R file

Processing data

Install and load packages to set the environment for processing and analysing Cyclistic data.

```
install.packages(c("tidyverse","dplyr","lubridate","ggplot2","skimr","janitor","rmarkdown"))
library(tidyverse)
library(dplyr)
library(lubridate)
library(ggplot2)
library(skimr)
library(janitor)
library(rmarkdown)
```

Import individual csv files and check for similarity in structure of each table.

```
jul22 <- read_csv("~/extracted files/202207-divvy-tripdata.csv")
jun22 <- read_csv("~/extracted files/202206-divvy-tripdata.csv")
may22 <- read_csv("~/extracted files/202205-divvy-tripdata.csv")
apr22 <- read_csv("~/extracted files/202204-divvy-tripdata.csv")
mar22 <- read_csv("~/extracted files/202203-divvy-tripdata.csv")
feb22 <- read_csv("~/extracted files/202202-divvy-tripdata.csv")
jan22 <- read_csv("~/extracted files/202201-divvy-tripdata.csv")
dec21 <- read_csv("~/extracted files/202112-divvy-tripdata.csv")
nov21 <- read_csv("~/extracted files/202111-divvy-tripdata.csv")
oct21 <- read_csv("~/extracted files/202110-divvy-tripdata.csv")
sep21 <- read_csv("~/extracted files/202109-divvy-tripdata.csv")
aug21 <- read_csv("~/extracted files/202108-divvy-tripdata.csv")
```

Union all the csv files into one data frame.

```
Yearly_Raw_Data <-
bind_rows(aug21,sep21,oct21,nov21,dec21,jan22,feb22,mar22,apr22,may22,jun22,jul22)
dim(Yearly_Raw_Data)
```

The output shows 13 columns with 5,901,463 rows in total.

Now adding new columns in order to make it easy to breakdown data to analyze. Use the date in order to break down the started_at datetime to date, month,year and hour of day.

```
Yearly_Raw_Data$date <- as.Date(Yearly_Raw_Data$started_at)
Yearly_Raw_Data$weekday <- format(as.Date(Yearly_Raw_Data$date),"%a")
Yearly_Raw_Data$month <- format(as.Date(Yearly_Raw_Data$date),"%m")
Yearly_Raw_Data$year <- format(as.Date(Yearly_Raw_Data$date),"%y")
Yearly_Raw_Data$starthour <- strftime(Yearly_Raw_Data$started_at,"%H")
```

Checking for any null values in dataframe.

```
sum(is.na(Yearly_Raw_Data))
```

~33% of data has NULL values.

```
colsums(is.na(Yearly_Raw_Data))
```

Output shows that majority of the NULLS are in start_station_name, start_station_id, end_station_name, end_station_id, end_lat, end_lng

Now, moving to remove NA from the Yearly_Raw_Data.

```
Yearly_Cleaned <- na.omit(Yearly_Raw_Data)
dim(Yearly_Cleaned)
```

Output has 4,629,230 rows.

Adding a new column - rideduration to the dataset. The ridedurationshall be in minutes.

```
Yearly_Cleaned <- Yearly_Cleaned %>%
  mutate(rideduration=difftime(ended_at,started_at,units="mins"))
```

Now checking rideduraiton data type

```
str(Yearly_Cleaned$rideduration)
summary(Yearly_Cleaned$rideduration)
is.numeric(Yearly_Cleaned$rideduration)
```

Output is FALSE for is.numeric

```
Yearly_Cleaned$rideduration <- as.numeric(as.character(Yearly_Cleaned$rideduration))
is.numeric(YEarly_Cleaned$rideduration)
```

Output is TRUE and ready for use

```
summary(Yearly_Cleaned$rideduration)
```

Minimum is -129 and Maximum is 41,629. So, we need to omit rideduration <=0 and greater than 16 hours (i.e. 960 minutes)

```
Yearly_Cleaned <- Yearly_Cleaned[!(Yearly_Cleaned$rideduration<=0),]
Yearly_Cleaned <- Yearly_Cleaned[!(Yearly_Cleaned$rideduration>960),]
summary(Yearly_Cleaned$rideduration)
```

4,627,392 rows are still in the data Yearly_Cleaned after filtering data.

Analyzing data

Now, stepping into analysis of the cleaned data without visualizations

- Summary of ride count, mean ride duration and median ride duration

```
Yearly_Cleaned %>%
   group_by(member_casual) %>%
   summarize(Number_of_rides = length(member_casual)
        , Percentage = (length(member_casual)/nrow(Yearly_Cleaned))*100
        , Meanrideduration = mean(rideduration)
        , medianrideduration = median(rideduration))

# A tibble: 2 × 5
  member_casual Number_of_rides Percentage Meanrideduration medianrideduration
  <chr>                   <int>      <dbl>            <dbl>              <dbl>
1 casual                1947845       42.1             24.9              15.1
2 member                2679547       57.9             12.5               9.17
```

- Summary of ride count, mean ride duration and median ride duration by month of year

```
  Yearly_Cleaned %>%
    group_by(member_casual,month) %>%
    summarize(Number_of_rides = length(member_casual)
              , meanrideduration = mean(rideduration)
              , medianrideduration = median(rideduration)) %>%
    print(n=24);
```

Output is a tibble 24 x 5.

