Google Data Analytics Capstone Case Study 1

2022-07-29

Ask

Guiding questions

- What is the problem you are trying to solve?
  - The goal is to build a marketing strategies to covert casual riders to annual members.
- How can your insights drive business decisions?
  - The insights will help the marketing team to increase the annual members.

Prepare

Guiding questions

- Where is your data located?
  - The data is located in the Google database; link is provided with the case study description
- How is the data organized?
  - The data is separated by month, each on its own csv (between July 2021 and June 2022).
- Are there issues with bias or credibility in this data? Does your data ROCCC?
  - There are not any issues with bias, since the population of the dataset is its own clients as bike riders. It's ROCC because it's reliable, original, comprehensive, current, and cited.
- How are you addressing licnesing, privacy, security, and accessibility?
  - The company has their own license over the dataset. The dataset does not have personal information about the riders.
- How did you verify the data's integrity?
  - All the files have consistent names, columns and each column has the correct data type.
- How does it help you answer your question?
  - It contains some key insights about the routine activities of riders.
- Are there any problems with the data?
  - It would be more helpful if there's more information about the riders.

Process

Guiding questions

- What tools are you choosing and why?
  - Here, I'm using R to merge the data of 12 months into 1 large data-frame because it's easier to merge a large dataset.
- Have you ensured your data's integrity?
  - Yes, the data is consistent throughout the columns.
- What steps have you taken to ensure that your data is clean?
  - First, I removed duplicates. Then, I re-format the date and time of the columns.
- How can you verify that your data is clean and ready to analyze?
  - It can be verified by this document.
- Have you documented your cleaning process so you can review and share those results?
  - Yes, it's all documented in this R notebook.

Code

Dependences

```
library(tidyverse)
library(dplyr)
```

Concatenating Concatenating csv files

```
csv_files <- list.files(path = "data", recursive = TRUE, full.names=TRUE)

df <- do.call(rbind, lapply(csv_files, read.csv))

head(df, 5)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 0A1B623926EF4E16   docked_bike 2021-07-02 14:44:36 2021-07-02 15:19:58
## 2 B2D5583A5A5E76EE  classic_bike 2021-07-07 16:57:42 2021-07-07 17:16:09
## 3 6F264597DDBF427A  classic_bike 2021-07-25 11:30:55 2021-07-25 11:48:45
## 4 379B58EAB20E8AA5  classic_bike 2021-07-08 22:08:30 2021-07-08 22:23:32
## 5 6615C1E4EB08E8FB electric_bike 2021-07-28 16:08:06 2021-07-28 16:27:09
##             start_station_name start_station_id             end_station_name
## 1 Michigan Ave & Washington St            13001   Halsted St & North Branch St
## 2   California Ave & Cortez St            17660          Wood St & Hubbard St
## 3         Wabash Ave & 16th St           SL-012          Rush St & Hubbard St
## 4   California Ave & Cortez St            17660        Carpenter St & Huron St
## 5   California Ave & Cortez St            17660 Elizabeth (May) St & Fulton St
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1   KA1504000117  41.88398 -87.62468 41.89937 -87.64848        casual
```

```
## 2          13432  41.90036 -87.69670 41.88990 -87.67147        casual
## 3   KA1503000044  41.86038 -87.62581 41.89017 -87.62619        member
## 4          13196  41.90036 -87.69670 41.89456 -87.65345        member
## 5          13197  41.90035 -87.69668 41.88659 -87.65839        casual
```

Data Cleaning

```
df_no_dups <- df[!duplicated(df), ]
print(paste("Removed", nrow(df) - nrow(df_no_dups), "duplicated rows"))
```

Removing duplicates

```
## [1] "Removed 0 duplicated rows"
```

```
df_no_dups$started_at <- as.POSIXct(df_no_dups$started_at, "%Y-%m-%d %H:%M:%S")
df_no_dups$ended_at <- as.POSIXct(df_no_dups$ended_at, "%Y-%m-%d %H:%M:%S")
```

Parse datetime columns

ride_time_minute The total ride time in minutes

```
df_no_dups <- df_no_dups %>%
  mutate(ride_time_minutes = as.numeric(df_no_dups$ended_at - df_no_dups$started_at)/ 60)

summary(df_no_dups$ride_time_minutes)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  -137.42     6.28    11.17    20.28    20.20 49107.15
```

year_month Separate the year and month might be helful

```
df_no_dups <- df_no_dups %>%
    mutate(year_month = paste(strftime(df_no_dups$started_at, "%Y"),
                              "-",
                              strftime(df_no_dups$started_at, "%m"),
                              "(", strftime(df_no_dups$started_at, "%b"), ")"))
unique(df_no_dups$year_month)
```

```
##  [1] "2021 - 07 ( Jul )" "2021 - 06 ( Jun )" "2021 - 08 ( Aug )"
##  [4] "2021 - 09 ( Sep )" "2021 - 10 ( Oct )" "2021 - 11 ( Nov )"
##  [7] "2021 - 12 ( Dec )" "2022 - 01 ( Jan )" "2022 - 02 ( Feb )"
## [10] "2022 - 03 ( Mar )" "2022 - 04 ( Apr )" "2022 - 05 ( May )"
## [13] "2022 - 06 ( Jun )"
```

weekday Show the weekday

```
df_no_dups <- df_no_dups %>%
  mutate(weekday = paste(strftime(df_no_dups$ended_at, "%u"),
                         "-",
                         strftime(df_no_dups$ended_at, "%a")))
unique(df_no_dups$weekday)
```

```
## [1] "5 - Fri" "3 - Wed" "7 - Sun" "4 - Thu" "6 - Sat" "1 - Mon" "2 - Tue"
```

start_hour Show the start hour

```
df_no_dups <- df_no_dups %>%
  mutate(start_hour = strftime(df_no_dups$started_at, "%H"))

unique(df_no_dups$start_hour)
```

```
##  [1] "07" "09" "04" "15" "10" "05" "11" "14" "13" "08" "00" "12" "02" "03" "06"
## [16] "22" "18" "01" "16" "21" "17" "19" "23" "20"
```

