

Exploratory Data Analysis – Cyclistic Bike Share

BI

2023-10-02

```
library(tidyverse)

## — Attaching core tidyverse packages — tidyverse
## 2.0.0 —
## ✓ dplyr      1.1.2      ✓ readr      2.1.4
## ✓ forcats    1.0.0      ✓ stringr    1.5.0
## ✓ ggplot2    3.4.2      ✓ tibble     3.2.1
## ✓ lubridate  1.9.2      ✓ tidyr      1.3.0
## ✓ purrr      1.0.1
## — Conflicts —
tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
## conflicts to become errors library(dplyr)

library(ggplot2)
Set the working directory
setwd("F:/TRAINING/DATCYCLIS") getwd()

## [1] "F:/TRAINING/DATCYCLIS"
```

Data Collection

[Import all the csv data files into Rstudio](#)

```
divvytrips1_Jan2021 <- read.csv("202101-divvy-tripdata.csv")

divvytrips1_Feb2021 <- read.csv("202102-divvy-tripdata.csv")

divvytrips1_Mar2021 <- read.csv("202103-divvy-tripdata.csv")

divvytrips1_Apr2021 <-read.csv("202104-divvy-tripdata.csv")

divvytrips1_May2021 <-read.csv("202105-divvy-tripdata.csv")

divvytrips1_Jun2021 <-read.csv("202106-divvy-tripdata.csv")

divvytrips1_Jul2021 <-read.csv("202107-divvy-tripdata.csv")
```

```
divvytrips1_Aug2021 <-read.csv("202108-divvy-tripdata.csv")
divvytrips1_Sep2021 <-read.csv("202109-divvy-tripdata.csv")
divvytrips1_Oct2021 <-read.csv("202110-divvy-tripdata.csv")
divvytrips1_Nov2021 <-read.csv("202111-divvy-tripdata.csv")
divvytrips1_Dec2021 <-read.csv("202112-divvy-tripdata.csv")
```

Data Cleaning and Preparation

Join all the above data frames for each month of the year into one

```
divvytrips_tot1 <- rbind(divvytrips1_Jan2021, divvytrips1_Feb2021,
divvytrips1_Mar2021, divvytrips1_Apr2021,
divvytrips1_May2021, divvytrips1_Jun2021,
divvytrips1_Jul2021, divvytrips1_Aug2021,
divvytrips1_Sep2021, divvytrips1_Oct2021,
divvytrips1_Nov2021, divvytrips1_Dec2021)
```

The merged data frames contains irrelevant variables to our analysis, we will remove them.

```
divvytrips_sub1 <- subset(divvytrips_tot1, select = -c(ride_id,
start_station_id, start_station_name, end_station_name,
end_station_id, start_lat, start_lng, end_lat, end_lng ))
str(divvytrips_sub1)
## 'data.frame': 5595063 obs. of 4 variables:
## $ rideable_type: chr "electric_bike" "electric_bike" "electric_bike"
"electric_bike" ...
## $ started_at : chr "2021-01-23 16:14:19" "2021-01-27 18:43:08"
"202101-21 22:35:54" "2021-01-07 13:31:13" ...
## $ ended_at : chr "2021-01-23 16:24:44" "2021-01-27 18:47:12"
"202101-21 22:37:14" "2021-01-07 13:42:55" ...
## $ member_casual: chr "member" "member" "member" "member" ...
```

View a summary of the data frame variables

```
summary(divvytrips_sub1)
```

## rideable_type	started_at	ended_at	member_casual
## Length:5595063	Length:5595063	Length:5595063	Length:5595063
## Class :character	Class :character	Class :character	Class :character
## Mode :character	Mode :character	Mode :character	Mode :character

Remove all the missing values - Drop all the NA

```
divvytrips_sub1 <- na.omit(divvytrips_sub1)
```

Data Manipulation

Convert the started_at and ended_at variables from character to date type

```
divvytrips_sub1$start2_date <- as.Date(divvytrips_sub1$started_at, format = "%Y-%m-%d")
divvytrips_sub1$ended2_date <- as.Date(divvytrips_sub1$ended_at, format = "%Y-%m-%d")
```

