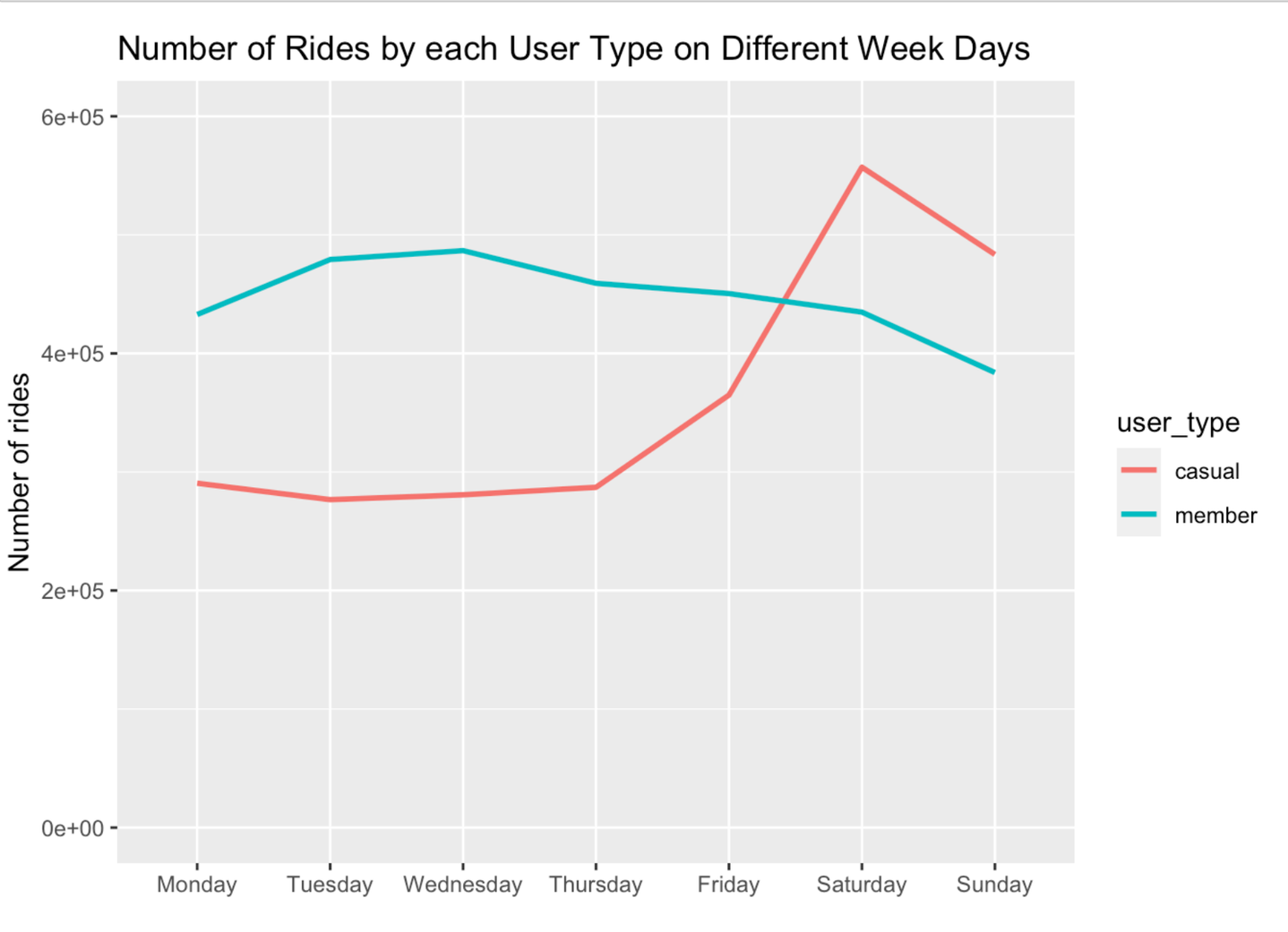
Average ride length by user type over the last 12 months

```r
df %>%
  group_by(month, user_type) %>%
  summarise(avg_ride_length = mean(ride_length)) %>%
  ggplot( aes(x = month, y=avg_ride_length, group=user_type, color=user_type)) +
  geom_line(size=1) + ylim(0, 50) + labs(title ="Average ride length by each User Type over the Year", x = "" , y
="Average ride length")
```



