Bike_share_case_study_md

Jordan Creenaune

8/21/2021

Google Data Analytics Certificate - Track 1 - Case study one.

This is a guided case study from module 8 of the Google Data Analytics Certificate.

Essential question - Case Study: How Does a Bike-Share Navigate Speedy Success?

- How are the different types of memberships using bikeshare programs represented in the data?
- What information can we draw from historical data that can help us understand the difference between casual riders and riders with memberships?

##STEP 1 - Install required packages

```
library(tidyverse)  #A series of packages used to wrangle data
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
## Registered S3 method overwritten by 'rvest':
##   method            from
##   read_xml.response xml2
```

```
## ── Attaching packages ─────────────────────────────────────────────────
──── tidyverse 1.2.1 ──
```

```
## ✔ ggplot2 3.1.1       ✔ purrr   0.3.2
## ✔ tibble  2.1.1       ✔ dplyr   0.8.0.1
## ✔ tidyr   0.8.3       ✔ stringr 1.4.0
## ✔ readr   1.3.1       ✔ forcats 0.4.0
```

```
## ── Conflicts ────────────────────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

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

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

```
## The following object is masked from 'package:base':
##
##     date
```

```
library(ggplot2)    #We will perform visualisations using this package
```

##STEP 2 - Import data sets and combine Set working directory and import datasets

```
setwd("/Users/jordancreenaune/Documents/R")

df_202007 <- read_csv("202007-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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()
## )
```

```
df_202008 <- read_csv("202008-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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()
## )
```

```
df_202009 <- read_csv("202009-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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()
## )
```

```
df_202010 <- read_csv("202010-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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()
## )
```

```
df_202011 <- read_csv("202011-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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()
## )
```

```
df_202012 <- read_csv("202012-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
df_202101 <- read_csv("202101-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
df_202102 <- read_csv("202102-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
df_202103 <- read_csv("202103-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
df_202104 <- read_csv("202104-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
df_202105 <- read_csv("202105-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

```
df_202106 <- read_csv("202106-divvy-tripdata.csv")
```

```
## Parsed with column specification:
## 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_character(),
##   end_station_name = col_character(),
##   end_station_id = col_character(),
##   start_lat = col_double(),
##   start_lng = col_double(),
##   end_lat = col_double(),
##   end_lng = col_double(),
##   member_casual = col_character()
## )
```

Combine datasets into a single df

```
tripdata <- rbind(df_202007,
                  df_202008,
                  df_202009,
                  df_202010,
                  df_202011,
                  df_202012,
                  df_202101,
                  df_202102,
                  df_202103,
                  df_202104,
                  df_202105,
                  df_202106)
glimpse(tripdata)
```

```
## Observations: 4,460,151
## Variables: 13
## $ ride_id            <chr> "762198876D69004D", "BEC9C9FBA0D4CF1B", "D2FD…
## $ rideable_type      <chr> "docked_bike", "docked_bike", "docked_bike", …
## $ started_at         <dttm> 2020-07-09 15:22:02, 2020-07-24 23:56:30, 20…
## $ ended_at           <dttm> 2020-07-09 15:25:52, 2020-07-25 00:20:17, 20…
## $ start_station_name <chr> "Ritchie Ct & Banks St", "Halsted St & Roscoe…
## $ start_station_id   <chr> "180", "299", "329", "181", "268", "635", "11…
## $ end_station_name   <chr> "Wells St & Evergreen Ave", "Broadway & Ridge…
## $ end_station_id     <chr> "291", "461", "156", "94", "301", "289", "140…
## $ start_lat          <dbl> 41.90687, 41.94367, 41.93259, 41.89076, 41.91…
## $ start_lng          <dbl> -87.62622, -87.64895, -87.63643, -87.63170, -…
## $ end_lat            <dbl> 41.90672, 41.98404, 41.93650, 41.91831, 41.90…
## $ end_lng            <dbl> -87.63483, -87.66027, -87.64754, -87.63628, -…
## $ member_casual      <chr> "member", "member", "casual", "casual", "memb…
```

##STEP 3- Clean Datasets Clean data - remove values that contain missing components in the data

```
# Remove rows with missing values
colSums(is.na(tripdata))
```

```
##            ride_id      rideable_type         started_at
##                  0                  0                  0
##           ended_at start_station_name   start_station_id
##                  0             282068             282694
##   end_station_name     end_station_id          start_lat
##             315109             315570                  0
##          start_lng            end_lat            end_lng
##                  0               5286               5286
##      member_casual
##                  0
```

```
# 5% of data with missing values will be removed
tripdata_cleaned <- tripdata[complete.cases(tripdata), ]

# data with started_at greater than ended_at will be removed - remove possible inconsist
encies
tripdata_cleaned <- tripdata_cleaned %>%
  filter(tripdata_cleaned$started_at < tripdata_cleaned$ended_at)
```

##STEP 4 - Analyse and manipulate data Create new column `ride_length` - Find the difference between ended_at and started_at