Extract month,day,year from start2_date

```
divvytrips_sub1$month2 <- month(divvytrips_sub1$start2_date)

divvytrips_sub1$day2 <- day(divvytrips_sub1$start2_date)

divvytrips_sub1$year2 <- year(divvytrips_sub1$start2_date)
```

Create a new column by extracting the day of the week from the start date

```
divvytrips_sub1$day_of_the_week2 <- format(divvytrips_sub1$start2_date, "%A")

divvytrips_sub1$month_name2 <- format(divvytrips_sub1$start2_date, "%b")
```

Calculate the ride_length (duration) in seconds

```
divvytrips_sub1$ride_length <- difftime(divvytrips_sub1$ended_at,
divvytrips_sub1$started_at )
```

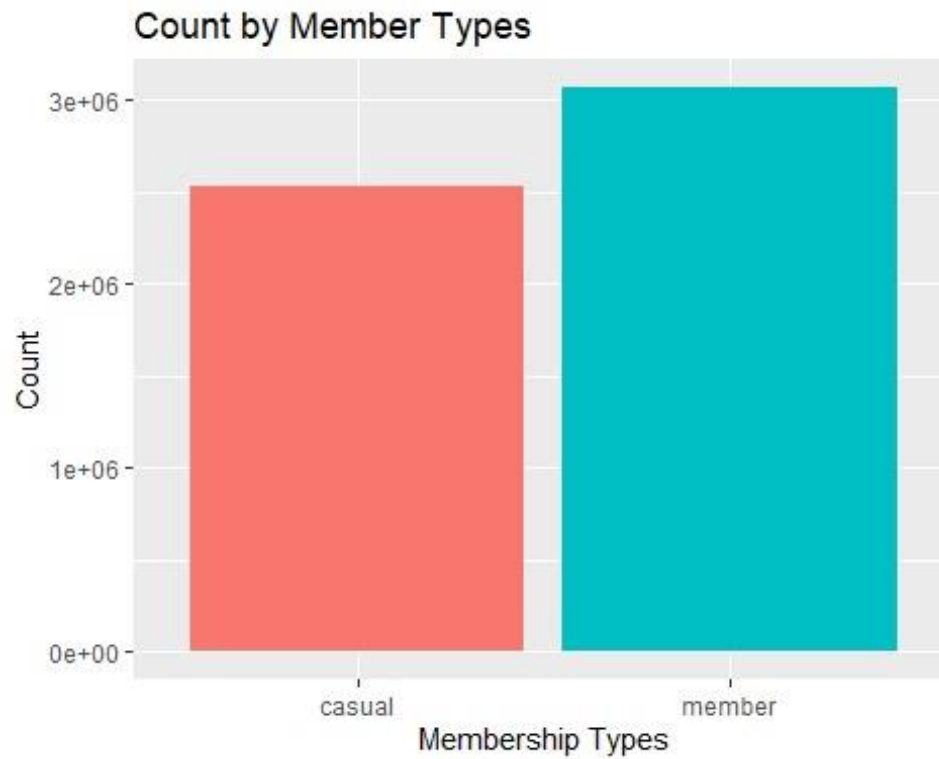
View the data with all the transformations

```
View(divvytrips_sub1)
```

