Bike Case

2022-09-30

Ask

Company: Cyclist Size: 5,824 Bicycles Year Launched: 2016

Number of Stations: 692

State of Operation: Chicago

Goal: Design marketing Strategies aimed at converting casual riders into annual members.

Business Task: To determine how do annual members and casual riders use Cyclistic bikes differently

#Prepare

The data collected would come from September 2021 to August 2022.

1. Best place to store the data would be in the Google Cloud Platform
2. This allows me to organize all the tables and compile them together with the use of Big Query.
3. This allows for easy access of data since everything is stored in the cloud.
4. This allows for protection of data ensuring that the data follows encryption practices.

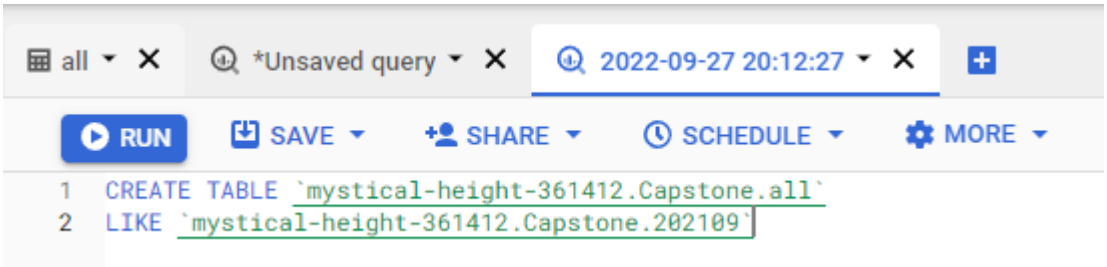
The data contains the following columns, this information would determine the ride habits of both members and non members.

1. Ride ID
2. Rideable Type
3. Started At
4. Ended At
5. Start Station Name
6. Start Station ID
7. End Station Name
8. End Station ID
9. Start Latitude
10. Start Longitude
11. End Latitude
12. End Longitude
13. Member Casual

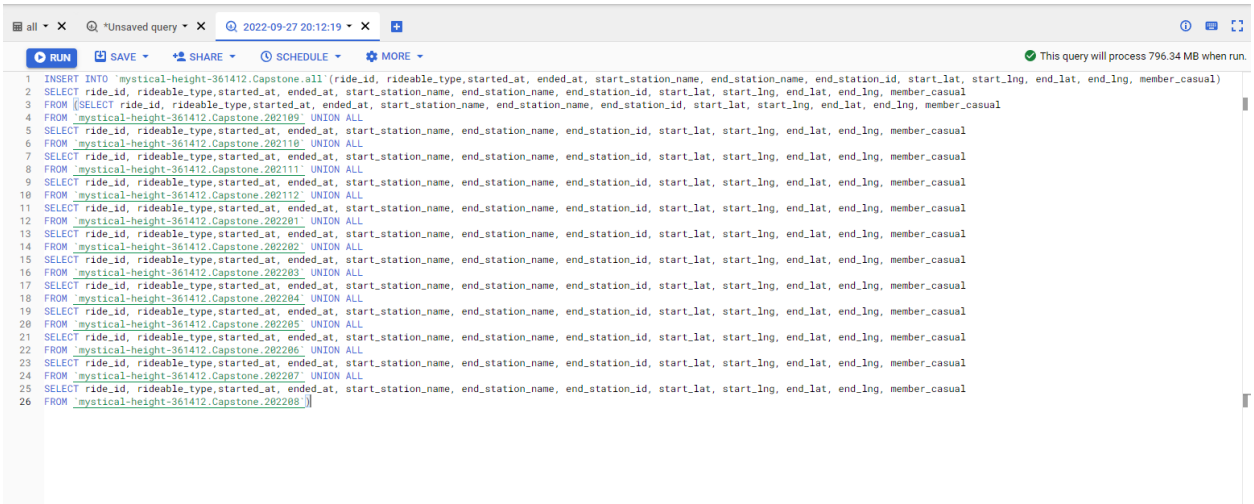
The data was analyzed and no discrepancies were encountered. The field names of the data and their corresponding file type were correct across all files.

The data from each month was compiled into a singular table in order to better analyze the data.

