BikeShare Analysis

1. Load packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readr)
library(dplyr)
library(ggplot2)
```

2. Read in data and merge tables

```
setwd("/Users/alishalaby/MyDocuments/Project Portfolio/BikeShareAnalysis/Data")

bike_data <- rbind(
  read_csv("202009-divvy-tripdata.csv"),
  read_csv("202010-divvy-tripdata.csv"),
  read_csv("202011-divvy-tripdata.csv"),
  read_csv("202012-divvy-tripdata.csv"),
  read_csv("202101-divvy-tripdata.csv"),
  read_csv("202102-divvy-tripdata.csv"),
  read_csv("202103-divvy-tripdata.csv"),
  read_csv("202104-divvy-tripdata.csv"),
  read_csv("202105-divvy-tripdata.csv"),
  read_csv("202106-divvy-tripdata.csv"),
  read_csv("202107-divvy-tripdata.csv"),
  read_csv("202108-divvy-tripdata.csv"))
```

```
## Rows: 532958 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.

## Rows: 388653 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.

## Rows: 259716 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.

## Rows: 131573 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.

## Rows: 96834 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.
## Rows: 49622 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.

## Rows: 228496 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.

## Rows: 337230 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.

## Rows: 531633 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.

## Rows: 729595 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.
```

```
## Rows: 822410 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.
```

```
## Rows: 804352 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.
```

2. Process Data

3. Understand data structure

```
head(bike_data)
```

```
## # A tibble: 6 x 13
##   ride_id rideable_type started_at          ended_at            start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 2B22BD~ electric_bike 2020-09-17 14:27:11 2020-09-17 14:44:24 Michigan Ave & ~
## 2 A7FB70~ electric_bike 2020-09-17 15:07:31 2020-09-17 15:07:45 W Oakdale Ave &~
## 3 86057F~ electric_bike 2020-09-17 15:09:04 2020-09-17 15:09:35 W Oakdale Ave &~
## 4 57F6DC~ electric_bike 2020-09-17 18:10:46 2020-09-17 18:35:49 Ashland Ave & B~
## 5 B9C471~ electric_bike 2020-09-17 15:16:13 2020-09-17 15:52:55 Fairbanks Ct & ~
## 6 378BBC~ electric_bike 2020-09-17 18:37:04 2020-09-17 19:23:28 Clark St & Armi~
## # ... with 8 more variables: 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>
```

```
str(bike_data)
```

```
## spec_tbl_df [4,913,072 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:4913072] "2B22BD5F95FB2629" "A7FB70B4AFC6CAF2" "86057FA01BAC778E" "57F
##  $ rideable_type     : chr [1:4913072] "electric_bike" "electric_bike" "electric_bike" "electric_bik
##  $ started_at        : POSIXct[1:4913072], format: "2020-09-17 14:27:11" "2020-09-17 15:07:31" ...
##  $ ended_at          : POSIXct[1:4913072], format: "2020-09-17 14:44:24" "2020-09-17 15:07:45" ...
##  $ start_station_name: chr [1:4913072] "Michigan Ave & Lake St" "W Oakdale Ave & N Broadway" "W Oakda
##  $ start_station_id  : chr [1:4913072] "52" NA NA "246" ...
##  $ end_station_name  : chr [1:4913072] "Green St & Randolph St" "W Oakdale Ave & N Broadway" "W Oakda
##  $ end_station_id    : chr [1:4913072] "112" NA NA "249" ...
##  $ start_lat         : num [1:4913072] 41.9 41.9 41.9 42 41.9 ...
##  $ start_lng         : num [1:4913072] -87.6 -87.6 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num [1:4913072] 41.9 41.9 41.9 42 41.9 ...
##  $ end_lng           : num [1:4913072] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr [1:4913072] "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_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>
```

4. Check NA values

```
sum(is.na(bike_data)) #total number of NA values
```

```
## [1] 1893790
```

```
bike_data %>% summarise_all(~ sum(is.na(.))) #number of NA values per column
```

```
## # A tibble: 1 x 13
##   ride_id rideable_type started_at ended_at start_station_name start_station_id
##     <int>         <int>      <int>    <int>              <int>            <int>
## 1       0             0          0        0             450045           450571
## # ... with 7 more variables: end_station_name <int>, end_station_id <int>,
## #   start_lat <int>, start_lng <int>, end_lat <int>, end_lng <int>,
## #   member_casual <int>
```

Evident that most na values are contained in station-based: 1. start_station_name 2. start_station_id 3. end_station_name 4. end_station_id We should leave the data as is and not replace NA values

5. Drop ‘lat’ and ‘long’ columns

