Bike_Share_2020_Q1

Hardikkumar Malaviya

2024-01-17

Install Packages:

```
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("lubridate")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("janitor")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("dplyr")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

##Load Packages:

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(lubridate)
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

```
##
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
library(dplyr)
library(tidyr)
library(ggplot2)
rm(list=ls())
```

Check is there any empty rows or colums

```
Bike_D1 <- read_csv("Divvy_Trips_2020_Q1.csv") %>%
remove_empty(which = c("cols")) %>%
remove_empty(which = c("rows"))
```

```
## Rows: 426887 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## dttm (2): started_at, ended_at
##
## 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.
```

Add columns with day of week and start / end time in hours and minutes

```
Bike_Data <- Bike_D1 %>%
  mutate(day_of_week=weekdays(started_at)) %>%
  mutate(start_hr=format(as.POSIXct(started_at), format = "%H:%M")) %>%
  mutate(end_hr=format(as.POSIXct(ended_at), format = "%H:%M"))
```

##Start time :

```
df1<-separate(Bike_Data,col=start_hr, into = c('start_h', 'start_m'), sep = ':',remove = TRUE, convert 
    mutate(Start_Time_minutes = (as.integer(start_h) * 60 + as.integer(start_m)))
```

##End Time :

```
df2<-
  separate(Bike_Data,col=end_hr, into = c('end_h', 'end_m'), sep = ':',remove = TRUE,
  convert = FALSE) %>%
    mutate(End_Time_minutes = (as.integer(end_h) * 60 + as.integer(end_m)))
```

##Merge data frams :

```
df3<-merge(x = df1, y = df2, all = TRUE)
```

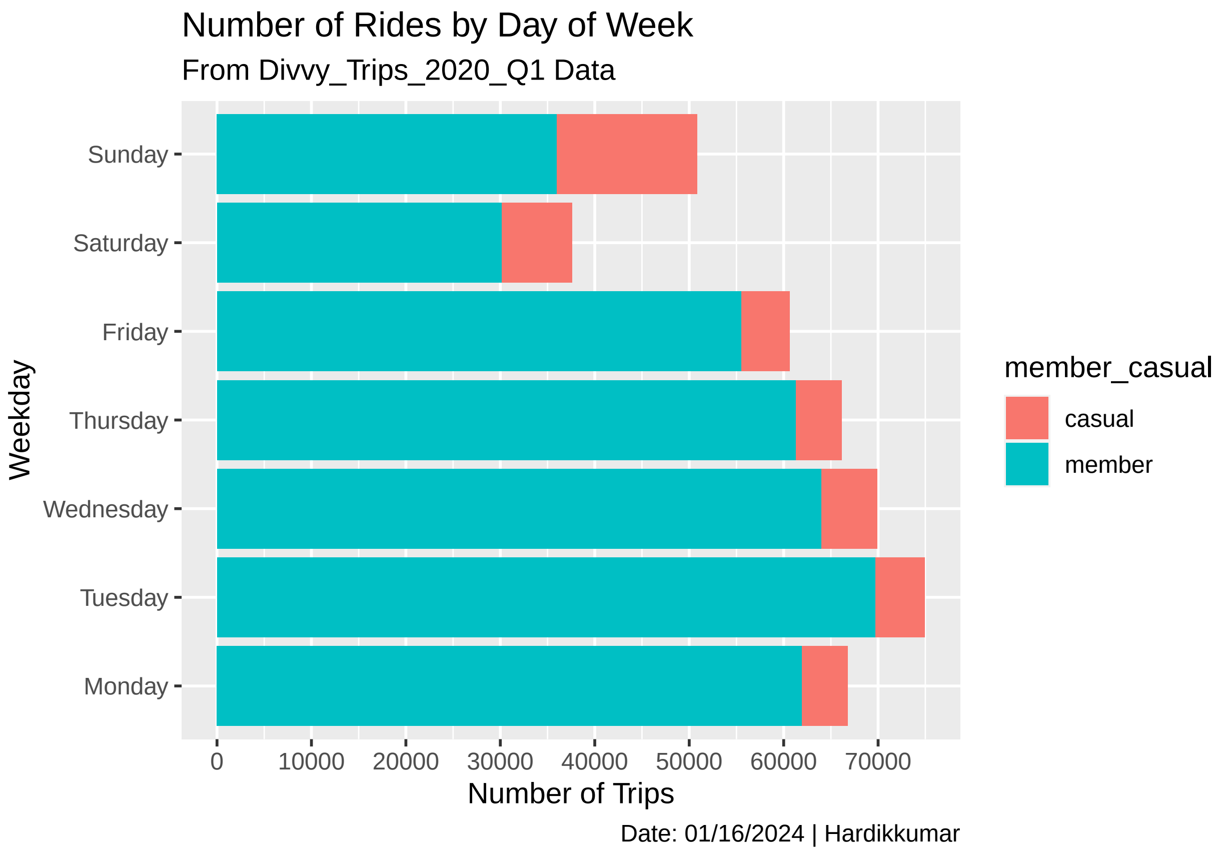
##Calculte ride length = end time - start time