Visualize the data

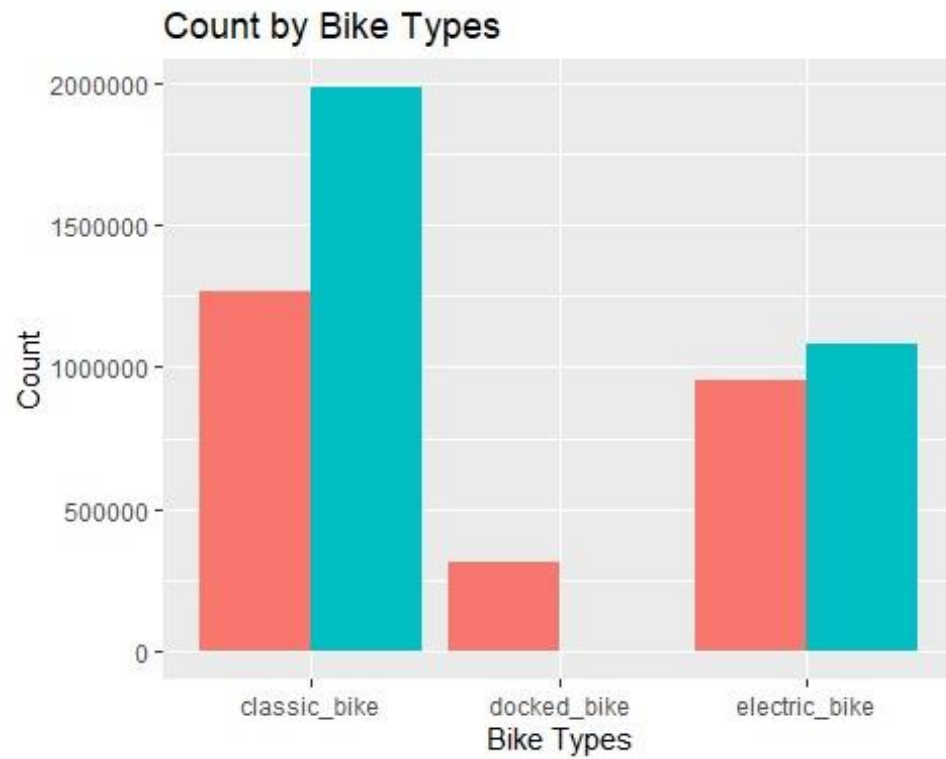
Visualize the count by membership types.

```
ggplot(data = divvytrips_sub1, aes(member_casual, fill = member_casual)) +
  geom_bar(position = "dodge") +
  labs(title = "Count by Member Types", x= "Membership Types", y = "Count") +
  theme(legend.position = "None")
```



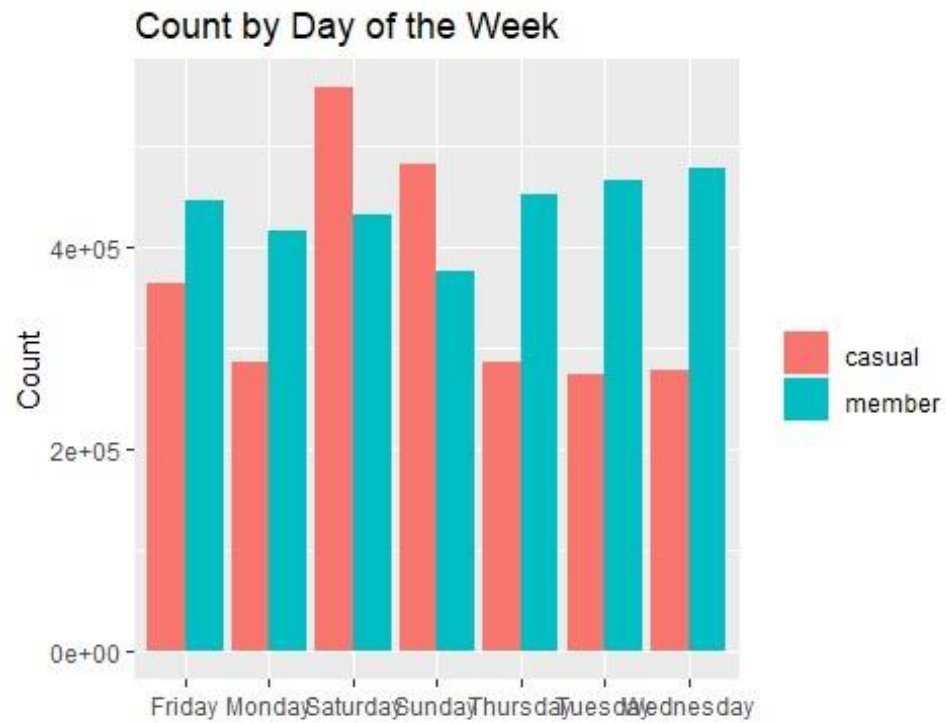
Visualize the count by the different types of bikes (rideable_type)

```
ggplot(data = divvytrips_sub1, aes(rideable_type, fill = member_casual)) +  
  geom_bar(position = "dodge") +  
  labs(title = "Count by Bike Types", x= "Bike Types", y = "Count") +  
  theme(legend.position = "None")
```