```
bike_data <- bike_data %>%
  select(-c(start_lat, start_lng, end_lat, end_lng))
```

6. Add columns ‘weekday’, ‘ride_length’ and ‘month’

```
bike_data <-
  bike_data %>%
  mutate(weekday = weekdays(as.Date(bike_data$started_at))) %>%
  mutate(ride_length = ended_at - started_at) %>%
  mutate(month = months(as.Date(bike_data$started_at)))
```

7. Convert ‘ride_length’ to minutes from second

```
glimpse(bike_data$ride_length)
```

```
##  ’difftime’ num [1:4913072] 1033 14 31 1503 ...
##  - attr(*, "units")= chr "secs"
```

```
bike_data$ride_length <- as.numeric(bike_data$ride_length)
bike_data$ride_length <- as.numeric(bike_data$ride_length/60)
```

8. Filter ‘bad’ data

```
bike_data <- bike_data %>%
  filter(ride_length >1) %>% #bikes with more than 1 min of use
  filter(ride_length <= 1440) # bikes atleast 1 day of use
```

3. Analyze

9. Summary statistics

```
summary(bike_data$ride_length)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    1.017    7.417   13.033   21.078   23.517 1439.900
```

10. Aggregate data based on casual vs member riders

```
bike_data %>%
  group_by(member_casual) %>%
  summarise(number_of_riders = n(),
            mean = mean(ride_length),
            median = median(ride_length))
```

```
## # A tibble: 2 x 4
##   member_casual number_of_riders  mean median
##   <chr>                    <int> <dbl>  <dbl>
## 1 casual                 2189511  29.2   17.4
## 2 member                 2638862  14.3   10.4
```

11. Average ride time day of the week members vs casuals

```
rider_weekday <- bike_data %>%
  group_by(member_casual, weekday) %>%
  summarise(average_ride_length = mean(ride_length), number_of_rides = n())
```

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

```
rider_weekday$weekday <- factor(rider_weekday$weekday, levels= c("Monday", "Tuesday", "Wednesday", "Thu
rider_weekday <- rider_weekday[order(rider_weekday$weekday),]
```

12. Analyze ridership data by type of vehicle

```
bike_type <- bike_data %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_riders = n())
```

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

13. Analyze station data

```
station_data <- bike_data %>%
  group_by(member_casual, start_station_name) %>%
  summarise(number_of_stations = n()) %>%
  arrange(desc(number_of_stations))
```

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

```
station_data <- station_data %>%
  rename(station_name = start_station_name)
```

14. Analyze ride_length by month

```
rider_month <- bike_data %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n(), average_ride_length = mean(ride_length))
```

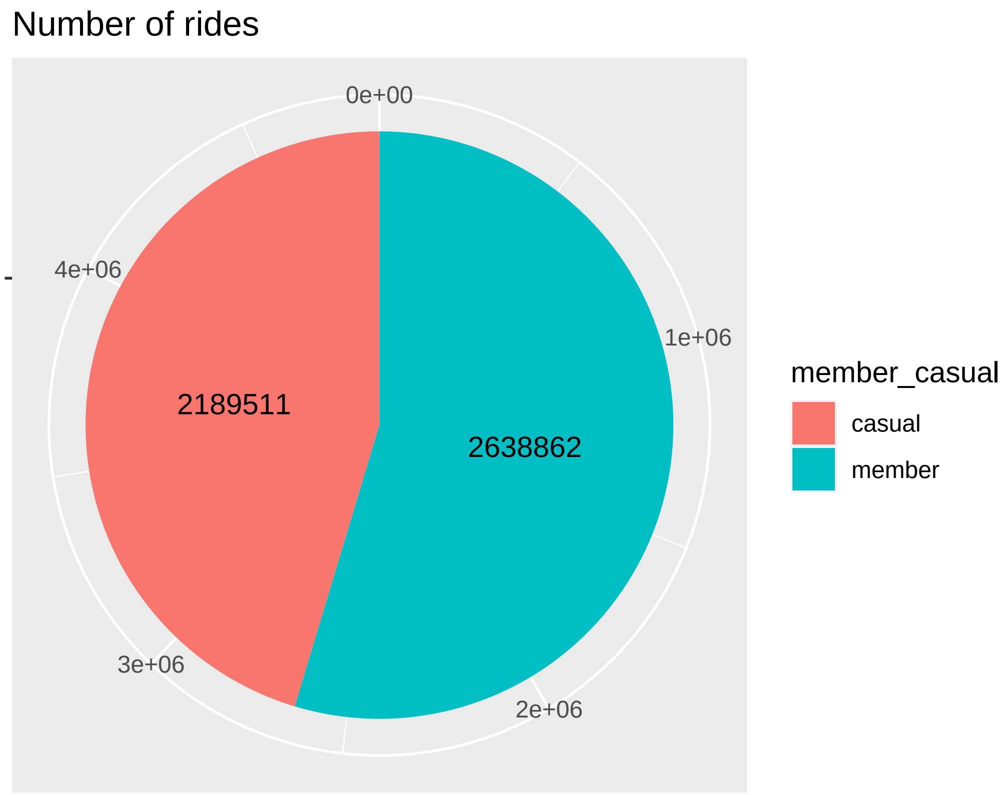
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.
```

```
rider_month$month <- factor(rider_month$month, levels= c("January", "February", "March", "April", "May"
rider_month <- rider_month[order(rider_month$month),]
```

