Manipulation, Cleaning, Exploration, Analysis and Visualization of Bikeshare Trip Data

Lamerck Kavuma

2023-12-28

OVERVIEW **Objective:**

Identifying differences in bike usage by annual member riders and casual riders

Findings:

1. Member riders prefer shorter rides over long rides while casual riders prefer long rides over short rides
2. Rider count peaks in the summer months and is lowest in the winter months with member riders predominant in these winter months
3. Casual riders dominate bike usage on weekend days while member riders dominate bike usage in the week days
4. Docked bikes are only used by by casual riders

Techniques Employed:

- Data Acquisition
- Data Manipulation
- Data Cleaning
- Data Exploration
- Data Analysis
- Data Visualization

DATA SOURCE This is a project undertaken at the end of the Google Data Analytics Professional Certificate Course and through Coursera, the data was made available by Motivate International Inc under this License from Lyft Bikes and Scooters, LLC.

```r
library(tidyverse)
```

SETTING UP MY R ENVIRONMENT

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## 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(readr)
library(dplyr)
library(lubridate)
library(ggplot2)
```

IMPORTING THE DATA TO BE USED INTO R The data was downloaded from this website in separate files, each corresponding to a month from January to December 2022. These files were then imported into R for analysis and exploration.

```
tripdata01 <- read.csv("~/Cycling trip datasets/tripdata01.csv")
tripdata02 <- read.csv("~/Cycling trip datasets/tripdata02.csv")
tripdata03 <- read.csv("~/Cycling trip datasets/tripdata03.csv")
tripdata04 <- read.csv("~/Cycling trip datasets/tripdata04.csv")
tripdata05 <- read.csv("~/Cycling trip datasets/tripdata05.csv")
tripdata06 <- read.csv("~/Cycling trip datasets/tripdata06.csv")
tripdata07 <- read.csv("~/Cycling trip datasets/tripdata07.csv")
tripdata08 <- read.csv("~/Cycling trip datasets/tripdata08.csv")
tripdata09 <- read.csv("~/Cycling trip datasets/tripdata09.csv")
tripdata10 <- read.csv("~/Cycling trip datasets/tripdata10.csv")
tripdata11 <- read.csv("~/Cycling trip datasets/tripdata11.csv")
tripdata12 <- read.csv("~/Cycling trip datasets/tripdata12.csv")
```

Because the datasets have the same variables in 13 columns, they are united into one data

Combining the Datasets

```
tripdatav1 <- rbind(tripdata01, tripdata02, tripdata03, tripdata04, tripdata05, tripdata06, tripdata07,
```

The new dataset is then saved into my working directory

Saving the Combined Dataset

```
saveRDS(tripdatav1, file = "tripdatav1.rds")
```

Reviewing the structure of the new dataset

```
str(tripdatav1)
```

```
## 'data.frame':    5667717 obs. of  13 variables:
##  $ ride_id           : chr  "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED4191054
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2022-01-13 11:59:47" "2022-01-10 08:41:56" "2022-01-25 04:53:40" "2022-0
##  $ ended_at          : chr  "2022-01-13 12:02:44" "2022-01-10 08:46:17" "2022-01-25 04:58:01" "2022-0
##  $ start_station_name: chr  "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave & Fu
##  $ start_station_id  : chr  "525" "525" "TA1306000016" "KA1504000151" ...
##  $ end_station_name  : chr  "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenview Ave & Fullerton
##  $ end_station_id    : chr  "RP-007" "RP-007" "TA1307000001" "TA1309000021" ...
##  $ start_lat         : num  42 42 41.9 42 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num  42 42 41.9 42 41.9 ...
##  $ end_lng           : num  -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr  "casual" "casual" "member" "casual" ...
```

DATA MANIPULATION AND CLEANING

Adding Columns

Adding 7 new columns with data to be used in the data exploration and analysis.

These include;

- ride_length which is the duration of the ride obtained by calculating the differenc between started_at and ended_at
- ride_length_group which are 20 groups in which rides are placed according to the ride_length with Set 1 having rides of shortest ride lengths.
- starting_month which is the month in which the trip started
- starting_date which is a date-only extract from the started_at date time varaibles.
- starting_hour which is the hour the trip started
- day_of_week which is the weekly day the trip started
- route which is a term used to identify specific routes that were used, obtained through combining the prefixes of start and end_station_names