- Keeping a proper order for the days of the week

```
 Yearly_Cleaned$weekday <- ordered(Yearly_Cleaned$weekday,
                                levels=c("Mon","Tue","Wed","Thu","Fri","Sat","Sun"))
```

- Summary of ride count, mean ride duration and median ride duration based on day of week

```
  Yearly_Cleaned %>%
    group_by(member_casual,weekday) %>%
    summarize(Number_of_rides = length(member_casual)
              , meanrideduration = mean(rideduration)
              , medianrideduration = median(rideduration)) %>%
    arrange(member_casual,weekday) %>%
    print(n=14);
```

Output is a tibble: 14 x 5

- Summary of ride count, mean ride duration and median ride duration based on hour of day

```
  Yearly_Cleaned %>%
    group_by(member_casual,starthour) %>%
    summarize(Number_of_rides = length(member_casual)
              , meanriderdurationinsec = mean(rideduration)
              , medianrideurationinsec = median(rideduration)) %>%
    print(n=48)
```

Output is a tibble 48 x 5

- Checking for rideable_type count, meanrideduration

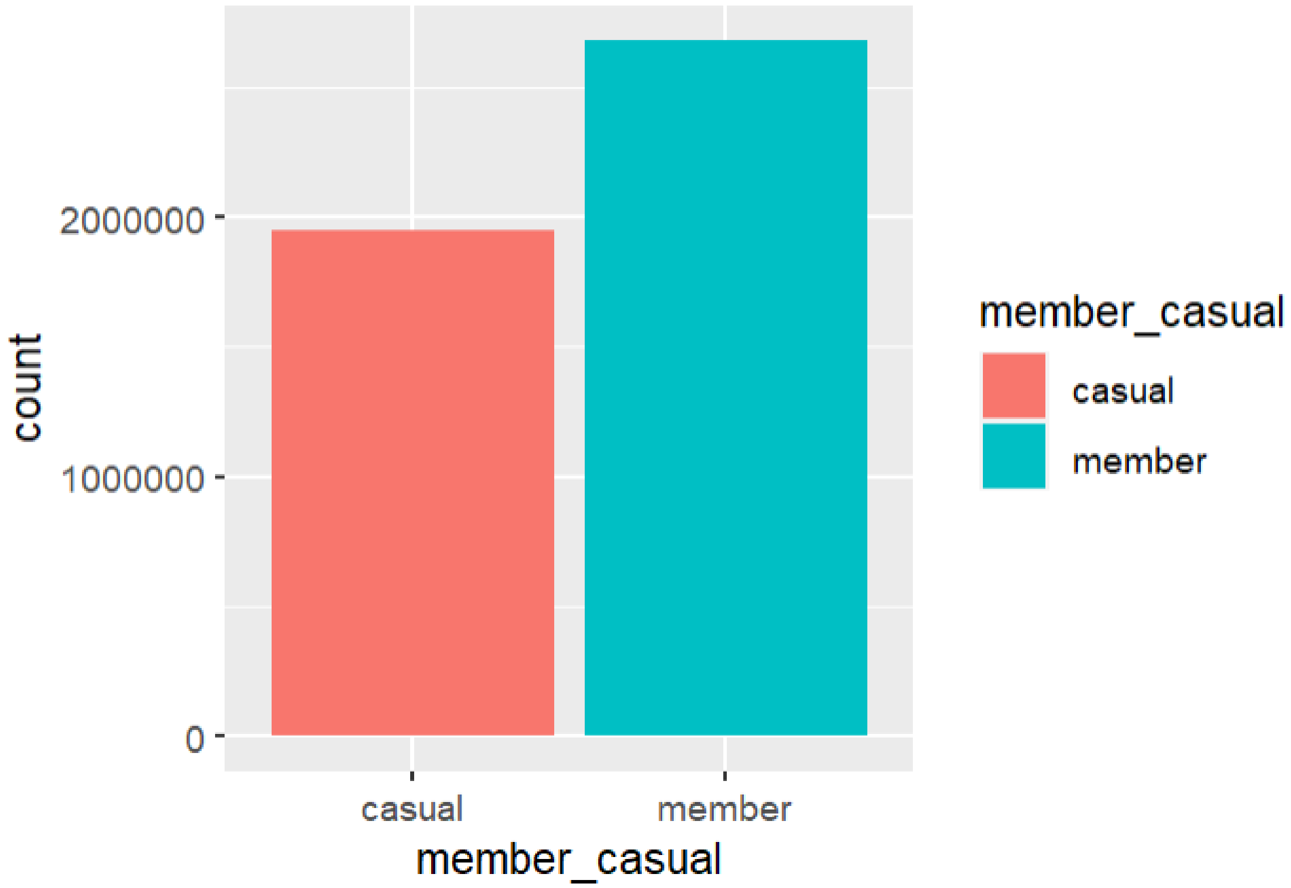
```
  Yearly_Cleaned %>%
    group_by(member_casual,rideable_type) %>%
    summarize(Number_of_rides_by_type = length(rideable_type)
              , meanrideduration inmin = mean(rideduration))

# A tibble: 5 × 4
# Groups:   member_casual [2]
  member_casual rideable_type Number_of_rides_by_type meanridedurationinmin
  <chr>         <chr>                           <int>                 <dbl>
1 casual        classic_bike                  1128313                  24.3
2 casual        docked_bike                    224026                  45.7
3 casual        electric_bike                  595506                  18.2
4 member        classic_bike                  1919882                  13.0
5 member        electric_bike                  759665                  11.3
```