4. Visualize

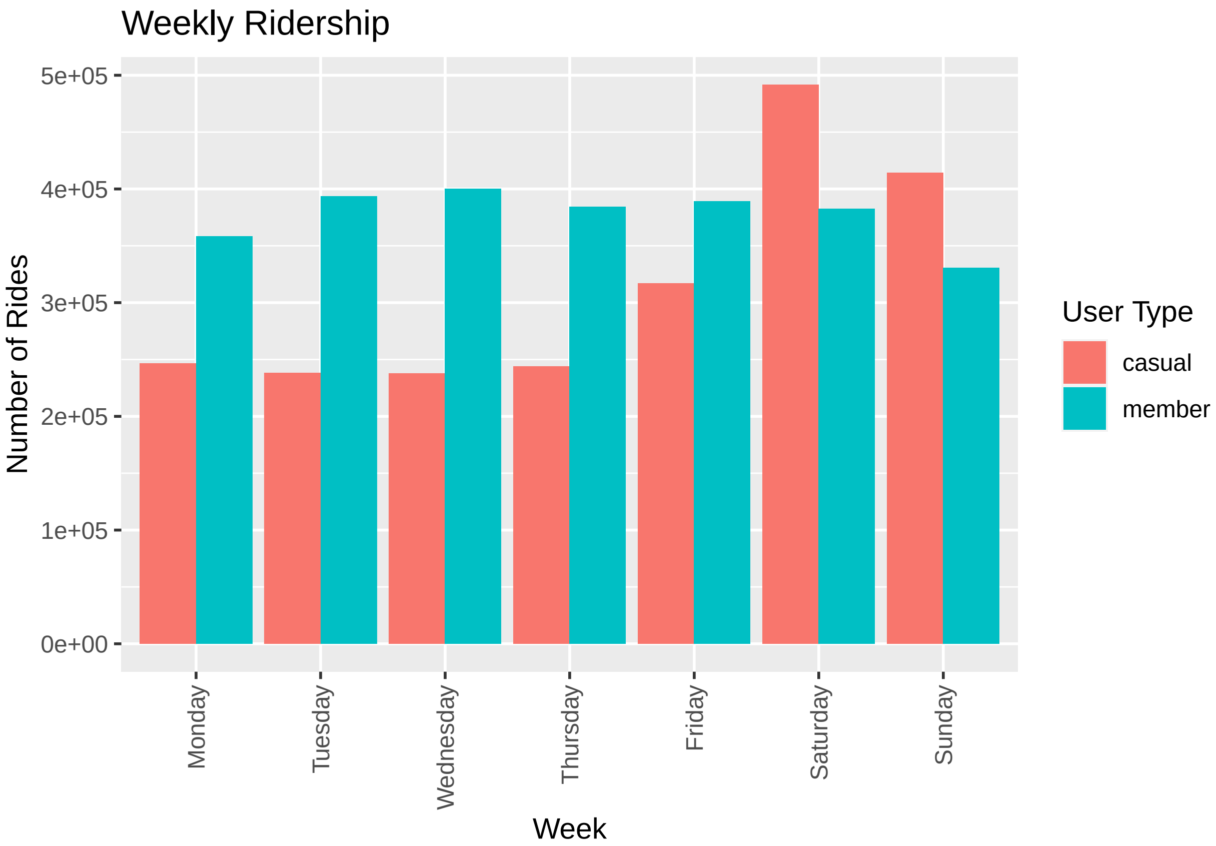
15. Visualize ridership distribution

```
bike_data %>%
  group_by(member_casual) %>%
  summarize(number_of_rides = n()) %>%
  ggplot(aes(x="", y=number_of_rides, fill=member_casual)) +
  geom_col() + labs(title="Number of rides", x="", y="") +
  geom_text(aes(label = number_of_rides), position = position_stack(vjust = 0.5)) +
  coord_polar("y")
```