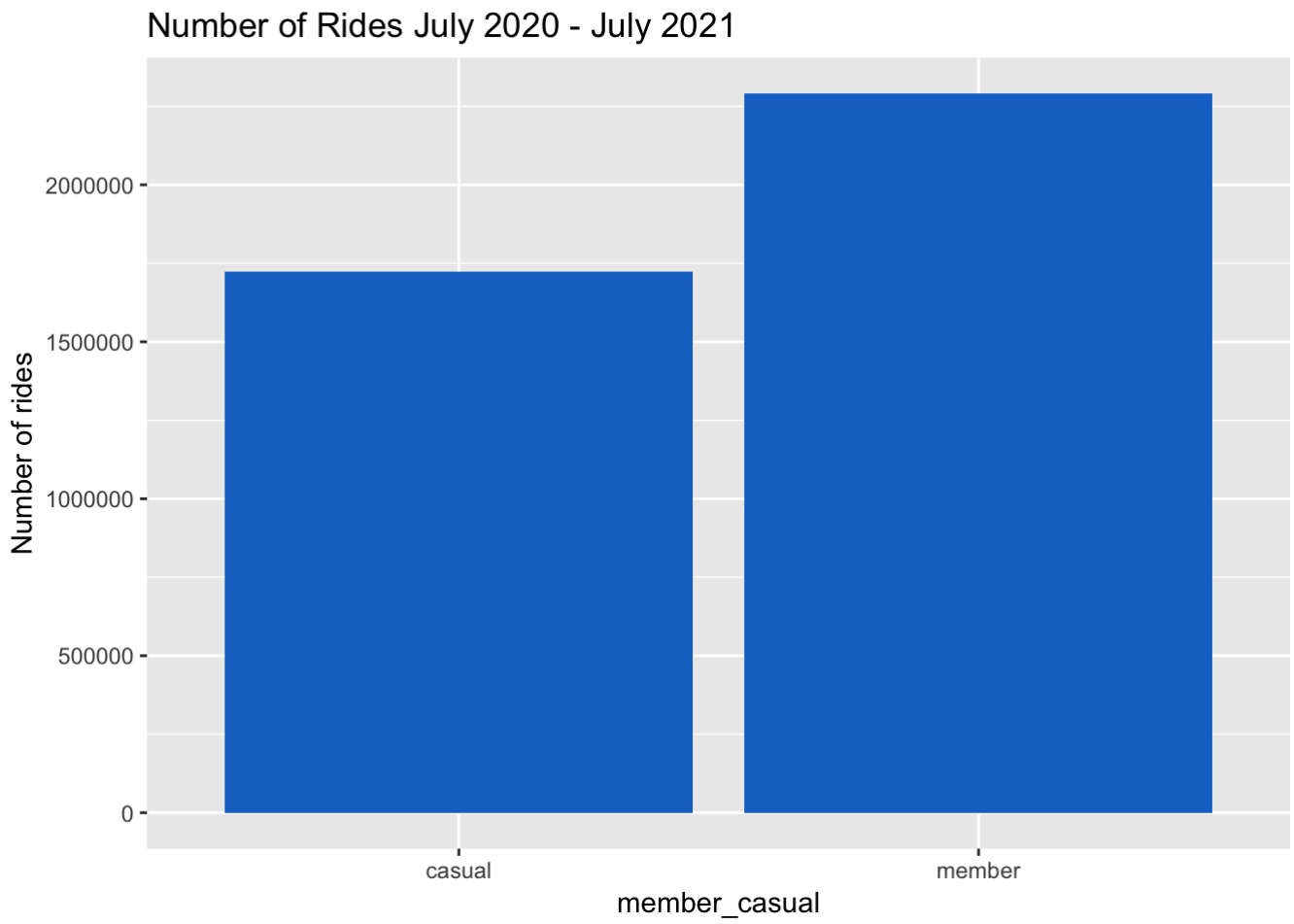
```
tripdata_cleaned$ride_length <- (tripdata_cleaned$ended_at - tripdata_cleaned$started_a
t)
head(tripdata_cleaned)
```

```
## # A tibble: 6 x 14
##   ride_id rideable_type started_at          ended_at
##   <chr>   <chr>         <dttm>              <dttm>
## 1 762198… docked_bike   2020-07-09 15:22:02 2020-07-09 15:25:52
## 2 BEC9C9… docked_bike   2020-07-24 23:56:30 2020-07-25 00:20:17
## 3 D2FD8E… docked_bike   2020-07-08 19:49:07 2020-07-08 19:56:22
## 4 54AE59… docked_bike   2020-07-17 19:06:42 2020-07-17 19:27:38
## 5 54025F… docked_bike   2020-07-04 10:39:57 2020-07-04 10:45:05
## 6 65636B… docked_bike   2020-07-28 16:33:03 2020-07-28 16:49:10
## # … with 10 more variables: start_station_name <chr>,
## #   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 <time>
```

Plot - demonstrating casual vs member riders for this data set

```
number_of_rides <- tripdata_cleaned %>%
                      group_by(member_casual) %>%
                      summarize(number_of_rides=n())


ggplot(number_of_rides, aes(x=member_casual, y=number_of_rides)) +
  geom_bar(stat = "identity",fill="dodgerblue3") +
  labs(title = "Number of Rides July 2020 - July 2021") +
  ylab("Number of rides")
```



Create 2 new columns `day_of_week` and 'day' using the lubridate package

```
tripdata_cleaned$day_of_week <- wday(tripdata_cleaned$started_at, label = FALSE)
tripdata_cleaned$day <- weekdays(as.Date(tripdata_cleaned$started_at))
```

Find mean - max - min - mode of ride_length

```
# mean of ride_length
mean_ride_length <- tripdata_cleaned %>%
  summarize(mean(ride_length))

# max ride_length
max_ride_length <- tripdata_cleaned %>%
  summarize(max(ride_length))

# min ride_length
min_ride_length <- tripdata_cleaned %>%
  summarize(min(ride_length))

#Mode of Column
#Create a mode function
getmode <- function(v) {
  uniqv <- unique(v)
  uniqv[which.max(tabulate(match(v, uniqv)))]
}

# Calculate the mode of day_of_week using the Mode function (from above)
day_mode <- getmode(tripdata_cleaned$day)
print(day_mode)
```

```
## [1] "Saturday"
```

```
day_of_week_mode <- getmode(tripdata_cleaned$day_of_week)
#print(day_of_week_mode)

data_summary <- data.frame(max_ride_length,min_ride_length,mean_ride_length,day_of_week_
mode)
print(data_summary)
```

```
##   max.ride_length. min.ride_length. mean.ride_length. day_of_week_mode
## 1     3356649 secs           1 secs     1587.753 secs                7
```

