Cyclistic Bike Share Analysis

Yousef Ayman

8/17/2022

```
library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse
1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.1.3

## Warning: package 'tibble' was built under R version 4.1.3

## Warning: package 'tidyr' was built under R version 4.1.3

## Warning: package 'readr' was built under R version 4.1.2

## Warning: package 'purrr' was built under R version 4.1.2

## Warning: package 'dplyr' was built under R version 4.1.3

## Warning: package 'stringr' was built under R version 4.1.2

## Warning: package 'forcats' was built under R version 4.1.3

## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

library(lubridate)

## Warning: package 'lubridate' was built under R version 4.1.3

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

library(ggplot2)
library(hydroTSM)

## Warning: package 'hydroTSM' was built under R version 4.1.3
## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.1.3

## 
## Attaching package: 'zoo'
## 
## The following objects are masked from 'package:base':
## 
##     as.Date, as.Date.numeric
## 
## Loading required package: xts

## Warning: package 'xts' was built under R version 4.1.3

## 
## Attaching package: 'xts'
## 
## The following objects are masked from 'package:dplyr':
## 
##     first, last
## 
## 
## Attaching package: 'hydroTSM'
## 
## The following object is masked from 'package:tidyr':
## 
##     extract

library(scales)

## Warning: package 'scales' was built under R version 4.1.3

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

Comments: Importing packages

```
rm(list = ls())
```

Comments: Clearing environment

```
a1 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202108-
divvy-tripdata.csv")
```

```
a2 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202109-
divvy-tripdata.csv")
a3 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202110-
divvy-tripdata.csv")
a4 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202111-
divvy-tripdata.csv")
a5 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202112-
divvy-tripdata.csv")
a6 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202201-
divvy-tripdata.csv")
a7 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202202-
divvy-tripdata.csv")
a8 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202203-
divvy-tripdata.csv")
a9 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202204-
divvy-tripdata.csv")
a10 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202205-
divvy-tripdata.csv")
a11 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202206-
divvy-tripdata.csv")
a12 <- read.csv("C:/Users/My PC/Desktop/google capstone/divvy-dataset/202207-
divvy-tripdata.csv")
```

Comments: importing data

```
data <- rbind(a1,a2,a3,a4,a5,a6,a7,a8,a9,a10,a11,a12)
```

Comments : Combining all the data into a single data frame

```
head(data)

##            ride_id rideable_type          started_at            ended_at
## 1 99103BB87CC6C1BB electric_bike 2021-08-10 17:15:49 2021-08-10 17:22:44
## 2 EAFCCCFB0A3FC5A1 electric_bike 2021-08-10 17:23:14 2021-08-10 17:39:24
## 3 9EF4F46C57AD234D electric_bike 2021-08-21 02:34:23 2021-08-21 02:50:36
## 4 5834D3208BFAF1DA electric_bike 2021-08-21 06:52:55 2021-08-21 07:08:13
## 5 CD825CB87ED1D096 electric_bike 2021-08-19 11:55:29 2021-08-19 12:04:11
## 6 612F12C94A964F3E electric_bike 2021-08-19 12:41:12 2021-08-19 12:47:47
##   start_station_name start_station_id end_station_name end_station_id
start_lat
## 1
41.77
## 2
41.77
## 3
41.95
## 4
41.97
## 5
41.79
## 6
```

```
41.81
##   start_lng end_lat end_lng member_casual
## 1    -87.68   41.77  -87.68        member
## 2    -87.68   41.77  -87.63        member
## 3    -87.65   41.97  -87.66        member
## 4    -87.67   41.95  -87.65        member
## 5    -87.60   41.77  -87.62        member
## 6    -87.61   41.80  -87.60        member
```