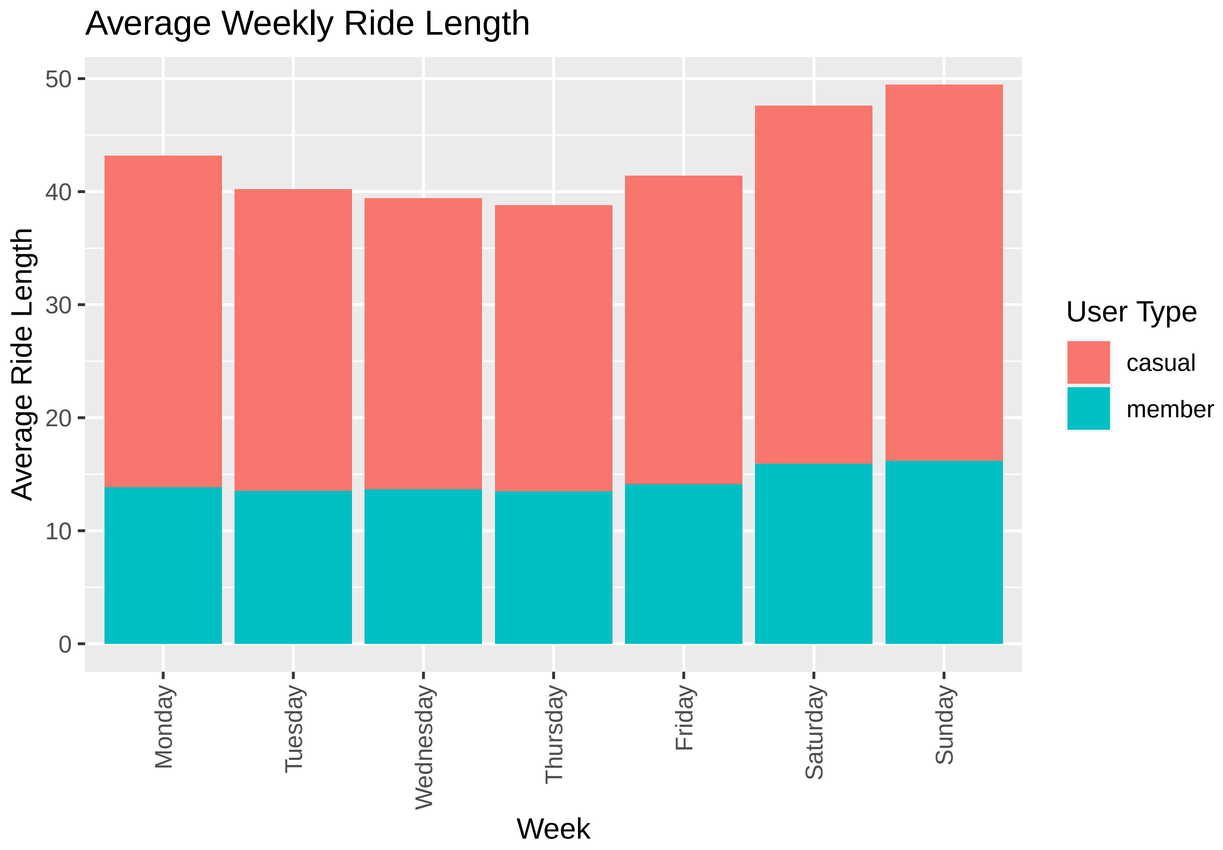
15. Visualize weekly ridership

```
rider_weekday %>%
  ggplot(aes(x=weekday, y=number_of_rides, fill=member_casual)) +
  geom_col(position="dodge") +
  labs(title="Weekly Ridership", x="Week", y="Number of Rides", fill="User Type") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



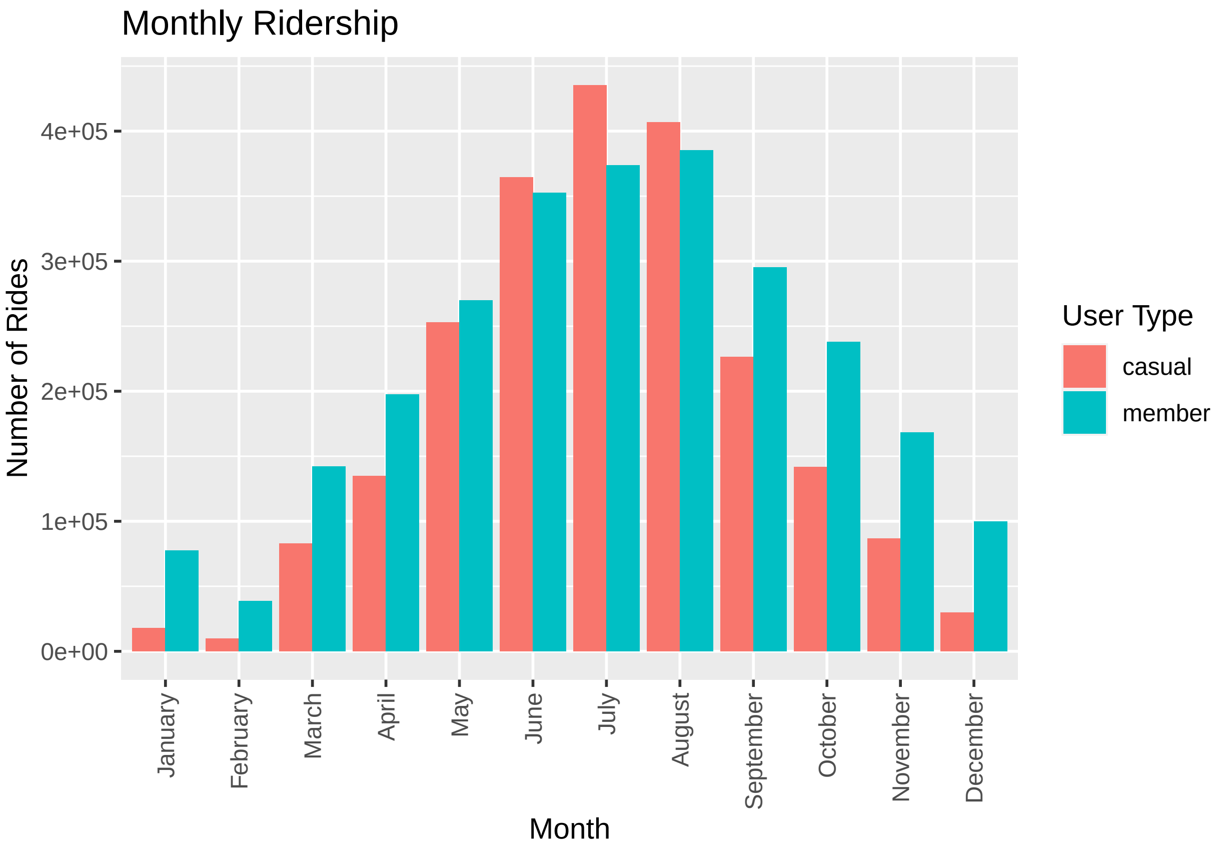
16. Visualize weekly average ride length

```
rider_weekday %>%
  ggplot(aes(x=weekday, y=average_ride_length, fill=member_casual)) +
  geom_col() +
  labs(title="Average Weekly Ride Length", x="Week", y="Average Ride Length", fill="User Type") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



17. Visualize monthly ridership

```
rider_month %>%
  ggplot(aes(x=month, y=number_of_rides, fill=member_casual)) +
  geom_col(position="dodge") +
  labs(title="Monthly Ridership", x="Month", y="Number of Rides", fill="User Type") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



5. Export

18.

```
# Get total riders and bike types in order to visualize on Tableau
total_riders <- data.frame(table(bike_data$member_casual))
total_types <- data.frame(table(bike_data$rideable_type))

write_csv(total_riders, "total_riders.csv")
write_csv(total_types, "total_types.csv")
write_csv(station_data, "station_data.csv")
write_csv(bike_data, "bike_data.csv")
write_csv(bike_type, "bike_type.csv")
write_csv(rider_month, "rider_month.csv")
write_csv(rider_weekday, "rider_weekday.csv")
```