# Google Capstone project

Code

The following R notebook describes my approach to the google case study capstone project.

Case study description: Here

First I downloaded the specific files from amazonaws.com. In my case the time period was 1/2022 to 12/2022. I unzipped all files in ONE directory. I simply used _copy *.csv merged.csv_ in *cmd* to merge all csv's to one file and imported that file into R Studio.

Hide

```r
library(tidyverse)
library(dplyr)
library(lubridate)
library(ggplot2)
```

Hide

```r
merged <- read_csv("merged.csv",
                   col_types = cols(start_station_name = col_skip(),
                                    start_station_id = col_skip(), end_station_name = col_skip(),
                                    end_station_id = col_skip(), start_lng = col_skip(),start_lat = col_skip(),

                                    end_lat = col_skip(), end_lng = col_skip()))
```

To minimize the data overhead multiple columns are not imported.

# Data cleaning and mutation

Hide

```r
Data <- subset(merged, ride_id != 'ride_id' )
```

Cleaning the Data. This can't be done in Excel cause the number of rows acceded 1048576 rows.

Hide

```r
Data <- Data %>%
  mutate(duration = ymd_hms(ended_at) - ymd_hms(started_at))
```

Adds a new column duration to the dataset.

Hide

```r
Data <- subset(Data, duration >= 60)
```

Removes every entry with a duration below 60 sec. Because trips under 60 seconds are not plausible.

Hide

```r
Data <- Data %>%
  mutate(weekday = weekdays(as.Date(started_at)))
```

Adds the column weekday to the dataset.

Hide

```r
Data <- Data %>%
  mutate(hour = hour(started_at))
```

Adds the column (starting) hour to the dataset.

Hide

```r
Data <- Data %>%
  mutate(month = month(started_at))
```

Adds the column month to the dataset.

Now comes a big chunck of sorting and transformations.

Hide

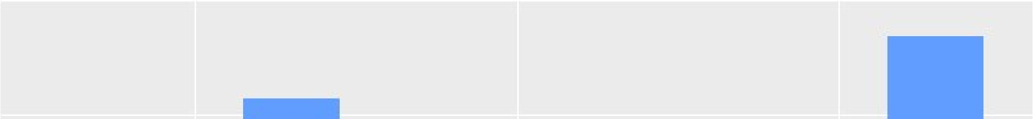
```r

DataC <- subset(Data, member_casual == "casual")
```