Average ride length between member and casual riders in seconds

```
# average ride_length for members and casual riders
tripdata_cleaned %>%
  group_by(member_casual) %>%
  summarize(mean(ride_length))
```

```
## # A tibble: 2 x 2
##   member_casual `mean(ride_length)`
##   <chr>         <time>
## 1 casual        2511.9573 secs
## 2 member         891.8916 secs
```

Average ride_length for users by day_of_week - inclusive of both membership types

```
average_d_o_w <- tripdata_cleaned %>%
                   group_by(day_of_week,day) %>%
                   summarize(mean(ride_length))

average_d_o_w$mean_ride_length <- round(average_d_o_w$`mean(ride_length)` ,digit=2)
average_d_o_w <- average_d_o_w[ -c(3) ]

average_d_o_w
```

```
## # A tibble: 7 x 3
## # Groups:   day_of_week [7]
##   day_of_week day       mean_ride_length
##         <dbl> <chr>     <time>
## 1           1 Sunday    2004.79 secs
## 2           2 Monday    1450.14 secs
## 3           3 Tuesday   1329.66 secs
## 4           4 Wednesday 1346.31 secs
## 5           5 Thursday  1362.01 secs
## 6           6 Friday    1526.52 secs
## 7           7 Saturday  1889.33 secs
```

##STEP 5- Visualisation

Visualisation of time series data Order data by date and time, isolate one column

```
date_order <- tripdata_cleaned[order(tripdata_cleaned$started_at),3,drop=FALSE ]
head(date_order) #Check data is in the correct order
```

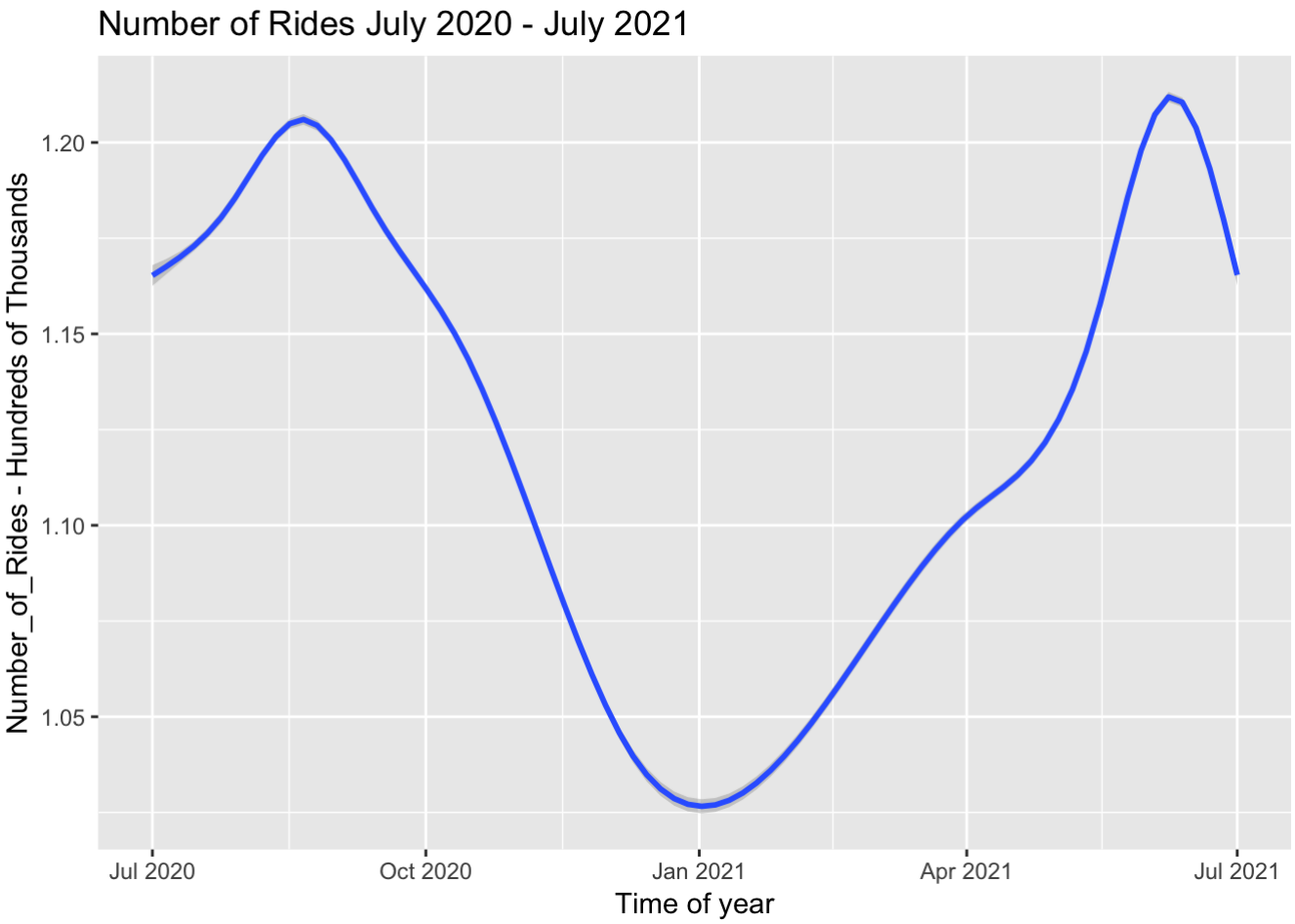
```
## # A tibble: 6 x 1
##   started_at
##   <dttm>
## 1 2020-07-01 00:00:14
## 2 2020-07-01 00:00:15
## 3 2020-07-01 00:00:49
## 4 2020-07-01 00:00:50
## 5 2020-07-01 00:01:11
## 6 2020-07-01 00:01:56
```