Visualizing data in R

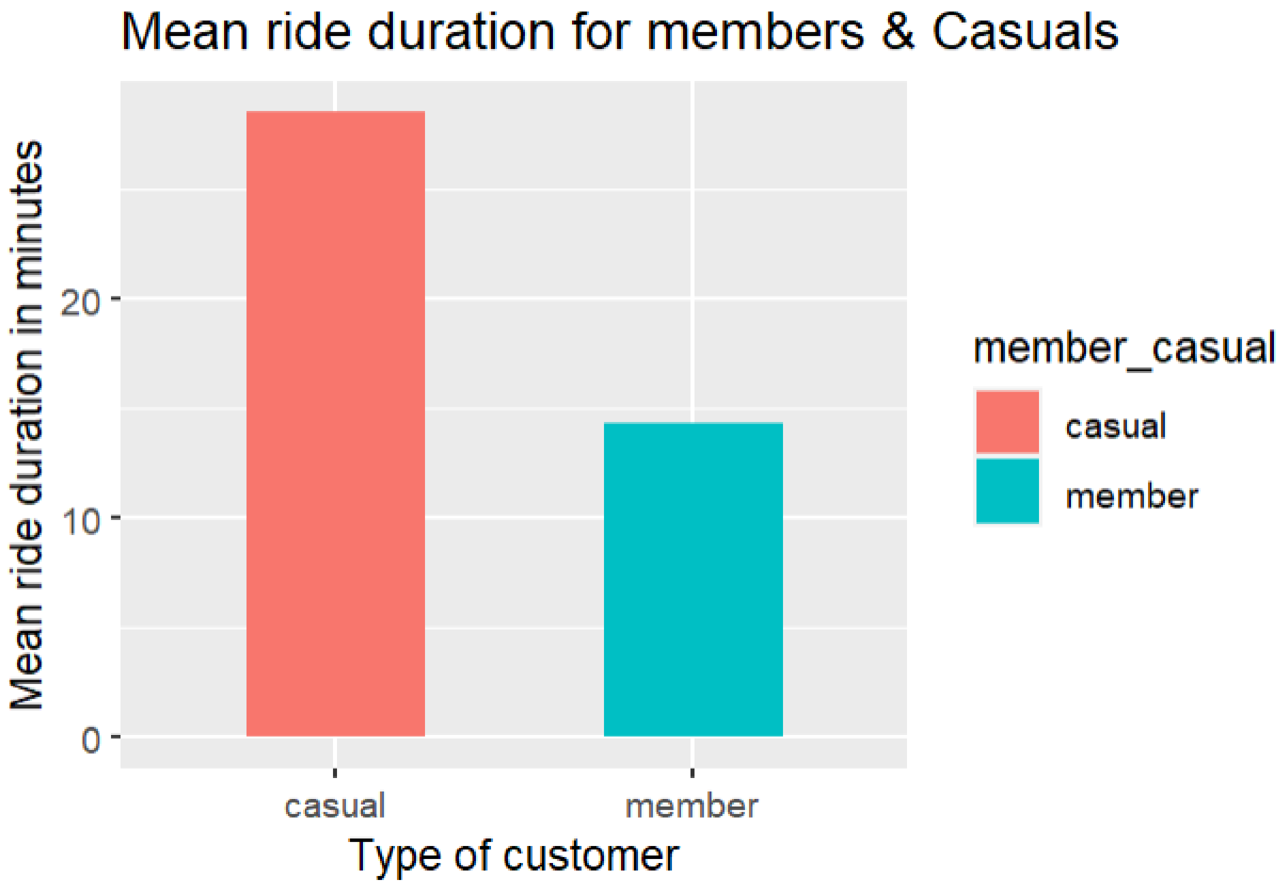
- Based on number of rides

```
ggplot(data=Yearly_Cleaned) +
   geom_bar(mapping=aes(x=member_casual, fill=member_casual)) +
   scale_y_continuous(labels = function(x) format(x, scientific = FALSE))
```