```r
tripdatav2 <- tripdatav1 %>%
  mutate(
    ride_length = as.numeric(difftime(ended_at, started_at, units = "secs")),
    ride_length_group = ntile(ride_length, 20),
    starting_month = month(started_at, label = TRUE),
    starting_date = as.Date(started_at),
    starting_hour = hour(started_at),
    day_of_week = wday(started_at, label = TRUE),
    route = paste(substr(start_station_name, 1, 3), substr(end_station_name, 1, 3), sep = "")
  )

str(tripdatav2)
```

```
## 'data.frame':    5667717 obs. of  20 variables:
##  $ ride_id           : chr  "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED4191054
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2022-01-13 11:59:47" "2022-01-10 08:41:56" "2022-01-25 04:53:40" "2022-0
##  $ ended_at          : chr  "2022-01-13 12:02:44" "2022-01-10 08:46:17" "2022-01-25 04:58:01" "2022-0
##  $ start_station_name: chr  "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Sheffield Ave & Fu
##  $ start_station_id  : chr  "525" "525" "TA1306000016" "KA1504000151" ...
##  $ end_station_name  : chr  "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenview Ave & Fullerton
##  $ end_station_id    : chr  "RP-007" "RP-007" "TA1307000001" "TA1309000021" ...
##  $ start_lat         : num  42 42 41.9 42 41.9 ...
##  $ start_lng         : num  -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num  42 42 41.9 42 41.9 ...
##  $ end_lng           : num  -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr  "casual" "casual" "member" "casual" ...
##  $ ride_length       : num  177 261 261 896 362 ...
##  $ ride_length_group : int  2 4 4 14 6 2 15 12 17 7 ...
##  $ starting_month    : Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ starting_date     : Date, format: "2022-01-13" "2022-01-10" ...
##  $ starting_hour     : int  11 8 4 0 1 18 18 12 7 15 ...
##  $ day_of_week       : Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<..: 5 2 3 3 5 3 1 7 2 6 ...
##  $ route             : chr  "GleCla" "GleCla" "SheGre" "ClaPau" ...
```

```r
head(tripdatav2)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 C2F7DD78E82EC875 electric_bike 2022-01-13 11:59:47 2022-01-13 12:02:44
## 2 A6CF8980A652D272 electric_bike 2022-01-10 08:41:56 2022-01-10 08:46:17
## 3 BD0F91DFF741C66D  classic_bike 2022-01-25 04:53:40 2022-01-25 04:58:01
## 4 CBB80ED419105406  classic_bike 2022-01-04 00:18:04 2022-01-04 00:33:00
## 5 DDC963BFDDA51EEA  classic_bike 2022-01-20 01:31:10 2022-01-20 01:37:12
## 6 A39C6F6CC0586C0B  classic_bike 2022-01-11 18:48:09 2022-01-11 18:51:31
##              start_station_name start_station_id              end_station_name
## 1      Glenwood Ave & Touhy Ave              525          Clark St & Touhy Ave
## 2      Glenwood Ave & Touhy Ave              525          Clark St & Touhy Ave
## 3 Sheffield Ave & Fullerton Ave     TA1306000016 Greenview Ave & Fullerton Ave
## 4      Clark St & Bryn Mawr Ave     KA1504000151     Paulina St & Montrose Ave
## 5   Michigan Ave & Jackson Blvd     TA1309000002        State St & Randolph St
## 6         Wood St & Chicago Ave              637       Honore St & Division St
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1         RP-007  42.01280 -87.66591 42.01256 -87.67437        casual
## 2         RP-007  42.01276 -87.66597 42.01256 -87.67437        casual
## 3   TA1307000001  41.92560 -87.65371 41.92533 -87.66580        member
## 4   TA1309000021  41.98359 -87.66915 41.96151 -87.67139        casual
## 5   TA1305000029  41.87785 -87.62408 41.88462 -87.62783        member
## 6   TA1305000034  41.89563 -87.67207 41.90312 -87.67394        member
##   ride_length ride_length_group starting_month starting_date starting_hour
## 1         177                 2            Jan    2022-01-13            11
## 2         261                 4            Jan    2022-01-10             8
## 3         261                 4            Jan    2022-01-25             4
## 4         896                14            Jan    2022-01-04             0
## 5         362                 6            Jan    2022-01-20             1
## 6         202                 2            Jan    2022-01-11            18
##   day_of_week  route
## 1         Thu GleCla
## 2         Mon GleCla
## 3         Tue SheGre
## 4         Tue ClaPau
## 5         Thu MicSta
## 6         Tue WooHon
```

Removing Columns

These columns include;

- start_station_name
- end_station_name
- start_station_id
- end_station_id
- start_lat
- start_lng
- end_lat
- end_lng

```
tripdatav2_clean <-tripdatav2 %>%
  select(ride_id, rideable_type, started_at, ended_at, member_casual, ride_length, ride_length_group, s
str(tripdatav2_clean)
```

