Cyclistic Data Analysis (Capstone Case Study)

Vinayak Kumar Pathak

December 2022

Setup

We'll first install and load all the necessary packages.

```
install.packages("tidyverse", repos = "http://cran.us.r-project.org")

library("dplyr")
library("lubridate")
library("ggplot2")
```

Import Data

We'll then import raw data into data frames

```
dec_21 <- read.csv("F:\\Capstone_CS1\\Datasets\\prev_12_m_csv\\tripdata_2021_12.csv")
str(dec_21)
```

```
## 'data.frame':    247540 obs. of  13 variables:
##  $ ride_id           : chr  "46F8167220E4431F" "73A77762838B32FD" "4CF42452054F59C5" "3278BA87BF698339" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "electric_bike" "classic_bike" ...
##  $ started_at        : chr  "2021-12-07 15:06:07" "2021-12-11 03:43:29" "2021-12-15 23:10:28" "2021-12-26 16:16:10" ...
##  $ ended_at          : chr  "2021-12-07 15:13:42" "2021-12-11 04:10:23" "2021-12-15 23:23:14" "2021-12-26 16:30:53" ...
##  $ start_station_name: chr  "Laflin St & Cullerton St" "LaSalle Dr & Huron St" "Halsted St & North Branch St" "Halsted St &
North Branch St" ...
##  $ start_station_id  : chr  "13307" "KP1705001026" "KA1504000117" "KA1504000117" ...
##  $ end_station_name  : chr  "Morgan St & Polk St" "Clarendon Ave & Leland Ave" "Broadway & Barry Ave" "LaSalle Dr & Huron S
t" ...
##  $ end_station_id    : chr  "TA1307000130" "TA1307000119" "13137" "KP1705001026" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.6 -87.6 -87.6 -87.7 ...
##  $ end_lat           : num  41.9 42 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "casual" "member" "member" ...
```

```
# jan_22 <- read.csv("F:\\Capstone_CS1\\Datasets\\prev_12_m_csv\\tripdata_2022_01.csv")
# str(jan_22)
# feb_22 <- read.csv("F:\\Capstone_CS1\\Datasets\\prev_12_m_csv\\tripdata_2022_02.csv")
# str(feb_22)
# ...
# nov_22 <- read.csv("F:\\Capstone_CS1\\Datasets\\prev_12_m_csv\\tripdata_2022_11.csv")
# str(nov_22)
```

Clean Data

Merge all data frames into one

We’ll merge all data frames into one to simplify the process of our analysis.

```
all_trips <- dec_21

# When we'll actually have data of all four months, we'll have to use bind_rows() function to merge all csv files into one.
# all_trips <- bind_rows(dec_21, jan_22, feb_22, mar_22, apr_22, may_22, jun_22, jul_22, aug_22, sep_22, oct_22, nov_22)

str(all_trips)
```

```
## 'data.frame':    247540 obs. of  13 variables:
##  $ ride_id           : chr  "46F8167220E4431F" "73A77762838B32FD" "4CF42452054F59C5" "3278BA87BF698339" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "electric_bike" "classic_bike" ...
##  $ started_at        : chr  "2021-12-07 15:06:07" "2021-12-11 03:43:29" "2021-12-15 23:10:28" "2021-12-26 16:16:10" ...
##  $ ended_at          : chr  "2021-12-07 15:13:42" "2021-12-11 04:10:23" "2021-12-15 23:23:14" "2021-12-26 16:30:53" ...
##  $ start_station_name: chr  "Laflin St & Cullerton St" "LaSalle Dr & Huron St" "Halsted St & North Branch St" "Halsted St &
North Branch St" ...
##  $ start_station_id  : chr  "13307" "KP1705001026" "KA1504000117" "KA1504000117" ...
##  $ end_station_name  : chr  "Morgan St & Polk St" "Clarendon Ave & Leland Ave" "Broadway & Barry Ave" "LaSalle Dr & Huron S
t" ...
##  $ end_station_id    : chr  "TA1307000130" "TA1307000119" "13137" "KP1705001026" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.6 -87.6 -87.6 -87.7 ...
##  $ end_lat           : num  41.9 42 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "casual" "member" "member" ...
```

```
# Deleting the individual data frames as they no longer are necessary.
rm(dec_21)
# rm(jan_22)
# rm(feb_22)
# ...
# rm(nov_22)
```

_

Remove obsolete Columns

We’ll then remove the columns which are not useful for this analysis.

```
all_trips <- select(all_trips, -c(start_station_name, start_station_id, end_station_name, end_station_id, start_lat, start_lng,
end_lat, end_lng))

str(all_trips)
```

```
## 'data.frame':    247540 obs. of  5 variables:
##  $ ride_id      : chr  "46F8167220E4431F" "73A77762838B32FD" "4CF42452054F59C5" "3278BA87BF698339" ...
##  $ rideable_type: chr  "electric_bike" "electric_bike" "electric_bike" "classic_bike" ...
##  $ started_at   : chr  "2021-12-07 15:06:07" "2021-12-11 03:43:29" "2021-12-15 23:10:28" "2021-12-26 16:16:10" ...
##  $ ended_at     : chr  "2021-12-07 15:13:42" "2021-12-11 04:10:23" "2021-12-15 23:23:14" "2021-12-26 16:30:53" ...
##  $ member_casual: chr  "member" "casual" "member" "member" ...
```

_

Correct Data Type

We’ll have to correct the data type of certain columns in order to obtain correct results.

```
all_trips <- mutate(all_trips, started_at = ymd_hms(started_at), ended_at = ymd_hms(ended_at))

str(all_trips)
```

```
## 'data.frame':    247540 obs. of  5 variables:
##  $ ride_id      : chr  "46F8167220E4431F" "73A77762838B32FD" "4CF42452054F59C5" "3278BA87BF698339" ...
##  $ rideable_type: chr  "electric_bike" "electric_bike" "electric_bike" "classic_bike" ...
##  $ started_at   : POSIXct, format: "2021-12-07 15:06:07" "2021-12-11 03:43:29" ...
##  $ ended_at     : POSIXct, format: "2021-12-07 15:13:42" "2021-12-11 04:10:23" ...
##  $ member_casual: chr  "member" "casual" "member" "member" ...
```

_

Add required Columns

We’ll then add some new columns which will be required for this analysis.

```
all_trips$ride_duration <- as.numeric(difftime(all_trips$ended_at, all_trips$started_at, units = "mins"))     # Computing ride duration in minutes

all_trips$day_of_week <- as.character(wday(all_trips$started_at, label = TRUE, abbr = FALSE))

all_trips$month_name <- as.character(month(all_trips$started_at, label = TRUE, abbr = FALSE))

all_trips$start_hour <- as.numeric(hour(all_trips$started_at))

all_trips$season <- as.character(with(all_trips, ifelse(month_name == "December" || month_name == "January" || month_name == "February", "Winter", ifelse(month_name == "March" || month_name == "April" || month_name == "May", "Spring", ifelse(month_name == "June" || month_name == "July" || month_name == "August", "Summer", ifelse(month_name == "September" || month_name == "October" || month_name == "November", "Fall", NA))))))

str(all_trips)
```

```
## 'data.frame':    247540 obs. of  10 variables:
##  $ ride_id      : chr  "46F8167220E4431F" "73A77762838B32FD" "4CF42452054F59C5" "3278BA87BF698339" ...
##  $ rideable_type: chr  "electric_bike" "electric_bike" "electric_bike" "classic_bike" ...
##  $ started_at   : POSIXct, format: "2021-12-07 15:06:07" "2021-12-11 03:43:29" ...
##  $ ended_at     : POSIXct, format: "2021-12-07 15:13:42" "2021-12-11 04:10:23" ...
##  $ member_casual: chr  "member" "casual" "member" "member" ...
##  $ ride_duration: num  7.58 26.9 12.77 14.72 20.27 ...
##  $ day_of_week  : chr  "Tuesday" "Saturday" "Wednesday" "Sunday" ...
##  $ month_name   : chr  "December" "December" "December" "December" ...
##  $ start_hour   : num  15 3 23 16 11 18 15 13 14 16 ...
##  $ season       : chr  "Winter" "Winter" "Winter" "Winter" ...
```

Transforming values of certain columns

```r
# Transforming values of 'member_casual' column
all_trips$member_casual[all_trips$member_casual == "member"] <- "Member"
all_trips$member_casual[all_trips$member_casual == "casual"] <- "Casual Rider"

# Transforming values of 'rideable_type' column
all_trips$rideable_type[all_trips$rideable_type == "classic_bike"] <- "Classic Bike"
all_trips$rideable_type[all_trips$rideable_type == "docked_bike"] <- "Docked Bike"
all_trips$rideable_type[all_trips$rideable_type == "electric_bike"] <- "Electric Bike"
```