Smooth line plot - demonstrates the amount of rides over the course of the year.

```
date_order_plot <-  date_order %>%
  group_by(started_at) %>%
  summarise(session_count = n()) %>%
  ggplot(aes(started_at, session_count)) +
  geom_smooth()+
  labs(x = "Time of year", y = "Number_of_Rides - Hundreds of Thousands",
       title = "Number of Rides July 2020 - July 2021")

date_order_plot
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Using this visualisation - we can clearly see that through both memberships and casual users, there are clear trends throughout the year for the use of the bikeshare program. As a result of the geographical location, Chicago, there is a significant decline in the winter months and many more users in the summer months.

Number of rides for users by day_of_week

```
tripdata_cleaned %>%
  group_by(ride_id, day_of_week) %>%
  summarize(number_of_rides=n())
```

```
## # A tibble: 4,015,456 x 3
## # Groups:   ride_id [4,015,456]
##    ride_id          day_of_week number_of_rides
##    <chr>                  <dbl>           <int>
##  1 000001004784CD35           4               1
##  2 000002EBE159AE82           3               1
##  3 00001A81D056B01B           4               1
##  4 00001DCF2BC423F4           1               1
##  5 00001E17DEF40948           4               1
##  6 00002279D7D315A5           7               1
##  7 0000370913F39D28           5               1
##  8 0000376F8A298CB2           6               1
##  9 000038F6910D8F7F           4               1
## 10 000039C9815A2F25           5               1
## # … with 4,015,446 more rows
```

```
head(tripdata_cleaned)
```

```
## # A tibble: 6 x 16
##   ride_id rideable_type started_at          ended_at
##   <chr>   <chr>         <dttm>              <dttm>
## 1 762198… docked_bike   2020-07-09 15:22:02 2020-07-09 15:25:52
## 2 BEC9C9… docked_bike   2020-07-24 23:56:30 2020-07-25 00:20:17
## 3 D2FD8E… docked_bike   2020-07-08 19:49:07 2020-07-08 19:56:22
## 4 54AE59… docked_bike   2020-07-17 19:06:42 2020-07-17 19:27:38
## 5 54025F… docked_bike   2020-07-04 10:39:57 2020-07-04 10:45:05
## 6 65636B… docked_bike   2020-07-28 16:33:03 2020-07-28 16:49:10
## # … with 12 more variables: start_station_name <chr>,
## #   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 <time>, day_of_week <dbl>, day <chr>
```

```
# average ride_length by type and day of week
counts <- aggregate(tripdata_cleaned$ride_length ~ tripdata_cleaned$member_casual +
                      tripdata_cleaned$day_of_week + tripdata_cleaned$day, FUN = mean)
                      names(counts)[1] <- "member_casual"
                      names(counts)[2] <- "day_of_week"
                      names(counts)[3] <- "day"
                      names(counts)[4] <- "mean_ride_length"

                      counts$mean_ride_length <- round(counts$mean_ride_length,digit=2)
counts <- counts[ -c(2) ]
counts$day <- factor(counts$day, levels= c("Sunday", "Monday",
                                           "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday"))
counts <- counts[order(counts$day), ]

head(counts)
```

```
##    member_casual     day mean_ride_length
## 7         casual  Sunday         2873.52
## 8         member  Sunday         1014.50
## 3         casual  Monday         2429.18
## 4         member  Monday          855.95
## 11        casual Tuesday         2225.78
## 12        member Tuesday          842.47
```

The above data frame demonstrates the mean_ride_length with regard to day of the week and the membership type (casual or member). We can see from the head of this dataframe that casual riders have significantly longer rides than members that encompasses any day of the week.

Average ride time by each day for members vs casual users

```
average_casual_vs_member <-
  aggregate(tripdata_cleaned$ride_length ~ tripdata_cleaned$member_casual + tripdata_cle
aned$day, FUN = mean)
  names(average_casual_vs_member)[1] <- "member_casual"
  names(average_casual_vs_member)[2] <- "day_of_week"
  names(average_casual_vs_member)[3] <- "mean_ride_length"
  average_casual_vs_member$mean_ride_length <- round(average_casual_vs_member$`mean_ride
_length` ,digit=2)
printrows <- average_casual_vs_member[1:14,]
printrows
```