```
## ’data.frame':    5667717 obs. of  12 variables:
##  $ ride_id          : chr  "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66D" "CBB80ED41910540
##  $ rideable_type    : chr  "electric_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at       : chr  "2022-01-13 11:59:47" "2022-01-10 08:41:56" "2022-01-25 04:53:40" "2022-0
##  $ ended_at         : chr  "2022-01-13 12:02:44" "2022-01-10 08:46:17" "2022-01-25 04:58:01" "2022-0
##  $ member_casual    : chr  "casual" "casual" "member" "casual" ...
##  $ ride_length      : num  177 261 261 896 362 ...
##  $ ride_length_group: int  2 4 4 14 6 2 15 12 17 7 ...
##  $ starting_month   : Ord.factor w/ 12 levels "Jan"<"Feb"<"Mar"<..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ starting_date    : Date, format: "2022-01-13" "2022-01-10" ...
##  $ starting_hour    : int  11 8 4 0 1 18 18 12 7 15 ...
##  $ day_of_week      : Ord.factor w/ 7 levels "Sun"<"Mon"<"Tue"<..: 5 2 3 3 5 3 1 7 2 6 ...
##  $ route            : chr  "GleCla" "GleCla" "SheGre" "ClaPau" ...
```

```
summary(tripdatav2_clean)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:5667717     Length:5667717     Length:5667717     Length:5667717
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  member_casual       ride_length      ride_length_group starting_month
##  Length:5667717     Min.   :-621201   Min.   : 1.0      Jul    : 823488
##  Class :character   1st Qu.:    349   1st Qu.: 5.0      Aug    : 785932
##  Mode  :character   Median :    617   Median :10.0      Jun    : 769204
##                     Mean   :   1167   Mean   :10.5      Sep    : 701339
##                     3rd Qu.:   1108   3rd Qu.:15.0      May    : 634858
##                     Max.   :2483235   Max.   :20.0      Oct    : 558685
##                                                         (Other):1394211
##  starting_date         starting_hour    day_of_week     route
##  Min.   :2022-01-01   Min.   : 0.00   Sun:776259   Length:5667717  
##  1st Qu.:2022-05-28   1st Qu.:11.00   Mon:751014   Class :character  
##  Median :2022-07-22   Median :15.00   Tue:782372   Mode  :character  
##  Mean   :2022-07-19   Mean   :14.22   Wed:798223                     
##  3rd Qu.:2022-09-16   3rd Qu.:18.00   Thu:841591                     
##  Max.   :2022-12-31   Max.   :23.00   Fri:801787                     
##                                       Sat:916471
```

The minimum value in the ride_length column is a negative which should not be the case. Reviewing the data to identify the potential cause(s) of this problem

```
tripdatav2_sort <- arrange(tripdatav2_clean, ride_length)
negative_ride_lengths <- filter(tripdatav2_sort, ride_length < 0)
nrow(negative_ride_lengths)
```

```
## [1] 100
```

```
head(negative_ride_lengths)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 E137518FFE807752 electric_bike 2022-09-28 11:04:32 2022-09-21 06:31:11
## 2 918F745F62CAC29E  classic_bike 2022-10-13 14:42:10 2022-10-13 11:53:28
## 3 38B9F148CE80499B electric_bike 2022-06-07 19:23:03 2022-06-07 17:05:38
## 4 B897BE02B21FA75E electric_bike 2022-06-07 19:15:39 2022-06-07 17:05:37
## 5 BF114472ABA0289C electric_bike 2022-06-07 19:14:47 2022-06-07 17:05:42
## 6 072E947E156D142D electric_bike 2022-06-07 19:14:46 2022-06-07 17:07:45
##   member_casual ride_length ride_length_group starting_month starting_date
## 1        member     -621201                 1            Sep    2022-09-28
## 2        member      -10122                 1            Oct    2022-10-13
## 3        casual       -8245                 1            Jun    2022-06-07
## 4        casual       -7802                 1            Jun    2022-06-07
## 5        member       -7745                 1            Jun    2022-06-07
## 6        casual       -7621                 1            Jun    2022-06-07
##   starting_hour day_of_week  route
## 1            11         Wed    Cor
## 2            14         Thu WilWil
## 3            19         Tue       
## 4            19         Tue    Kos
## 5            19         Tue BasW A
## 6            19         Tue W AW A
```

Deleting Rows

There are 100 observations with a negative ride length and in all cases, starting_date is greater than ending_date. These rows were dropped.

```
tripdatav3 <- tripdatav2_clean[tripdatav2_clean$ride_length >= 0, ]
```

Saving the clean Dataset

Saving the clean dataset in the working directory.