```
glimpse(data)
```

```
## Rows: 5,901,463
## Columns: 13
## $ ride_id            <chr> "99103BB87CC6C1BB", "EAFCCCFB0A3FC5A1",
"9EF4F46C57~
## $ rideable_type      <chr> "electric_bike", "electric_bike",
"electric_bike", ~
## $ started_at         <chr> "2021-08-10 17:15:49", "2021-08-10 17:23:14",
"2021~
## $ ended_at           <chr> "2021-08-10 17:22:44", "2021-08-10 17:39:24",
"2021~
## $ start_station_name <chr> "", "", "", "", "", "", "", "", "", "", "", "",
"",~
## $ start_station_id   <chr> "", "", "", "", "", "", "", "", "", "", "", "",
"",~
## $ end_station_name   <chr> "", "", "", "", "", "", "", "Clark St & Grace
St", ~
## $ end_station_id     <chr> "", "", "", "", "", "", "", "TA1307000127", "",
"",~
## $ start_lat          <dbl> 41.77000, 41.77000, 41.95000, 41.97000,
41.79000, 4~
## $ start_lng          <dbl> -87.68000, -87.68000, -87.65000, -87.67000, -
87.600~
## $ end_lat            <dbl> 41.77000, 41.77000, 41.97000, 41.95000,
41.77000, 4~
## $ end_lng            <dbl> -87.68000, -87.63000, -87.66000, -87.65000, -
87.620~
## $ member_casual      <chr> "member", "member", "member", "member",
"member", "~
```

```
str(data)
```

```
## 'data.frame':    5901463 obs. of  13 variables:
##  $ ride_id           : chr  "99103BB87CC6C1BB" "EAFCCCFB0A3FC5A1"
"9EF4F46C57AD234D" "5834D3208BFAF1DA" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike"
"electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2021-08-10 17:15:49" "2021-08-10 17:23:14"
"2021-08-21 02:34:23" "2021-08-21 06:52:55" ...
##  $ ended_at          : chr  "2021-08-10 17:22:44" "2021-08-10 17:39:24"
"2021-08-21 02:50:36" "2021-08-21 07:08:13" ...
```

```
##  $ start_station_name: chr  "" "" "" "" ...
##  $ start_station_id  : chr  "" "" "" "" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.8 41.8 42 42 41.8 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num  41.8 41.8 42 42 41.8 ...
##  $ end_lng           : num  -87.7 -87.6 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...
```

```
summary(data)
```

```
##    ride_id          rideable_type       started_at          ended_at        
##  Length:5901463     Length:5901463     Length:5901463     Length:5901463    
##  Class  :character  Class  :character  Class  :character  Class  :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##  start_station_name start_station_id   end_station_name   end_station_id    
##  Length:5901463     Length:5901463     Length:5901463     Length:5901463    
##  Class  :character  Class  :character  Class  :character  Class  :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##    start_lat       start_lng         end_lat          end_lng      
##  Min.   :41.64   Min.   :-87.84   Min.   :41.39   Min.   :-88.97  
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66  
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64  
##  Mean   :41.90   Mean   :-87.65   Mean   :41.90   Mean   :-87.65  
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63  
##  Max.   :45.64   Max.   :-73.80   Max.   :42.37   Max.   :-87.50  
##                                   NA's   :5590    NA's   :5590    
##  member_casual     
##  Length:5901463    
##  Class  :character  
##  Mode  :character  
##                    
##                    
##                    
##                    
```

Comments : Examining the data

```
x <- nrow(data) # checking number of rows before removing duplicates

data <- distinct(data) # removing duplicate rows
```

```
y <- nrow(data) # checking number of rows after removing duplicates

if(x==y){
  print("There is no duplicate rows in the data")
}else{
  print(paste("The number of duplicate rows in the data is " , (x-y)))
}

## [1] "There is no duplicate rows in the data"

data <- data %>%
            select(2,3,4,13) #selecting the date i need

unique(data$rideable_type) #seeing the unique values of the ride type

## [1] "electric_bike" "classic_bike"  "docked_bike"

unique(data$member_casual) #seeing the unique values of riders

## [1] "member" "casual"
```

Comments : Cleaning the data

```
data <- data %>%
              mutate(ride_length =
difftime(data$ended_at,data$started_at))#calculate the duration of the ride

sapply(data , class) #checking of data types of my columns

## rideable_type    started_at      ended_at member_casual   ride_length
##   "character"   "character"   "character"   "character"    "difftime"

data$date <- as.Date(data$started_at) #adding date column

data$year <- format(as.Date(data$date), "%Y") #adding year column

data$month <-  months(data$date) #adding month column

data$day_of_week <- format(as.Date(data$date), "%A") #adding day  column

data <- data %>%
  mutate(season = time2season(date,
                              out.fmt = "seasons")) # Convert dates to
seasons

data <- data %>%
  arrange(date) #sorting the data by date

data$day_of_week <- ordered(data$day_of_week, levels=c("Sunday", "Monday",
"Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")) # ordering day of
the week
```