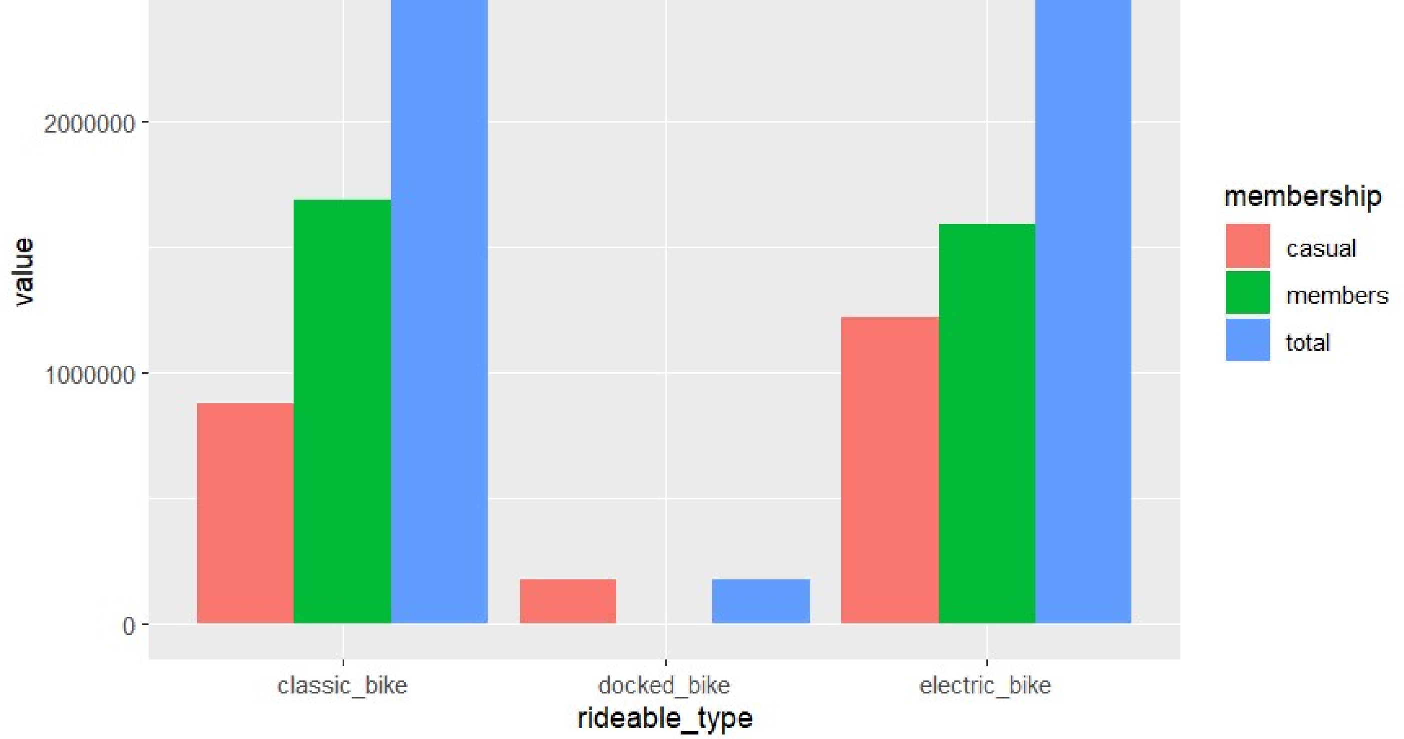
```
DataM <- subset(Data, member_casual == "member")

HoursAll <- Data %>%
  count(hour)

HoursC <- DataC %>%
  count(hour)

HoursM <- DataM %>%
  count(hour)

Hours <- bind_cols(HoursAll,HoursC$n,HoursM$n)

colnames(Hours) <- c("hour","total","casual","members")

#weekdays

WeekdaysAll <- Data %>%
  count(weekday)

WeekdaysC <- DataC %>%
  count(weekday)

WeekdaysM <- DataM %>%
  count(weekday)

Weekdays <- bind_cols(WeekdaysAll,WeekdaysC$n,WeekdaysM$n)

colnames(Weekdays) <-c("weekday","total","casual","members")


#months

MonthsAll <- Data %>%
  count(month)

MonthsC <- DataC %>%
  count(month)

MonthsM <- DataM %>%
  count(month)

Months <- bind_cols(MonthsAll,MonthsC$n,MonthsM$n)

colnames(Months) <- c("month","total","casual","members")

#rideables

RideablesAll <- Data %>%
  count(rideable_type)

RideablesC <- DataC %>%
  count(rideable_type)

RideablesM <- DataM %>%
  count(rideable_type)

Rideables <- data.frame(rideable_type = c("classic_bike","docked_bike","electric_bike"),
                        total = c(RideablesAll$n[1],RideablesAll$n[2],RideablesAll$n[3]),
                        members = c(RideablesM$n[1],0,RideablesM$n[2]),
                        casual = c(RideablesC$n[1],RideablesC$n[2],RideablesC$n[3]))
```

The following plot shows the preferred rideables types.

Hide

```
dfm <- pivot_longer(Rideables, -rideable_type, names_to="membership", values_to="value")

ggplot(dfm,aes(x = rideable_type,y = value)) +
  geom_bar(aes(fill = membership),stat = "identity",position = "dodge")+
  scale_y_continuous(labels = ~ format(.x, scientific = FALSE))
```

