Google Data Analytics Capstone Project: Cyclistic

Install required packages

```
library(tidyverse)
library(lubridate)
library(ggplot2)
library(janitor)
```

STEP 1: COLLECT DATA

```
sep_22_trip_data <- read_csv("202209-divvy-tripdata.csv")
```

```
## Rows: 701339 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
aug_22_trip_data <- read_csv("202208-divvy-tripdata.csv")
```

```
## Rows: 785932 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
jul_22_trip_data <- read_csv("202207-divvy-tripdata.csv")
```

```
## Rows: 823488 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
jun_22_trip_data <- read_csv("202206-divvy-tripdata.csv")
```

```
## Rows: 769204 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
may_22_trip_data <- read_csv("202205-divvy-tripdata.csv")
```

```
## Rows: 634858 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
apr_22_trip_data <- read_csv("202204-divvy-tripdata.csv")
```

```
## Rows: 371249 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
mar_22_trip_data <- read_csv("202203-divvy-tripdata.csv")
```

```
## Rows: 284042 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
feb_22_trip_data <- read_csv("202202-divvy-tripdata.csv")
```

```
## Rows: 115609 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
jan_22_trip_data <- read_csv("202201-divvy-tripdata.csv")
```

```
## Rows: 103770 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
```

```
## 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.
```

```
dec_21_trip_data <- read_csv("202112-divvy-tripdata.csv")
```

```
## Rows: 247540 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
nov_21_trip_data <- read_csv("202111-divvy-tripdata.csv")
```

```
## Rows: 359978 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

```
oct_21_trip_data <- read_csv("202110-divvy-tripdata.csv")
```

```
## Rows: 631226 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## 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.
```

STEP 2: WRANGLE DATA AND COMBINE INTO A SINGLE FILE

```
colnames(sep_22_trip_data)
```

```
##  [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"
```

```
colnames(aug_22_trip_data)
```

```
##  [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"
```

```
colnames(jul_22_trip_data)
```

```
##  [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"
```

```
colnames(jun_22_trip_data)
```

```
##  [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"
```

```
colnames(may_22_trip_data)
```

```
##  [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"
```

```
colnames(apr_22_trip_data)
```

```
##  [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"
```

```
colnames(mar_22_trip_data)
```

```
##  [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"
```

```
colnames(feb_22_trip_data)
```

```
##  [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"
```

```
colnames(jan_22_trip_data)
```

```
##  [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"
```

```
colnames(dec_21_trip_data)
```

```
##  [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"
```

```
colnames(nov_21_trip_data)
```

```
##  [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"
```

```
colnames(oct_21_trip_data)
```

```
##  [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"
```

Compare column datatype across dataframes to check for error

```
compare_df_cols(sep_22_trip_data, aug_22_trip_data, jul_22_trip_data,
    jun_22_trip_data, may_22_trip_data, apr_22_trip_data, mar_22_trip_data,
    feb_22_trip_data, jan_22_trip_data, dec_21_trip_data, nov_21_trip_data,
    oct_21_trip_data, return = "mismatch")
```

```
##  [1] column_name      sep_22_trip_data aug_22_trip_data jul_22_trip_data
##  [5] jun_22_trip_data may_22_trip_data apr_22_trip_data mar_22_trip_data
##  [9] feb_22_trip_data jan_22_trip_data dec_21_trip_data nov_21_trip_data
## [13] oct_21_trip_data
## <0 rows> (or 0-length row.names)
```

Stack individual months's data frames into one big data frame

```
all_trips <- bind_rows(sep_22_trip_data, aug_22_trip_data, jul_22_trip_data,
    jun_22_trip_data, may_22_trip_data, apr_22_trip_data, mar_22_trip_data,
    feb_22_trip_data, jan_22_trip_data, dec_21_trip_data, nov_21_trip_data,
    oct_21_trip_data)
```

STEP 3: CLEAN UP AND ADD DATA TO PREPARE FOR ANALYSIS

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

Inspect the new table that has been created