```
Final_Data <- df3 %>%
  mutate(ride_length = End_Time_minutes- Start_Time_minutes)
```

Use. factor funtion,

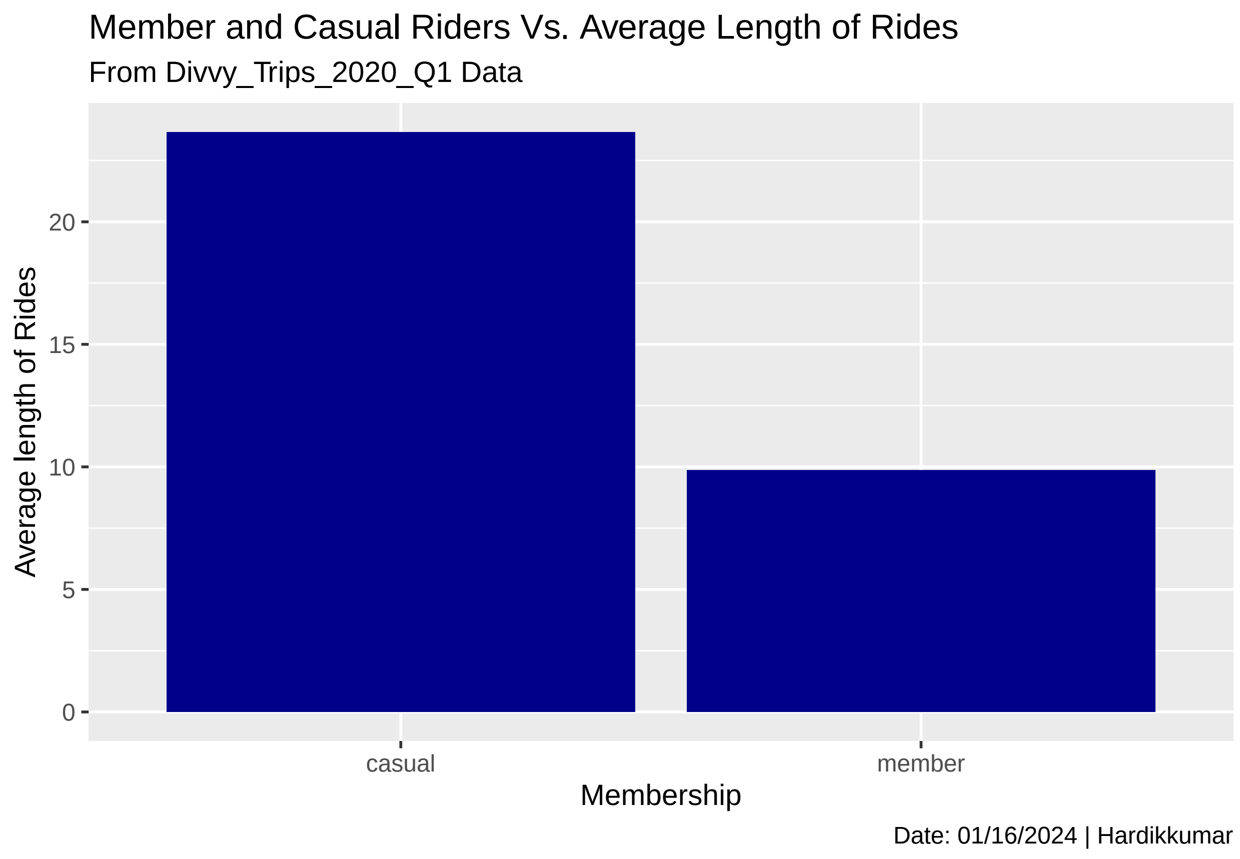
```
Final_Data$day_of_week <- factor(Final_Data$day_of_week, c("Monday", "Tuesday", "Wednesday","Thursday",