```
data$ride_length <- as.numeric(as.character(data$ride_length)) #converting
column data type to numeric

data$ride_length <- data$ride_length/60 #converting ride length from sec to
mins

data <- data %>%
        filter(!(ride_length < 0))#filtering data
```

Comments : Transforming the data

```
head(data)

##   rideable_type          started_at            ended_at member_casual
## 1 electric_bike 2021-08-01 18:11:35 2021-08-01 18:17:05        member
## 2 electric_bike 2021-08-01 18:26:59 2021-08-01 18:32:23        member
## 3 electric_bike 2021-08-01 08:16:41 2021-08-01 08:46:14        member
## 4 electric_bike 2021-08-01 16:38:02 2021-08-01 16:55:43        member
## 5 electric_bike 2021-08-01 14:19:54 2021-08-01 14:22:48        member
## 6 electric_bike 2021-08-01 18:09:44 2021-08-01 18:35:33        member
##   ride_length       date year  month day_of_week season
## 1     5.50000 2021-08-01 2021 August      Sunday summer
## 2     5.40000 2021-08-01 2021 August      Sunday summer
## 3    29.55000 2021-08-01 2021 August      Sunday summer
## 4    17.68333 2021-08-01 2021 August      Sunday summer
## 5     2.90000 2021-08-01 2021 August      Sunday summer
## 6    25.81667 2021-08-01 2021 August      Sunday summer

aggregate(data$ride_length ~ data$member_casual, FUN = max)# Comparing
members and casual users max

##   data$member_casual data$ride_length
## 1             casual         41629.17
## 2             member          1559.90

aggregate(data$ride_length ~ data$member_casual, FUN = min)# Comparing
members and casual users min

##   data$member_casual data$ride_length
## 1             casual                0
## 2             member                0

aggregate(data$ride_length ~ data$member_casual, FUN = median)# Comparing
members and casual users median

##   data$member_casual data$ride_length
## 1             casual        14.400000
## 2             member         9.016667

aggregate(data$ride_length ~ data$member_casual, FUN = mean) # Comparing
members and casual users mean
```

```
##   data$member_casual data$ride_length
## 1             casual         29.21285
## 2             member         12.93272
```

```
#calculating total number of rides for each season
num_of_rides_season <- data %>%
  group_by(member_casual, data$season) %>%
  summarise(number_of_rides = n())
```

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

```
num_of_rides_season
```

```
## # A tibble: 8 x 3
## # Groups:   member_casual [2]
##   member_casual `data$season` number_of_rides
##   <chr>         <chr>                   <int>
## 1 casual        autumm                 728023
## 2 casual        spring                 496711
## 3 casual        summer                1187752
## 4 casual        winter                 109674
## 5 member        autumm                1019239
## 6 member        spring                 793435
## 7 member        summer                1209235
## 8 member        winter                 357245
```

```
#calculating total number of rides for each month
num_of_rides_month <- data %>%
  group_by(member_casual, data$month) %>%
  summarise(number_of_rides = n())
```

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

```
num_of_rides_month
```

```
## # A tibble: 24 x 3
## # Groups:   member_casual [2]
##    member_casual `data$month` number_of_rides
##    <chr>         <chr>                  <int>
##  1 casual        April                 126417
##  2 casual        August                412662
##  3 casual        December               69738
##  4 casual        February               21416
##  5 casual        January                18520
##  6 casual        July                  406046
##  7 casual        June                  369044
##  8 casual        March                  89880
##  9 casual        May                   280414
## 10 casual        November              106898
## # ... with 14 more rows
## # i Use `print(n = ...)` to see more rows
```

```
#calculating total number of rides for each day
num_of_rides_day <- data %>%
  group_by(member_casual, data$day_of_week) %>%
  summarise(number_of_rides = n())
```

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