```
## spec_tbl_df [5,828,235 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5828235] "5156990AC19CA285" "E12D4A16BF51C274" "A02B53CD7DB72DD7" "C82
##  $ rideable_type     : chr [1:5828235] "electric_bike" "electric_bike" "electric_bike" "electric_bik
##  $ started_at        : POSIXct[1:5828235], format: "2022-09-01 08:36:22" "2022-09-01 17:11:29" ...
##  $ ended_at          : POSIXct[1:5828235], format: "2022-09-01 08:39:05" "2022-09-01 17:14:45" ...
```

```
##  $ start_station_name: chr [1:5828235] NA NA NA NA ...
##  $ start_station_id  : chr [1:5828235] NA NA NA NA ...
##  $ end_station_name  : chr [1:5828235] "California Ave & Milwaukee Ave" NA NA NA ...
##  $ end_station_id    : chr [1:5828235] "13084" NA NA NA ...
##  $ start_lat         : num [1:5828235] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:5828235] -87.7 -87.6 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num [1:5828235] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:5828235] -87.7 -87.6 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr [1:5828235] "casual" "casual" "casual" "casual" ...
##  - 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_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()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
summary(all_trips)  #Statistical summary of data. Mainly for numeric
```

```
##    ride_id          rideable_type        started_at
##  Length:5828235     Length:5828235     Min.   :2021-10-01 00:00:09.00
##  Class :character   Class :character   1st Qu.:2022-02-28 19:21:08.50
##  Mode  :character   Mode  :character   Median :2022-06-08 06:41:28.00
##                                        Mean   :2022-05-06 21:39:18.18
##                                        3rd Qu.:2022-08-02 11:26:01.00
##                                        Max.   :2022-09-30 23:59:56.00
##
##     ended_at                      start_station_name start_station_id
##  Min.   :2021-10-01 00:03:11.0   Length:5828235     Length:5828235
##  1st Qu.:2022-02-28 19:34:02.5   Class :character   Class :character
##  Median :2022-06-08 06:55:07.0   Mode  :character   Mode  :character
##  Mean   :2022-05-06 21:58:54.2
##  3rd Qu.:2022-08-02 11:46:26.0
##  Max.   :2022-10-05 19:53:11.0
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:5828235     Length:5828235     Min.   :41.64   Min.   :-87.84
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64
##                                        Mean   :41.90   Mean   :-87.65
##                                        3rd Qu.:41.93   3rd Qu.:-87.63
##                                        Max.   :45.64   Max.   :-73.80
##
##     end_lat         end_lng       member_casual
##  Min.   :41.39   Min.   :-88.97   Length:5828235
```

```
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character
##  Median :41.90   Median :-87.64   Mode  :character
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.37   Max.   :-87.30
##  NA's   :5844    NA's   :5844
```

Add columns that list the date, month, year and time of each ride

```
all_trips$date <- as.Date(all_trips$started_at)
all_trips$month <- floor_date(all_trips$date, "month")
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A")
all_trips$time <- format(as.POSIXct(all_trips$started_at), format = "%H:%M")
```

Add column that categorizes ride start time into time periods (Morning/Afternoon/Evening/Night)

```
all_trips <- all_trips %>%
    mutate(time_hour = hour(strptime(time, format = "%H:%M"))) %>%
    mutate(time_period = case_when(time_hour >= 6 & time_hour <
        12 ~ "Morning", time_hour >= 12 & time_hour < 18 ~ "Afternoon",
        time_hour >= 18 & time_hour < 23 ~ "Evening", time_hour >=
            23 | time_hour <= 6 ~ "Night"))
```

Add a "ride_length" calculation to all_trips (in seconds)

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

Convert "ride_length" from Factor to numeric so we can run calculations on the data

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

Remove "bad" data as the dataframe contains ride_length that was negative or equal to 0

```
all_trips <- subset(all_trips, ride_length > 0)
```

Change "ride_length units to minutes

```
all_trips$ride_length <- all_trips$ride_length/60
```

Trim dataset by removing station name, lat, long from data set