–

Check Data

We’ll the check data for any errors and outliers.

```r
summary(all_trips)
```

```
##    ride_id          rideable_type        started_at
##  Length:247540      Length:247540      Min.   :2021-12-01 00:00:01.00
##  Class :character   Class :character   1st Qu.:2021-12-06 12:51:05.25
##  Mode  :character   Mode  :character   Median :2021-12-13 13:04:54.50
##                                        Mean   :2021-12-13 23:39:29.21
##                                        3rd Qu.:2021-12-20 10:14:01.00
##                                        Max.   :2021-12-31 23:59:48.00
##     ended_at                      member_casual      ride_duration
##  Min.   :2021-12-01 00:02:40.00   Length:247540      Min.   :    0.000
##  1st Qu.:2021-12-06 13:02:03.50   Class :character   1st Qu.:    4.967
##  Median :2021-12-13 13:18:39.00   Mode  :character   Median :    8.433
##  Mean   :2021-12-13 23:54:00.61                      Mean   :   14.523
##  3rd Qu.:2021-12-20 10:24:38.25                      3rd Qu.:   14.733
##  Max.   :2022-01-03 17:32:18.00                      Max.   :30400.550
##  day_of_week         month_name          start_hour       season
##  Length:247540      Length:247540      Min.   : 0.00   Length:247540
##  Class :character   Class :character   1st Qu.:10.00   Class :character
##  Mode  :character   Mode  :character   Median :14.00   Mode  :character
##                                        Mean   :13.67
##                                        3rd Qu.:17.00
##                                        Max.   :23.00
```

```
# Inspecting rows with ride duration greater than 1000 minutes.
all_trips[all_trips$ride_duration > 1000, ]
```

_

Remove unwanted rows

We’ll then remove the all the erroneous rows.

```
# Upon merging all datasets, we'll observe that there are entries with negative ride duration. This must be due to some sort of
error. Hence, we'll remove them.
# all_trips <- subset(all_trips, ride_duration > 0)

all_trips <- na.omit(all_trips)     # Removing any row with NA values
```

_

Splitting data frames

We'll split the data frames into two. One will contain main data for our analysis, and another with outliers.

```
all_trips_core <- subset(all_trips, ride_duration <= 1000)

all_trips_outliers <- subset(all_trips, ride_duration > 1000)

# Deleting original data frame as it is no longer necessary
rm(all_trips)
```

Analysis

We'll now conduct some basic descriptive analysis.

```
# Comparing Members and Casual Riders based on their average Ride duration
aggregate(all_trips_core$ride_duration ~ all_trips_core$member_casual, FUN = mean)
```

```
##   all_trips_core$member_casual all_trips_core$ride_duration
## 1                 Casual Rider                     17.59971
## 2                       Member                     10.78726
```

```
# Comparing Type of Bikes based on their average Ride duration
aggregate(all_trips_core$ride_duration ~ all_trips_core$rideable_type, FUN = mean)
```

```
##   all_trips_core$rideable_type all_trips_core$ride_duration
## 1                 Classic Bike                     13.39888
## 2                  Docked Bike                     36.25133
## 3                Electric Bike                     11.40622
```

```
# Sorting days of the week in the desired order. We have to do this because R sorts the data alphabetically by default.
all_trips_core$day_of_week <- ordered(all_trips_core$day_of_week, levels=c("Monday", "Tuesday", "Wednesday", "Thursday", "Frida
y", "Saturday", "Sunday"))

# Analyzing ridership data based on Type of Rider and Day of the Week
all_trips_core %>%
  group_by(member_casual, day_of_week) %>%  # Groups by Type of Rider and Day of the Week
  summarise(number_of_rides = n() ,average_duration = mean(ride_duration)) %>%  # Calculate Number of rides and Average duratio
n
  arrange(member_casual, day_of_week) #Sorts the data
```

```
## # A tibble: 14 × 4
## # Groups:   member_casual [2]
##    member_casual day_of_week number_of_rides average_duration
##    <chr>         <ord>                 <int>            <dbl>
##  1 Casual Rider  Monday                 7703             18.0
##  2 Casual Rider  Tuesday                6224             15.9
##  3 Casual Rider  Wednesday             10679             16.5
##  4 Casual Rider  Thursday              12568             16.7
##  5 Casual Rider  Friday                12923             17.5
##  6 Casual Rider  Saturday              11061             19.1
##  7 Casual Rider  Sunday                 8411             19.4
##  8 Member        Monday                22482             10.6
##  9 Member        Tuesday               22145             10.6
## 10 Member        Wednesday             34036             10.4
## 11 Member        Thursday              35176             10.7
## 12 Member        Friday                29482             10.8
## 13 Member        Saturday              19063             11.3
## 14 Member        Sunday                15391             11.5
```

If we had data of other months, we could perform following analysis also.

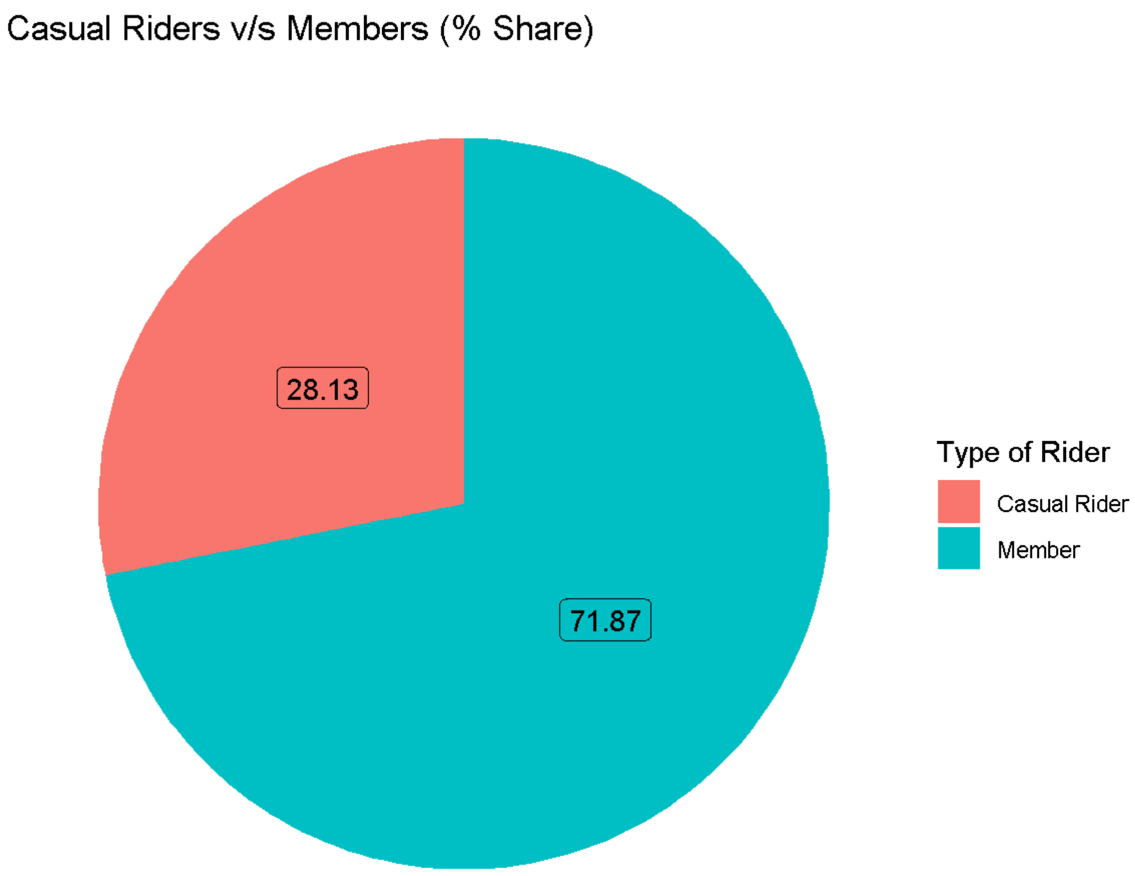
```
# Sorting months in the desired order.
all_trips_core$month_name <- ordered(all_trips_core$month_name, levels=c("December", "January", "February", "March", "April",
"May", "June", "July", "August", "September", "October", "November"))

# Analyzing ridership data based on Type of Rider and Month
all_trips_core %>%
  group_by(member_casual, month_name) %>%  # Groups by Type of Rider and Month
  summarise(number_of_rides = n() ,average_duration = mean(ride_duration)) %>%  # Calculate Number of rides and Average duratio
n
  arrange(member_casual, month_name) #Sorts the data

# Sorting seasons in the desired order.
all_trips_core$season <- ordered(all_trips_core$season, levels=c("Winter", "Spring", "Summer", "Fall"))

# Analyzing ridership data based on Type of Rider and Season
all_trips_core %>%
  group_by(member_casual, season) %>%  # Groups by Type of Rider and Season
  summarise(number_of_rides = n() ,average_duration = mean(ride_duration)) %>%  # Calculate Number of rides and Average duratio
n
  arrange(member_casual, season) #Sorts the data
```