First a new table was created to store the data



Then a combination of UNION ALL statements were used to compile the data



Process

Step 1: Install Packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(tinytex)
```

Step 2: Read CSV and Convert Values to String

```
processed_tripdata <- read_csv(file = paste("bikedata.csv", sep=""))
```

```
## Rows: 5920599 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (8): ride_id, rideable_type, started_at, ended_at, start_station_name, e...
## dbl (4): start_lat, start_lng, end_lat, end_lng
## lgl (1): start_station_id
##
## 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.
```

```
str(processed_tripdata)
```

```
## spec_tbl_df [5,920,599 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5920599] "A535F2E03A1D8A04" "8F1611A8E1DA38D4" "B9B7D65816A280A3" "8D5
##  $ rideable_type     : chr [1:5920599] "classic_bike" "classic_bike" "docked_bike" "classic_bike" ..
##  $ started_at        : chr [1:5920599] "2021-09-21 19:57:57 UTC" "2022-03-26 14:51:34 UTC" "2022-03-
##  $ ended_at          : chr [1:5920599] "2021-09-22 13:09:20 UTC" "2022-03-27 15:51:29 UTC" "2022-03-
##  $ start_station_name: chr [1:5920599] "Sheffield Ave & Wellington Ave" "Blackstone Ave & Hyde Park
##  $ start_station_id  : logi [1:5920599] NA NA NA NA NA NA ...
##  $ end_station_name  : chr [1:5920599] NA NA NA NA ...
##  $ end_station_id    : chr [1:5920599] NA NA NA NA ...
##  $ start_lat         : num [1:5920599] 41.9 41.8 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:5920599] -87.7 -87.6 -87.6 -87.6 -87.6 ...
```

```
##  $ end_lat           : num [1:5920599] NA NA NA NA NA NA NA NA NA NA ...
##  $ end_lng           : num [1:5920599] NA NA NA NA NA NA NA NA NA NA ...
##  $ member_casual     : chr [1:5920599] "casual" "casual" "casual" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_character(),
##   ..   ended_at = col_character(),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_logical(),
##   ..   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>
```

Step 3: Format Columns

```
processed_tripdata$started_at <- ymd_hms(processed_tripdata$started_at)
processed_tripdata$ended_at <- ymd_hms(processed_tripdata$ended_at)
processed_tripdata$duration <- as.numeric(difftime(processed_tripdata$ended_at, processed_tripdata$star
processed_tripdata$month <- format(processed_tripdata$started_at, format="%B")
#Formats into month name
processed_tripdata$day_of_week <- format(processed_tripdata$started_at, format="%A")
#Formats into weekday name
processed_tripdata$hour <- format(processed_tripdata$started_at, format="%H")
#Formats into hour in 24 hour format
processed_tripdata <- processed_tripdata %>% filter(duration > 0)
#Remove trips that have a duration of 0 and less
processed_tripdata$member_casual<-replace(processed_tripdata$member_casual,processed_tripdata$member_ca
processed_tripdata$member_casual<-replace(processed_tripdata$member_casual,processed_tripdata$member_ca
#Change the casing of the member_casual variable
options(scipen=999)
#To use decimals instead of scientific notation
```

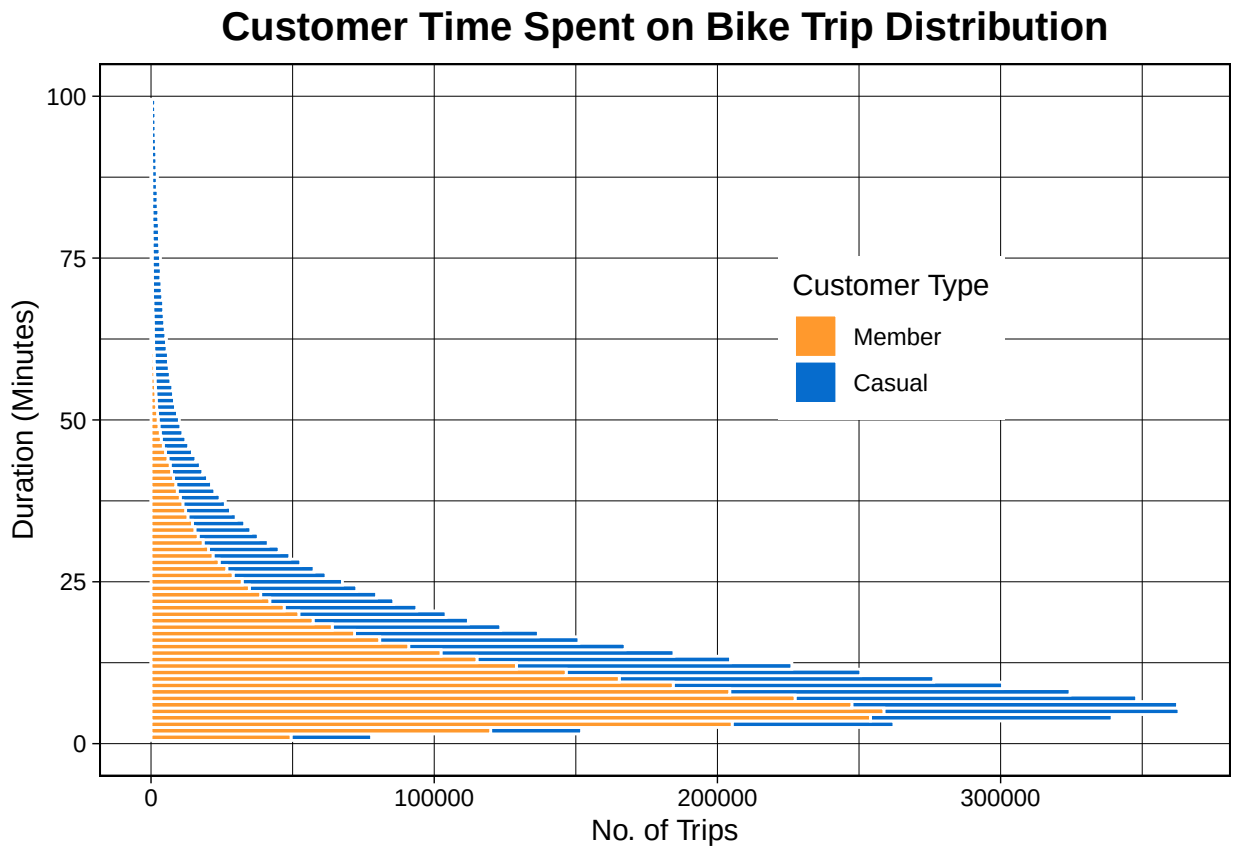
Analyze