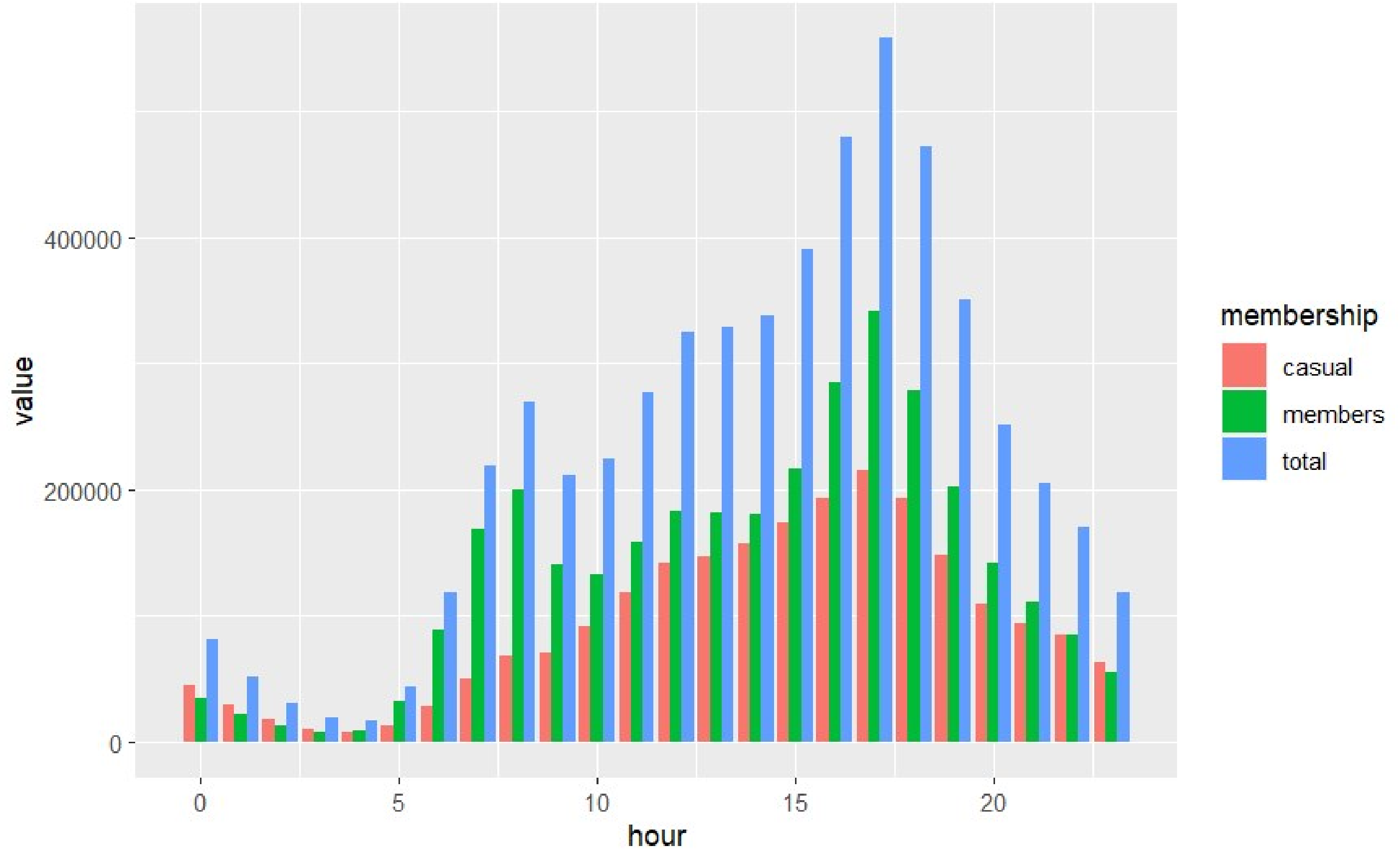




The following plot shows how the riding hours are distributed.