```
all_trips_trim <- all_trips %>%
    select(-c(started_at, ended_at, start_station_name, start_station_id,
        end_station_name, end_station_id, start_lat, start_lng,
        end_lat, end_lng))
```

STEP 4: CONDUCT DESCRIPTIVE ANALYSIS

Descriptive analysis on ride_length (all figures in seconds)

```
summary(all_trips_trim$ride_length)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.02     5.93    10.48    19.61    18.85 40705.02
```

Compare members and casual users

```
aggregate(all_trips_trim$ride_length ~ all_trips_trim$member_casual,
    FUN = mean)
```

```
##   all_trips_trim$member_casual all_trips_trim$ride_length
## 1                       casual                   29.36362
## 2                       member                   12.76948
```

```
aggregate(all_trips_trim$ride_length ~ all_trips_trim$member_casual,
    FUN = median)
```

```
##   all_trips_trim$member_casual all_trips_trim$ride_length
## 1                       casual                  13.450000
## 2                       member                   8.883333
```

```
aggregate(all_trips_trim$ride_length ~ all_trips_trim$member_casual,
    FUN = max)
```

```
##   all_trips_trim$member_casual all_trips_trim$ride_length
## 1                       casual                   40705.02
## 2                       member                    1559.90
```

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

```
##   all_trips_trim$member_casual all_trips_trim$ride_length
## 1                       casual                 0.01666667
## 2                       member                 0.01666667
```

See the average ride time by each day for members vs casual users

```
all_trips_trim$day_of_week <- ordered(all_trips_trim$day_of_week,
    levels = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday",
        "Friday", "Saturday"))
aggregate(all_trips_trim$ride_length ~ all_trips_trim$member_casual +
    all_trips_trim$day_of_week, FUN = mean)
```

```
##    all_trips_trim$member_casual all_trips_trim$day_of_week
## 1                       casual                     Sunday
## 2                       member                     Sunday
## 3                       casual                     Monday
## 4                       member                     Monday
## 5                       casual                    Tuesday
## 6                       member                    Tuesday
## 7                       casual                  Wednesday
## 8                       member                  Wednesday
## 9                       casual                   Thursday
## 10                      member                   Thursday
## 11                      casual                     Friday
## 12                      member                     Friday
```