```
##    member_casual day_of_week mean_ride_length
## 1         casual      Friday         2407.36
## 2         member      Friday          875.91
## 3         casual      Monday         2429.18
## 4         member      Monday          855.95
## 5         casual    Saturday         2661.35
## 6         member    Saturday          982.67
## 7         casual      Sunday         2873.52
## 8         member      Sunday         1014.50
## 9         casual    Thursday         2280.23
## 10        member    Thursday          840.44
## 11        casual     Tuesday         2225.78
## 12        member     Tuesday          842.47
## 13        casual   Wednesday         2282.19
## 14        member   Wednesday          844.05
```

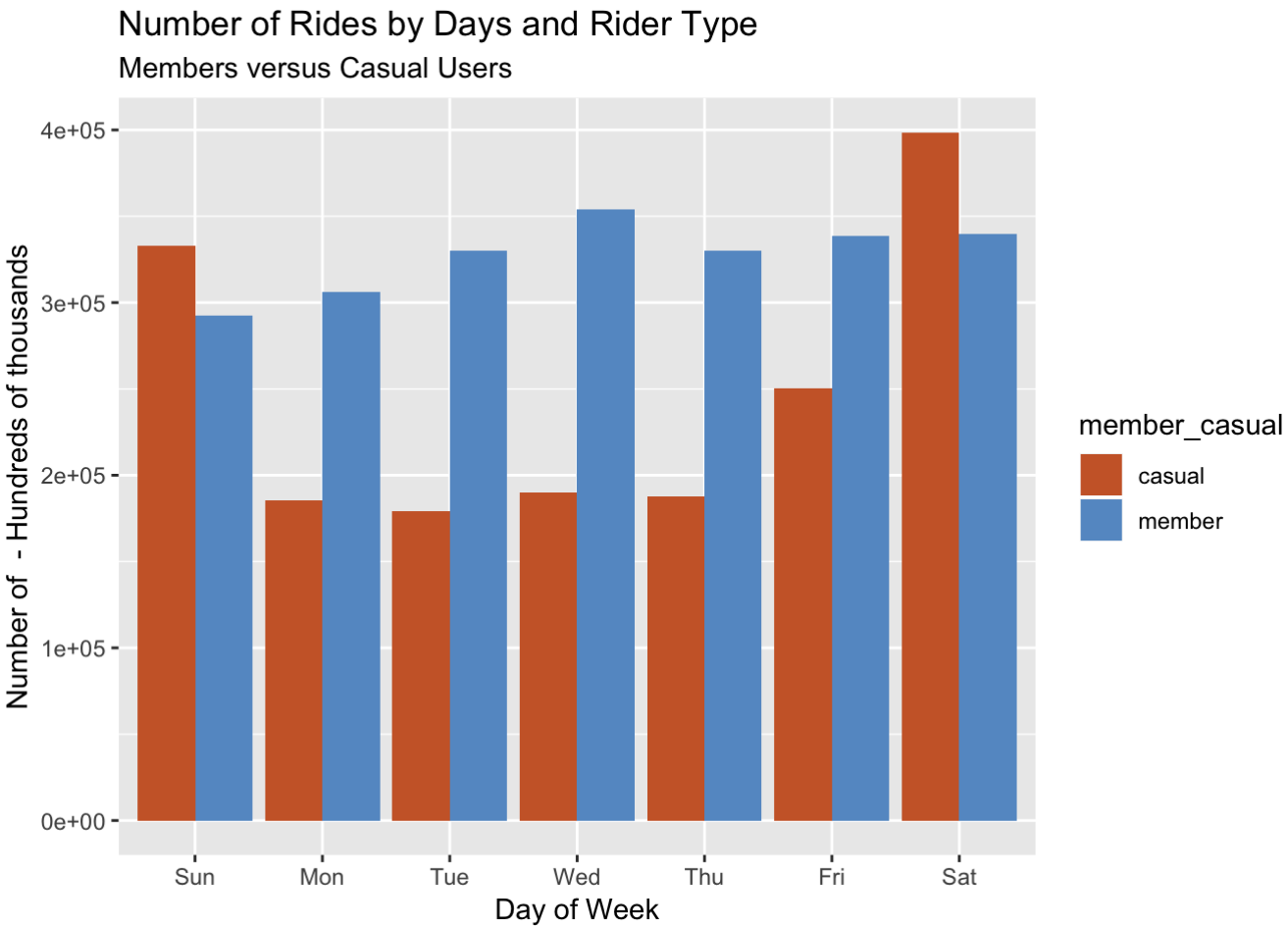
The above dataframe demonstrates the mean ride_length between casual and member riders. It is clear through this dataframe that casual riders have significantly longer duration of rides than riders that hold memberships.

Analyze and visualise ridership data by type and weekday

```
tripdata_cleaned %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarize(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int> <time>
##  1 casual        Sun              333130 2873.5199 secs
##  2 casual        Mon              185675 2429.1839 secs
##  3 casual        Tue              179455 2225.7803 secs
##  4 casual        Wed              189873 2282.1869 secs
##  5 casual        Thu              187616 2280.2335 secs
##  6 casual        Fri              250310 2407.3610 secs
##  7 casual        Sat              398686 2661.3511 secs
##  8 member        Sun              292236 1014.4982 secs
##  9 member        Mon              305938  855.9480 secs
## 10 member        Tue              330083  842.4719 secs
## 11 member        Wed              353789  844.0461 secs
## 12 member        Thu              330300  840.4405 secs
## 13 member        Fri              338882  875.9073 secs
## 14 member        Sat              339483  982.6687 secs
```