```r
saveRDS(tripdatav3, file = "tripdatav3.rds")
```

DATA EXPLORATION Calculating the **total number of rides** in 2022 followed by total number of member rides and then total number of casual rides

```r
nrow(tripdatav3)
```

```
## [1] 5667617
```

```r
member_rides <- filter(tripdatav3, member_casual == 'member')
nrow(member_rides)
```

```
## [1] 3345640
```

```r
casual_rides <- filter(tripdatav3, member_casual == 'casual')
nrow(casual_rides)
```

```
## [1] 2321977
```

Calculating the **average number of rides per day** for the entire year

```r
total_rides <- nrow(tripdatav3)
total_days <- n_distinct(tripdatav3$starting_date)
average_daily_rides <- total_rides/total_days
```

Calculating the general **average ride length** followed the average ride length for member rides and then average ride length for casual rides

```r
total_length <- sum(tripdatav3$ride_length)
avg_ride_length <- total_length/total_rides
cat("The general average ride length is:", avg_ride_length, "seconds\n")
```

```
## The general average ride length is: 1166.757 seconds
```

```r
rides_by_members <- nrow(member_rides)
member_total_length <- sum(filter(tripdatav3, member_casual == 'member')$ride_length)
avg_member_length <- member_total_length/rides_by_members
cat("The average ride length for members is:", avg_member_length, "seconds\n")
```

```
## The average ride length for members is: 762.8406 seconds
```

```r
rides_by_casuals <- nrow(casual_rides)
casual_total_length <- sum(filter(tripdatav3, member_casual == 'casual')$ride_length)
avg_casual_length <- casual_total_length/rides_by_casuals
cat("The average ride length for casuals is:", avg_casual_length, "seconds\n")
```

```
## The average ride length for casuals is: 1748.743 seconds
```

What is the most **popular route** in general, then for members and then for casuals?

```
mode_route <- sort(-table(tripdatav3$route))
head(mode_route)
```

```
##
##          ClaCla     Cla     She  SheShe     Bro
## -427441  -68866  -67235  -38587  -33419  -32296
```

```
mode_route_members <- sort(-table(filter(tripdatav3, member_casual == 'member')$route))
head(mode_route_members)
```

```
##
##          ClaCla     Cla     She  EllEll  SheShe
## -234991  -39883  -39855  -21935  -20650  -19148
```

```
mode_route_casuals <- sort(-table(filter(tripdatav3, member_casual == 'casual')$route))
head(mode_route_casuals)
```

```
##
##          ClaCla     Cla  MicMic  DuSDuS     She
## -192450  -28983  -27380  -19506  -18417  -16652
```

What is the most **popular day of the week** in general, then for members and then for casuals?

```
popular_days <- sort(-table(tripdatav3$day_of_week))
head(popular_days)
```

```
##
##     Sat     Thu     Fri     Wed     Tue     Sun
## -916459 -841582 -801781 -798221 -782349 -776219
```

```
members_popular_days <- sort(-table(filter(tripdatav3, member_casual == 'member')$day_of_week))
tibble(members_popular_days)
```

```
## # A tibble: 7 x 1
##   members_popular_days
##   <table[1d]>
## 1 -532255
## 2 -523867
## 3 -518618
## 4 -473335
## 5 -467083
## 6 -443274
## 7 -387208
```

```
casuals_popular_days <- sort(-table(filter(tripdatav3, member_casual == 'casual')$day_of_week))
tibble(casuals_popular_days)
```

```
## # A tibble: 7 x 1
##   casuals_popular_days
##   <table[1d]>
## 1 -473185
## 2 -389011
## 3 -334698
## 4 -309327
## 5 -277671
## 6 -274354
## 7 -263731
```

What is the most **popular hour of the day** in general, then for members and then for casuals?

```
popular_hours <- sort(-table(tripdatav3$starting_hour))
head(popular_hours)
```

```
##
##      17      16      18      15      19      14
## -569587 -489489 -482170 -399775 -357728 -344964
```

```
members_popular_hours <- sort(-table(filter(tripdatav3, member_casual == 'member')$starting_hour))
head(members_popular_hours)
```

```
##
##      17      16      18      15      19       8
## -349432 -291777 -284618 -221566 -206349 -204534
```

```
casuals_popular_hours <- sort(-table(filter(tripdatav3, member_casual == 'casual')$starting_hour))
head(casuals_popular_hours)
```

```
##
##      17      16      18      15      14      19
## -220155 -197712 -197552 -178209 -159956 -151379
```

What is the most **popular month** in general, then for members and then for casuals?

```
popular_months <- sort(-table(tripdatav3$starting_month))
head(popular_months)
```

```
##
##     Jul     Aug     Jun     Sep     May     Oct
## -823472 -785917 -769192 -701330 -634857 -558681
```

```
members_popular_months <- sort(-table(filter(tripdatav3, member_casual == 'member')$starting_month))
head(members_popular_months)
```

```
##
##     Aug     Jul     Sep     Jun     May     Oct
## -427000 -417426 -404636 -400148 -354443 -349693
```

```
casuals_popular_months <- sort(-table(filter(tripdatav3, member_casual == 'casual')$starting_month))
head(casuals_popular_months)
```

```
##
##     Jul     Jun     Aug     Sep     May     Oct
## -406046 -369044 -358917 -296694 -280414 -208988
```

Which **bicycle type** is most commonly used?