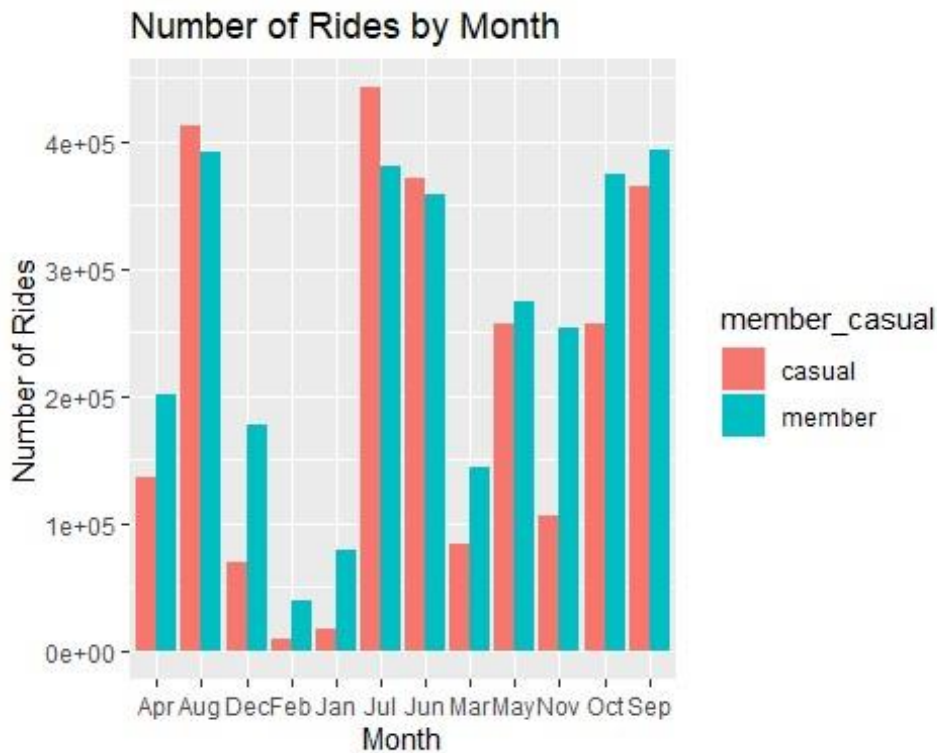
Visualize the general count for diferent days of the week

```
ggplot(data = divvytrips_sub1, aes(day_of_the_week2, fill = member_casual)) +  
  geom_bar(position = "dodge") +  
  labs(title = "Count by Day of the Week", x= " ", y = "Count")+  
  theme(legend.title = element_blank())
```



Visualize the count by month by member type (2)

```
ggplot(data = divvytrips_sub1, aes(month_name2, fill = member_casual)) +
  geom_bar(position = "dodge") +
  labs(title = "Number of Rides by Month", x= "Month", y = "Number of Rides")
```