Visualization

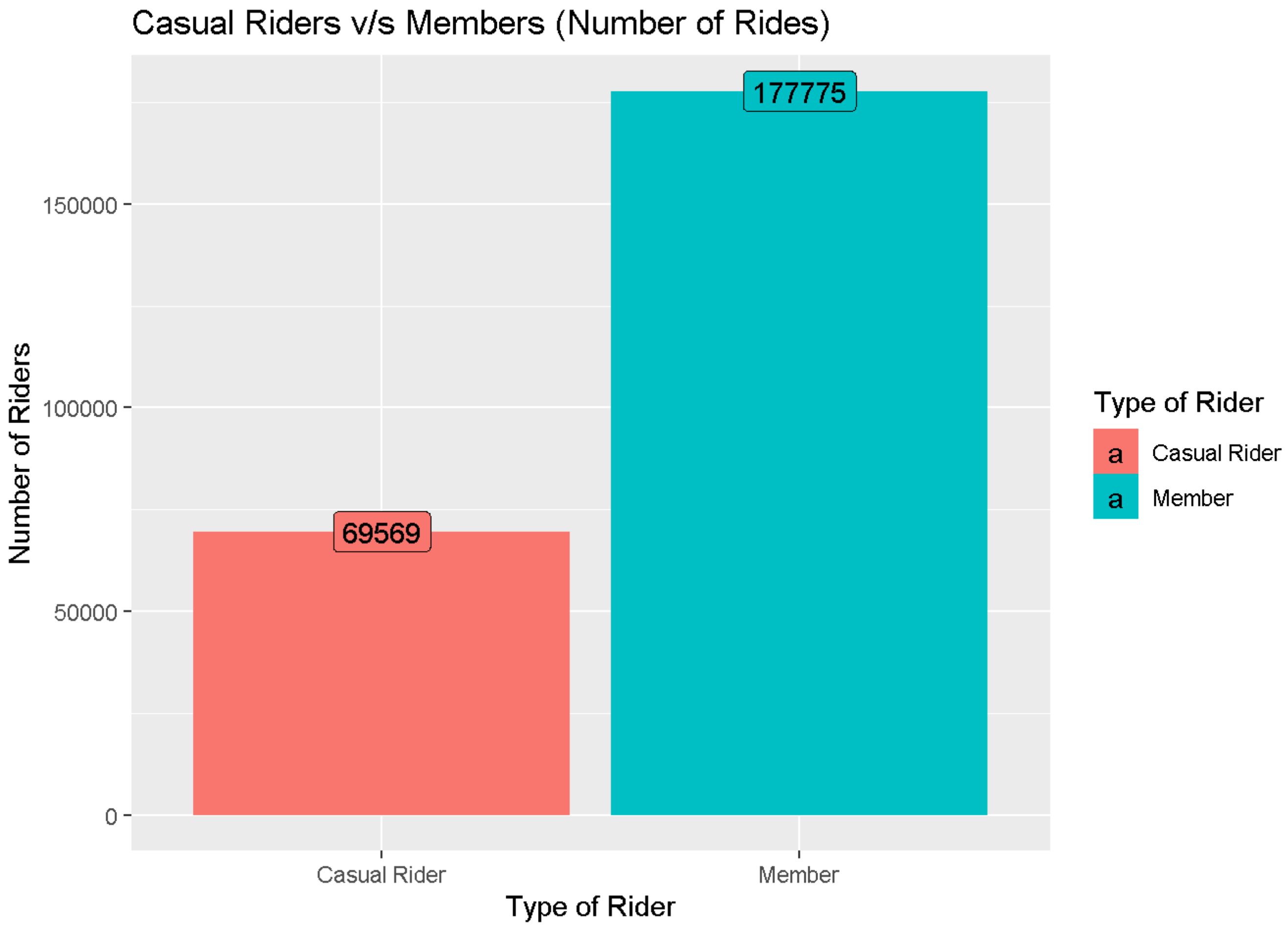
We’ll now visualize our data to get a better understanding of it.

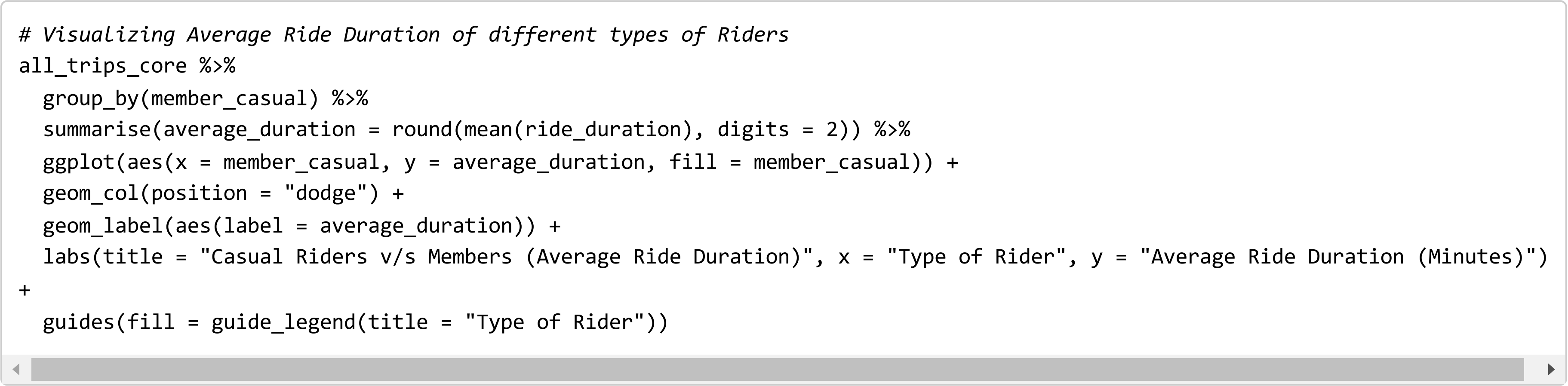
First we’ll visualize Number of Riders and Average Ride Duration based on different Type of Riders.

```
# Visualizing Percentage share of Type of Riders
all_trips_core %>%
  group_by(member_casual) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = "", y = number_of_rides, fill = member_casual)) +
  geom_col() +
  coord_polar(theta = "y", start = 0) +
  geom_label(aes(label = round(((number_of_rides)/nrow(all_trips_core))*100, digits = 2)), position = position_stack(vjust = 0.
5), show.legend = FALSE) +
  labs(title = "Casual Riders v/s Members (% Share)", x = "", y = "Number of Riders") +
  guides(fill = guide_legend(title = "Type of Rider")) +
  theme_void()
```