Save the file

```
df_no_dups %>% write.csv("cleaned_data.csv")
```

Analyze

Code

```
head(df_no_dups)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 0A1B623926EF4E16   docked_bike 2021-07-02 14:44:36 2021-07-02 15:19:58
## 2 B2D5583A5A5E76EE  classic_bike 2021-07-07 16:57:42 2021-07-07 17:16:09
## 3 6F264597DDBF427A  classic_bike 2021-07-25 11:30:55 2021-07-25 11:48:45
## 4 379B58EAB20E8AA5  classic_bike 2021-07-08 22:08:30 2021-07-08 22:23:32
## 5 6615C1E4EB08E8FB electric_bike 2021-07-28 16:08:06 2021-07-28 16:27:09
## 6 62DC2B32872F9BA8 electric_bike 2021-07-29 17:09:08 2021-07-29 17:15:00
##             start_station_name start_station_id              end_station_name
## 1 Michigan Ave & Washington St            13001   Halsted St & North Branch St
## 2   California Ave & Cortez St            17660           Wood St & Hubbard St
## 3         Wabash Ave & 16th St           SL-012           Rush St & Hubbard St
## 4   California Ave & Cortez St            17660        Carpenter St & Huron St
## 5   California Ave & Cortez St            17660 Elizabeth (May) St & Fulton St
## 6   California Ave & Cortez St            17660  Albany Ave & Bloomingdale Ave
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1   KA1504000117  41.88398 -87.62468 41.89937 -87.64848        casual
## 2          13432  41.90036 -87.69670 41.88990 -87.67147        casual
## 3   KA1503000044  41.86038 -87.62581 41.89017 -87.62619        member
## 4          13196  41.90036 -87.69670 41.89456 -87.65345        member
```

```
## 5          13197  41.90035 -87.69668 41.88659 -87.65839        casual
## 6          15655  41.90033 -87.69674 41.91389 -87.70513        casual
##   ride_time_minutes        year_month weekday start_hour
## 1         35.366667 2021 - 07 ( Jul ) 5 - Fri         07
## 2         18.450000 2021 - 07 ( Jul ) 3 - Wed         09
## 3         17.833333 2021 - 07 ( Jul ) 7 - Sun         04
## 4         15.033333 2021 - 07 ( Jul ) 4 - Thu         15
## 5         19.050000 2021 - 07 ( Jul ) 3 - Wed         09
## 6          5.866667 2021 - 07 ( Jul ) 4 - Thu         10
```

```
summary(df_no_dups)
```

```
##    ride_id          rideable_type        started_at                    
##  Length:5900385     Length:5900385     Min.   :2021-07-01 00:00:22.00  
##  Class :character   Class :character   1st Qu.:2021-08-26 07:57:58.00  
##  Mode  :character   Mode  :character   Median :2021-10-27 17:35:55.00  
##                                        Mean   :2021-12-12 00:11:36.51  
##                                        3rd Qu.:2022-04-25 13:41:23.00  
##                                        Max.   :2022-06-30 23:59:58.00  
##                                                                        
##     ended_at                      start_station_name start_station_id  
##  Min.   :2021-07-01 00:04:51.00   Length:5900385     Length:5900385    
##  1st Qu.:2021-08-26 08:11:00.00   Class :character   Class :character  
##  Median :2021-10-27 17:49:46.00   Mode  :character   Mode  :character  
##  Mean   :2021-12-12 00:31:53.47                                        
##  3rd Qu.:2022-04-25 13:57:17.00                                        
##  Max.   :2022-07-13 04:21:06.00                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:5900385     Length:5900385     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      ride_time_minutes
##  Min.   :41.39   Min.   :-88.97   Length:5900385     Min.   : -137.42  
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character   1st Qu.:    6.28  
##  Median :41.90   Median :-87.64   Mode  :character   Median :   11.17  
##  Mean   :41.90   Mean   :-87.65                      Mean   :   20.28  
##  3rd Qu.:41.93   3rd Qu.:-87.63                      3rd Qu.:   20.20  
##  Max.   :42.17   Max.   :-87.49                      Max.   :49107.15  
##  NA's   :5374    NA's   :5374                                         
##   year_month          weekday           start_hour       
##  Length:5900385     Length:5900385     Length:5900385    
##  Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character  
##                                                          
##                                                          
##                                                          
## 
```

Function to resize the plots

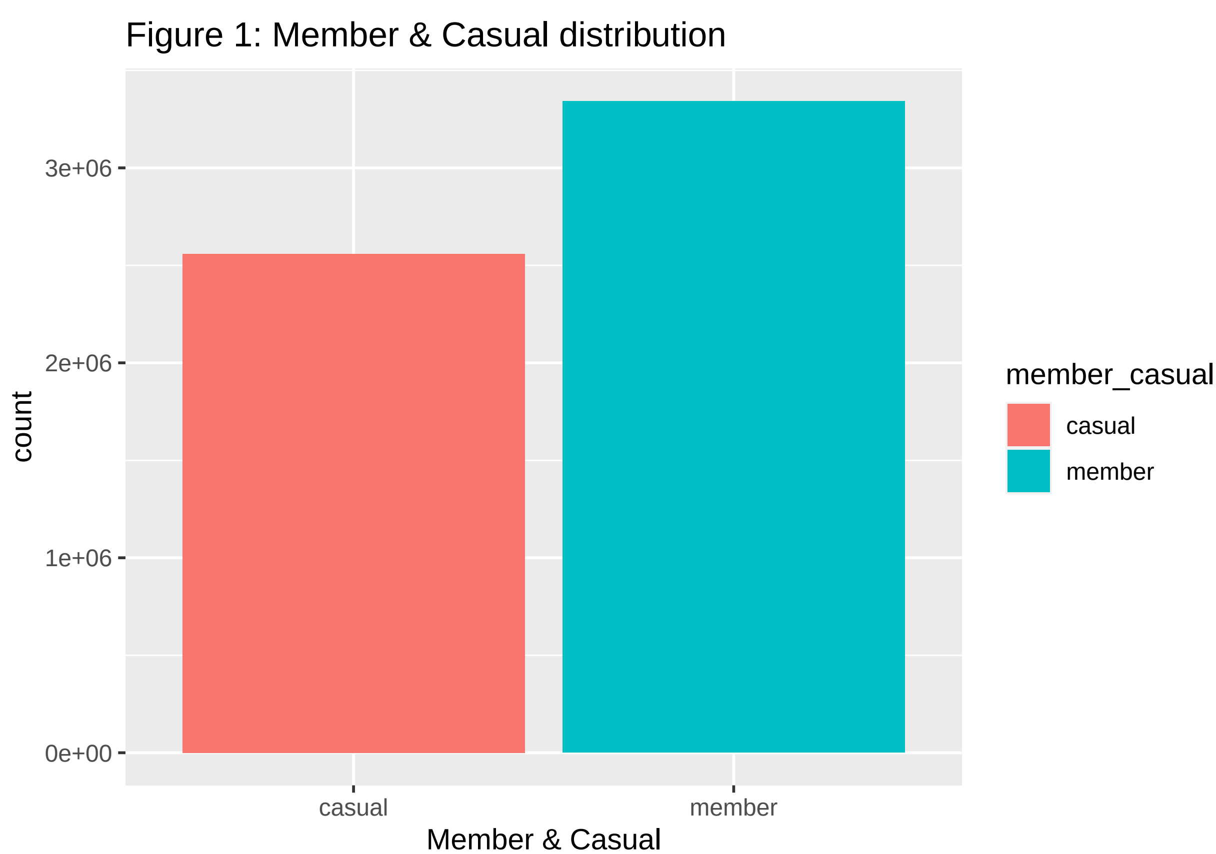
```r
fig <- function(width, height) {
  options(repr.plot.width = width,
          repr.plot.height = height)
}
```

Distribution of Members and Casual riders

```r
df_no_dups %>% group_by(member_casual) %>%
  summarise(freq = length(ride_id),
            percent_total = length(ride_id)/ nrow(df_no_dups) *100)
```

```
## # A tibble: 2 x 3
##   member_casual    freq percent_total
##   <chr>           <int>         <dbl>
## 1 casual        2558227          43.4
## 2 member        3342158          56.6
```

```r
fig(16,8)
ggplot(df_no_dups, aes(member_casual, fill = member_casual)) +
  geom_bar() +
  labs(x= "Member & Casual", title = "Figure 1: Member & Casual distribution")
```