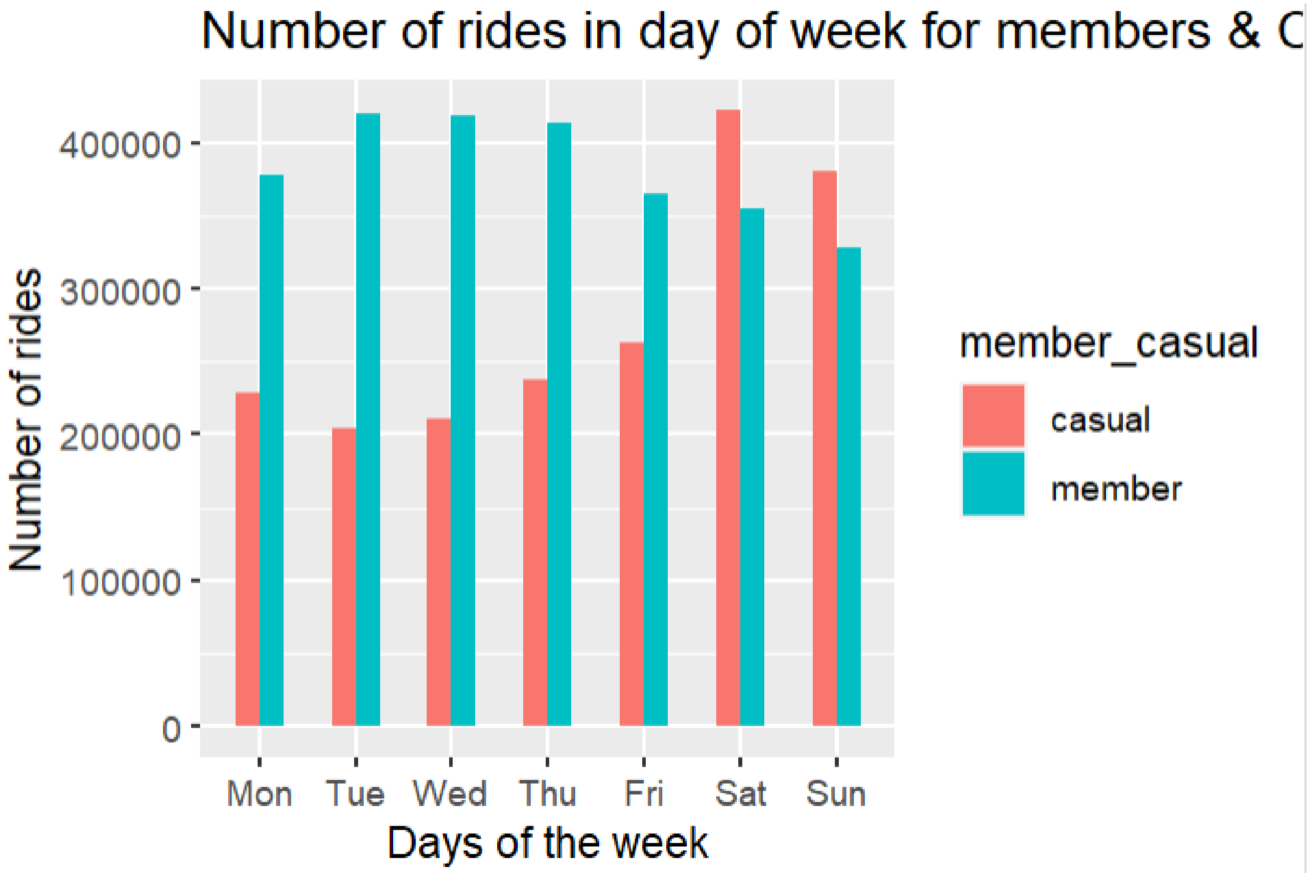
- Based on ride duration

```
  Yearly_Cleaned %>%
    group_by(member_casual,weekday) %>%
    summarise(Ride_Duration=mean(rideduration),.groups="drop") %>%
    ggplot(aes(x=member_casual,y=Ride_Duration,fill=member_casual))+
    geom_col(width = 0.5,position = position_dodge(width = 0.5))+
    labs(title="Mean ride duration for members & Casuals") +
    xlab("Type of customer") +
    ylab("Mean ride duration in minutes") +
    scale_y_continuous(labels = function(x)format(x,scientific=FALSE))
```