```
#This plot shows the distribution of customers both members and casual riders, across the duration of e
ggplot(processed_tripdata, aes(x=duration, fill=member_casual)) +
    geom_histogram(binwidth = 1, color="white") +
    xlim(0, 100) +
    xlab("Duration (Minutes)") +
    ylab("No. of Trips") +
    theme_linedraw() +
```

```
    ggtitle("Customer Time Spent on Bike Trip Distribution") +
    theme(plot.title = element_text(size = 15, face = "bold", hjust = 0.5),
            legend.justification = c(1,0),
            legend.position = c(0.8,0.5)
         ) +
    guides(fill=guide_legend(title="Customer Type")) +
    scale_fill_manual(values=c('#FF992D','#066CCD'), limits = c("Member", "Casual")) +
    coord_flip()
```

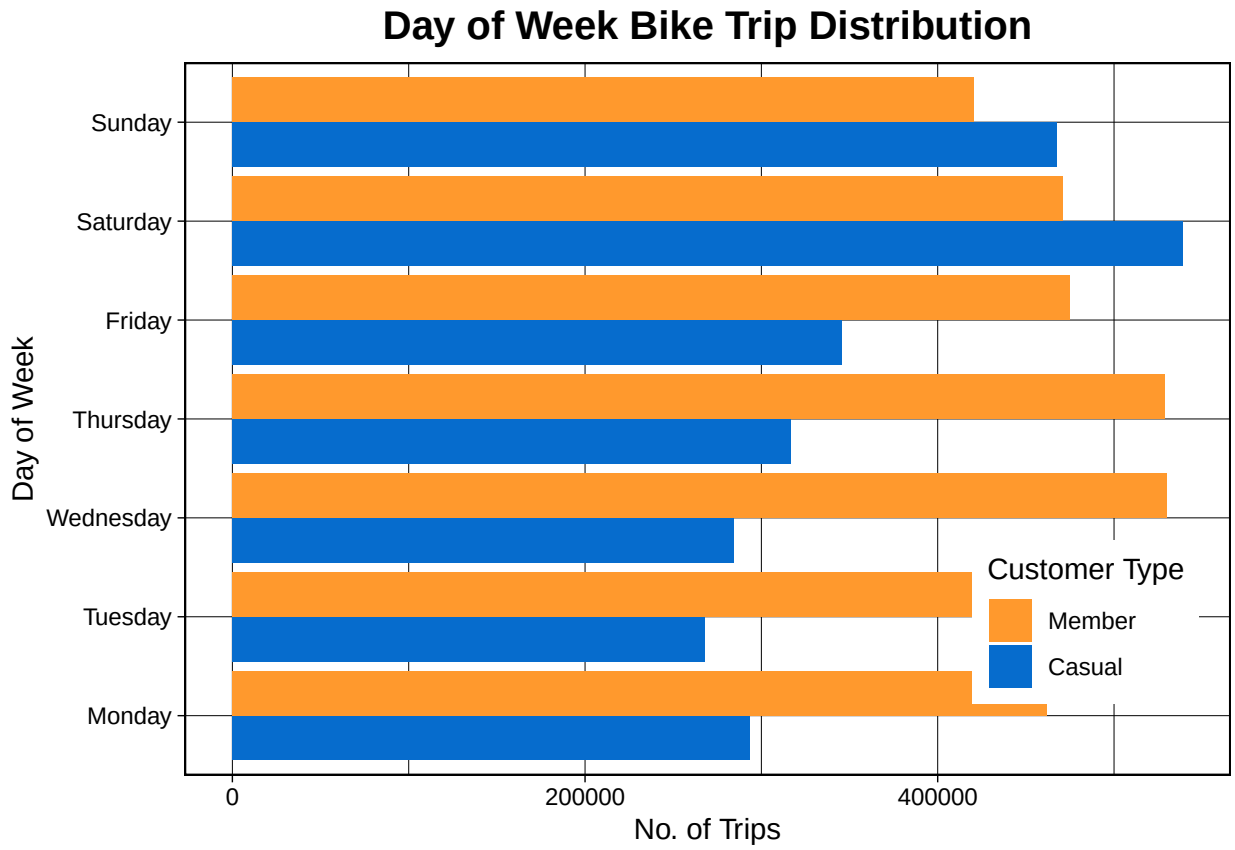
```