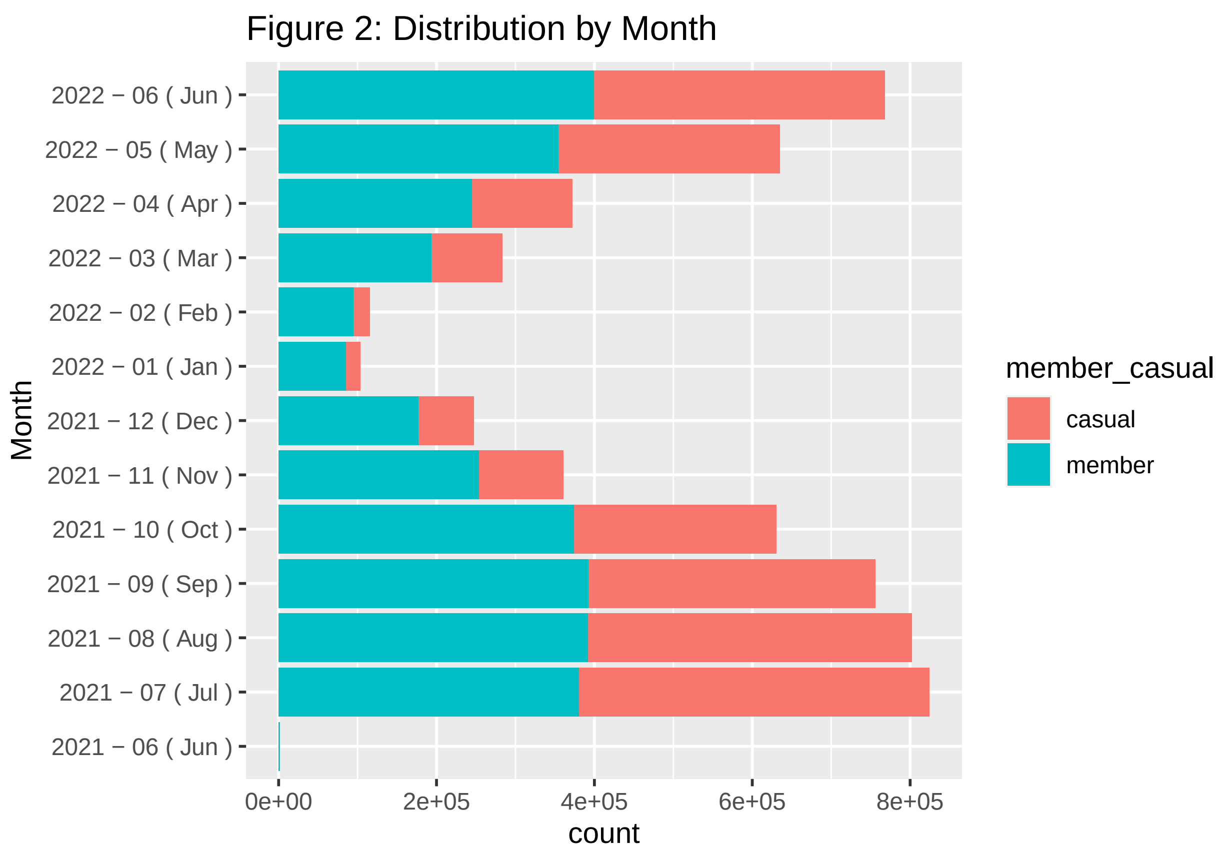


By Year and Month

```r
df_no_dups %>%
  group_by(year_month) %>%
  summarise(freq = length(ride_id),
            percent_total = length(ride_id)/ nrow(df_no_dups) *100,
            'member_%' = sum(member_casual == "member")/ length(ride_id) *100,
            'casual_%' = sum(member_casual == "casual")/ length(ride_id) *100,
            'member_casual_diff' = (sum(member_casual == "member") - sum(member_casual == "casual"))/ le
```

```
## # A tibble: 13 x 6
##    year_month          freq percent_total `member_%` `casual_%` member_casual_d~
##    <chr>              <int>         <dbl>      <dbl>      <dbl>            <dbl>
##  1 2021 - 06 ( Jun )   1393        0.0236       57.3       42.7            14.6
##  2 2021 - 07 ( Jul ) 824227       14.0          46.2       53.8            -7.65
##  3 2021 - 08 ( Aug ) 802503       13.6          48.8       51.2            -2.43
##  4 2021 - 09 ( Sep ) 756238       12.8          51.9       48.1             3.75
##  5 2021 - 10 ( Oct ) 630747       10.7          59.3       40.7            18.5
##  6 2021 - 11 ( Nov ) 360513        6.11         70.3       29.7            40.7
##  7 2021 - 12 ( Dec ) 247111        4.19         71.7       28.3            43.3
##  8 2022 - 01 ( Jan ) 103464        1.75         82.6       17.4            65.2
##  9 2022 - 02 ( Feb ) 115986        1.97         81.5       18.5            62.9
## 10 2022 - 03 ( Mar ) 283417        4.80         68.3       31.7            36.6
## 11 2022 - 04 ( Apr ) 371945        6.30         65.9       34.1            31.7
## 12 2022 - 05 ( May ) 634768       10.8          55.9       44.1            11.7
## 13 2022 - 06 ( Jun ) 768073       13.0          52.0       48.0             4.00
```

```r
df_no_dups %>%
  ggplot(aes(year_month, fill = member_casual)) +
  geom_bar() +
  labs(x="Month", title = "Figure 2: Distribution by Month") +
  coord_flip()
```