```
# visualize number of rides by rider type
tripdata_cleaned %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarize(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") +
  scale_fill_manual(values = c("#CC6633","#6699CC")) +
  labs(title = "Number of Rides by Days and Rider Type",
       subtitle = "Members versus Casual Users") +
  ylab("Number of  - Hundreds of thousands") +
  xlab("Day of Week")
```

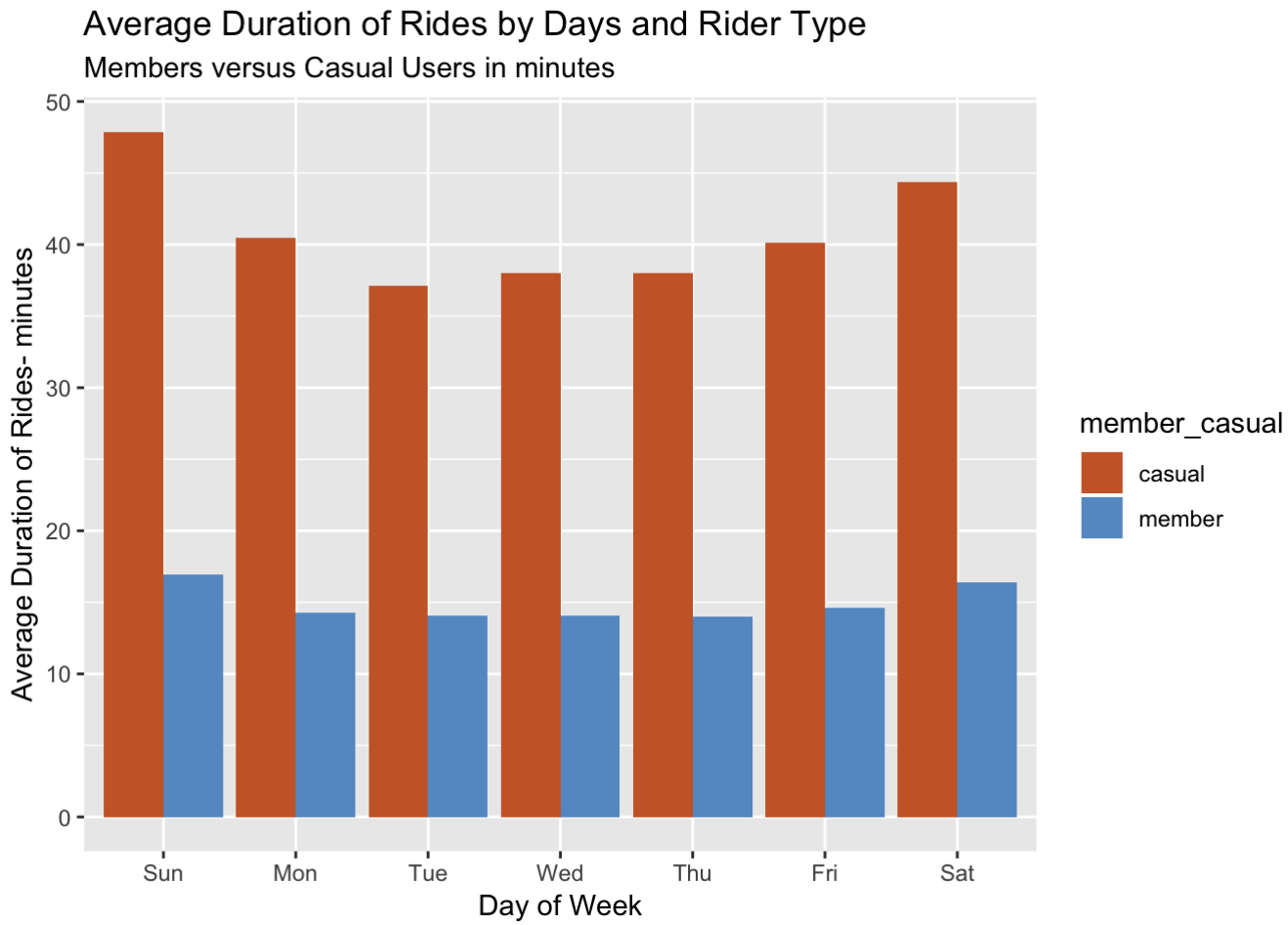


This visusalistion demonstrates the number of rides between casual and those who hold memberships. It is clear that during the week, riders who hold memberships dominate usage throughout the work week (Monday to Friday). Those riders who are casual riders have more riders throughout weekends and what we have previously learned that they also have a longer ride duration.

Visualization for average duration - with regard to membership status (casual vs member)

```
tripdata_cleaned %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarize(average_duration = mean(ride_length)/60) %>%
  arrange(member_casual, weekday) %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  scale_fill_manual(values = c("#CC6633","#6699CC")) +
  labs(title = "Average Duration of Rides by Days and Rider Type",
       subtitle = "Members versus Casual Users in minutes") +
  ylab("Average Duration of Rides- minutes") +
  xlab("Day of Week")
```