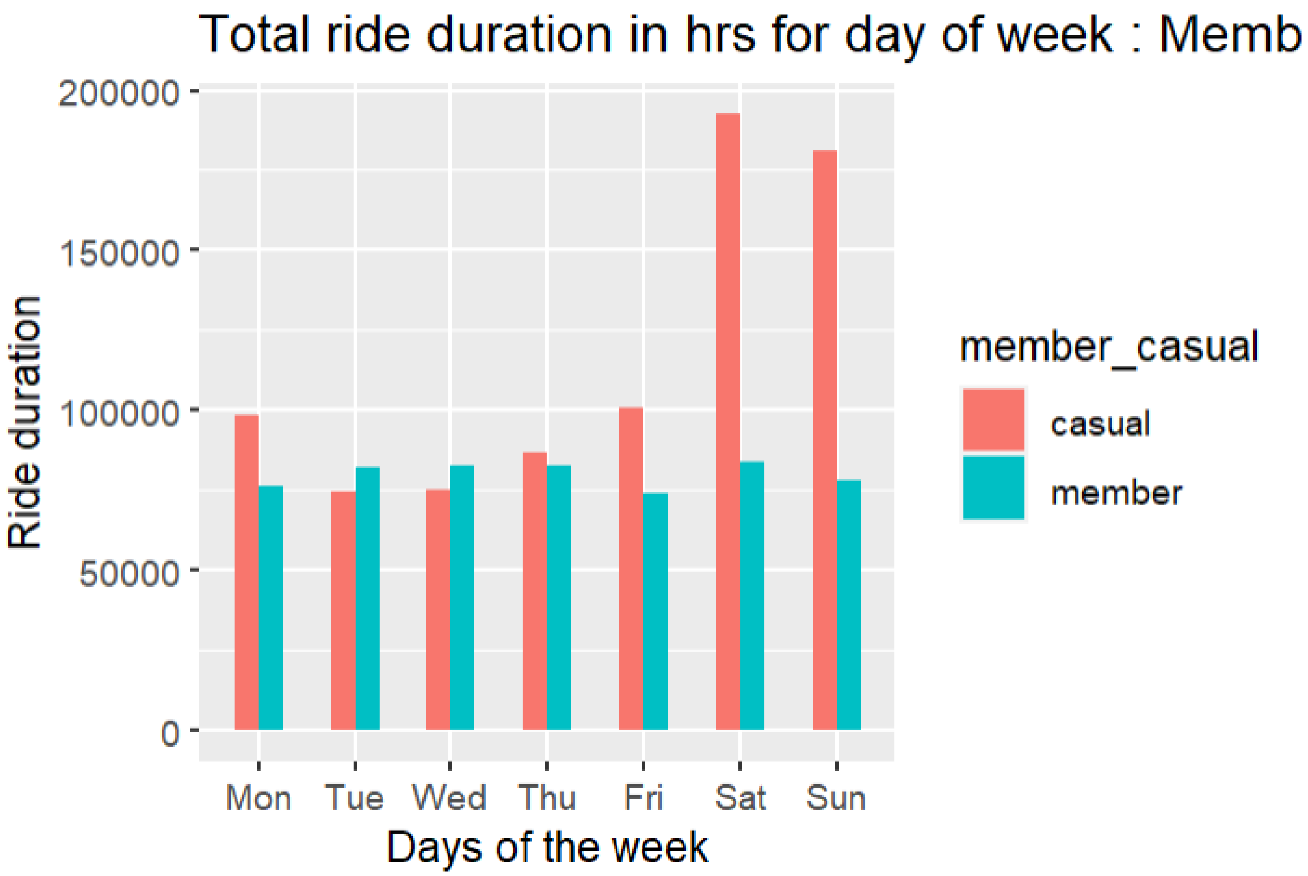
- Break up of number of rides over the day of the week

```
Yearly_Cleaned %>%
  group_by(member_casual,weekday) %>%
  summarise(Number_of_rides=length(ride_id),.groups="drop") %>%
  arrange(member_casual,weekday) %>%
  ggplot(aes(x=weekday,y=Number_of_rides,fill=member_casual))+
  geom_col(width = 0.5,position = position_dodge(width = 0.5))+
  labs(title="Number of rides in day of week for members & Casuals") +
  xlab("Days of the week") +
  ylab("Number of rides") +
  scale_y_continuous(labels = function(x)format(x,scientific=FALSE))
```



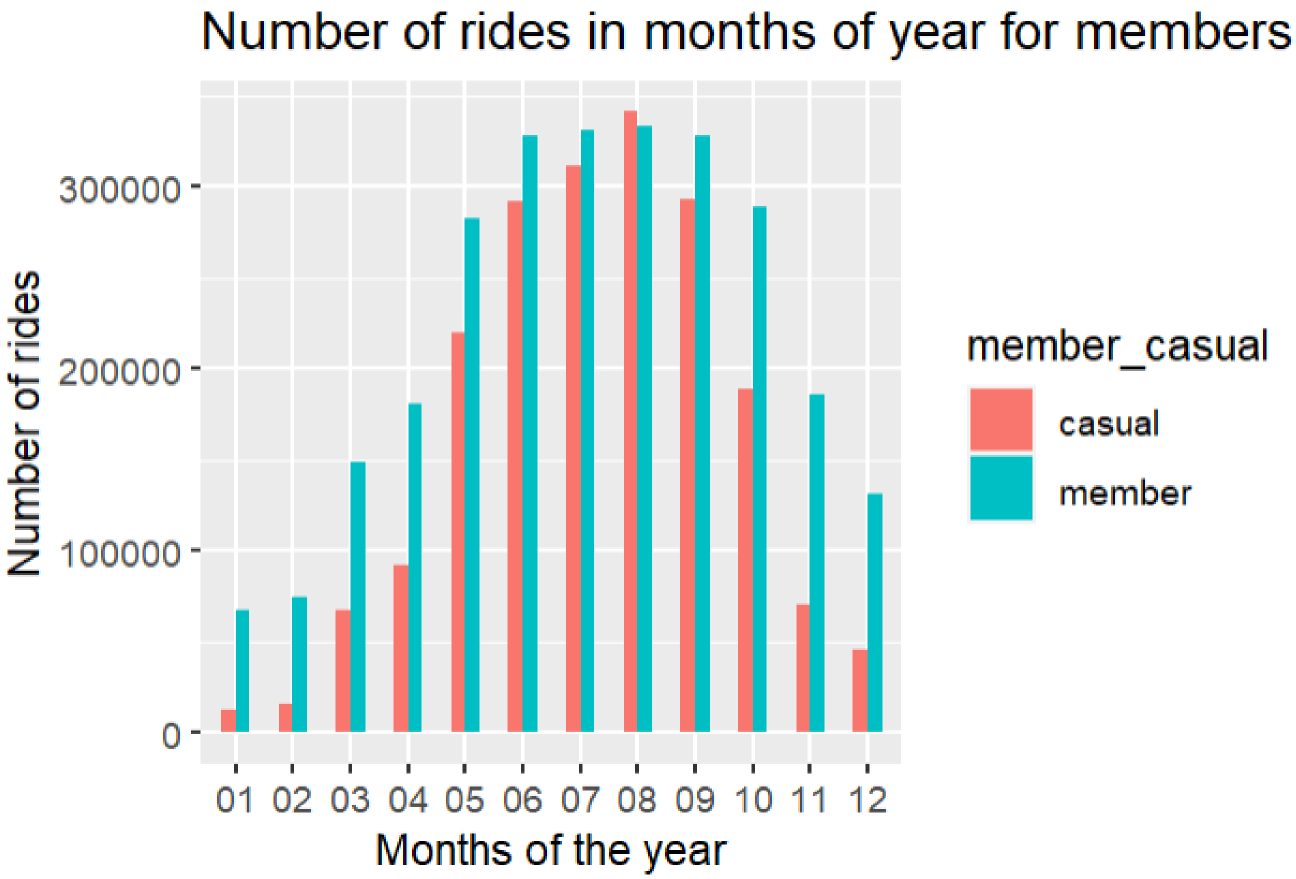
- Break up of ride duration based on the day of the week

```
Yearly_Cleaned %>%
  group_by(member_casual,weekday) %>%
  summarise(Total_ride_duration_hrs=sum(rideduration)/60,.groups="drop") %>%
  arrange(member_casual,weekday) %>%
  ggplot(aes(x=weekday,y=Total_ride_duration_hrs,fill=member_casual))+
  geom_col(width = 0.5,position = position_dodge(width = 0.5))+
  labs(title="Total ride duration in hrs for day of week : Members & Casuals") +
  xlab("Days of the week") +
  ylab("Ride duration") +
  scale_y_continuous(labels = function(x)format(x,scientific=FALSE))
```