## Warning: Removed 75269 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 4 rows containing missing values (geom_bar).
```



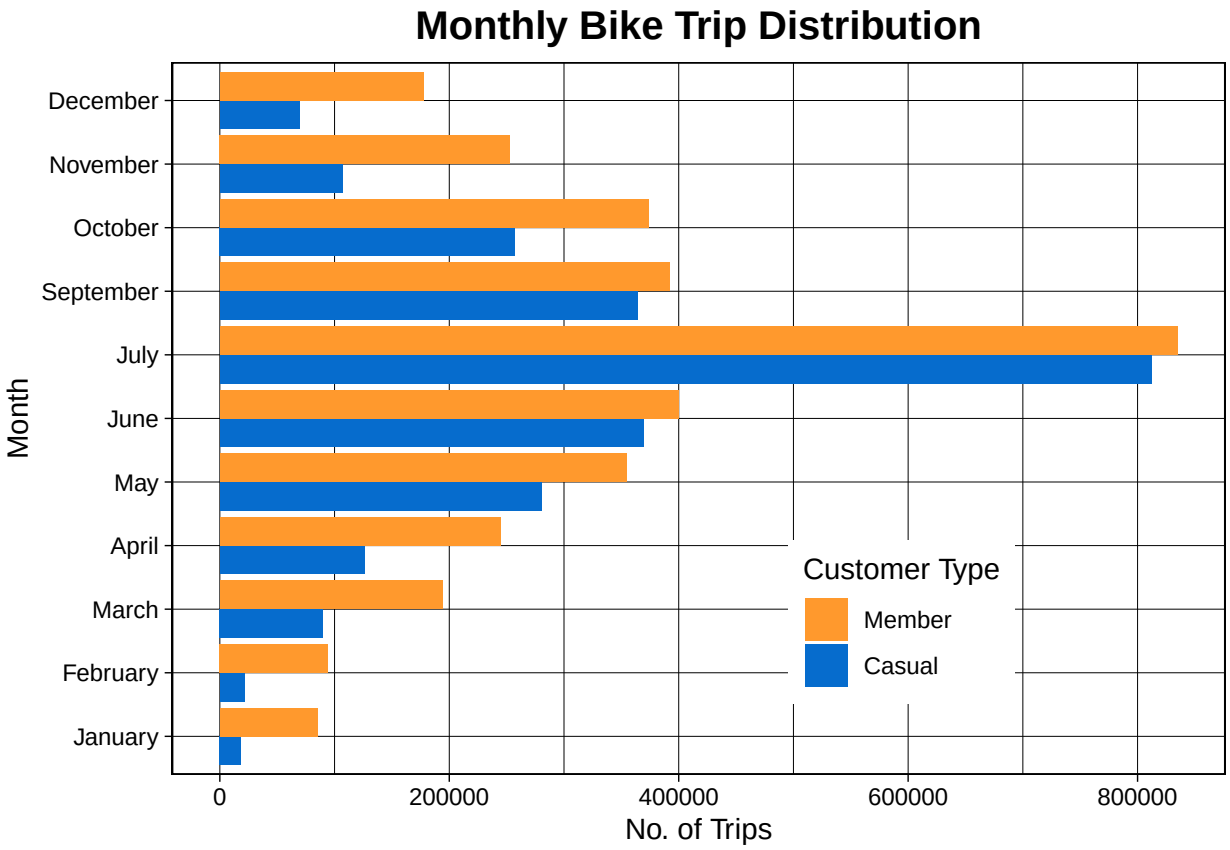

```
#This plot shows the distribution of customers both members and casual riders, across the numbers of tr
processed_tripdata %>%
    select(day_of_week, member_casual) %>%
    group_by(day_of_week, member_casual) %>%
    count() %>%
    ggplot(aes(x=factor(day_of_week, level=c('Monday', 'Tuesday', 'Wednesday', 'Thursday',
                                             'Friday', 'Saturday', 'Sunday')),
               y=n, fill=member_casual)) +
        geom_bar(stat="identity", position=position_dodge())+
        coord_flip()+
        scale_fill_manual(values=c('#FF992D','#066CCD'), limits = c("Member", "Casual")) +
```