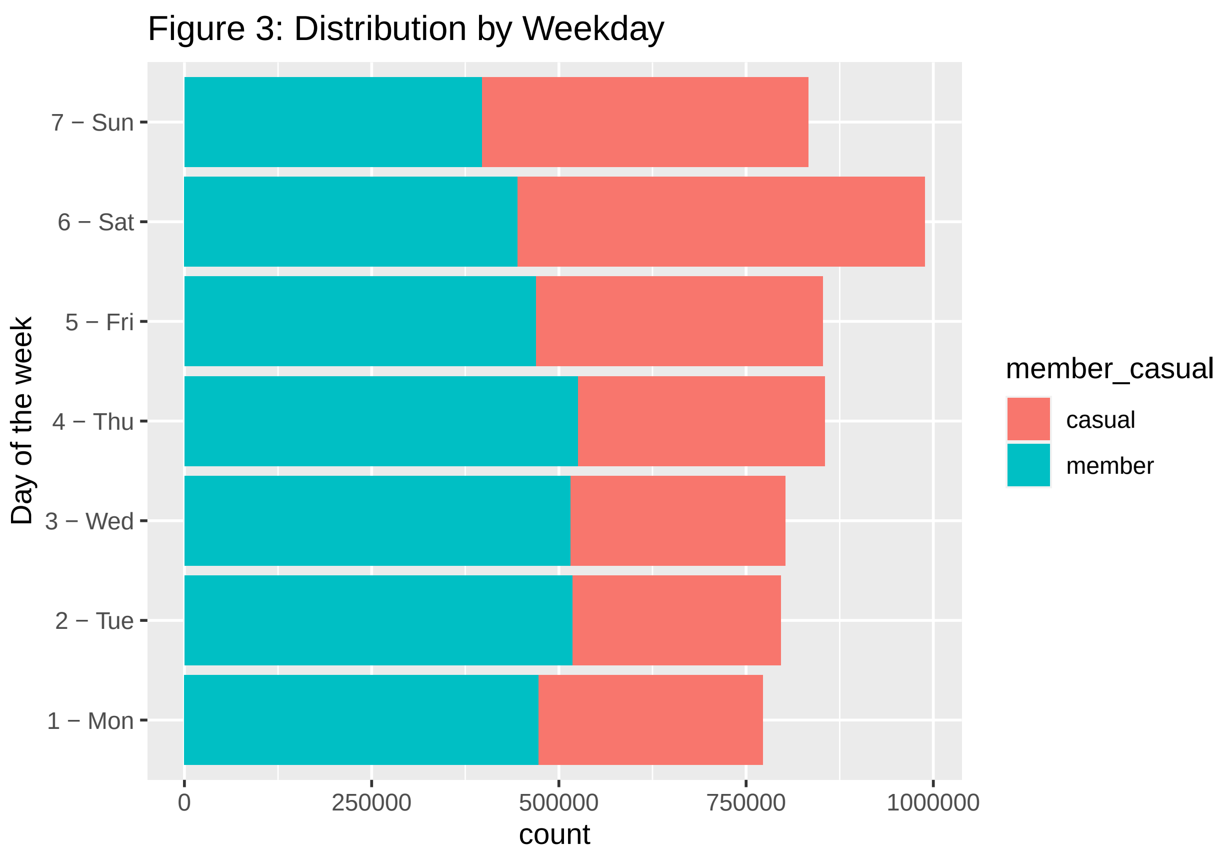


By Weekday

```
df_no_dups %>%
  group_by(weekday) %>%
  summarise(freq = length(ride_id),
            percent_total = length(ride_id)/ nrow(df_no_dups) *100,
            'member_%' = sum(member_casual == "member")/ length(ride_id) *100,
            'casual_%' = sum(member_casual == "casual")/ length(ride_id) *100,
            'member_casual_diff' = (sum(member_casual == "member") - sum(member_casual == "casual"))/ le
```

```
## # A tibble: 7 x 6
##   weekday   freq percent_total `member_%` `casual_%` member_casual_diff
##   <chr>    <int>         <dbl>      <dbl>      <dbl>              <dbl>
## 1 1 - Mon 772094          13.1       61.2       38.8              22.5
## 2 2 - Tue 796199          13.5       65.1       34.9              30.1
## 3 3 - Wed 802366          13.6       64.2       35.8              28.5
## 4 4 - Thu 855572          14.5       61.4       38.6              22.7
## 5 5 - Fri 852309          14.4       55.1       44.9              10.1
## 6 6 - Sat 988579          16.8       45.0       55.0             -10.0
## 7 7 - Sun 833266          14.1       47.7       52.3              -4.69
```

```
df_no_dups %>%
  ggplot(aes(weekday, fill = member_casual)) +
  geom_bar() +
  labs(x="Day of the week", title = "Figure 3: Distribution by Weekday") +
  coord_flip()
```