```
num_of_rides_day
```

```
## # A tibble: 14 x 3
## # Groups:   member_casual [2]
##    member_casual `data$day_of_week` number_of_rides
##    <chr>         <ord>                        <int>
##  1 casual        Sunday                      475591
##  2 casual        Monday                      299653
##  3 casual        Tuesday                     273810
##  4 casual        Wednesday                   281783
##  5 casual        Thursday                    316118
##  6 casual        Friday                      347637
##  7 casual        Saturday                    527568
##  8 member        Sunday                      417953
##  9 member        Monday                      472387
## 10 member        Tuesday                     523377
## 11 member        Wednesday                   522617
## 12 member        Thursday                    522658
## 13 member        Friday                      466676
## 14 member        Saturday                    453486
```

```
#calculating total number of ride type
num_of_rideable_type <- data %>%
  group_by(member_casual, data$rideable_type) %>%
  summarise(number_of_rides = n())
```

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

```
num_of_rideable_type
```

```
## # A tibble: 5 x 3
## # Groups:   member_casual [2]
##   member_casual `data$rideable_type` number_of_rides
##   <chr>         <chr>                          <int>
## 1 casual        classic_bike                 1132868
## 2 casual        docked_bike                   226723
## 3 casual        electric_bike                1162569
## 4 member        classic_bike                 1922698
## 5 member        electric_bike                1456456
```

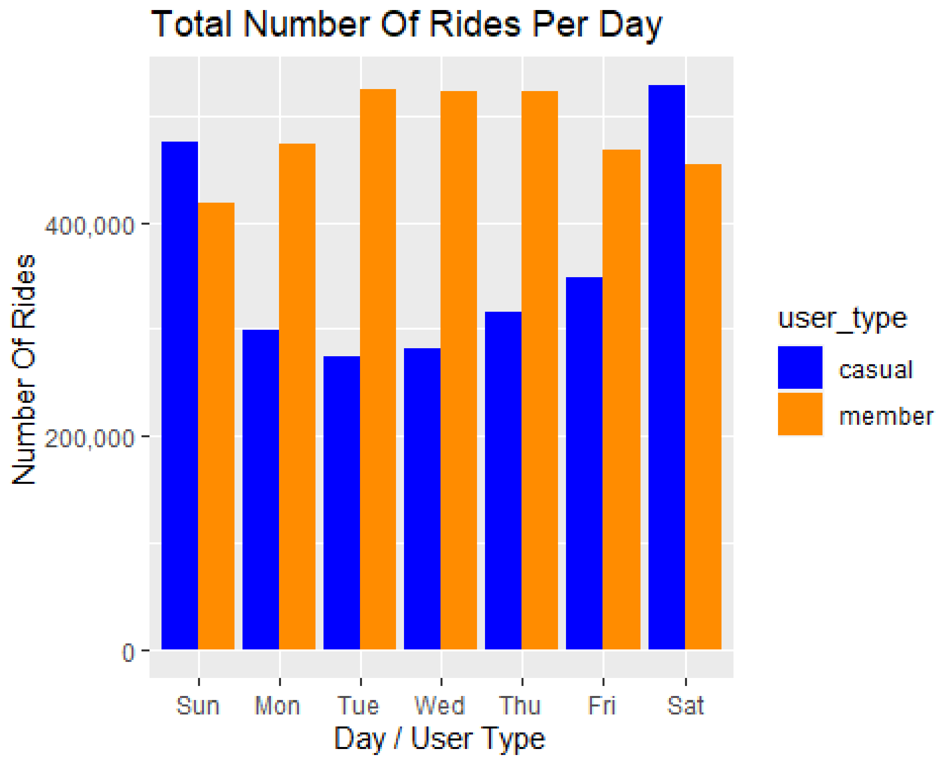
```
#calculating average  time of rides for each day
avg_day <- aggregate(data$ride_length ~ data$member_casual +
data$day_of_week, FUN = mean)
#calculating average  time of rides for each month
avg_month <- aggregate(data$ride_length ~ data$member_casual + data$month,
FUN = mean)
#calculating average  time of rides for each season
avg_season <- aggregate(data$ride_length ~ data$member_casual + data$season,
FUN = mean)
#calculating average  time of rides for eachride type
avg_rideable_type <- aggregate(data$ride_length ~ data$rideable_type +
data$member_casual, FUN = mean)