```
        xlab("Day of Week") +
        ylab("No. of Trips") +
        theme_linedraw() +
        ggtitle("Day of Week Bike Trip Distribution") +
        guides(fill=guide_legend(title="Customer Type")) +
        theme(plot.title = element_text(size = 15, face = "bold", hjust = 0.5),
                legend.justification = c(1,0),
                legend.position = c(0.97,0.1)
            )
```



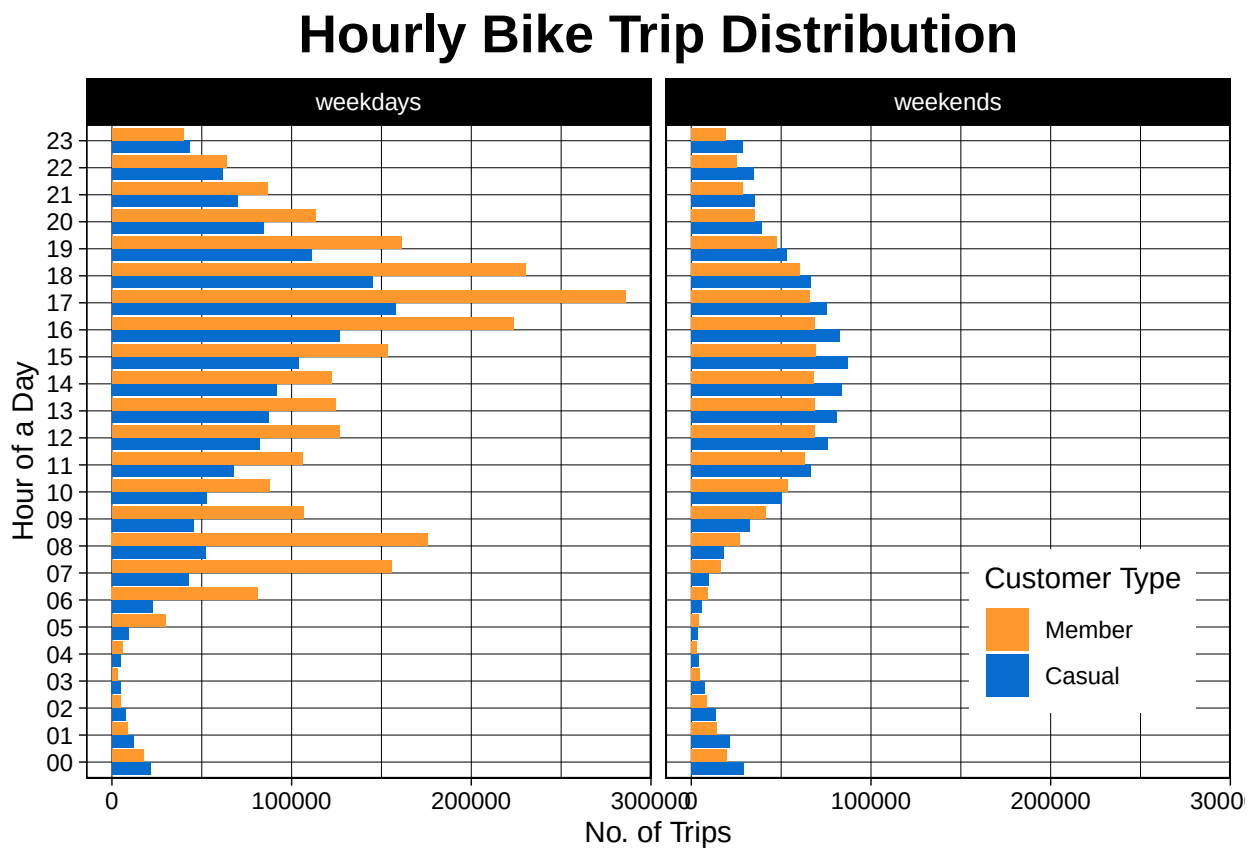

```
#This plot shows the distribution of customers both members and casual riders, across the numbers of tr
processed_tripdata %>%
    select(month, member_casual) %>%
    group_by(month, member_casual) %>%
    count() %>%
    ggplot(aes(x=factor(month, level=c('January', 'February', 'March', 'April', 'May',
                                       'June', 'July', 'August', 'September', 'October', 'November',
                        y=n, fill=member_casual)) +
        geom_bar(stat="identity", position=position_dodge())+
        coord_flip()+
        scale_fill_manual(values=c('#FF992D','#066CCD'), limits = c("Member", "Casual")) +
        xlab("Month") +
        ylab("No. of Trips") +
        theme_linedraw() +
        ggtitle("Monthly Bike Trip Distribution") +
```

```
        guides(fill=guide_legend(title="Customer Type")) +
        theme(plot.title = element_text(size = 15, face = "bold", hjust = 0.5),
                legend.justification = c(1,0),
                legend.position = c(0.8,0.1)
            )
```