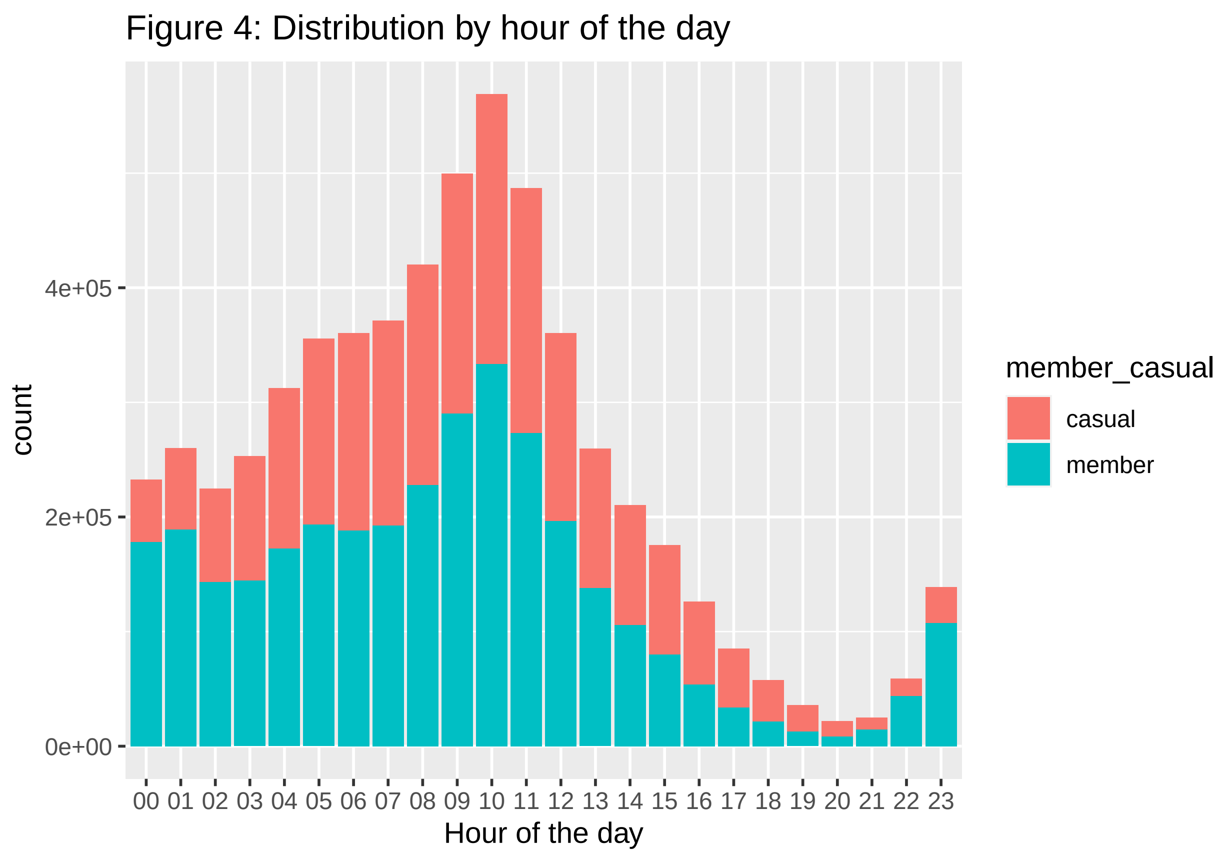


By Hour of the day

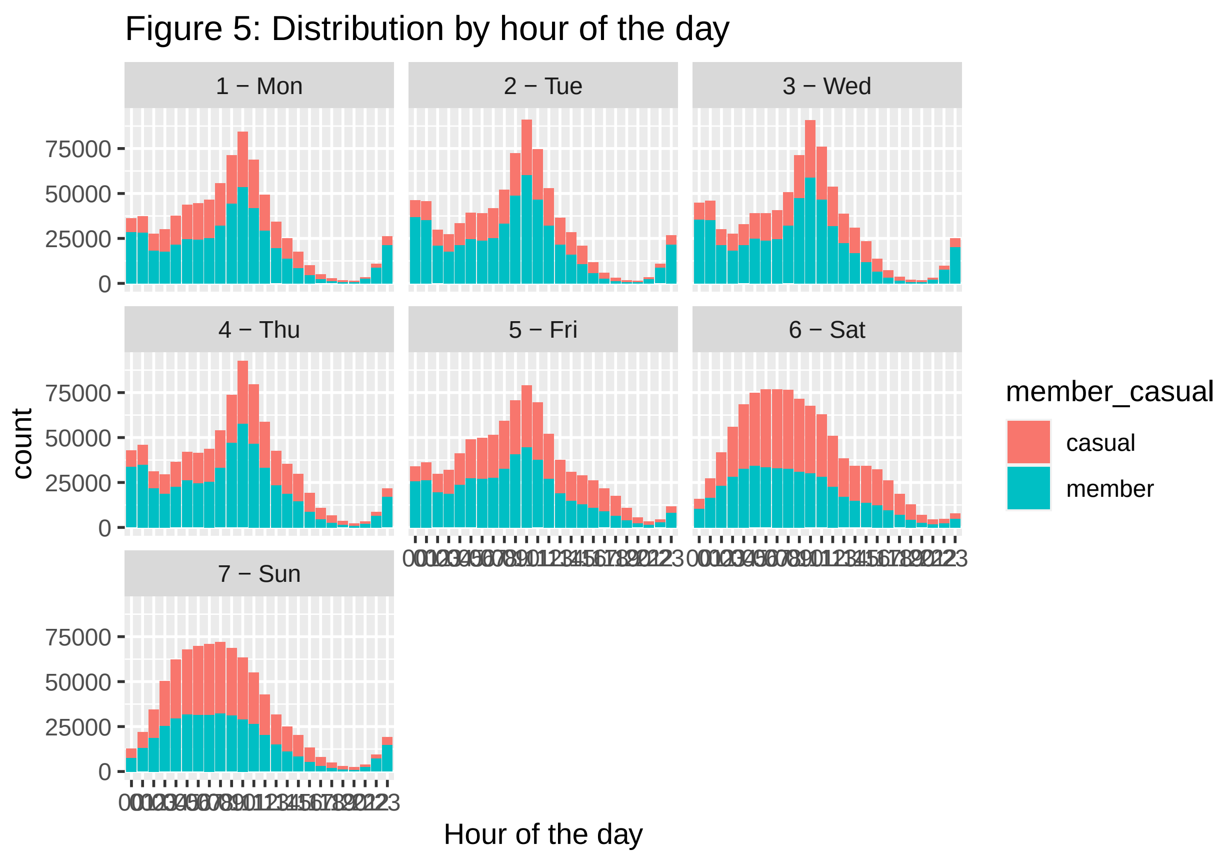
```
df_no_dups %>%
  group_by(start_hour) %>%
  summarise(freq = length(ride_id),
            percent_total = length(ride_id)/ nrow(df_no_dups) *100,
            'member_%' = sum(member_casual == "member")/ length(ride_id) *100,
            'casual_%' = sum(member_casual == "casual")/ length(ride_id) *100,
            'member_casual_diff' = (sum(member_casual == "member") - sum(member_casual == "casual"))/ le
```

```
## # A tibble: 24 x 6
##    start_hour   freq percent_total `member_%` `casual_%` member_casual_diff
##    <chr>       <int>         <dbl>      <dbl>      <dbl>              <dbl>
##  1 00         232611          3.94       76.6       23.4              53.3
##  2 01         259915          4.41       72.7       27.3              45.5
##  3 02         224574          3.81       63.7       36.3              27.5
##  4 03         253004          4.29       57.0       43.0              14.1
##  5 04         312587          5.30       55.1       44.9              10.2
##  6 05         355835          6.03       54.3       45.7               8.57
##  7 06         360254          6.11       52.2       47.8               4.49
##  8 07         371462          6.30       51.8       48.2               3.67
##  9 08         420045          7.12       54.2       45.8               8.46
## 10 09         499525          8.47       58.1       41.9              16.2
## # ... with 14 more rows
```

```
df_no_dups %>%
  ggplot(aes(start_hour, fill = member_casual)) +
  geom_bar() +
  labs(x="Hour of the day", title = "Figure 4: Distribution by hour of the day")
```



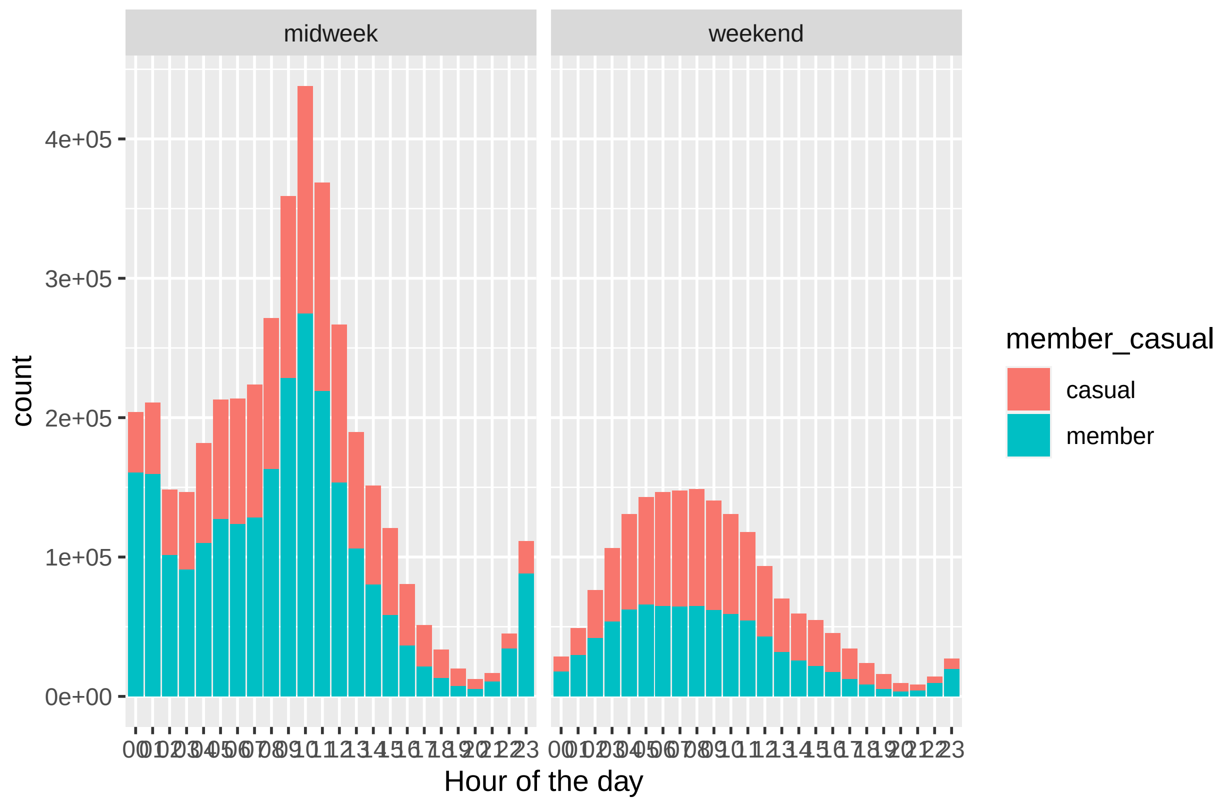

```
df_no_dups %>%
  ggplot(aes(start_hour, fill = member_casual)) +
  geom_bar() +
  labs(x="Hour of the day", title = "Figure 5: Distribution by hour of the day") +
  facet_wrap(~weekday)
```