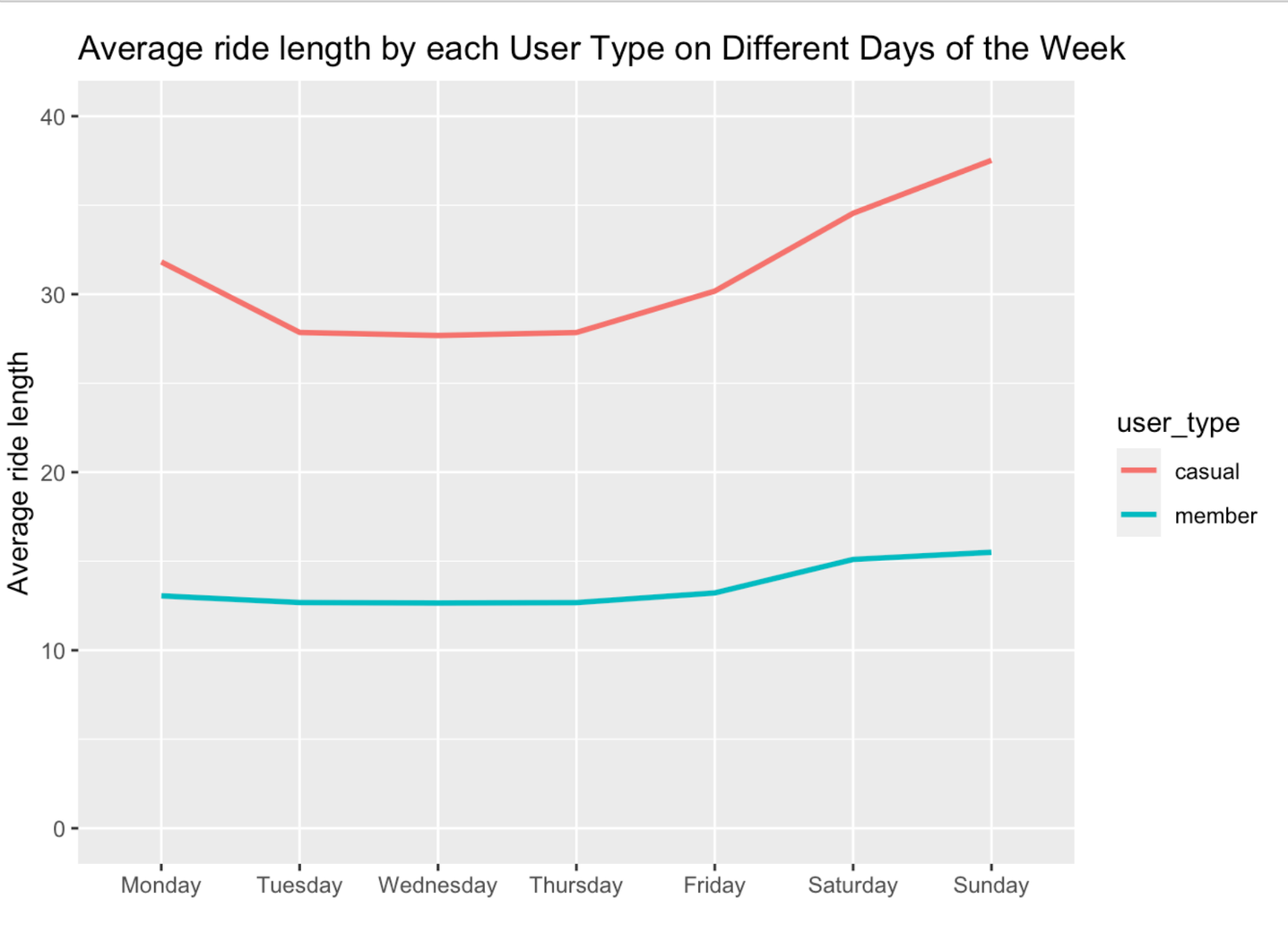
Rides by user type on the different days of the week

```r
df %>% group_by(week_day, user_type) %>%
  summarise(number_of_rides = n()) %>%
  ggplot( aes(x = week_day, y=number_of_rides, group = user_type, color =user_type)) +
  geom_line(size=1) + ylim(0, 600000) +  labs(title ="Number of Rides by each User Type on Different Week Days",
x = "" , y="Number of rides")
```