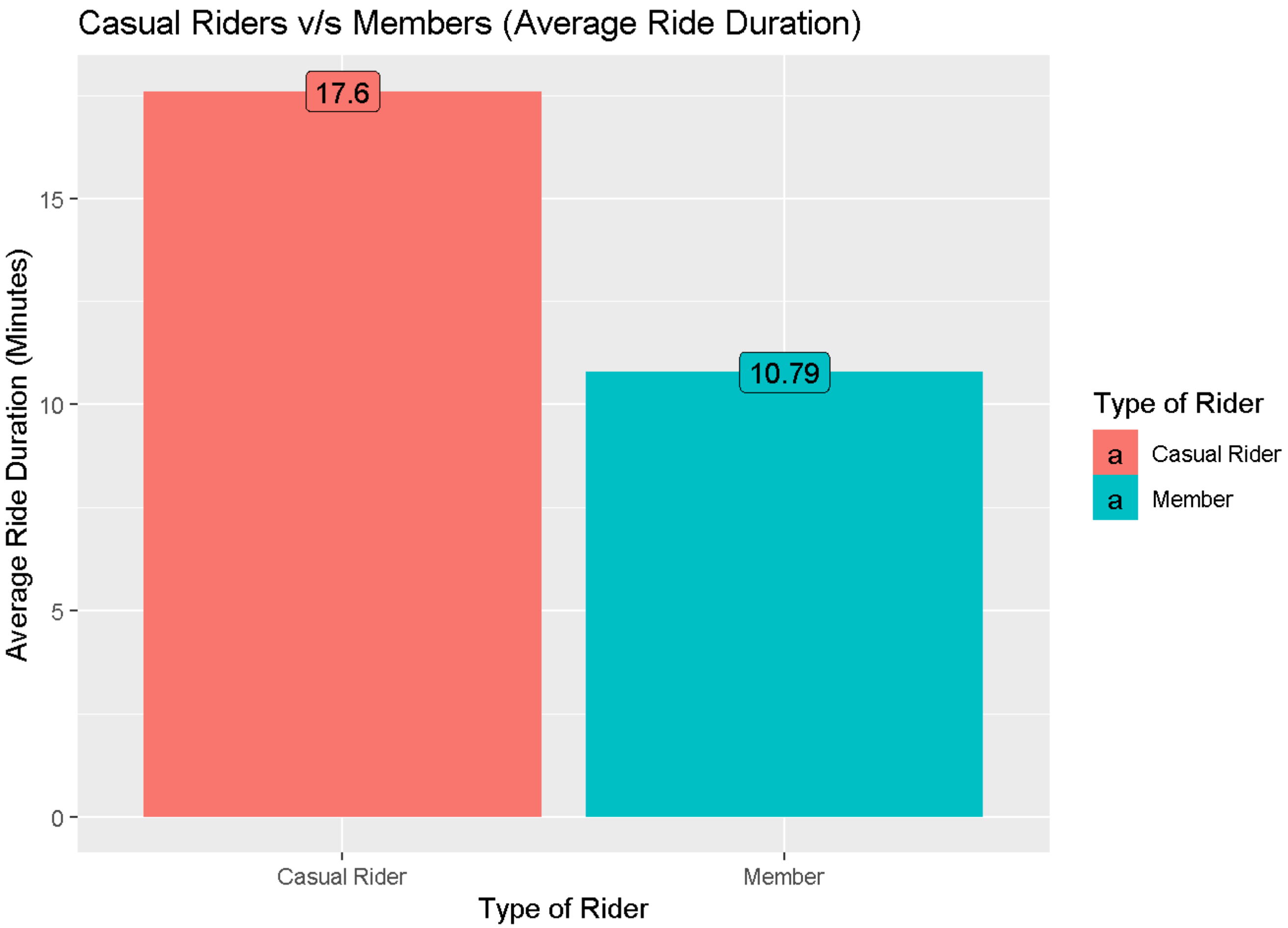


```
# Visualizing Number of different types of Riders
all_trips_core %>%
  group_by(member_casual) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = member_casual, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  geom_label(aes(label = number_of_rides)) +
  labs(title = "Casual Riders v/s Members (Number of Rides)", x = "Type of Rider", y = "Number of Riders") +
  guides(fill = guide_legend(title = "Type of Rider"))
```



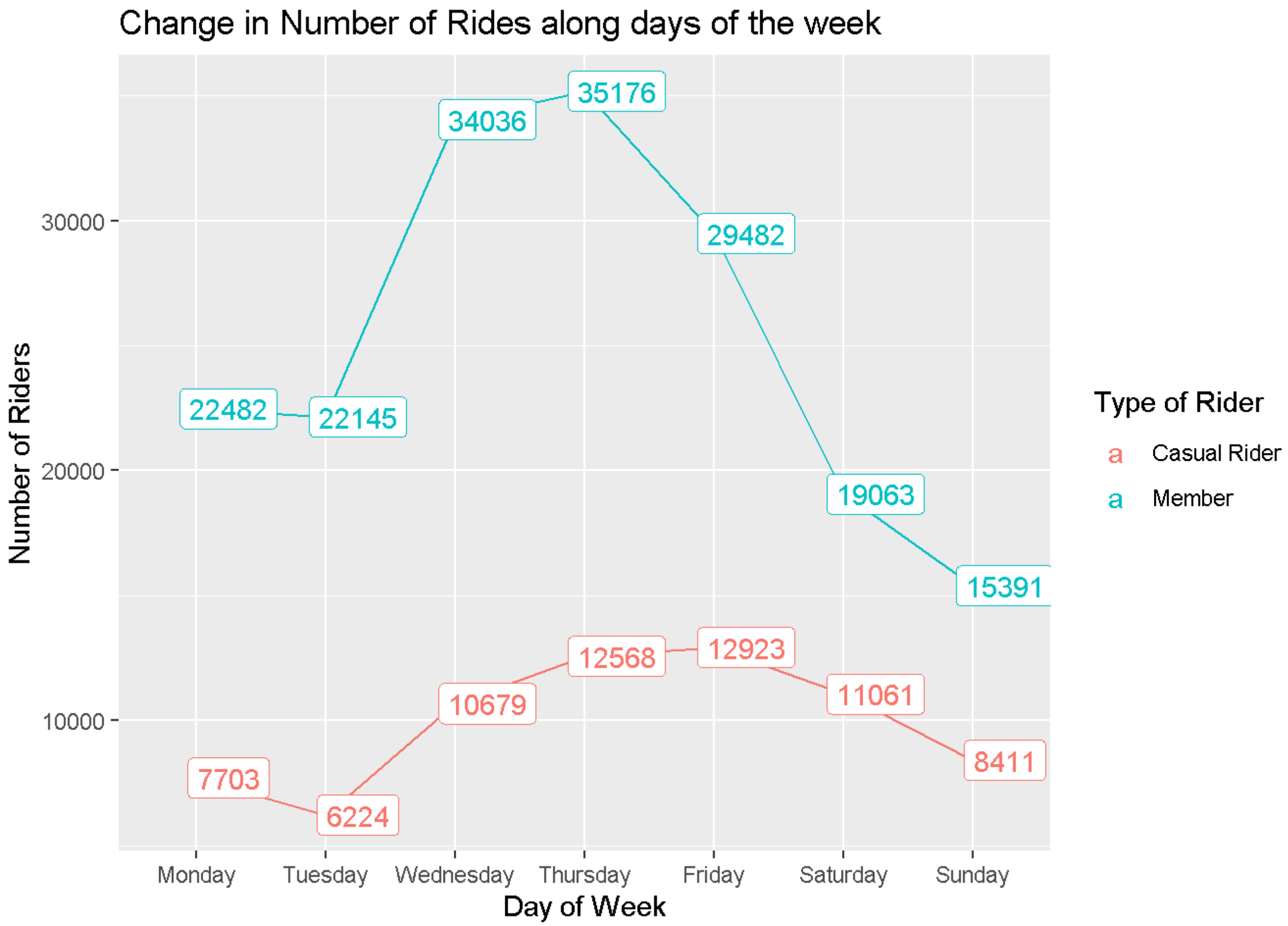


```
# Visualizing Average Ride Duration of different types of Riders
all_trips_core %>%
  group_by(member_casual) %>%
  summarise(average_duration = round(mean(ride_duration), digits = 2)) %>%
  ggplot(aes(x = member_casual, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  geom_label(aes(label = average_duration)) +
  labs(title = "Casual Riders v/s Members (Average Ride Duration)", x = "Type of Rider", y = "Average Ride Duration (Minutes)")
+
  guides(fill = guide_legend(title = "Type of Rider"))
```