There's a difference of riders types between weekend and mid_week

```
df_no_dups %>%
    mutate(type_of_weekday = ifelse(weekday == '6 - Sat' | weekday == '7 - Sun',
                                     'weekend',
                                     'midweek')) %>%
    ggplot(aes(start_hour, fill=member_casual)) +
    labs(x="Hour of the day", title="Figure 6 - Distribution by hour of the day in the midweek") +
    geom_bar() +
    facet_wrap(~ type_of_weekday)
```

Figure 6 − Distribution by hour of the day in the midweek



The two plots differs in some key ways:

- While the weekends have a rather smooth curve, the midweek have a more steep change in the number of riders.
- For midweek, there's big increase during the mid-day then it falls towards the night. While that, For the weekend, the number of riders flow smoothly throughout the day, it starts off low then increase gradually towards 6-9am then starting falling.

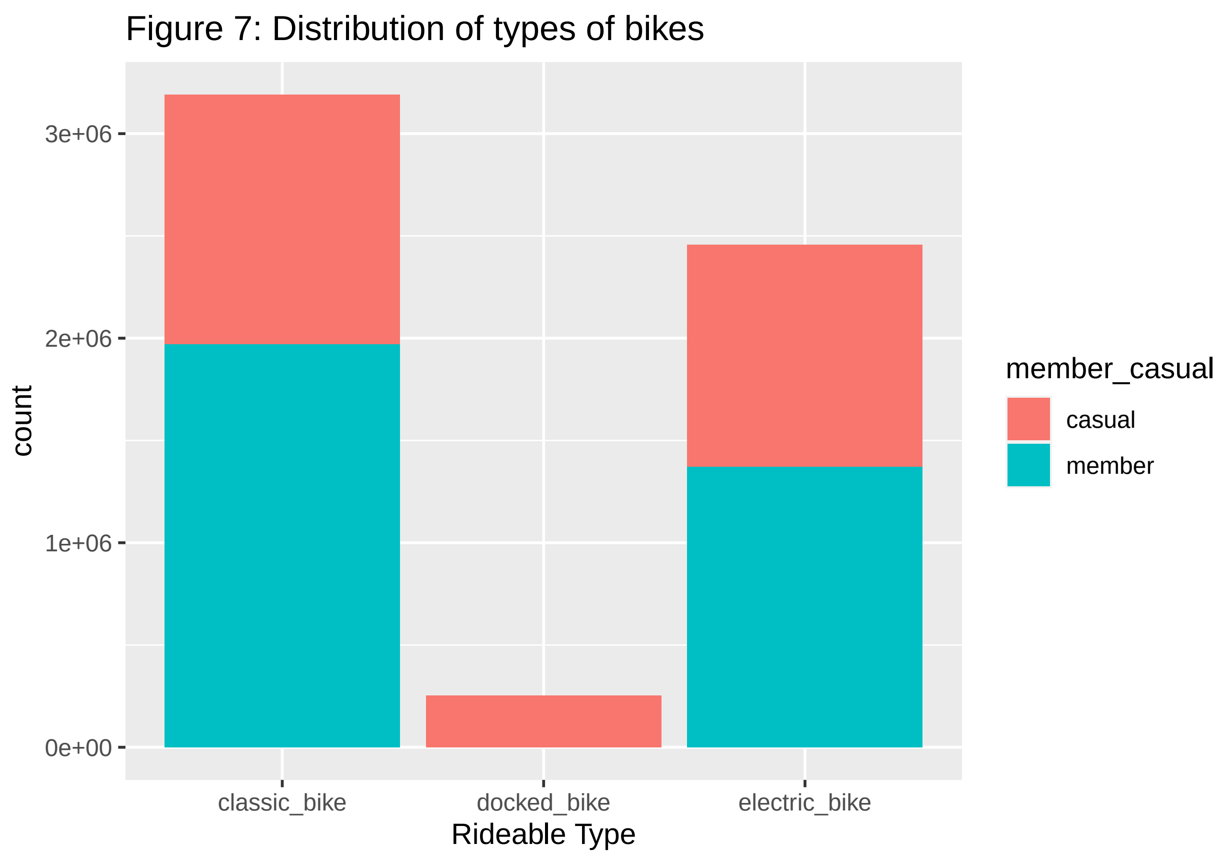
It is important to question which type of bike used by which type of riders use during the day. From there, we can somewhat find out for what reasons they might use the bike for.

Ridedable type

```
df_no_dups %>%
  group_by(rideable_type) %>%
  summarise(freq = length(ride_id),
            percent_total = length(ride_id)/ nrow(df_no_dups) *100,
            'member_%' = sum(member_casual == "member")/ length(ride_id) *100,
            'casual_%' = sum(member_casual == "casual")/ length(ride_id) *100,
            'member_casual_diff' = (sum(member_casual == "member") - sum(member_casual == "casual"))/ l
```

```
## # A tibble: 3 x 6
##   rideable_type    freq percent_total `member_%` `casual_%` member_casual_diff
##   <chr>           <int>         <dbl>      <dbl>      <dbl>              <dbl>
## 1 classic_bike  3189377         54.1        61.8       38.2               23.6
## 2 docked_bike    253371          4.29        0        100               -100
## 3 electric_bike 2457637         41.7        55.8       44.2               11.6
```

```r
ggplot(df_no_dups, aes(rideable_type, fill = member_casual)) +
  labs(x="Rideable Type", title = "Figure 7: Distribution of types of bikes") +
  geom_bar()
```



Other variables

Ride_time_m

```r
summary(df_no_dups$ride_time_minutes)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##  -137.42     6.28    11.17    20.28    20.20 49107.15
```

The max and min may give some problem when plotting some charts, since the min of ride time gives a negative value.

```r
quantiles <- quantile(df_no_dups$ride_time_minutes, seq(0,1, by=0.05))
quantiles
```

```
##           0%           5%          10%          15%          20%          25%
##  -137.416667     2.583333     3.716667     4.600000     5.433333     6.283333
##          30%          35%          40%          45%          50%          55%
##     7.133333     8.033333     8.983333    10.033333    11.166667    12.450000
##          60%          65%          70%          75%          80%          85%
##    13.900000    15.616667    17.666667    20.200000    23.483333    27.916667
##          90%          95%         100%
##    34.833333    50.166667 49107.150000
```

Based on the output, We can see that:

- The difference between 100% and 0% is 49244 minutes.
- The difference 95% and 5% is 47.5 minutes. Because of that, in the analysis of this variable, we are going to subset out the outliers. This subset will contain 95% of the dataset.