Average ride duration by user type on the different days of the week

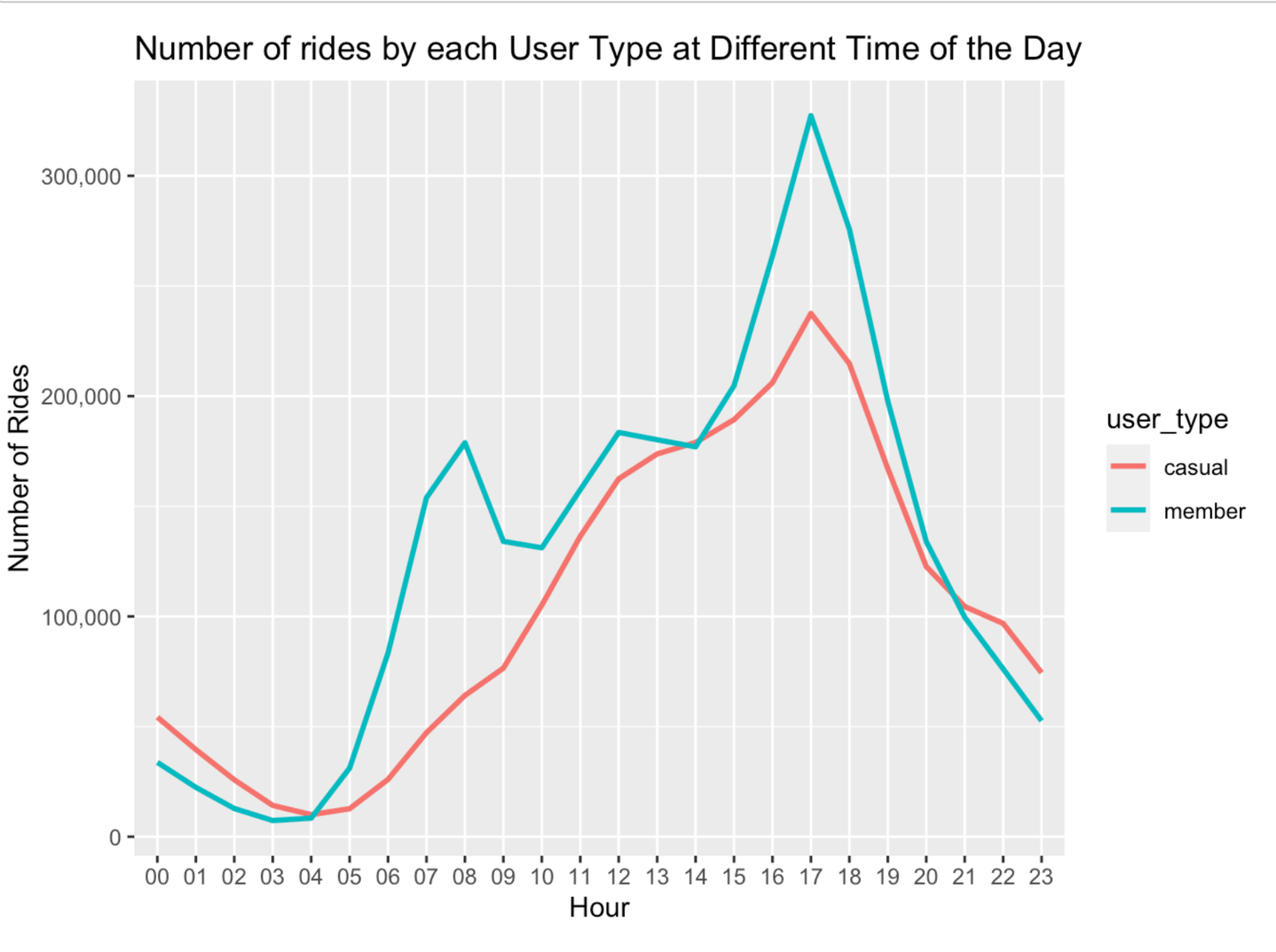
```r
df %>% group_by(week_day, user_type) %>%
  summarise(avg_ride_length = mean(ride_length)) %>%
  ggplot( aes(x = week_day, y=avg_ride_length, group = user_type, color =user_type)) +
  geom_line(size=1) + ylim(0, 40) +  labs(title ="Average ride length by each User Type on Different Days of the
Week", x = "" , y="Average ride length")
```



Rides by user type over the hours of the day

```r
df %>% group_by(hours, user_type) %>%
  summarise(number_of_rides = n()) %>%
  ggplot( aes(x = hours, y=number_of_rides, group = user_type, color =user_type)) +
  geom_line(size=1) +  labs(title ="Number of rides by each User Type at Different Time of the Day", x = "Hour" ,
y="Number of Rides") + scale_y_continuous(labels = comma)
```