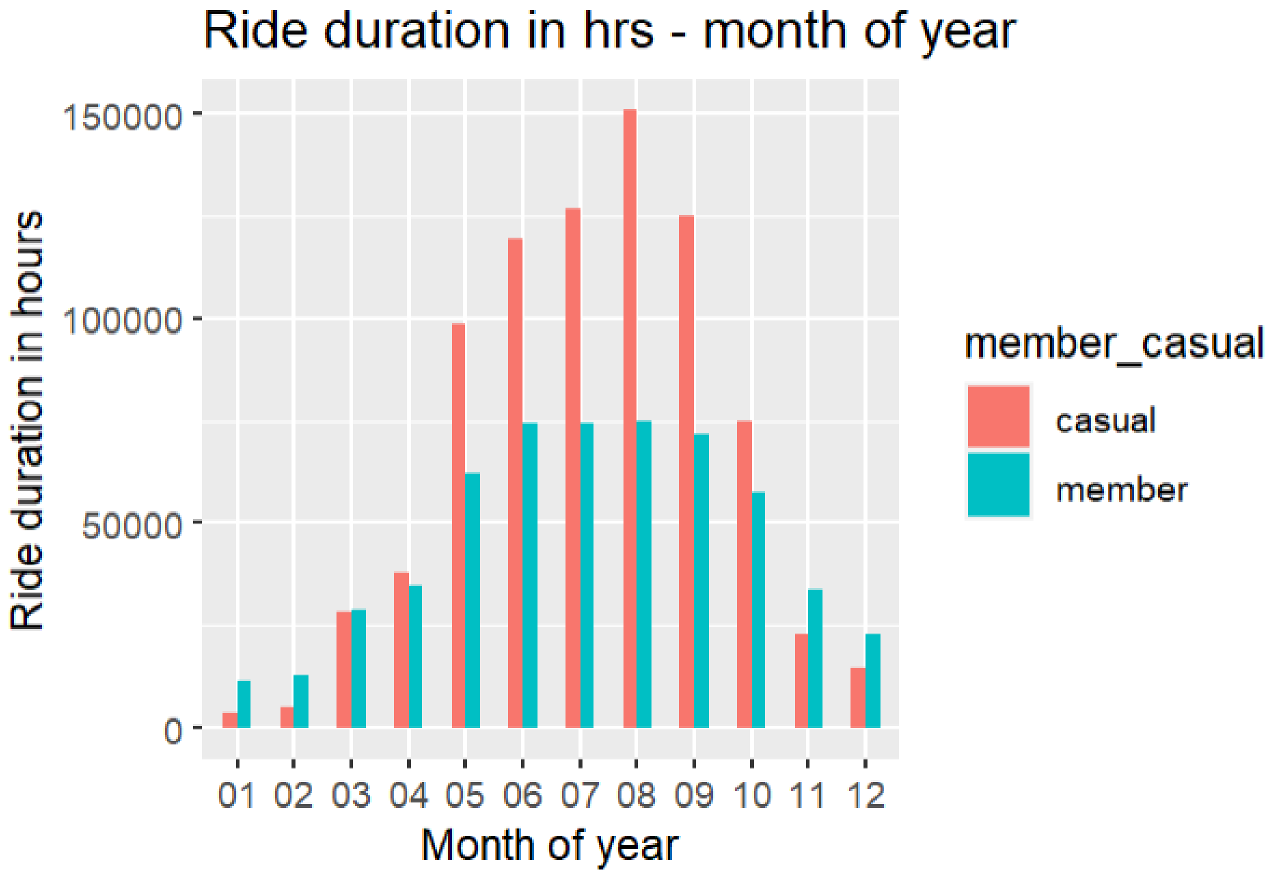
- Break up of number of rides over the month of the year

```
Yearly_Cleaned %>%
  group_by(member_casual,month) %>%
  summarise(Number_of_rides = length(ride_id)) %>%
  arrange(member_casual,month) %>%
  ggplot(aes(x=month,y=Number_of_rides,fill=member_casual))+
    geom_col(width = 0.5,position = position_dodge(width = 0.5))+
    labs(title="Number of rides in months of year for members & Casuals") +
    xlab("Months of the year")+ ylab("Number of rides")+
    scale_y_continuous(labels = function(x)format(x,scientific=FALSE))
```