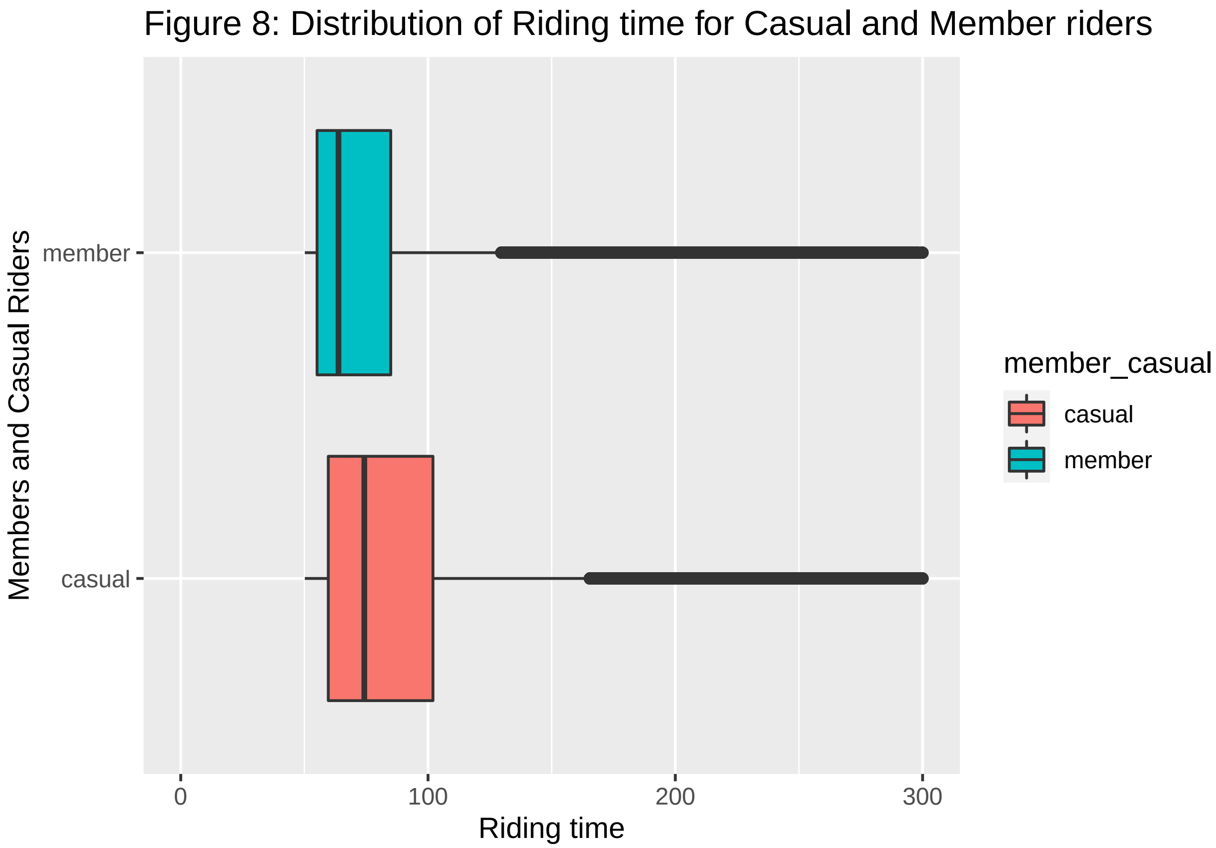
```
df_no_outliers <- df_no_dups %>%
  filter(ride_time_minutes > as.numeric(quantiles["5%"])) %>%
  filter(ride_time_minutes > as.numeric(quantiles["95%"]))

print(paste("Removed", nrow(df_no_dups) - nrow(df_no_outliers), "rows as outliers"))
```

```
## [1] "Removed 5605456 rows as outliers"
```

```
df_no_outliers %>%
  group_by(member_casual) %>%
  summarise(mean = mean(ride_time_minutes),
            "first_quarter" = as.numeric(quantile(ride_time_minutes, 0.25)),
            "median" = median(ride_time_minutes),
            "third_quarter" = as.numeric(quantile(ride_time_minutes, 0.75)),
            "IR" = third_quarter - first_quarter)
```

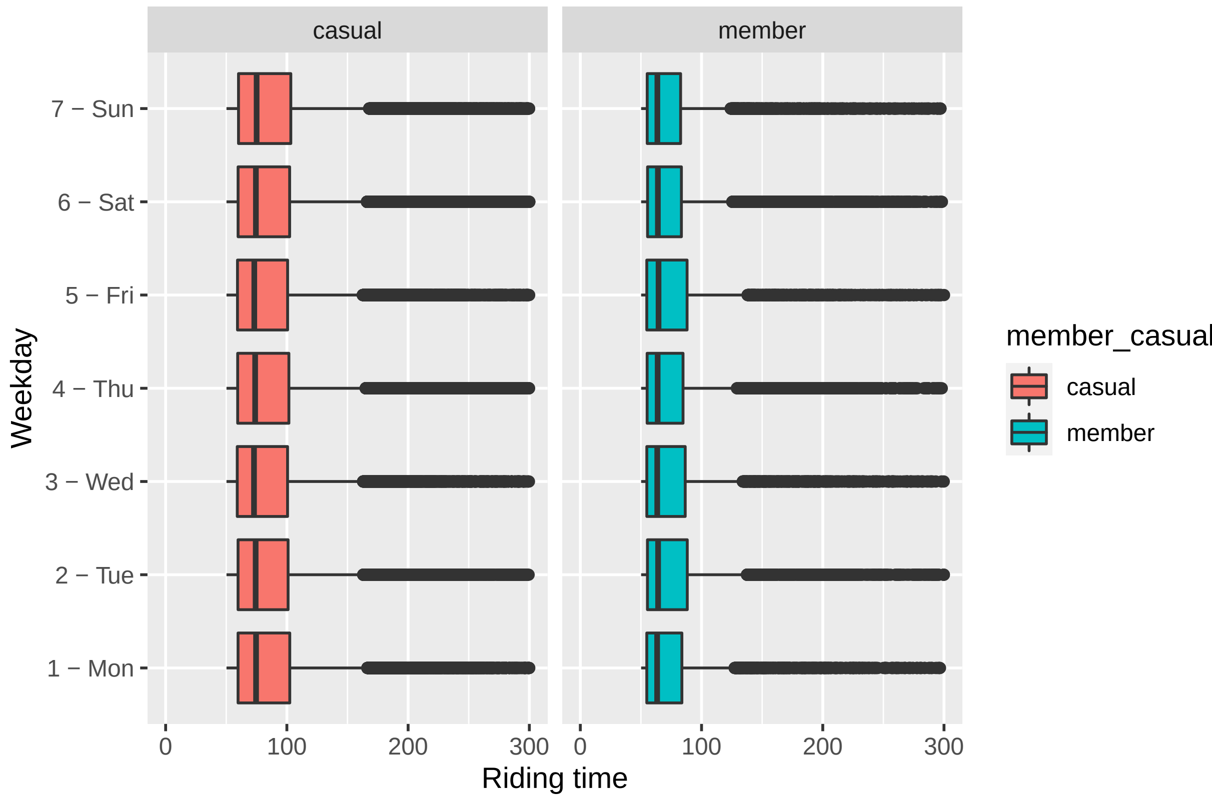
```
## # A tibble: 2 x 6
##   member_casual  mean first_quarter median third_quarter    IR
##   <chr>         <dbl>         <dbl>  <dbl>         <dbl> <dbl>
## 1 casual         152.          60.2   75.9         108.   47.6
## 2 member         124.          55.5   65.4          92.8  37.3
```

```
df_no_outliers %>%
  ggplot(aes(x=member_casual, y= ride_time_minutes, fill = member_casual)) +
  labs(x="Members and Casual Riders", y="Riding time", title = "Figure 8: Distribution of Riding time fo
  geom_boxplot() +
  coord_flip() +
  scale_y_continuous(limits = c(0,300))
```