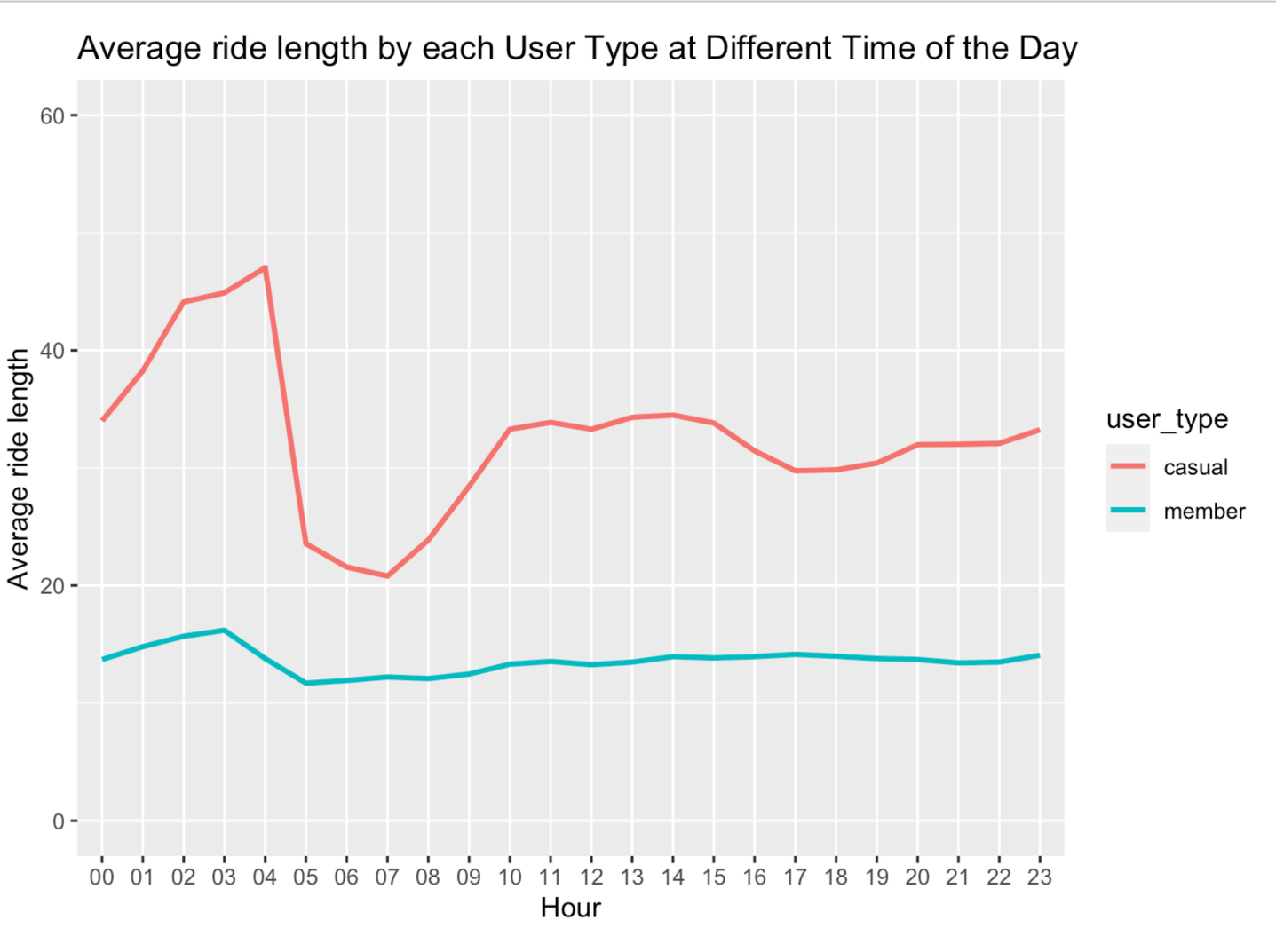
```
## `summarise()` has grouped output by 'hours'. You can override using the
## `.groups` argument.
```



Average ride duration by user type over the different days of the day

```r
df %>%
  group_by(hours, user_type) %>%
  summarise(avg_ride_length = mean(ride_length)) %>%
  ggplot( aes(x = hours, y=avg_ride_length, group = user_type, color =user_type)) +
  geom_line(size=1) + ylim(0, 60) +  labs(title ="Average ride length by each User Type at Different Time of the
Day", x = "Hour" , y="Average ride length")
```

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




~~~
We are done with R Studio. The rest of the analysis will be done on other platforms and the report will be produc
es.


```r
#write_csv(df, "cyclistitic_full_year_clean")
~~~