```
bicycle_type_freq <- sort(-table(tripdatav3$rideable_type))
head(bicycle_type_freq)
```

```
##
## electric_bike  classic_bike   docked_bike
##      -2888957      -2601186       -177474
```

How many member riders use these different bikes?

Note: *eb* for *Electric Bikes*, *cb* for *Classic Bikes*, and *db* for *Docked Bikes*

```
members_eb_freq <- filter(filter(tripdatav3, rideable_type == 'electric_bike'), member_casual == 'membe
nrow(members_eb_freq)
```

```
## [1] 1635897
```

```
members_cb_freq <- filter(filter(tripdatav3, rideable_type == 'classic_bike'), member_casual == 'member
nrow(members_cb_freq)
```

```
## [1] 1709743
```

```
members_db_freq <- filter(filter(tripdatav3, rideable_type == 'docked_bike'), member_casual == 'member'
nrow(members_db_freq)
```

```
## [1] 0
```

Then for casual riders

```
casuals_eb_freq <- filter(filter(tripdatav3, rideable_type == 'electric_bike'), member_casual == 'casua
nrow(casuals_eb_freq)
```

```
## [1] 1253060
```

```
casuals_cb_freq <- filter(filter(tripdatav3, rideable_type == 'classic_bike'), member_casual == 'casual
nrow(casuals_cb_freq)
```

```
## [1] 891443
```

```
casuals_db_freq <- filter(filter(tripdatav3, rideable_type == 'docked_bike'), member_casual == 'casual')
nrow(casuals_db_freq)
```

```
## [1] 177474
```

What ride type is more likely to use **one off routes**?

Note: One off routes are the routes that where used once only

```
unique_routes_per_type <- tripdatav3 %>%
  group_by(member_casual) %>%
  summarise(unique_routes = n_distinct(route))

total_rides_per_type <- tripdatav3 %>%
  count(member_casual)

unique_route_proportion <- unique_routes_per_type %>%
  inner_join(total_rides_per_type, by = "member_casual") %>%
  mutate(proportion_one_off = (unique_routes / n) * 100)
print(unique_route_proportion)
```

```
## # A tibble: 2 x 4
##   member_casual unique_routes       n proportion_one_off
##   <chr>                 <int>   <int>              <dbl>
## 1 casual                20127 2321977              0.867
## 2 member                19928 3345640              0.596
```

On which day of the week are casual riders most likely to use the different bike types?

```
rideable_type_per_day <- tripdatav3 %>%
  filter(member_casual == 'casual') %>%
  group_by (rideable_type, day_of_week) %>%
  summarise(day_count = n())
```

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

```
print(rideable_type_per_day)
```

```
## # A tibble: 21 x 3
## # Groups:   rideable_type [3]
##    rideable_type day_of_week day_count
##    <chr>         <ord>           <int>
##  1 classic_bike  Sun            158573
##  2 classic_bike  Mon            104257
##  3 classic_bike  Tue             96119
##  4 classic_bike  Wed             98363
##  5 classic_bike  Thu            113837
##  6 classic_bike  Fri            123125
##  7 classic_bike  Sat            197169
##  8 docked_bike   Sun             35729
##  9 docked_bike   Mon             22535
## 10 docked_bike   Tue             17756
## # i 11 more rows
```

```
eb_count_per_day <- rideable_type_per_day %>%
  filter(rideable_type == 'electric_bike') %>%
  arrange(desc(day_count))
print(eb_count_per_day)
```

```
## # A tibble: 7 x 3
## # Groups:   rideable_type [1]
##   rideable_type day_of_week day_count
##   <chr>         <ord>           <int>
## 1 electric_bike Sat            235058
## 2 electric_bike Sun            194709
## 3 electric_bike Fri            188186
## 4 electric_bike Thu            175716
## 5 electric_bike Wed            158656
## 6 electric_bike Mon            150879
## 7 electric_bike Tue            149856
```

```
db_count_per_day <- rideable_type_per_day %>%
  filter(rideable_type == 'docked_bike') %>%
  arrange(desc(day_count))
print(db_count_per_day)
```

```
## # A tibble: 7 x 3
## # Groups:   rideable_type [1]
##   rideable_type day_of_week day_count
##   <chr>         <ord>           <int>
## 1 docked_bike   Sat             40958
## 2 docked_bike   Sun             35729
## 3 docked_bike   Fri             23387
## 4 docked_bike   Mon             22535
## 5 docked_bike   Thu             19774
## 6 docked_bike   Tue             17756
## 7 docked_bike   Wed             17335
```