Average Ride Length by Membership Type

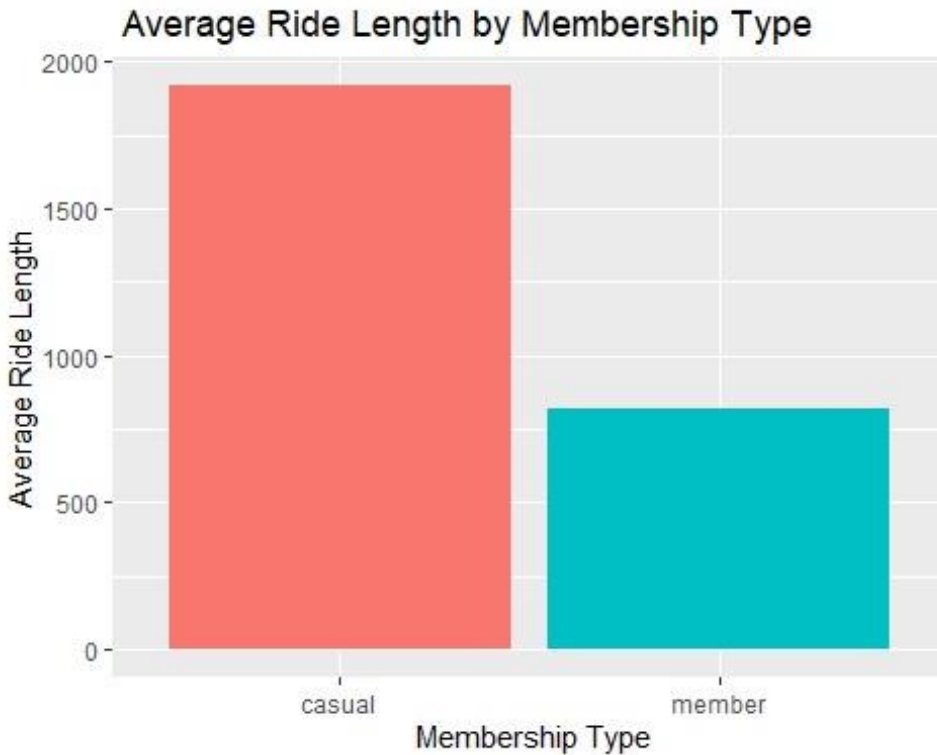
```
divvytrips_sub1 %>% group_by(member_casual) %>% summarise(average_ride_length = mean(ride_length))
```

```
## # A tibble: 2 × 2
##   member_casual average_ride_length
##   <chr>         <drtn>
## 1 casual      1920.0896 secs    ##
## 2 member      817.9822 secs
```

```
divvytrips_sub1 %>% group_by(member_casual) %>% summarise(average_ride_length = mean(ride_length)) %>%
```

```
ggplot(aes(x= member_casual, fill = member_casual, y= average_ride_length)) + geom_col(position = "dodge") +
  labs(title = " Average Ride Length by Membership Type", x= "
Membership Type", y = " Average Ride Length ") + theme(legend.position
= "None")
```

```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



Group the resulting data by the column `casual_member`

```
divvytrips_sub1_by_member_casual <- group_by(divvytrips_sub1, member_casual)
```

Count the number of members in each group

```
summarize(divvytrips_sub1_by_member_casual, quantitybymember = n())
```

```
## # A tibble: 2 × 2
##   member_casual quantitybymember
##   <chr>          <int>
## 1 casual          2529005
## 2 member          3066058
```

Summarize to Calculate the average ride length by group