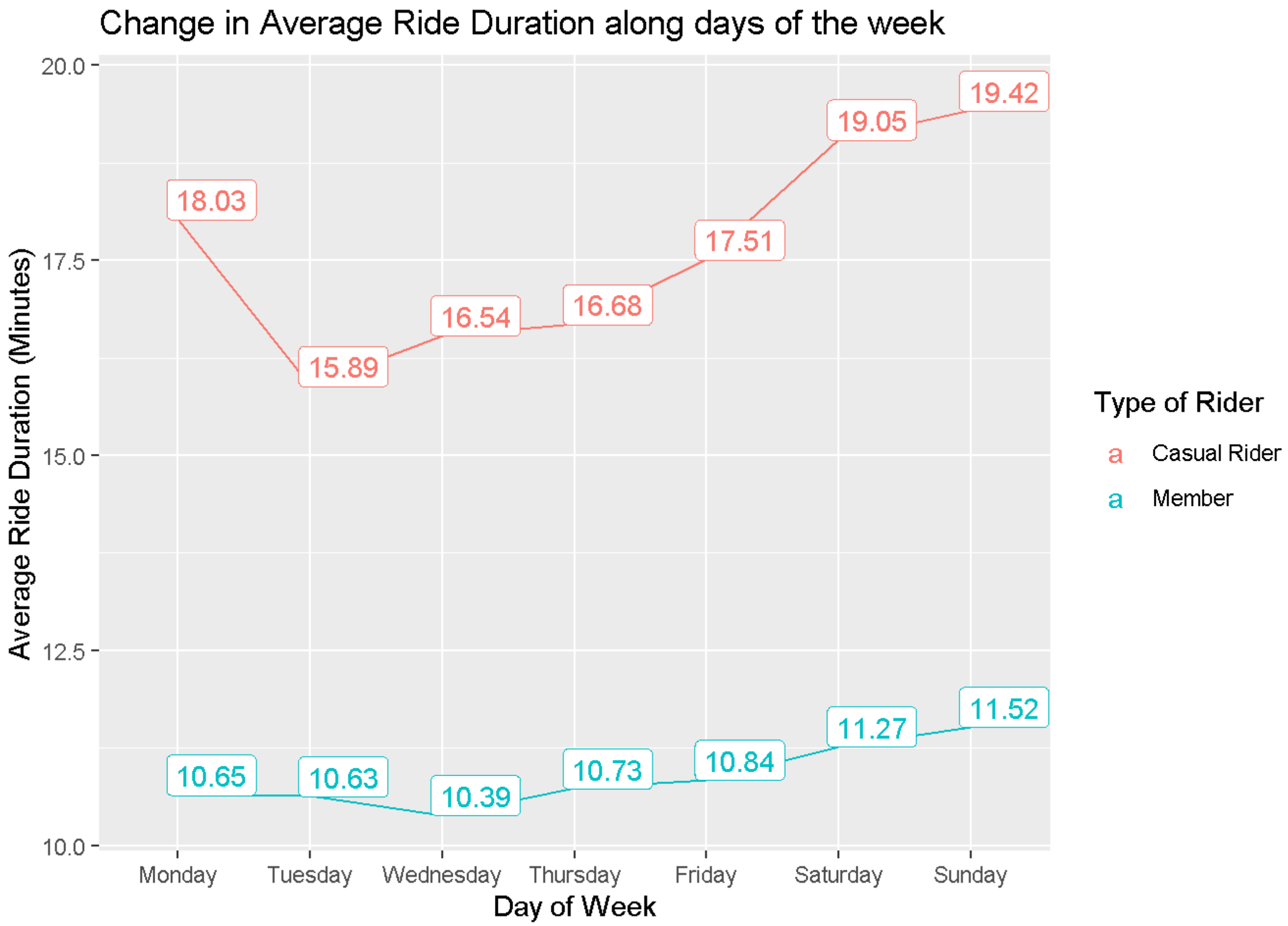


Casual Riders v/s Members (Average Ride Duration)
Average Ride Duration (Minutes)
0
5
10
15
17.6
10.79
Casual Rider
Member
Type of Rider
Type of Rider
a Casual Rider
a Member

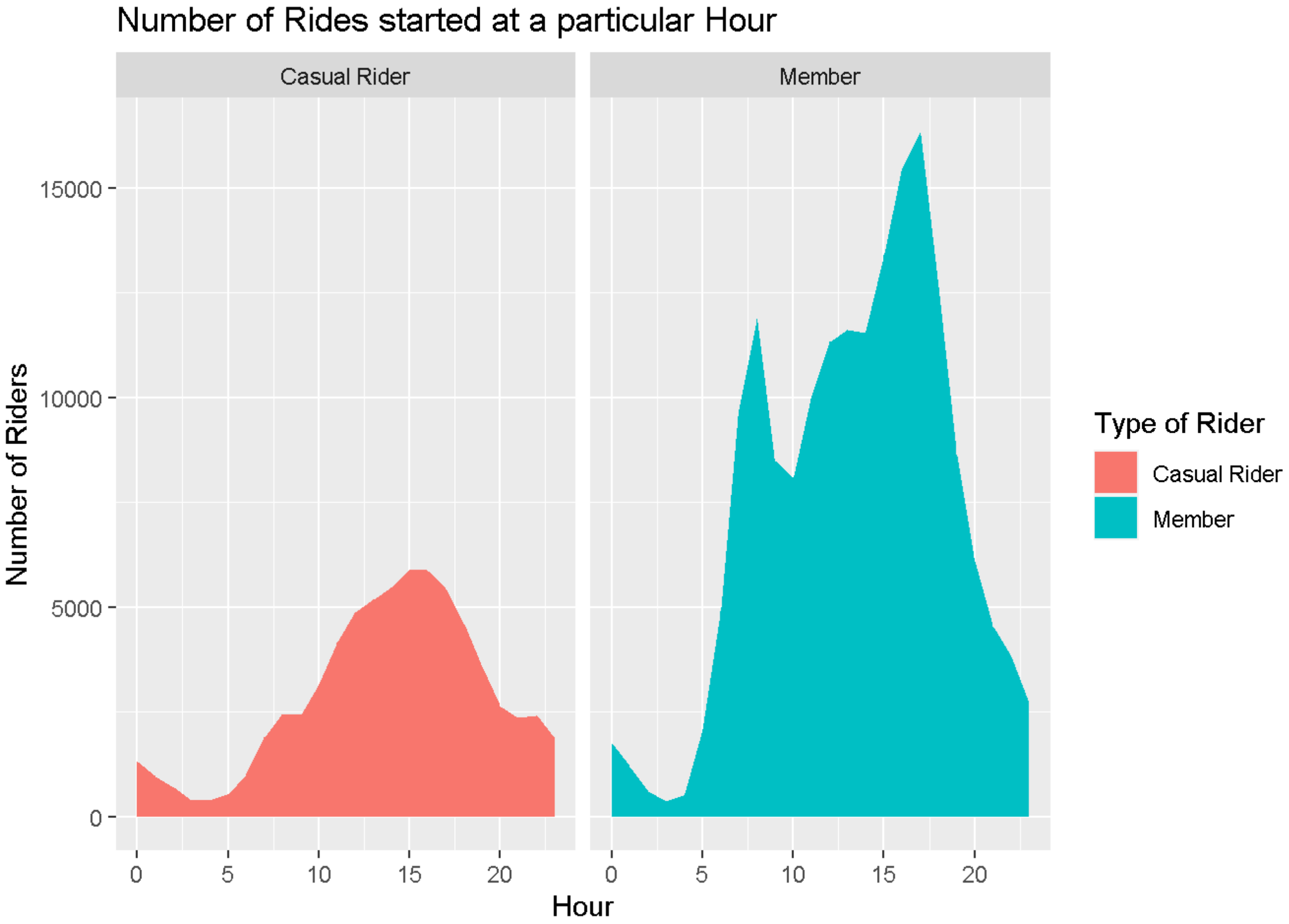
```r
# Sorting days of the week in the desired order. We have to do this because R sorts the data alphabetically by default.
all_trips_core$day_of_week <- ordered(all_trips_core$day_of_week, levels=c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"))

# Visualizing Number of Riders of each Rider type and Day of the Week
all_trips_core %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, group = member_casual, color = member_casual)) +
  geom_line() +
  geom_label(aes(label = number_of_rides), nudge_x = 0.25, nudge_y = 0.25, check_overlap = TRUE) +
  labs(title = "Change in Number of Rides along days of the week", x = "Day of Week", y = "Number of Riders", color = "Type of Rider")
```



```
# Visualizing Average Ride Duration of each Rider type and Day of the Week
all_trips_core %>%
  group_by(member_casual, day_of_week) %>%
  summarise(average_duration = round(mean(ride_duration), digits = 2)) %>%
  arrange(member_casual, day_of_week)  %>%
  ggplot(aes(x = day_of_week, y = average_duration, group = member_casual, color = member_casual)) +
  geom_line() +
  geom_label(aes(label = average_duration), nudge_x = 0.25, nudge_y = 0.25, check_overlap = TRUE) +
  labs(title = "Change in Average Ride Duration along days of the week", x = "Day of Week", y = "Average Ride Duration (Minute
s)", color = "Type of Rider")
```



```
# Visualizing Number of rides started at a particular hour
all_trips_core %>%
  group_by(member_casual, start_hour) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual, start_hour)  %>%
  ggplot(aes(x = start_hour, y = number_of_rides, fill = member_casual)) +
  geom_area() +
  labs(title = "Number of Rides started at a particular Hour", x = "Hour", y = "Number of Riders", color = "Type of Rider") +
  guides(fill = guide_legend(title = "Type of Rider")) +
  facet_grid(.~member_casual)
```



_

If we had data for whole year, we could visualize Monthly and Seasonal trends also with the following code.