# analyze ridership data by type and weekday
data %>%
  group_by(member_casual, day_of_week) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n()                           #calculates the
number of rides and average duration
            ,average_duration = mean(ride_length)) %>%      # calculates the
average duration
  arrange(member_casual, day_of_week)   # sorts
```

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

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual day_of_week number_of_rides average_duration
##    <chr>         <ord>                 <int>            <dbl>
##  1 casual        Sunday               475591             34.0
##  2 casual        Monday               299653             29.7
##  3 casual        Tuesday              273810             25.5
##  4 casual        Wednesday            281783             25.0
##  5 casual        Thursday             316118             26.2
##  6 casual        Friday               347637             27.4
##  7 casual        Saturday             527568             31.8
##  8 member        Sunday               417953             14.6
##  9 member        Monday               472387             12.6
## 10 member        Tuesday              523377             12.1
## 11 member        Wednesday            522617             12.2
## 12 member        Thursday             522658             12.4
## 13 member        Friday               466676             12.6
## 14 member        Saturday             453486             14.5
```

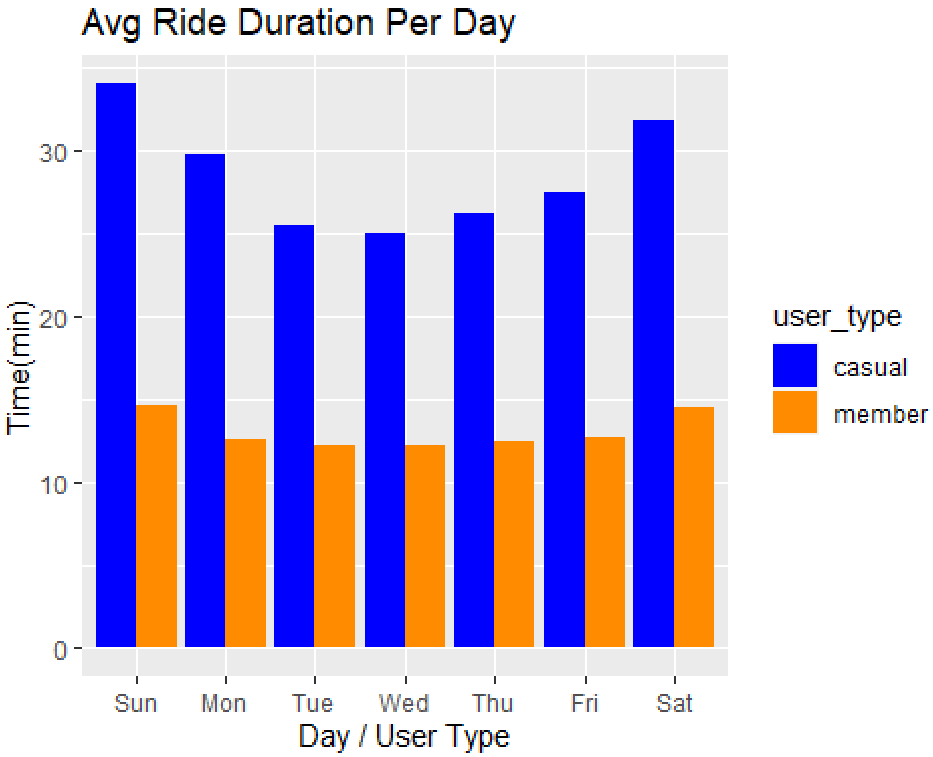
Comments: Analyzing the data

```
data %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  rename(user_type = member_casual) %>%
  group_by(user_type, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(user_type, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = user_type)) +
  geom_col(position = "dodge") +
  scale_fill_manual(values = c("#0000ff", "#ff8c00")) +
  scale_y_continuous(labels = comma) +
  labs(title = "Total Number Of Rides Per Day" , x = "Day / User Type",
       y = "Number Of Rides")

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



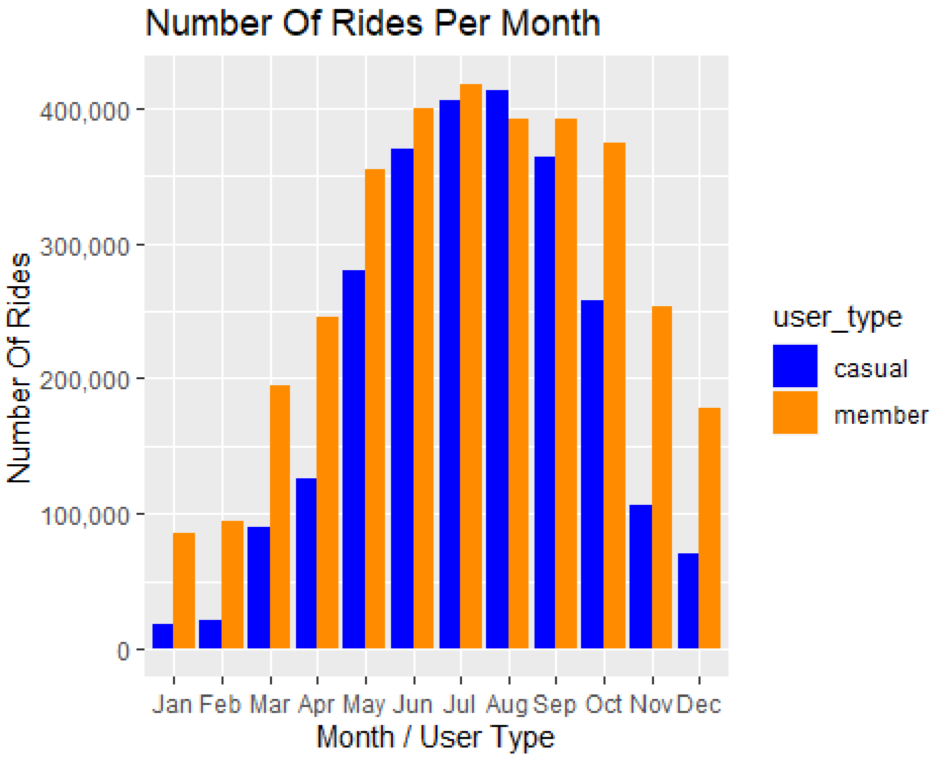
Comments: Visualize the number of rides per day