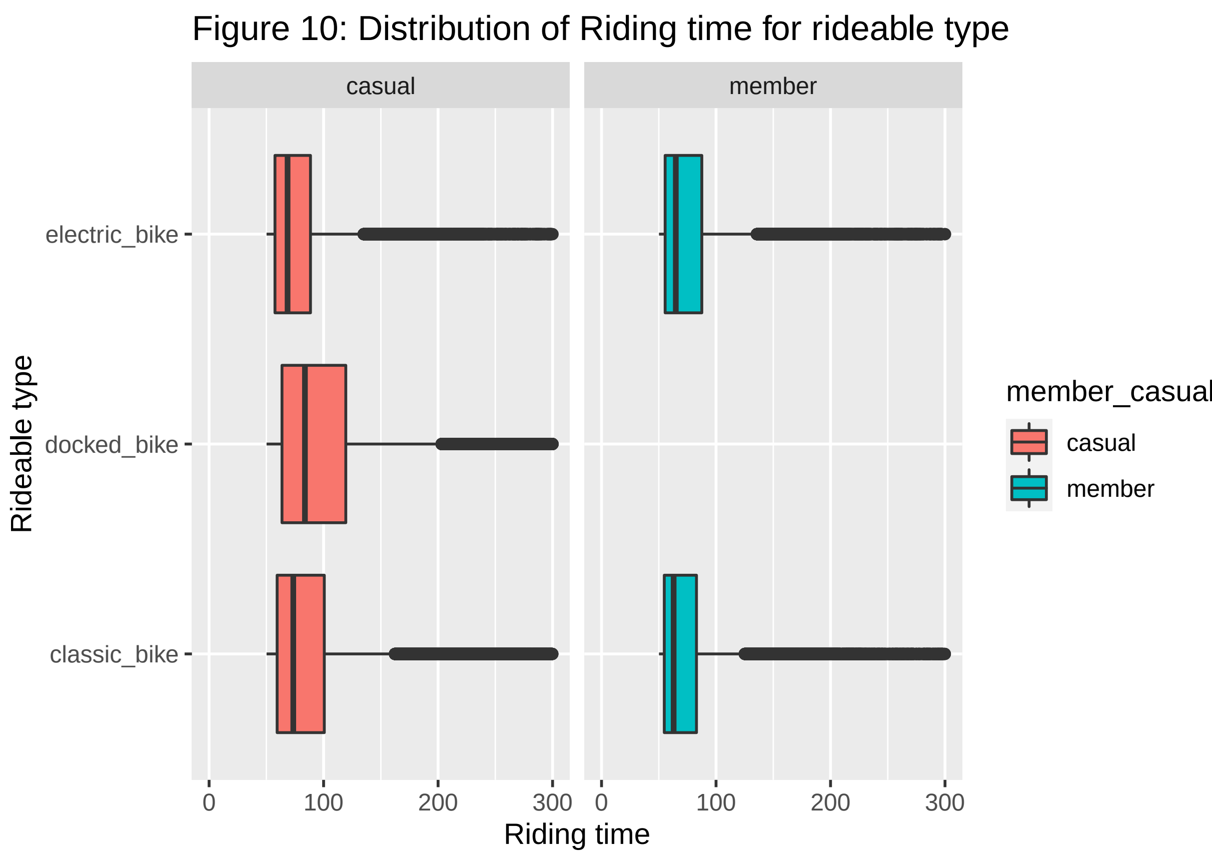



```
ggplot(df_no_outliers, aes(x=weekday, y= ride_time_minutes, fill = member_casual)) +
labs(x="Weekday", y="Riding time", title = "Figure 9: Distribution of Riding time for Casual and Membe
geom_boxplot() +
facet_wrap(~ member_casual)+
coord_flip() +
scale_y_continuous(limits = c(0,300))
```

Figure 9: Distribution of Riding time for Casual and Member riders




```
ggplot(df_no_outliers, aes(x=rideable_type, y=ride_time_minutes, fill=member_casual)) +
    geom_boxplot() +
    facet_wrap(~ member_casual) +
    labs(x="Rideable type", y="Riding time", title="Figure 10: Distribution of Riding time for rideable
    coord_flip() +
    scale_y_continuous(limits = c(0,300))
```



Guding questions

- How should you organize your data to perform analysis on it?
  - The data has been organized into a single CSV called data.csv by concatenating all csv files (between July 2021 and June 2022) from the database given.
- Has your data been properly formatted?
  - Yes, all the columns have been proper formatted into their correct data types.
- What surprises did you discover in the data?
  - One of the main surprises is how members differ from casual riders when analysed from weekday. Furthermore, the members have less riding time than the casual riders.
- What trends or relationships did you find in the data?
  - There are more members than casual riders from the dataset.
  - There's a significant difference between the flow of members and casual from weekends to mid-weeks.
  - Members have less riding time.
  - Members do not use docked bikes.
- How will these insights help answer your business questions?
  - The insights helps to build a profile for members.

Share

The share phase is usually done by building a presentation. But since this is only a showcase case study, this notebook can be seen as a presentation.

Guiding questions

- Were you able to answer the question of how annual members and casual riders use Cyclistic bike differently?
  - Yes. The data shows several differences between casual and member riders
- What story does your data tell?
  - The main story the data tells is that members follows a set schedule as seen on Figure 6. Members also have less riding time because they have a set route to take. Furthermore, members do not use docked bike to travel.
- How do your finding relate to your original question?
  - The findings build a profile for members, relating to "Finding the key differences between casual and annual riders", also knowing why they use the bikes helps to find "How digital media could influence them".
- Who is your audience? What is the best way to communicate with them?
  - The main target audience is the marketing analytics team. The best way to communicate is through a slide presentation of the findings.
- Can data visualization help you share your finding?
  - Yes, the important part of the findings is through data visualization.
- Is your presentation accessible to you audience?
  - Yes, the charts were made using vibrant colors and correct labels.

Act

The act phase would be done by the marketing team of the company. The main takeaway will be the top three recommendations for the marketing.

Guiding questions

- What is your final conclusion based on your analysis?
  - Members and casual riders have different routine activities when using the bikes. The conclusion is further stated in the share phase.
- How could your team and business apply your insights?
  - The insights could be implemented when preparing a marketing campaign for converting casual to members. The marketing team can have a focus on workers as a green way to get to work.
- What next steps would you or your stakeholders take based on your findings?
  - Further analysis is needed to improve findings. However, the marketing team can take the key information from this analysis to build a marketing campaign.
- Is there additional data you could use to expand on your findings?
  - Climate data
  - More information about members