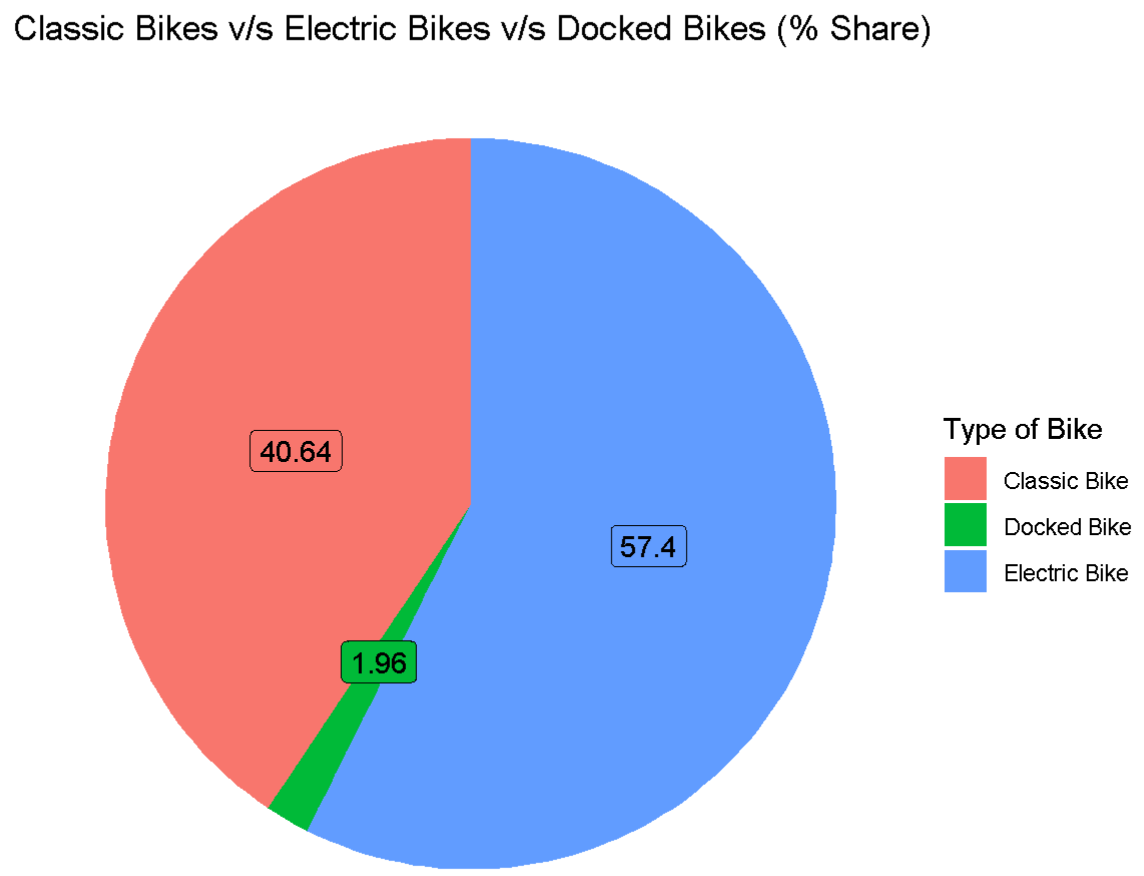
```r
# Sorting months in the desired order.
all_trips_core$month_name <- ordered(all_trips_core$month_name, levels=c("December", "January", "February", "March", "April",
"May", "June", "July", "August", "September", "October", "November"))

# Visualizing Number of Riders of each Rider type and Month
all_trips_core %>%
  group_by(member_casual, month_name) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual, month_name)  %>%
  ggplot(aes(x = month_name, y = number_of_rides, group = member_casual, color = member_casual)) +
  geom_line() +
  geom_label(aes(label = number_of_rides), nudge_x = 0.25, nudge_y = 0.25, check_overlap = TRUE) +
  labs(title = "Monthly Change in Number of Rides", x = "Month", y = "Number of Riders", color = "Type of Rider")

# Visualizing Average Ride Duration of each Rider type and Month
all_trips_core %>%
  group_by(member_casual, month_name) %>%
  summarise(average_duration = round(mean(ride_duration), digits = 2)) %>%
  arrange(member_casual, month_name)  %>%
  ggplot(aes(x = month_name, y = average_duration, group = member_casual, color = member_casual)) +
  geom_line() +
  geom_label(aes(label = average_duration), nudge_x = 0.25, nudge_y = 0.25, check_overlap = TRUE) +
  labs(title = "Monthly Change Average Ride Duration", x = "Month", y = "Average Ride Duration (Minutes)", color = "Type of Rid
er")

# Sorting seasons in the desired order.
all_trips_core$season <- ordered(all_trips_core$season, levels=c("Winter", "Spring", "Summer", "Fall"))

# Visualizing Number of Riders of each Rider type and Season
all_trips_core %>%
  group_by(member_casual, season) %>%
  summarise(number_of_rides = n()) %>%
  arrange(member_casual, season)  %>%
  ggplot(aes(x = season, y = number_of_rides, group = member_casual, color = member_casual)) +
  geom_line() +
  geom_label(aes(label = number_of_rides), nudge_x = 0.25, nudge_y = 0.25, check_overlap = TRUE) +
  labs(title = "Seasonal Change in Number of Rides", x = "Season", y = "Number of Riders", color = "Type of Rider")
```

```
# Visualizing Average Ride Duration of each Rider type and Season
all_trips_core %>%
  group_by(member_casual, season) %>%
  summarise(average_duration = round(mean(ride_duration), digits = 2)) %>%
  arrange(member_casual, season)  %>%
  ggplot(aes(x = season, y = average_duration, group = member_casual, color = member_casual)) +
  geom_line() +
  geom_label(aes(label = average_duration), nudge_x = 0.25, nudge_y = 0.25, check_overlap = TRUE) +
  labs(title = "Seasonal Change Average Ride Duration", x = "Season", y = "Average Ride Duration (Minutes)", color = "Type of R
ider")
```