```
summarize(divvytrips_sub1_by_member_casual, Average_Ride_Length =
mean(ride_length))
```

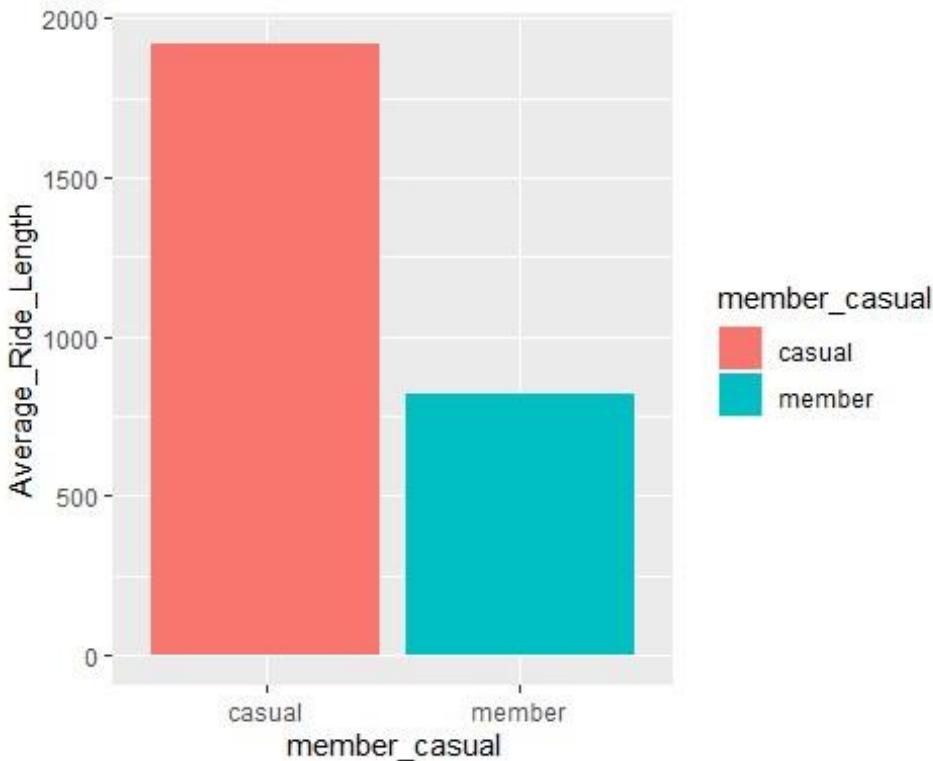
```
## # A tibble: 2 × 2
##   member_casual Average_Ride_Length
##   <chr>          <drtn>
## 1 casual          1920.0896 secs
## 2 member           817.9822 secs
```

```
Avgridebyyear <- summarize(divvytrips_sub1_by_member_casual,
Average_Ride_Length = mean(ride_length))
```

Plot the result dataframe `Avgridebyyear`

```
Avgridebyyear %>% ggplot(aes(x = member_casual, fill = member_casual, y =
Average_Ride_Length)) + geom_col(position = "dodge")
```

```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



Group the resulting data by the column `casual_member` and `month_name2`

```
divvytrips_sub1_by_member_casual_month <- group_by(divvytrips_sub1,
member_casual, month_name2)
```

Plot `member_casual`, `month_name2`, `Average_Ride_Length`

```
summarize(divvytrips_sub1_by_member_casual_month, Average_Ride_Length =
mean(ride_length))
```

```
## `summarise()` has grouped output by 'member_casual'. You can override
using the
```

```
## `.groups` argument.
```

```
## # A tibble: 24 × 3
```

```
## # Groups:   member_casual [2]
```

```
##   member_casual month_name2 Average_Ride_Length
##   <chr>          <chr>      <drtn>
## 1 casual      Apr       2281.379 secs
## 2 casual      Aug       1727.182 secs
## 3 casual      Dec       1409.657 secs
## 4 casual      Feb       2962.394 secs
## 5 casual      Jan       1541.075 secs
```

```
## 6 casual      Jul      1967.410 secs
## 7 casual      Jun      2227.286 secs
## 8 casual      Mar      2286.739 secs
## 9 casual      May      2293.858 secs
## 10 casual     Nov      1389.780 secs
## # i 14 more rows
```

```
Avgridebymonth <- summarize(divvytrips_sub1_by_member_casual_month,
  Average_Ride_Length = mean(ride_length))
```