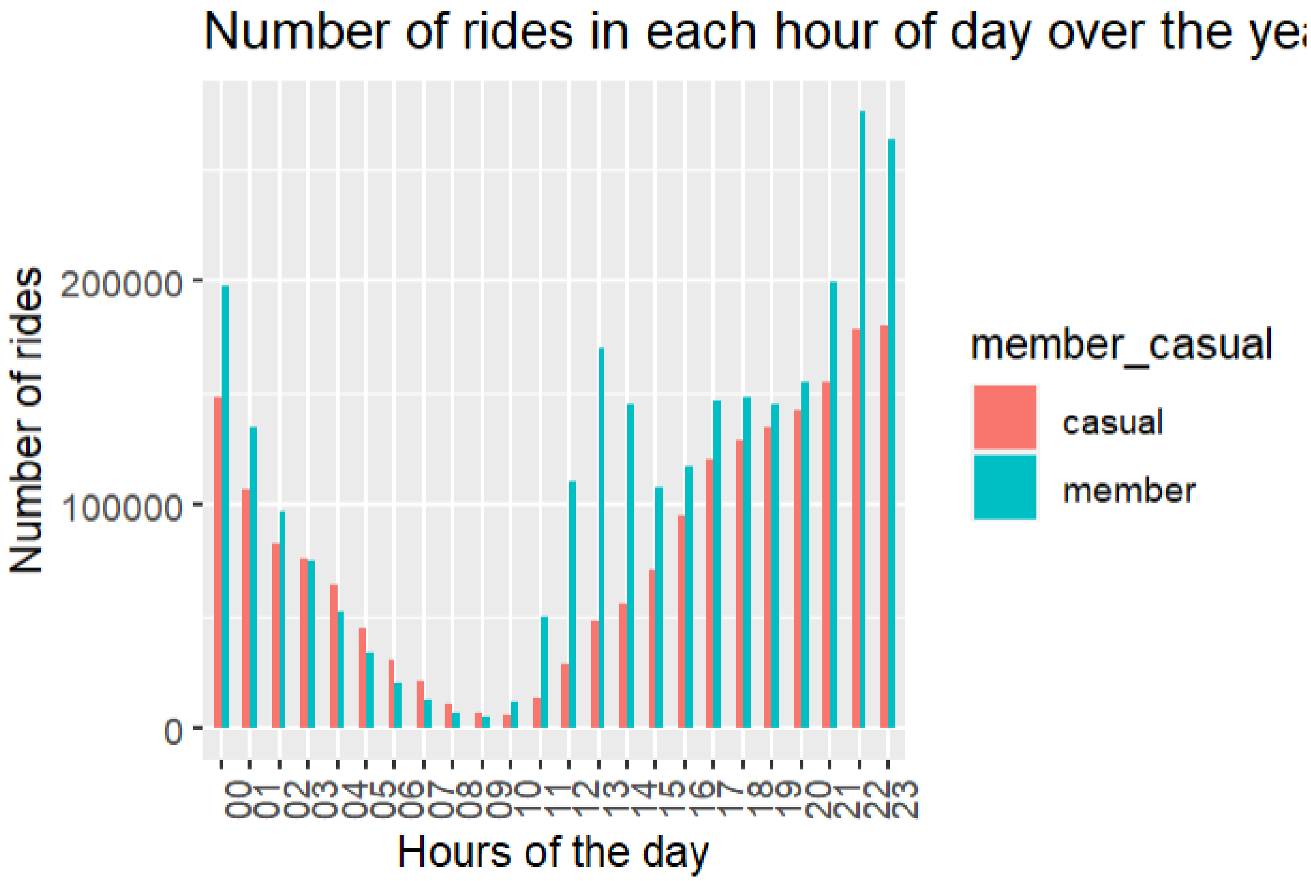
- Break up of ride duration over the month of the year

```
Yearly_Cleaned %>%
  group_by(member_casual,month) %>%
  arrange(member_casual,month) %>%
  summarize(Number_of_hours = sum(rideduration)/60) %>%
  ggplot(aes(x=month,y=Number_of_hours,fill=member_casual))+
  geom_col(width = 0.5,position = position_dodge(width = 0.5))+
  labs(title="Ride duration in hrs - month of year") +
  xlab("Month of year") +
  ylab("Ride duration in hours") +
  scale_y_continuous(labels = function(x)format(x,scientific=FALSE))
```