```
cb_count_per_day <- rideable_type_per_day %>%
  filter(rideable_type == 'classic_bike') %>%
  arrange(desc(day_count))
print(cb_count_per_day)
```

```
## # A tibble: 7 x 3
## # Groups:   rideable_type [1]
##   rideable_type day_of_week day_count
##   <chr>         <ord>           <int>
## 1 classic_bike  Sat            197169
## 2 classic_bike  Sun            158573
## 3 classic_bike  Fri            123125
## 4 classic_bike  Thu            113837
## 5 classic_bike  Mon            104257
## 6 classic_bike  Wed             98363
## 7 classic_bike  Tue             96119
```

In which hour are casual riders most likely to use a certain rideable type?

```
rideable_type_per_hour <- tripdatav3 %>%
  filter(member_casual == 'casual') %>%
  group_by (rideable_type, starting_hour) %>%
  summarise(hour_count = n())
```

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

```
db_count_per_hour <- rideable_type_per_hour %>%
  filter(rideable_type == 'docked_bike') %>%
  arrange(desc(hour_count))
head(db_count_per_hour)
```

```
## # A tibble: 6 x 3
## # Groups:   rideable_type [1]
##   rideable_type starting_hour hour_count
##   <chr>                 <int>      <int>
## 1 docked_bike              15      16296
## 2 docked_bike              16      16223
## 3 docked_bike              14      15832
## 4 docked_bike              17      14965
## 5 docked_bike              13      14646
## 6 docked_bike              12      13736
```

```
eb_count_per_hour <- rideable_type_per_hour %>%
  filter(rideable_type == 'electric_bike') %>%
  arrange(desc(hour_count))
head(eb_count_per_hour)
```

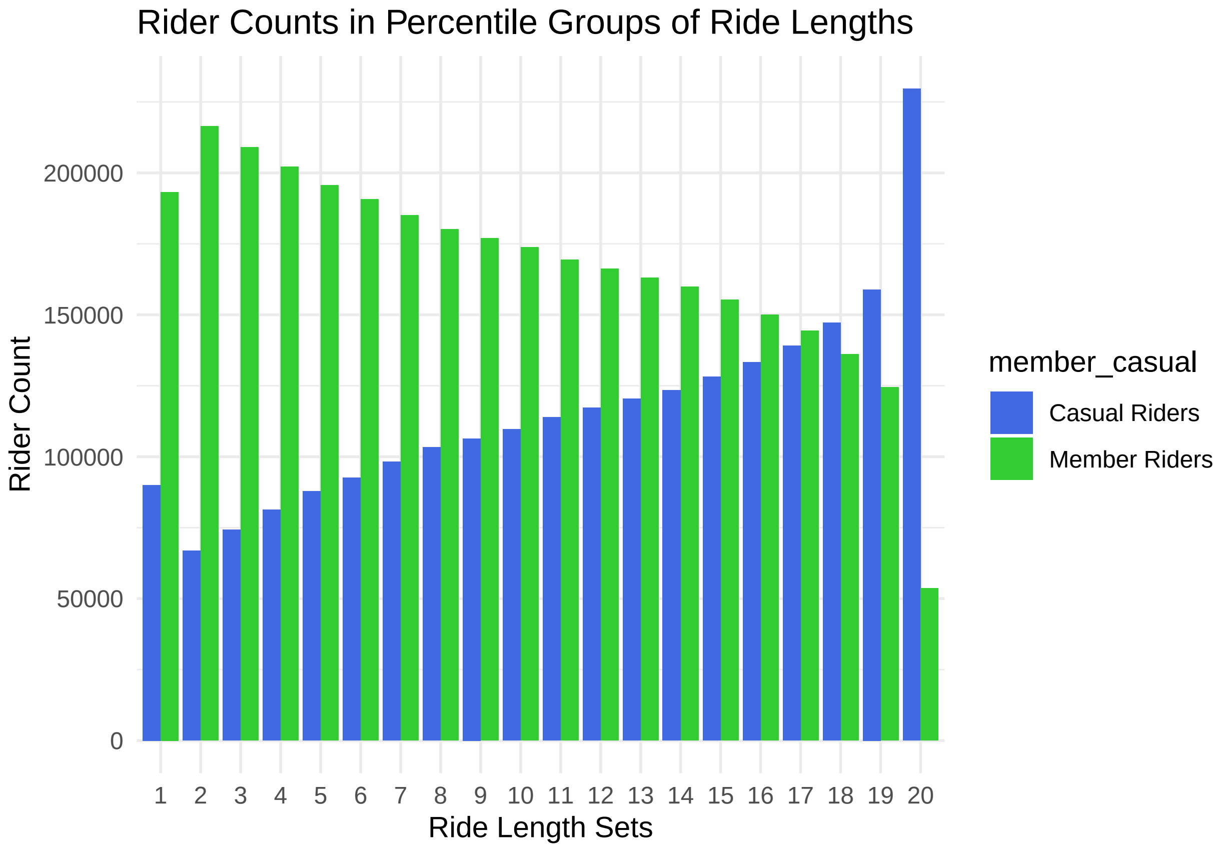
```
## # A tibble: 6 x 3
## # Groups:   rideable_type [1]
##   rideable_type starting_hour hour_count
##   <chr>                 <int>      <int>
## 1 electric_bike            17     117834
## 2 electric_bike            16     107231
## 3 electric_bike            18     102915
## 4 electric_bike            15      95024
## 5 electric_bike            14      82454
## 6 electric_bike            19      78723
```