```
#These plots show the number of hours in a day members and casual customers ride bikes on both weekdays
weekend <- processed_tripdata %>%
    select(day_of_week, hour, member_casual) %>%
    filter(day_of_week == 'Saturday' | day_of_week == 'Sunday') %>%
    group_by(hour, member_casual) %>%
    count()

weekend$weekend_weekday = 'weekends'

weekday <- processed_tripdata %>%
    select(day_of_week, hour, member_casual) %>%
    filter(day_of_week != 'Saturday' & day_of_week != 'Sunday') %>%
    group_by(hour, member_casual) %>%
    count()

weekday$weekend_weekday = 'weekdays'

weekend <- rbind(weekend, weekday)
```

```
weekend %>%
    ggplot(aes(x=hour, y=n, fill=member_casual)) +
        geom_bar(stat="identity", position=position_dodge())+
        coord_flip()+
        scale_fill_manual(values=c('#FF992D','#066CCD'), limits = c("Member", "Casual")) +
        xlab("Hour of a Day") +
        ylab("No. of Trips") +
        theme_linedraw() +
        ggtitle("Hourly Bike Trip Distribution") +
        guides(fill=guide_legend(title="Customer Type")) +
        theme(plot.title = element_text(size = 20, face = "bold", hjust = 0.5),
                legend.justification = c(1,0),
                legend.position = c(0.97,0.1)
             ) +
        facet_wrap(~weekend_weekday)
```

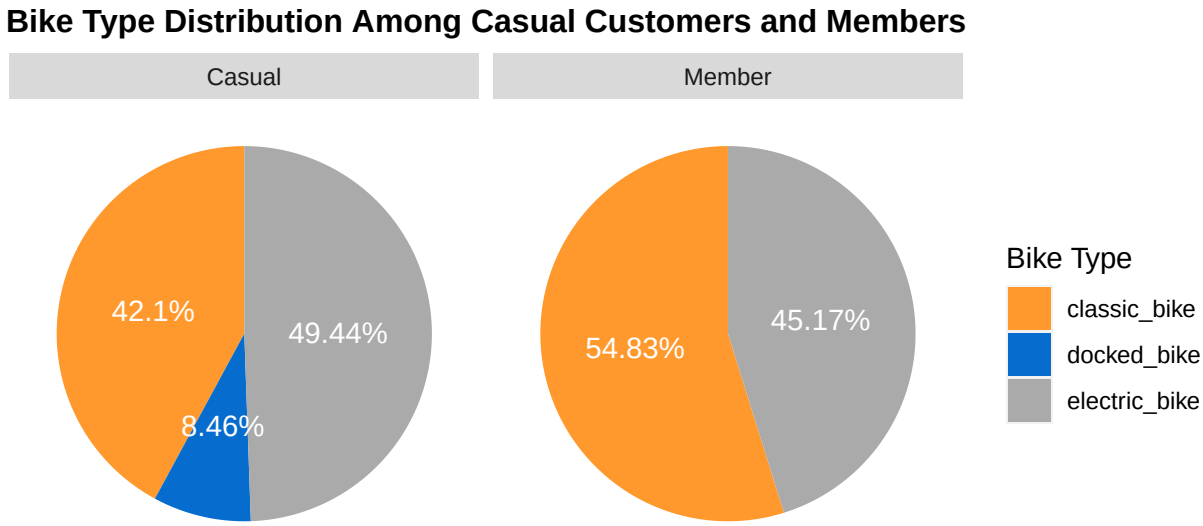



```
#These plots shows the distribution of customers both members and casual riders, across the different b
casual <- processed_tripdata %>%
    select(member_casual, rideable_type) %>%
    filter(member_casual == "Casual") %>%
    group_by(rideable_type, member_casual) %>%
    count()