Final_Data %>%
  ggplot() + geom_bar(aes(y=day_of_week, fill= member_casual)) +
  labs(title = "Number of Rides by Day of Week",subtitle = "From Divvy_Trips_2020_Q1 Data", caption = "
 scale_x_continuous(breaks = c(0, 10000, 20000, 30000, 40000, 50000,60000,70000,80000))
```



##Plot 2:

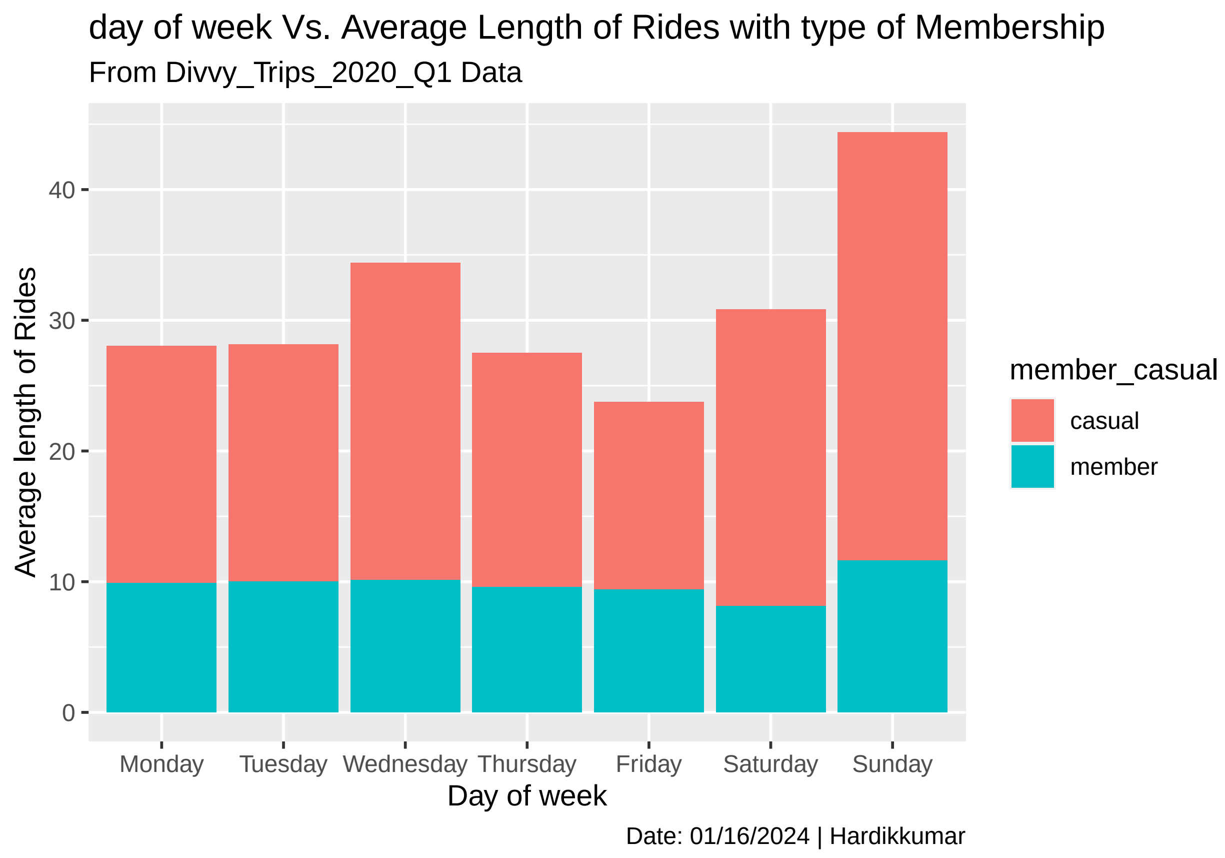
```
Final_Data %>%
  group_by(member_casual) %>%
  arrange(member_casual,desc(member_casual)) %>%
  summarise(mean_ride=mean(ride_length)) %>%
 ggplot() + geom_col(aes(y=mean_ride, x= member_casual), fill= "darkblue") +
  labs(title = "Member and Casual Riders Vs. Average Length of Rides",subtitle = "From Divvy_Trips_2020_
```



##Plot 3:

```
Final_Data$day_of_week <- factor(Final_Data$day_of_week, c("Monday", "Tuesday", "Wednesday","Thursday",
Final_Data %>%
  group_by(day_of_week,member_casual) %>%
  summarise(mean_ride=mean(ride_length)) %>%
 ggplot() + geom_col(aes(y=mean_ride, x= day_of_week, fill= member_casual) )+
labs(title = "day of week Vs. Average Length of Rides with type of Membership",subtitle = "From Divvy_T
```

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



##Plot 4:

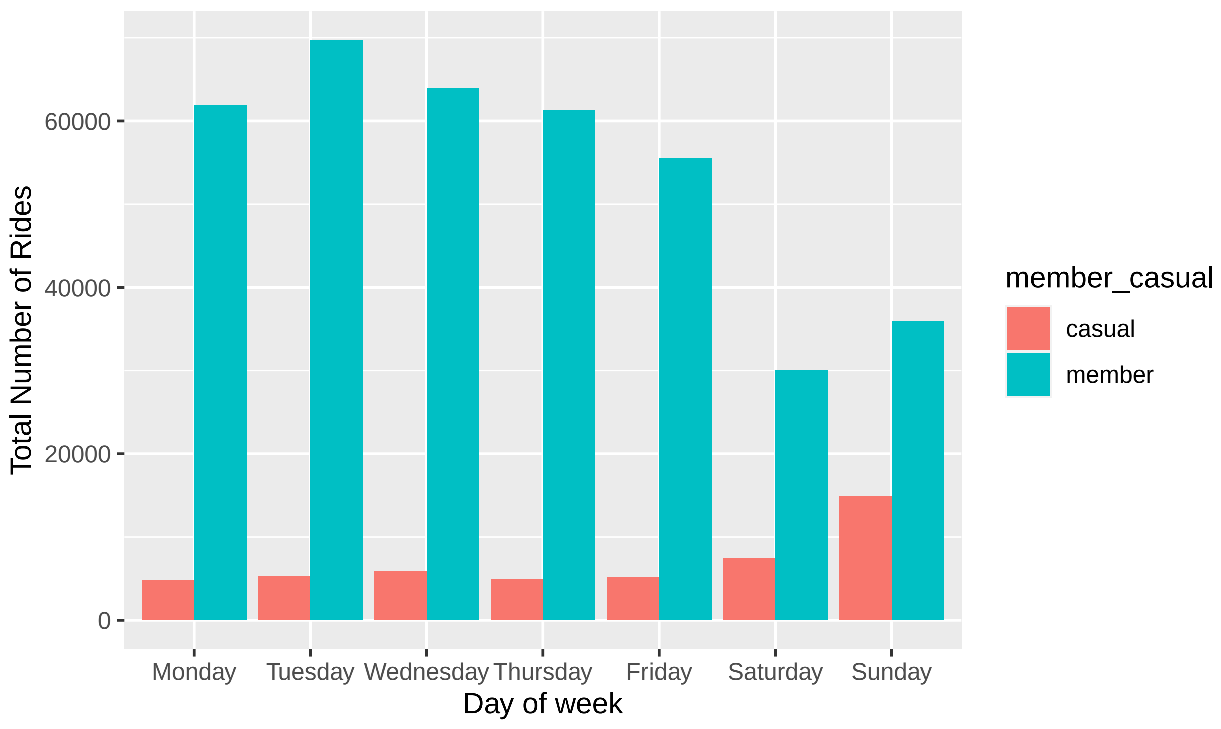
```
Final_Data$day_of_week <- factor(Final_Data$day_of_week, c("Monday", "Tuesday", "Wednesday","Thursday",
Final_Data %>%

 ggplot() + geom_bar(aes(x=day_of_week, fill= member_casual), position = "dodge")+


labs(title = "Day of Week Vs. Average Length of Rides with type of Membership",subtitle = "From Divvy_T
```

Day of Week Vs. Average Length of Rides with type of Membership

From Divvy_Trips_2020_Q1 Data



Date: 01/16/2024 | Hardikkumar