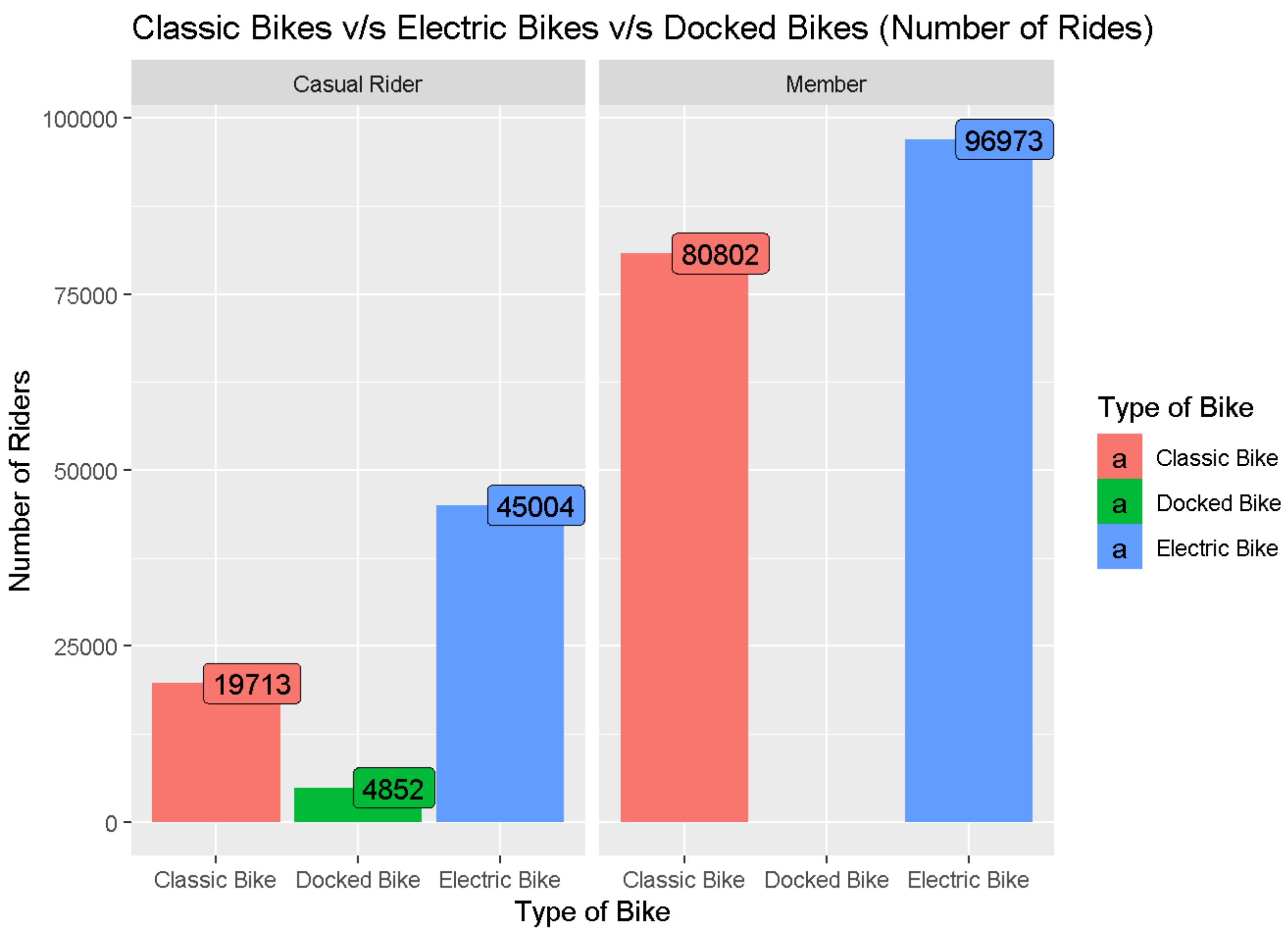
_

Now we’ll visualize Number of Riders and Average Ride Duration based on different Type of Bikes.

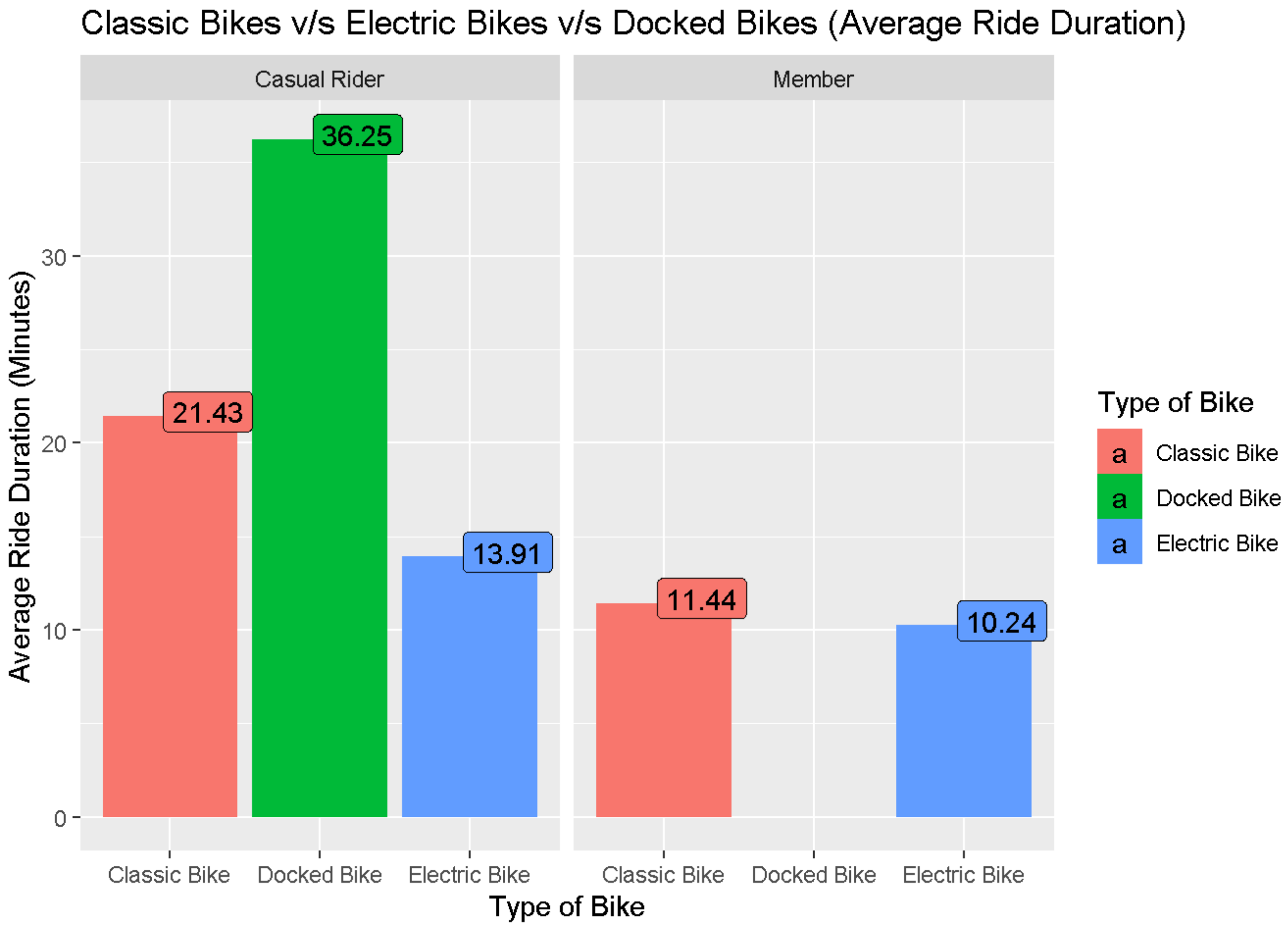
```
# Visualizing Percentage share of Type of Bikes
all_trips_core %>%
  group_by(rideable_type) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = "", y = number_of_rides, fill = rideable_type)) +
  geom_col() +
  coord_polar(theta = "y", start = 0) +
  geom_label(aes(label = round(((number_of_rides)/nrow(all_trips_core))*100, digits = 2)), position = position_stack(vjust = 0.
5), show.legend = FALSE) +
  labs(title = "Classic Bikes v/s Electric Bikes v/s Docked Bikes (% Share)", x = "", y = "Number of Riders") +
  guides(fill = guide_legend(title = "Type of Bike")) +
  theme_void()
```



```
# Visualizing Number of different types of Bikes among different types of Riders
all_trips_core %>%
  group_by(rideable_type, member_casual) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = rideable_type, y = number_of_rides, fill = rideable_type)) +
  geom_col(position = "dodge") +
  geom_label(aes(label = number_of_rides), nudge_x = 0.25, nudge_y = 0.25, check_overlap = TRUE) +
  labs(title = "Classic Bikes v/s Electric Bikes v/s Docked Bikes (Number of Rides)", x = "Type of Bike", y = "Number of Rider
s") +
  guides(fill = guide_legend(title = "Type of Bike")) +
  facet_grid(.~member_casual)
```