```
cb_count_per_hour <- rideable_type_per_hour %>%
  filter(rideable_type == 'classic_bike') %>%
  arrange(desc(hour_count))
head(cb_count_per_hour)
```

```
## # A tibble: 6 x 3
## # Groups:   rideable_type [1]
##   rideable_type starting_hour hour_count
##   <chr>                 <int>      <int>
## 1 classic_bike             17      87356
## 2 classic_bike             18      81748
## 3 classic_bike              16       74258
## 4 classic_bike              15       66889
## 5 classic_bike              19       62586
## 6 classic_bike              14       61670
```

VISUALIZATION Visualizing the **relationship between annual and casual rider frequency and increasing ride length** by plotting a bar graph of rider count against ride_length_group

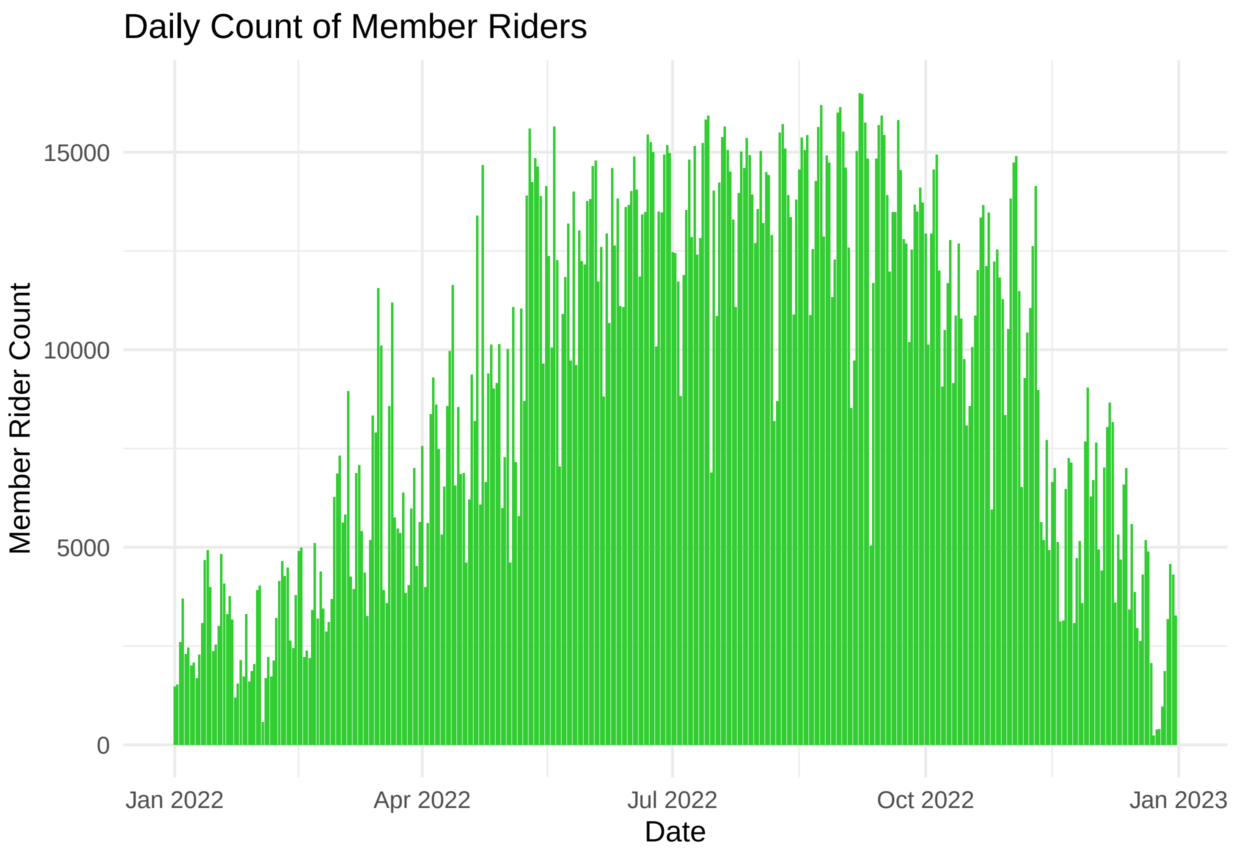
```
ggplot(tripdatav3, aes(x = as.factor(ride_length_group), fill = member_casual)) +