```
data %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  rename(user_type = member_casual) %>%
  group_by(user_type, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(user_type, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = user_type)) +
  geom_col(position = "dodge") +
  scale_fill_manual(values = c("#0000ff", "#ff8c00")) +
  labs(title = "Avg Ride Duration Per Day" ,x = "Day / User Type",
       y = "Time(min)")

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



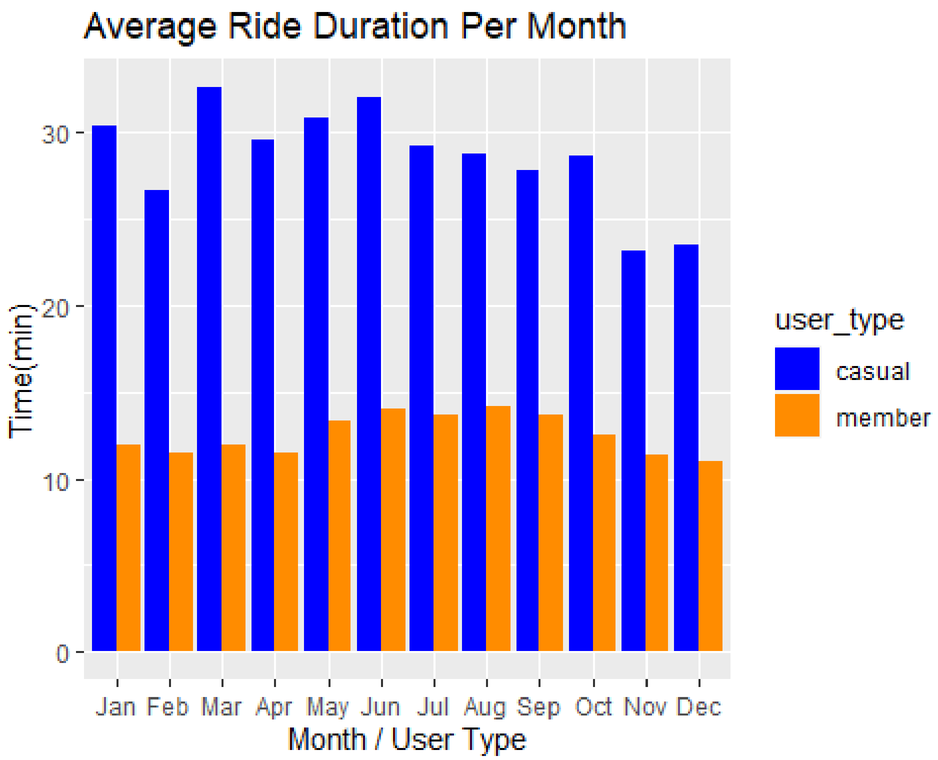
Comments: Creating a visualization for average duration per day

```
data %>%
  mutate(months= month(started_at, label = TRUE)) %>%
  rename(user_type = member_casual) %>%
  group_by(months, user_type) %>%
  summarise(number_of_rides = n()) %>%
  arrange(user_type, months)  %>%
  ggplot(aes(x = months, y = number_of_rides, fill = user_type)) +