```
## Don't know how to automatically pick scale for object of type difftime. Defaulting to
continuous.
```



This visualisation demonstrates the average duration of rides with regard to days and rider type. It is evident through this analysis that casual riders have significantly longer rides than members each day. Combined with information that we have already learned about membership types, members are more likely to ride more often during the week and have shorter rides. This could be due to the convenience and the ease at which they are able to find and rent a bike to get to their destination.

Average ride_length and type and month

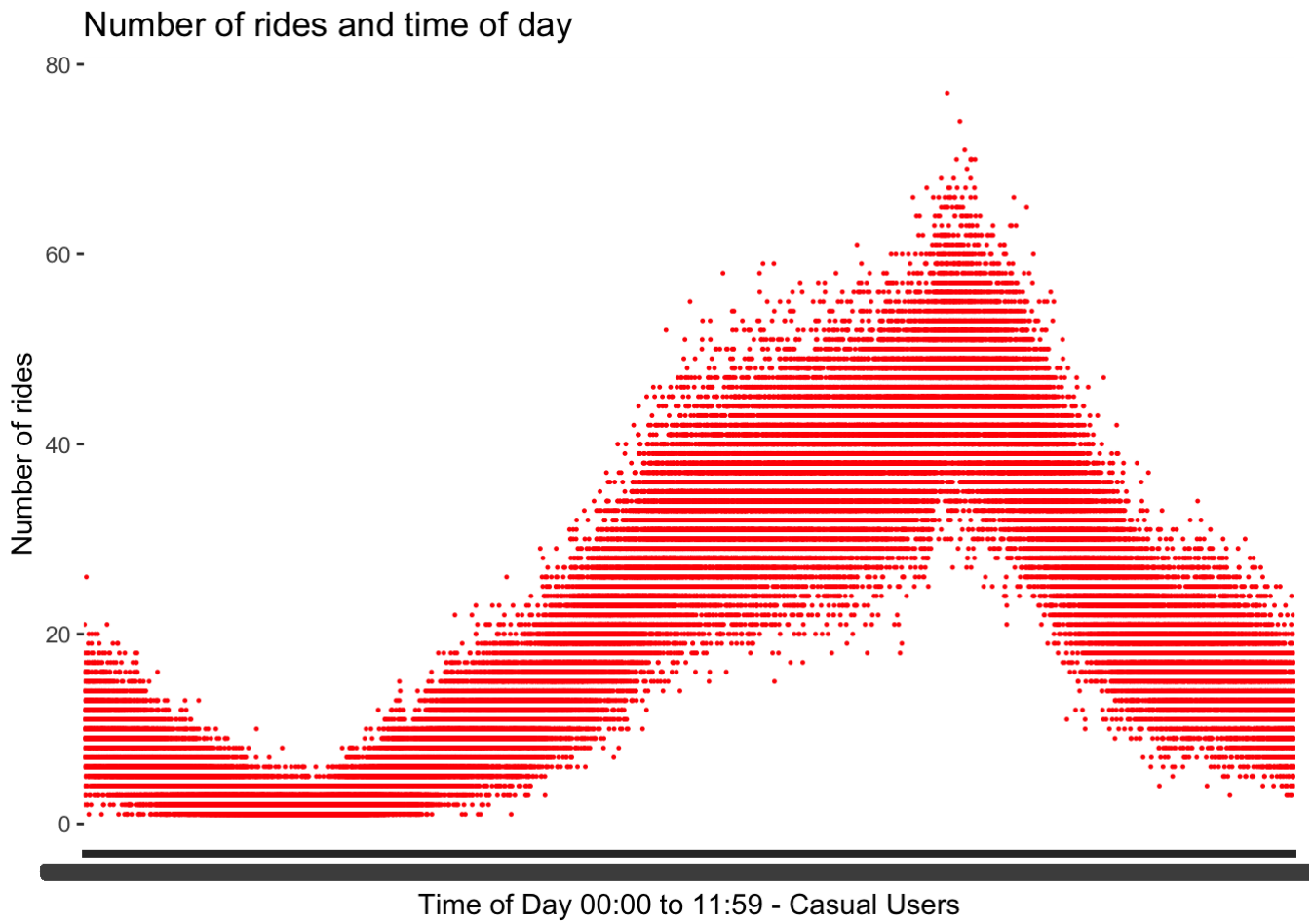
```
tripdata_cleaned$month <- month(tripdata_cleaned$started_at, label = TRUE)
rides <- aggregate(tripdata_cleaned$ride_length ~ tripdata_cleaned$member_casual +
                     tripdata_cleaned$month,FUN = mean)
```

The ways in which casual and member riders use this service are different in a variety of ways. The last two graphs will demonstrate how these types of riders behave in terms of the time of day that they're using the service.

Casual Riders time of day

```
casual_rider <- subset(tripdata_cleaned, member_casual == "casual")
#separate date and time into two columns
casual_time <- separate(casual_rider, started_at, into = c("date", "time"), sep = " ",)
#Drop non essential columns
casual_rider_time <- casual_time[order(casual_time$time),4,drop=FALSE ]
#Aggregate time and count instances in a new column
casual_time_count <- casual_rider_time %>% count(time)