Hide

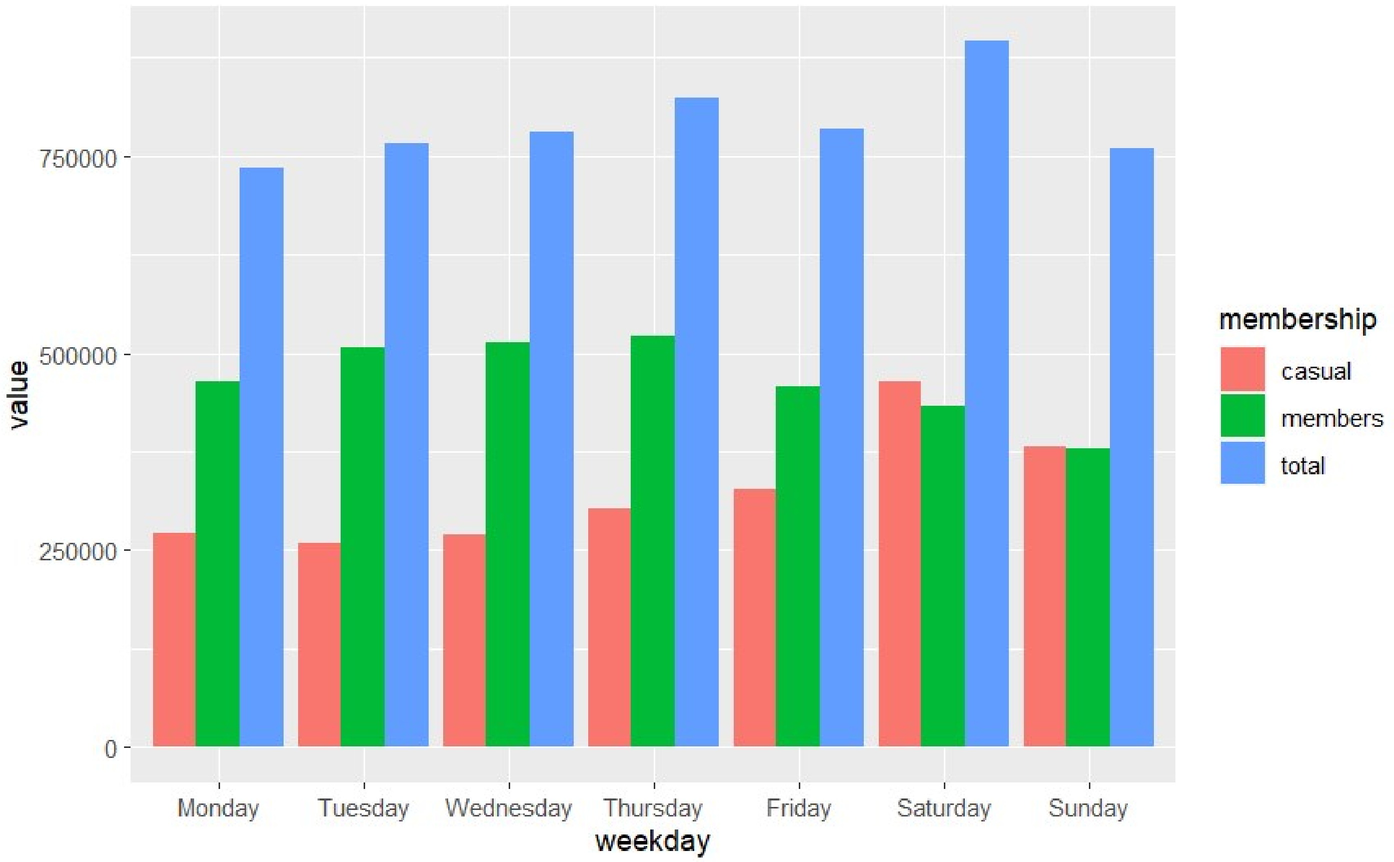
```
dfm <- pivot_longer(Hours, -hour, names_to="membership", values_to="value")

ggplot(dfm,aes(x = hour,y = value)) +
  geom_bar(aes(fill = membership),stat = "identity",position = "dodge")+
  scale_y_continuous(labels = ~ format(.x, scientific = FALSE))
```



Here are the weekdays.