  geom_col(position = "dodge") +
  scale_fill_manual(values = c("#0000ff", "#ff8c00")) +
  scale_y_continuous(labels = comma) +
 labs(title = "Number Of Rides Per Month", x = "Month / User Type",
      y = "Number Of Rides")

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



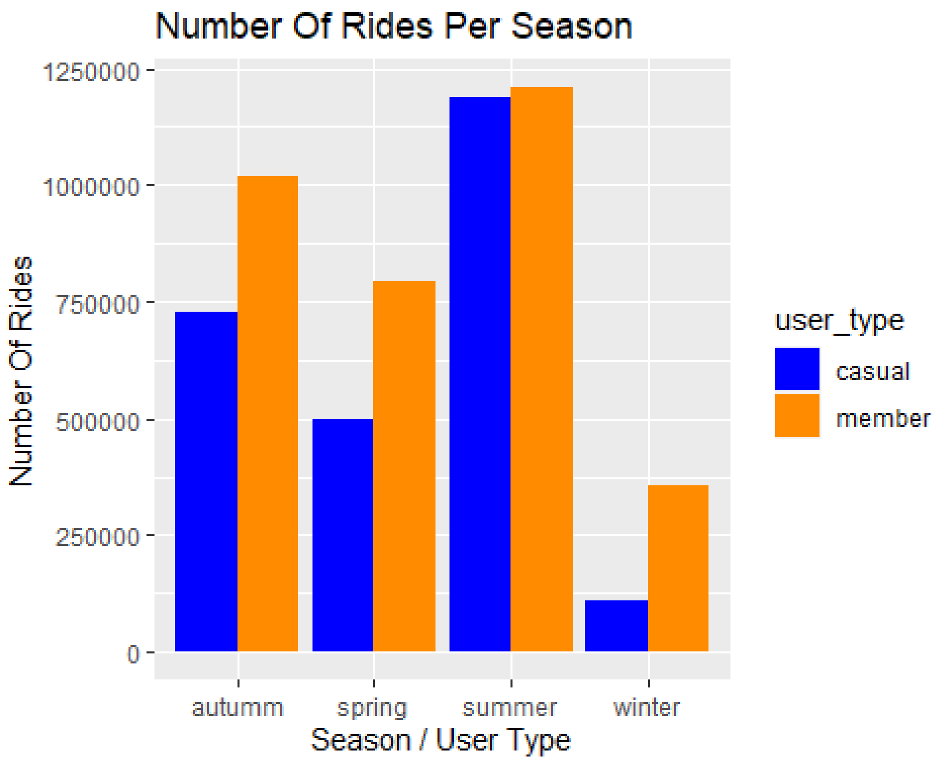
Comments: Creating a visualization for number of rides per month

```
data %>%
  mutate(months= month(started_at, label = TRUE )) %>%
  rename(user_type = member_casual) %>%
  group_by(months, user_type) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(user_type, months)  %>%
  ggplot(aes(x = months, y = average_duration, fill = user_type)) +
  geom_col(position = "dodge") +
  scale_fill_manual(values = c("#0000ff", "#ff8c00")) +
  labs(title = "Average Ride Duration Per Month", x = "Month / User Type",
       y = "Time(min)")