```
## 13                       casual                    Saturday
## 14                       member                    Saturday
##    all_trips_trim$ride_length
## 1                    34.36728
## 2                    14.21568
## 3                    29.73078
## 4                    12.32844
## 5                    25.81165
## 6                    12.16383
## 7                    25.03590
## 8                    12.12346
## 9                    25.68210
## 10                   12.29492
## 11                   28.01435
## 12                   12.52916
## 13                   32.71292
## 14                   14.26471
```

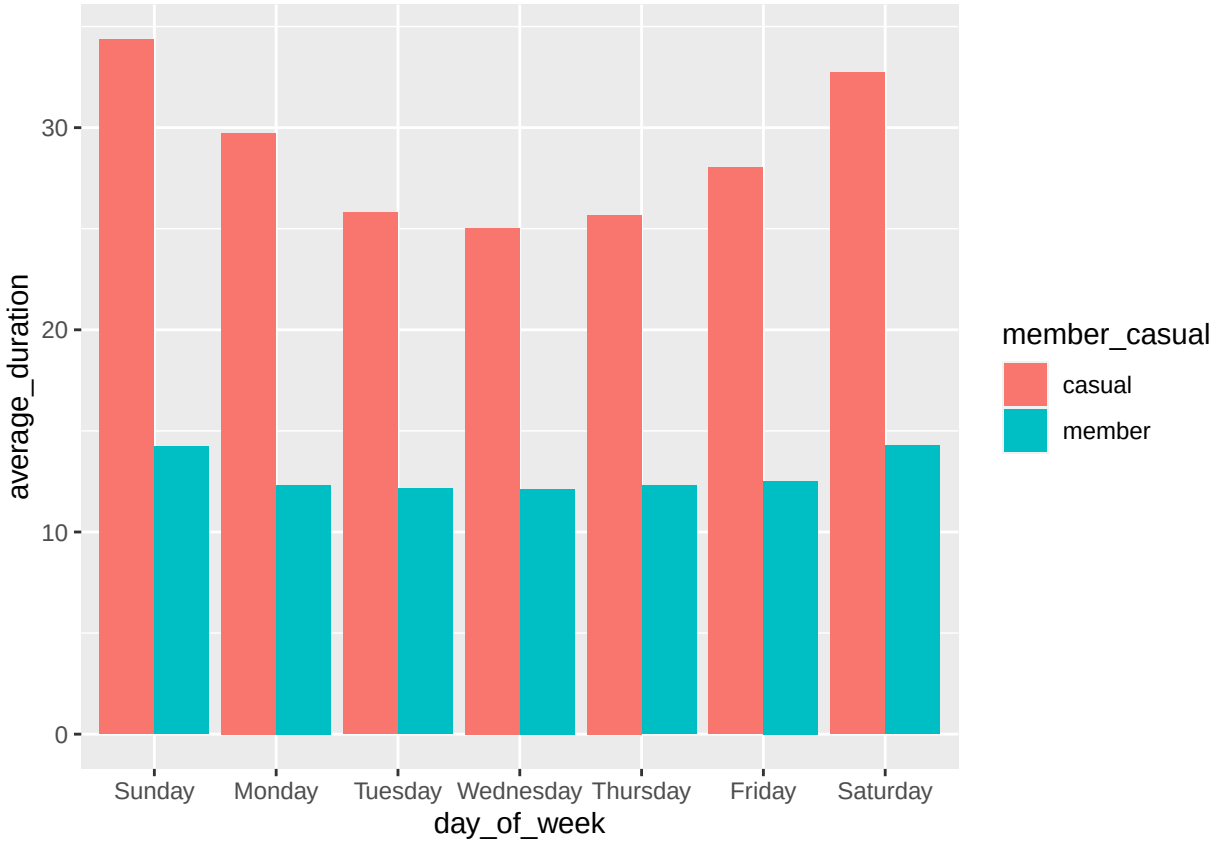
analyze ridership data by type and days of week

```
all_trips_trim %>%
    group_by(member_casual, day_of_week) %>%
    summarize(number_of_rides = n(), average_duration = mean(ride_length)) %>%
    arrange(member_casual, day_of_week)
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual day_of_week number_of_rides average_duration
##    <chr>         <ord>                 <int>            <dbl>
##  1 casual        Sunday               404977             34.4
##  2 casual        Monday               279762             29.7
##  3 casual        Tuesday              275745             25.8
##  4 casual        Wednesday            281640             25.0
##  5 casual        Thursday             306662             25.7
##  6 casual        Friday               352466             28.0
##  7 casual        Saturday             499739             32.7
##  8 member        Sunday               393568             14.2
##  9 member        Monday               473027             12.3
## 10 member        Tuesday              541484             12.2
## 11 member        Wednesday            538459             12.1
## 12 member        Thursday             530510             12.3
## 13 member        Friday               491436             12.5
## 14 member        Saturday             458189             14.3
```

visualize the average number of rides by rider type

```
all_trips_trim %>%
    group_by(member_casual, day_of_week) %>%
    summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
    arrange(member_casual, day_of_week) %>%
    ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
    geom_col(position = "dodge")
```



STEP 5: EXPORT SUMMARY FILE FOR FURTHER ANALYSIS

Create table of the total number of departures of casual members from station names with lat-lon data station IDs

```
station_list_by_membertype <- all_trips %>%
    group_by(start_station_name, member_casual) %>%
    summarize(lat = round(mean(start_lat), 3), lon = round(mean(start_lng),
        3), num_departures = n_distinct(ride_id), avg_ride = mean(ride_length))
```

Create table of summary data aggregated by days

```
daily_trips <- all_trips_trim %>%
    group_by(date, time_period, member_casual, day_of_week, time_hour,
        rideable_type) %>%
    summarize(mean = mean(ride_length), sum = sum(ride_length))
```

Export csv files

```
write.csv(station_list_by_membertype, file = "~/Desktop/station_list_by_membertype.csv")
write.csv(daily_trips, file = "~/Desktop/daily_trips.csv")
```