member <- processed_tripdata %>%
    select(member_casual, rideable_type) %>%
```

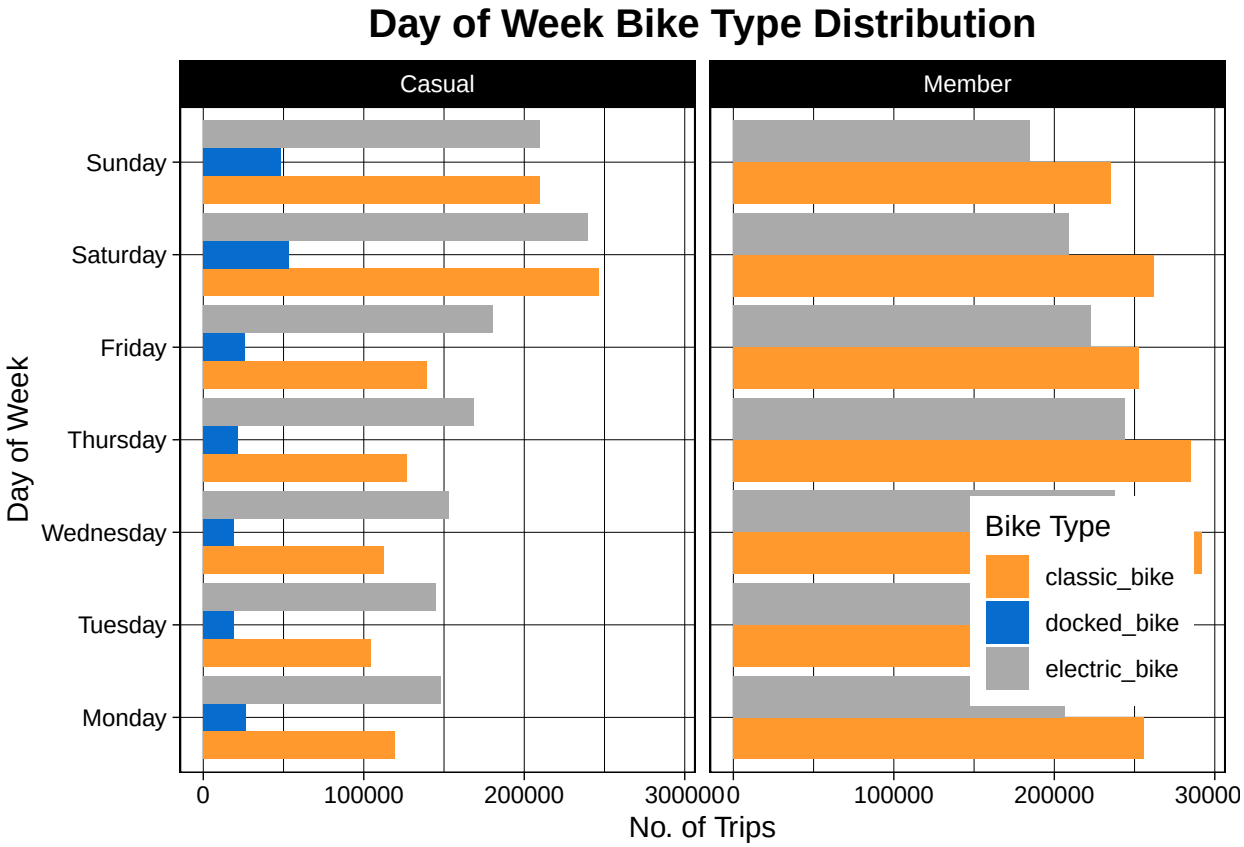
```
    filter(member_casual == "Member") %>%
    group_by(rideable_type, member_casual) %>%
    count()

customer <- rbind((casual%>%mutate(countT= sum(casual$n)) %>%
                group_by(rideable_type, add=TRUE) %>%
                mutate(per=n/countT, per_label=paste0(round(100*n/countT,2),"%"))),
               (member%>%mutate(countT= sum(member$n)) %>%
                group_by(rideable_type, add=TRUE) %>%
                mutate(per=n/countT, per_label=paste0(round(100*n/countT,2),"%"))))
```