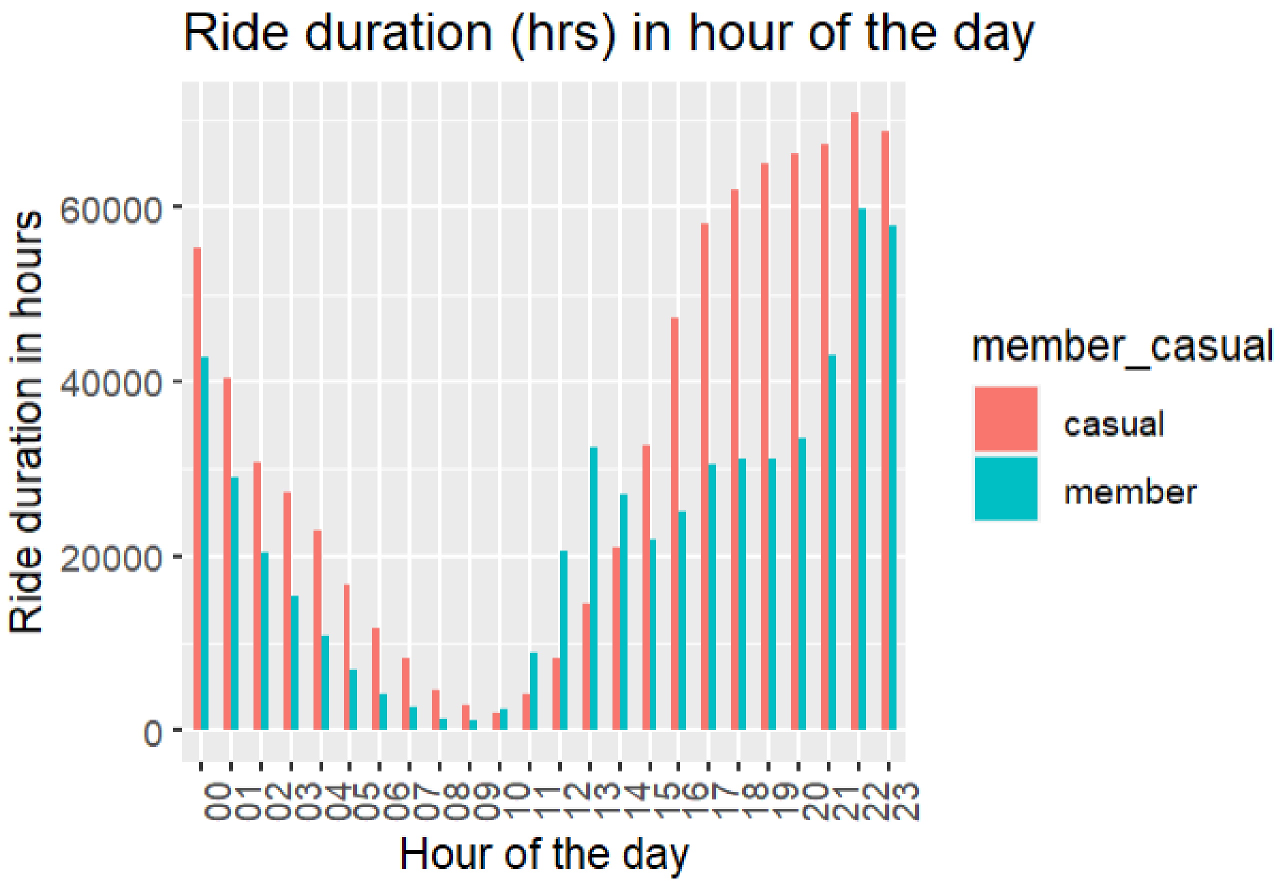
- Break up of number of rides over the hour of the day

```
Yearly_Cleaned %>%
  group_by(member_casual,starthour) %>%
  summarise(Number_of_rides = length(ride_id)) %>%
  arrange(member_casual,starthour) %>%
  ggplot(aes(x=starthour,y=Number_of_rides,fill=member_casual))+
  geom_col(width = 0.5,position = position_dodge(width = 0.5))+
  labs(title="Number of rides in each hour of day over the year") +
  xlab("Hours of the day")+ ylab("Number of rides")+
  theme(axis.text.x = element_text(angle = 90))+
  scale_y_continuous(labels = function(x)format(x,scientific=FALSE))
```



- Break up of ride duration over the hour of the day

```
Yearly_Cleaned %>%
  group_by(member_casual,starthour) %>%
  arrange(member_casual,starthour) %>%
  summarize(Number_of_hours = sum(rideduration)/60) %>%
  ggplot(aes(x=starthour,y=Number_of_hours,fill=member_casual))+
  geom_col(width = 0.5,position = position_dodge(width = 0.5))+
  labs(title="Ride duration (hrs) in hour of the day") +
  xlab("Hour of the day")+ ylab("Ride duration in hours")+
  theme(axis.text.x = element_text(angle = 90)) +
  scale_y_continuous(labels = function(x)format(x,scientific=FALSE))
```


Ride duration (hrs) in hour of the day
Ride duration in hours
0
20000
40000
60000
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Hour of the day
member_casual
casual
member