#Plot - Number of riders - time of day Casual Users
ggplot() +
  geom_point(data=casual_time_count, aes(time, n),colour="red")+
  update_geom_defaults("point",list(size=0.2))+
  labs(x = "Time of Day 00:00 to 11:59 - Casual Users", y = "Number of rides",
       title = "Number of rides and time of day")
```

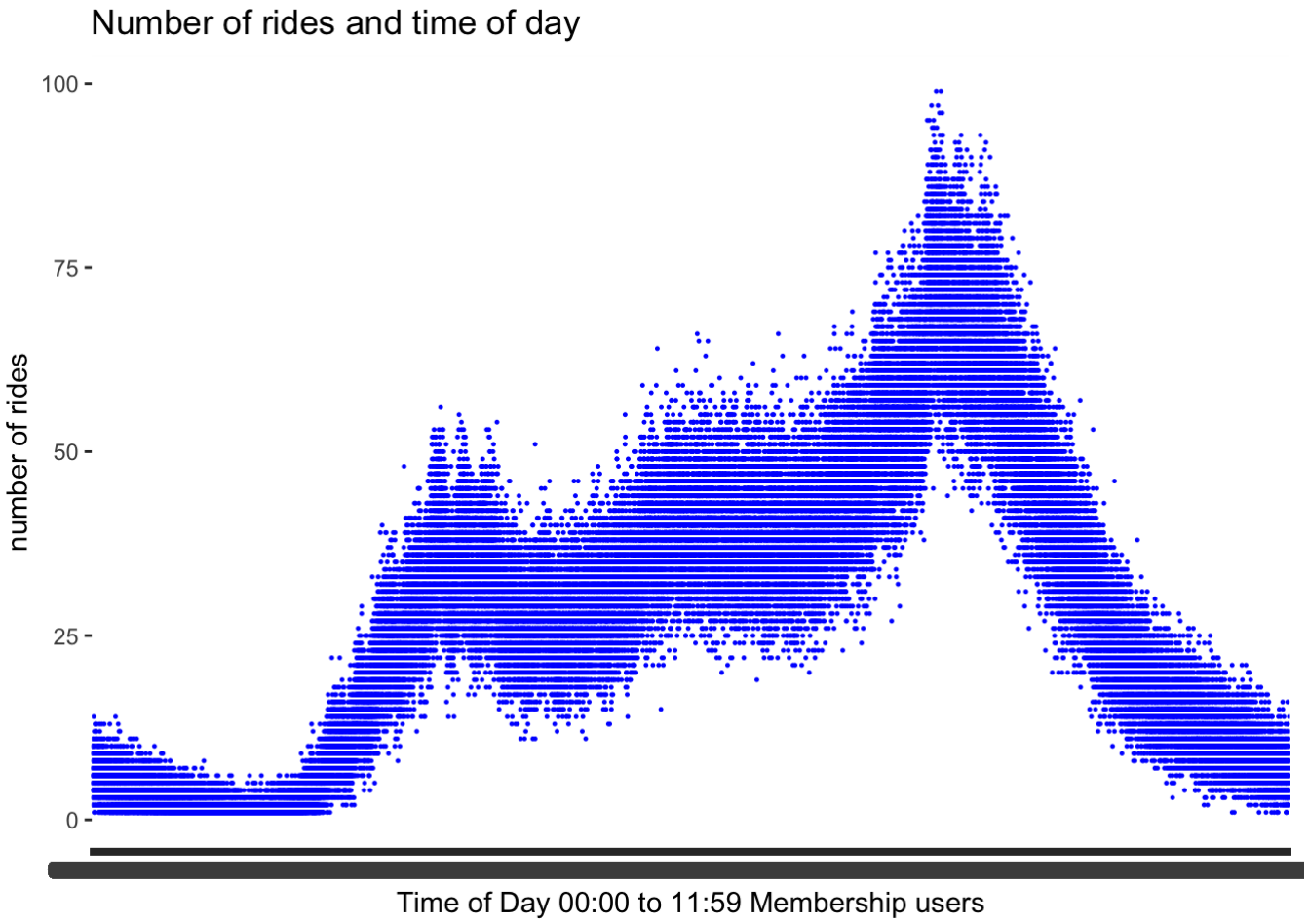


This plot demonstrates the time at which casual riders started their trip during a 24 hr period over the course of a year. Each point represents a time of day from 00:00 to 11:59 and the count at which that time occurs throughout the dataset. Casual riders tend to have more frequent rides during the evening and there are strong trends from around 5pm to 10pm. Keeping in mind previous learning from this data set that included casual riders using the service more on weekends and having longer rides than those who hold memberships.

```
member_rider <- subset(tripdata_cleaned, member_casual == "member")
#separate date and time into two columns
member_time <- separate(member_rider, started_at, into = c("date", "time"), sep = " ",)
#Drop non essential columns
member_rider_time <- member_time[order(member_time$time),4,drop=FALSE ]
#Aggregate time and count instances in a new column
member_time_count <- member_rider_time %>% count(time)



#Plot - Number of riders - time of day membership Users
ggplot() +
  geom_point(data=member_time_count, aes(time, n),colour="blue")+
  update_geom_defaults("point",list(size=0.2))+
  labs(x = "Time of Day 00:00 to 11:59 Membership users", y = "number of rides",
       title = "Number of rides and time of day")
```



This plot demonstrates the time at which member riders started their trip during a 24 hr period over the course of a year. Each point represents a time of day from 00:00 to 11:59pm and the count at which that time occurs throughout the dataset. Member riders tend to use this service at clear points during the morning rush hour, in the middle of the day and in the evening around 5-6pm. This coupled with previous learnings, indicates that member riders take more frequent and shorter rides particularly throughout the week commuting to and from work.

Dataset is ordered and exported to a csv for further analysis to be imported to tableau or PowerBI

```
alltrips <- tripdata_cleaned %>%
  select(-day_of_week)
alltrips$day_of_week <- wday(alltrips$started_at, label = TRUE)

alltrips_ordered <- alltrips[order(alltrips$started_at),]

head(alltrips_ordered )
```

```
## # A tibble: 6 x 17
##   ride_id rideable_type started_at          ended_at
##   <chr>   <chr>         <dttm>              <dttm>
## 1 C66CC4… docked_bike   2020-07-01 00:00:14 2020-07-01 01:28:12
## 2 BD6363… docked_bike   2020-07-01 00:00:15 2020-07-01 02:44:58
## 3 185629… docked_bike   2020-07-01 00:00:49 2020-07-01 00:45:04
## 4 06B27D… docked_bike   2020-07-01 00:00:50 2020-07-01 02:52:16
## 5 7F17B8… docked_bike   2020-07-01 00:01:11 2020-07-01 00:08:03
## 6 78DDAA… docked_bike   2020-07-01 00:01:56 2020-07-01 00:24:27
## # … with 13 more variables: start_station_name <chr>,
## #   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 <time>, day <chr>, month <ord>,
## #   day_of_week <ord>
```

```
#write.csv(alltrips_ordered, file = "all_trips.csv", row.names = FALSE)
```