```
## Warning: The ‘add‘ argument of ‘group_by()‘ is deprecated as of dplyr 1.0.0.
## Please use the ‘.add‘ argument instead.
```

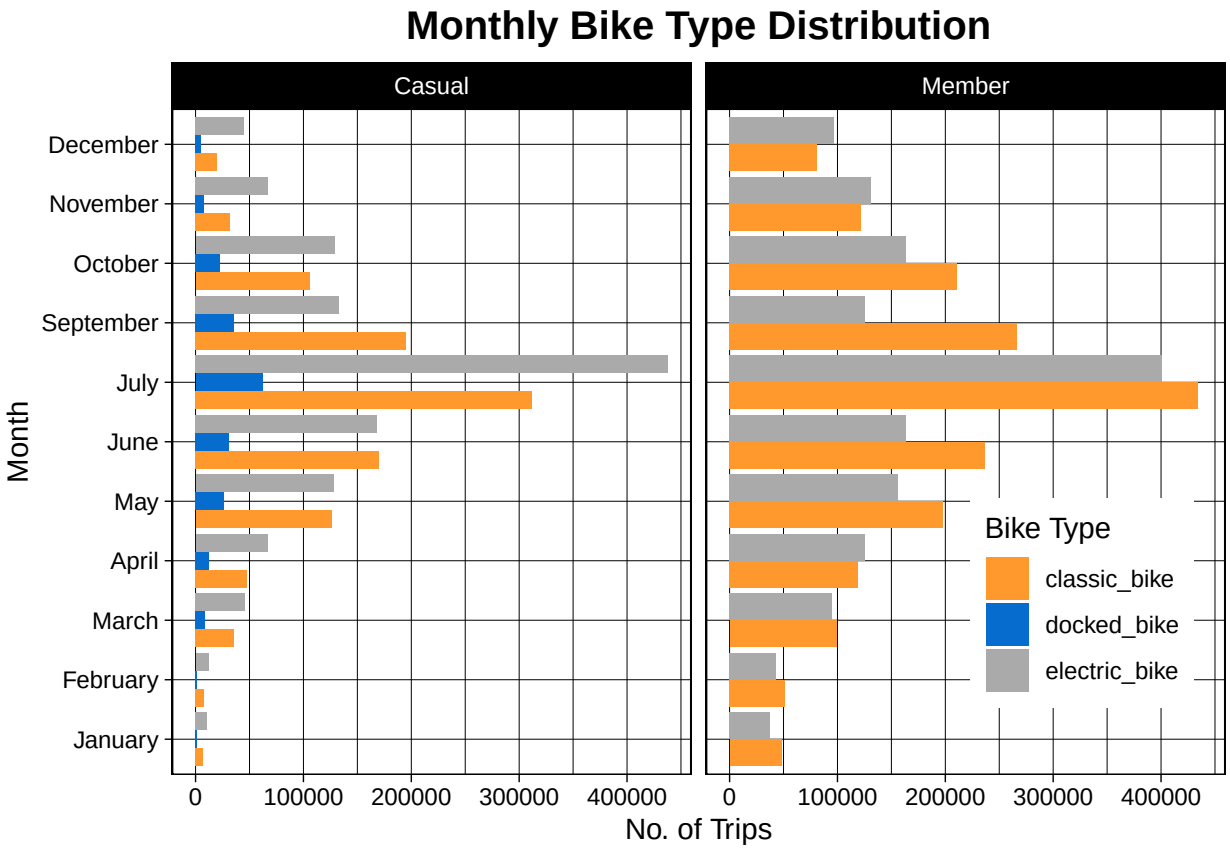
```
ggplot(customer, aes(x="", y=per, fill=rideable_type)) +
        geom_col() +
        coord_polar(theta = "y") +
        scale_y_continuous(labels = scales::percent) +
        geom_text(aes(label = per_label), position = position_stack(vjust = 0.5), color="white") +
        scale_fill_manual(values=c('#FF992D','#066CCD','#AAAAAA'), limits = c("classic_bike", "docked_b
        theme(axis.ticks = element_blank(),
            axis.title = element_blank(),
            axis.text = element_text(size = 0),
            panel.background = element_rect(fill = "white"),
            plot.title = element_text(size = 12, face = "bold", hjust = 0.5)) +
        ggtitle("Bike Type Distribution Among Casual Customers and Members") +
        guides(fill=guide_legend(title="Bike Type")) +
        facet_wrap(~member_casual)
```




```
#These plots shows the distribution of the different bike types based on the day of the week
processed_tripdata %>%
    select(day_of_week, rideable_type, member_casual) %>%
    group_by(day_of_week, rideable_type, member_casual) %>%
    count() %>%
    ggplot(aes(x=factor(day_of_week, level=c('Monday', 'Tuesday', 'Wednesday', 'Thursday',
                                             'Friday', 'Saturday', 'Sunday')),
               y=n, fill=rideable_type)) +
        geom_bar(stat="identity", position=position_dodge())+
        coord_flip()+
        scale_fill_manual(values=c('#FF992D','#066CCD','#AAAAAA'), limits = c("classic_bike","docked_bi
        xlab("Day of Week") +
        ylab("No. of Trips") +
        theme_linedraw() +
        ggtitle("Day of Week Bike Type Distribution") +
        guides(fill=guide_legend(title="Bike Type")) +
        theme(plot.title = element_text(size = 15, face = "bold", hjust = 0.5),
              legend.justification = c(1,0),
              legend.position = c(0.97,0.1)
             ) +
        facet_wrap(~member_casual)
```




```
#These plots shows the distribution of the different bike types based on the month
processed_tripdata %>%
    select(month, rideable_type, member_casual) %>%
    group_by(month, rideable_type, member_casual) %>%
    count() %>%
    ggplot(aes(x=factor(month, level=c('January', 'February', 'March', 'April', 'May',
                                       'June', 'July', 'August', 'September', 'October', 'November',
               y=n, fill=rideable_type)) +
        geom_bar(stat="identity", position=position_dodge())+
        coord_flip()+
        scale_fill_manual(values=c('#FF992D','#066CCD','#AAAAAA'), limits = c("classic_bike", "docked_b
        xlab("Month") +
        ylab("No. of Trips") +
        theme_linedraw() +
        ggtitle("Monthly Bike Type Distribution") +
        guides(fill=guide_legend(title="Bike Type")) +
        theme(plot.title = element_text(size = 15, face = "bold", hjust = 0.5),
              legend.justification = c(1,0),
              legend.position = c(0.97,0.1)
             ) +
        facet_wrap(~member_casual)
```


Monthly Bike Type Distribution
Casual
Member
December
November
October
September
July
June
May
April
March
February
January
Month
0
100000
200000
300000
400000
No. of Trips
Bike Type
classic_bike
docked_bike
electric_bike