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



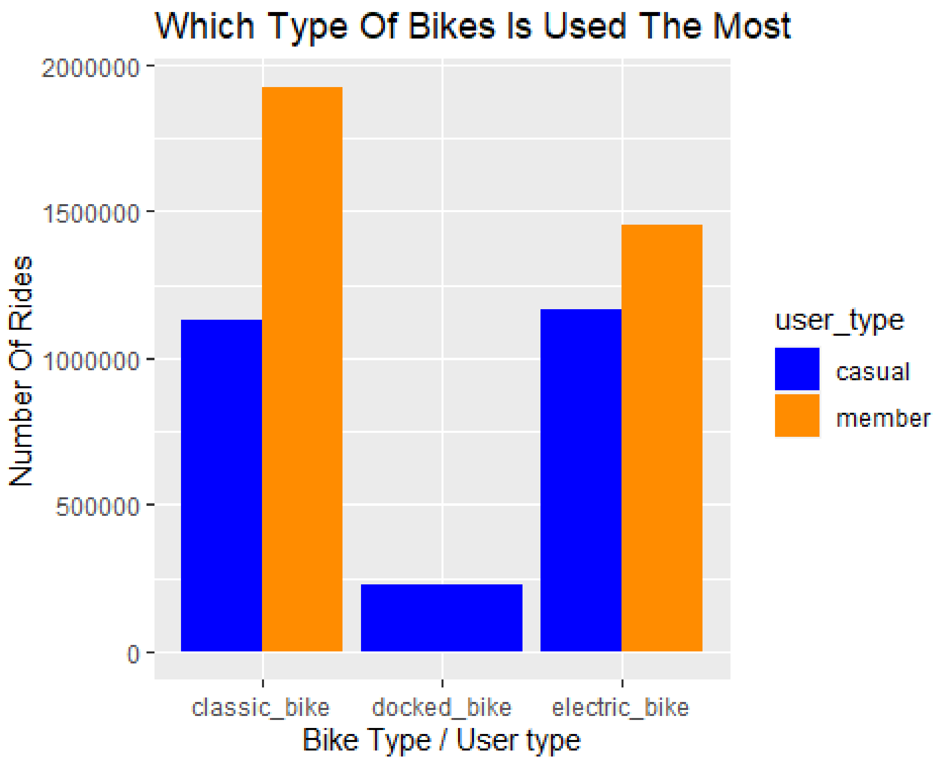
Comments: Creating a visualization for average duration per month

```
data %>%
  rename(user_type = member_casual) %>%
  group_by(season,user_type) %>%
  summarise(number_of_rides = n()) %>%
  arrange(user_type,season) %>%
  ggplot(aes(x = season , y = number_of_rides , fill = user_type)) +
  geom_col(position = "dodge") +
  scale_fill_manual(values = c("#0000ff", "#ff8c00")) +
  labs(title = "Number Of Rides Per Season" , x = "Season / User Type"
       , y = "Number Of Rides")

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



Comments: Creating a visualization for number of rides per season

```
data %>%
  rename(user_type = member_casual) %>%
  group_by(rideable_type,user_type) %>%
  summarise(number_of_rides = n()) %>%
  arrange(user_type,rideable_type) %>%
  ggplot(aes(x = rideable_type , y = number_of_rides , fill = user_type )) +
  geom_col(position = "dodge") +
  scale_fill_manual(values = c("#0000ff", "#ff8c00")) +
  labs(title = "Which Type Of Bikes Is Used The Most" , x = "Bike Type / User
type"
       , y = "Number Of Rides")

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



Comments: Creating a visualization for which type of bikes is used the most

```
write.csv(data , file =
"C:/Google_Capstone_Project/Exported_data/Cyclistic_bike_share_cleaned.csv")
write.csv(avg_season , file =
"C:/Google_Capstone_Project/Exported_data/ride_season_avg_length.csv")
write.csv(avg_rideable_type , file =
"C:/Google_Capstone_Project/Exported_data/rideable_type_avg_length.csv")
write.csv(avg_month , file =
"C:/Google_Capstone_Project/Exported_data/ride_month_avg_length.csv")
write.csv(avg_day, file =
"C:/Google_Capstone_Project/Exported_data/ride_day_avg_length.csv")
write.csv(num_of_rides_season, file =
"C:/Google_Capstone_Project/Exported_data/ride_season_total_length.csv")
write.csv(num_of_rides_month, file =
"C:/Google_Capstone_Project/Exported_data/ride_month_total_length.csv")
write.csv(num_of_rides_day, file =
"C:/Google_Capstone_Project/Exported_data/ride_day_total_length.csv")
write.csv(num_of_rideable_type, file
="C:/Google_Capstone_Project/Exported_data/ride_type_total_length.csv")
```

Comments : Export the data for data vz or future analysis