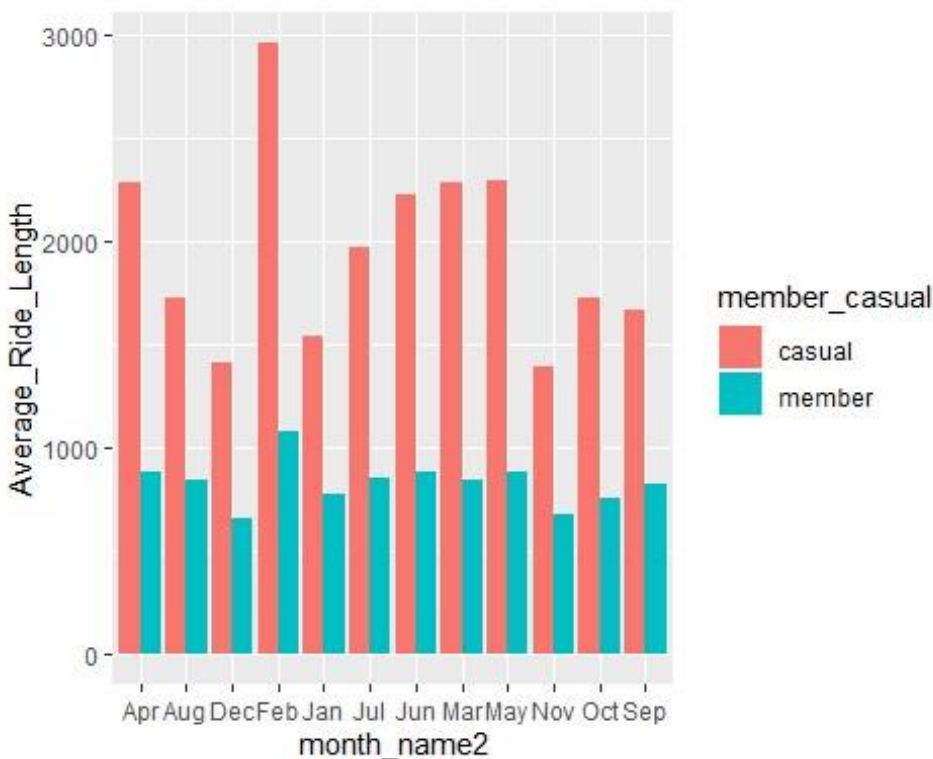
```
## `summarise()` has grouped output by 'member_casual'. You can override
using the
```

```
## `.groups` argument.
```

Plot the result dataframe Avgridebymonth

```
Avgridebymonth %>% ggplot(aes(x = month_name2, fill = member_casual, y =
  Average_Ride_Length)) + geom_col(position = "dodge")
```

```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



Group the resulting data by the column casual_member and day_of_the_week2

```
divvytrips_sub1_by_member_casual_day_of_the_week2 <-
  group_by(divvytrips_sub1, member_casual, day_of_the_week2)
```

To Plot member_casual, day_of_the_week2, Average_Ride_Length

```

summarize(divvytrips_sub1_by_member_casual_day_of_the_week2,
Average_Ride_Length = mean(ride_length))

## `summarise()` has grouped output by 'member_casual'. You can override
using the
## `.groups` argument.

## # A tibble: 14 × 3
## # Groups:   member_casual [2]
##   member_casual day_of_the_week2 Average_Ride_Length
##   <chr>         <chr>         <drtn>
## 1 casual      Friday          1820.8806 secs
## 2 casual      Monday          1912.4939 secs
## 3 casual      Saturday         2082.2740 secs
## 4 casual      Sunday          2253.9274 secs
## 5 casual      Thursday         1662.2206 secs
## 6 casual      Tuesday          1678.2879 secs
## 7 casual      Wednesday         1659.4264 secs
## 8 member      Friday           799.4854 secs
## 9 member      Monday           794.8362 secs
## 10 member     Saturday          915.8778 secs
## 11 member     Sunday            939.3469 secs
## 12 member     Thursday           766.5641 secs
## 13 member     Tuesday           767.2800 secs      ## 14 member
Wednesday          769.0844 secs

```

Let us put the result in a variable `AvgridebyDayoftheWeek2` to see it better

```

AvgridebyDayoftheWeek2 <-
summarize(divvytrips_sub1_by_member_casual_day_of_the_week2,
Average_Ride_Length = mean(ride_length))

## `summarise()` has grouped output by 'member_casual'. You can override
using the
## `.groups` argument.

```

Plot the result dataframe `AvgridebyDayoftheWeek2`

```

AvgridebyDayoftheWeek2 %>% ggplot(aes(x = day_of_the_week2, fill =
member_casual, y = Average_Ride_Length)) + geom_col(position = "dodge")

## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.

```