Hide

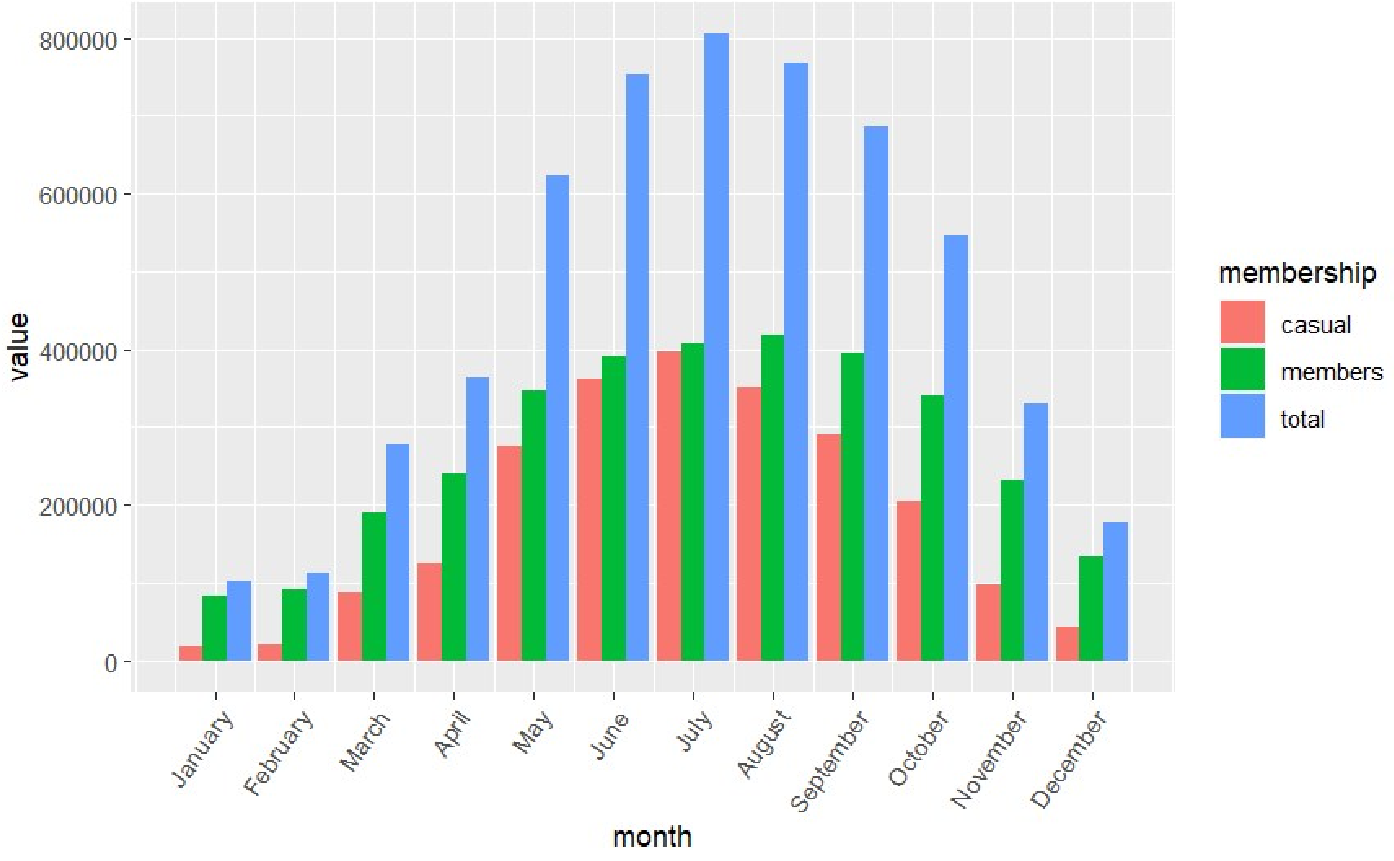
```
dfm <- pivot_longer(Weekdays, -weekday, names_to="membership", values_to="value")

xf = c("Monday","Tuesday","Wednesday","Thursday","Friday","Saturday","Sunday")

ggplot(dfm,aes(x = weekday,y = value)) +
  geom_bar(aes(fill = membership),stat = "identity",position = "dodge")+
  scale_y_continuous(labels = ~ format(.x, scientific = FALSE))+
  scale_x_discrete(limits = xf)
```



And finally over a years period.

Hide

```
dfm <- pivot_longer(Months, -month, names_to="membership", values_to="value")

ggplot(dfm,aes(x =month,y = value)) +
  geom_bar(aes(fill = membership),stat = "identity",position = "dodge")+
  scale_y_continuous(labels = ~ format(.x, scientific = FALSE))+
  scale_x_continuous(breaks = seq_along(month.name), labels = month.name)+
  theme(axis.text.x = element_text(angle = 55, hjust = 1))
```



# Conclusion

Not surprisingly the demand for bikes is higher in summer and lower in winter. When it comes to casual riders the demand is higher on weekends. Also the daytime distribution is in line with real life experience.

# Advice

Target casual riders for membership conversion, when their demand is the highest on weekends and in summer. Try to lift up the usage of casual riders during the week, through advertisements or through lowering the rates.

# Afterthoughts

During this project I realized that one major drawback of R is that it’s single threaded. For this project a workaround for parallelization was out of my scope, but if you have even a bigger dataset, I highly recommend to use forks of libraries which allow parallelization like multidplyr etc.