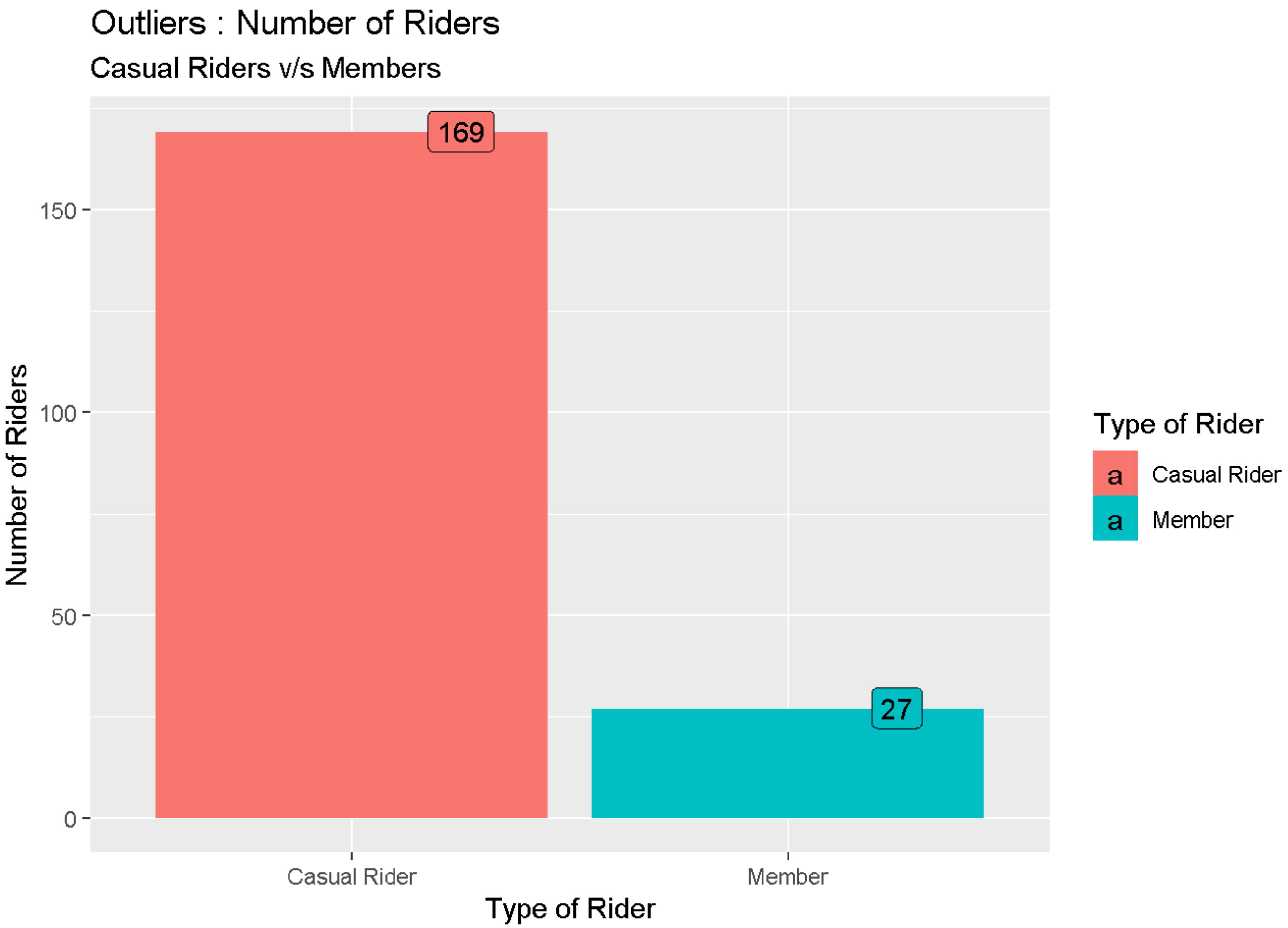
```
# Visualizing Average Ride Duration of different types of Bikes among different types of Riders
all_trips_core %>%
  group_by(rideable_type, member_casual) %>%
  summarise(average_duration = round(mean(ride_duration), digits = 2)) %>%
  ggplot(aes(x = rideable_type, y = average_duration, fill = rideable_type)) +
  geom_col(position = "dodge") +
  geom_label(aes(label = average_duration), nudge_x = 0.25, nudge_y = 0.25, check_overlap = TRUE) +
  labs(title = "Classic Bikes v/s Electric Bikes v/s Docked Bikes (Average Ride Duration)", x = "Type of Bike", y = "Average Ri
de Duration (Minutes)") +
  guides(fill = guide_legend(title = "Type of Bike")) +
  facet_grid(.~member_casual)
```



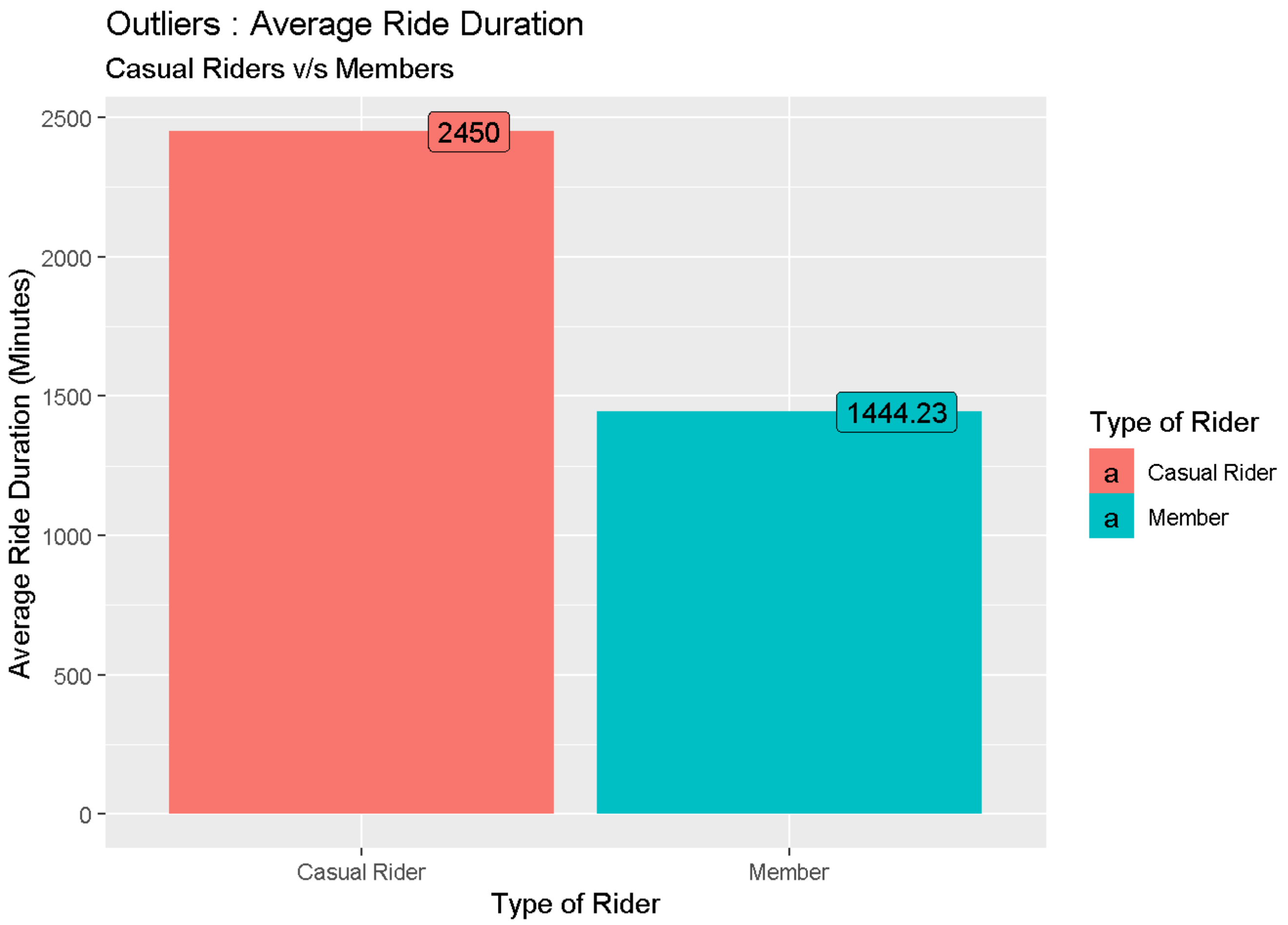
–

We can visualize Outliers also with the following code to better understand their position and significance.

```r
# Visualizing Outliers

# Visualizing Number of different types of Riders
all_trips_outliers %>%
  group_by(member_casual) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = member_casual, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  geom_label(aes(label = number_of_rides), nudge_x = 0.25, nudge_y = 0.25, check_overlap = TRUE) +
  labs(title = "Outliers : Number of Riders", subtitle = "Casual Riders v/s Members", x = "Type of Rider", y = "Number of Riders") +
  guides(fill = guide_legend(title = "Type of Rider")) +
  scale_fill_discrete(labels = c("Casual Rider", "Member"))
```


Outliers : Number of Riders
Casual Riders v/s Members
169
27
150
100
50
0
Number of Riders
Casual Rider
Member
Type of Rider
Type of Rider
a Casual Rider
a Member

```
# Visualizing Average Ride Duration of different types of Riders
all_trips_outliers %>%
  group_by(member_casual) %>%
  summarise(average_duration = round(mean(ride_duration), digits = 2)) %>%
  ggplot(aes(x = member_casual, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  geom_label(aes(label = average_duration), nudge_x = 0.25, nudge_y = 0.25, check_overlap = TRUE) +
  labs(title = "Outliers : Average Ride Duration", subtitle = "Casual Riders v/s Members", x = "Type of Rider", y = "Average Ri
de Duration (Minutes)") +
  guides(fill = guide_legend(title = "Type of Rider")) +
  scale_fill_discrete(labels = c("Casual Rider", "Member"))
```



** Thank You **