  geom_bar(position = "dodge") +
  labs(x = "Ride Length Sets", y = "Rider Count") +
  ggtitle("Rider Counts in Percentile Groups of Ride Lengths") +
  scale_fill_manual(values = c("royalblue", "limegreen"), labels = c("Casual Riders", "Member Riders"))
  theme_minimal()
```



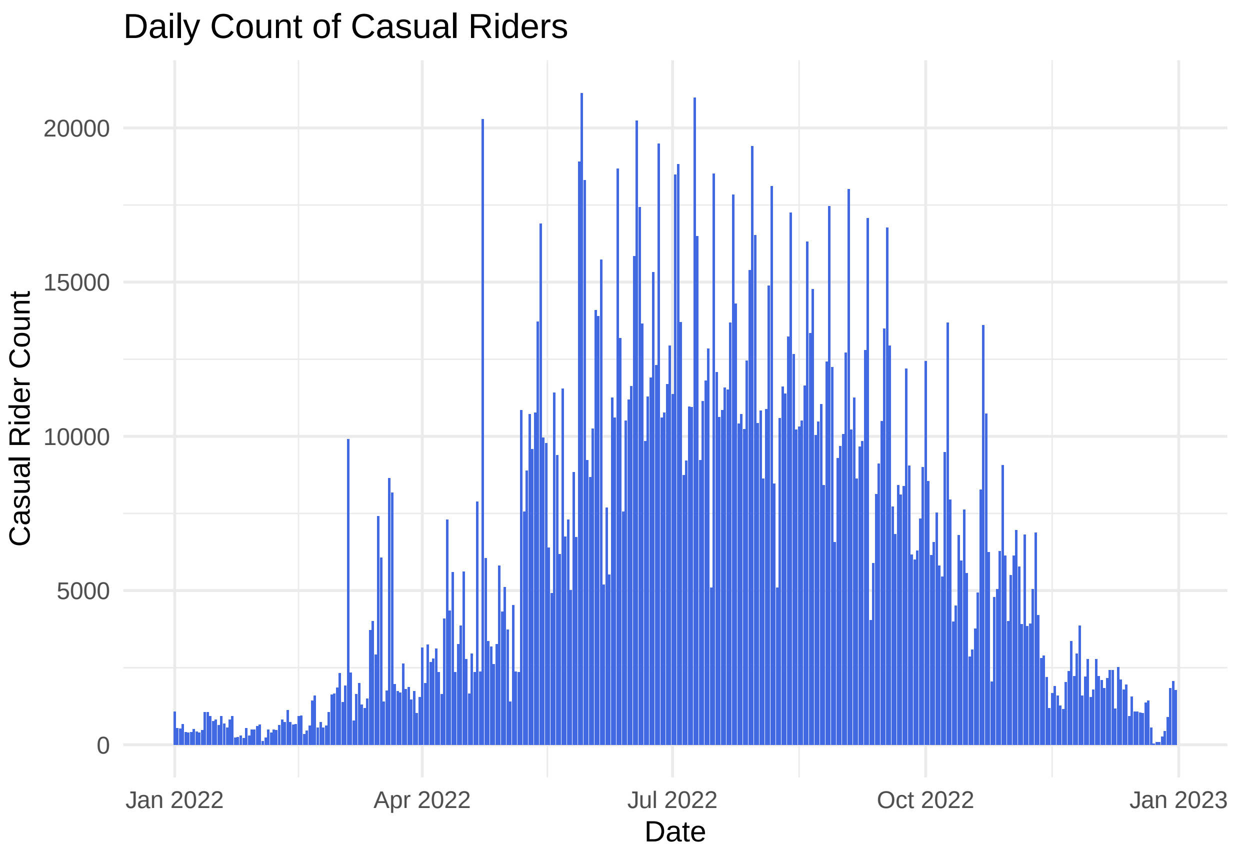
- Member riders prefer short rides over long rides while Casual riders prefer long rides over short ones

Visualizing the **trend in daily count of member and casual riders throughout the year** by plotting a bar graph of rider count per day for either each rider type

```
ggplot(subset(tripdatav3, member_casual == "member"), aes(x = starting_date)) +
  geom_bar(position = "dodge", fill = "limegreen") +
  labs(x = "Date", y = "Member Rider Count", title = "Daily Count of Member Riders") +
  theme_minimal()
```



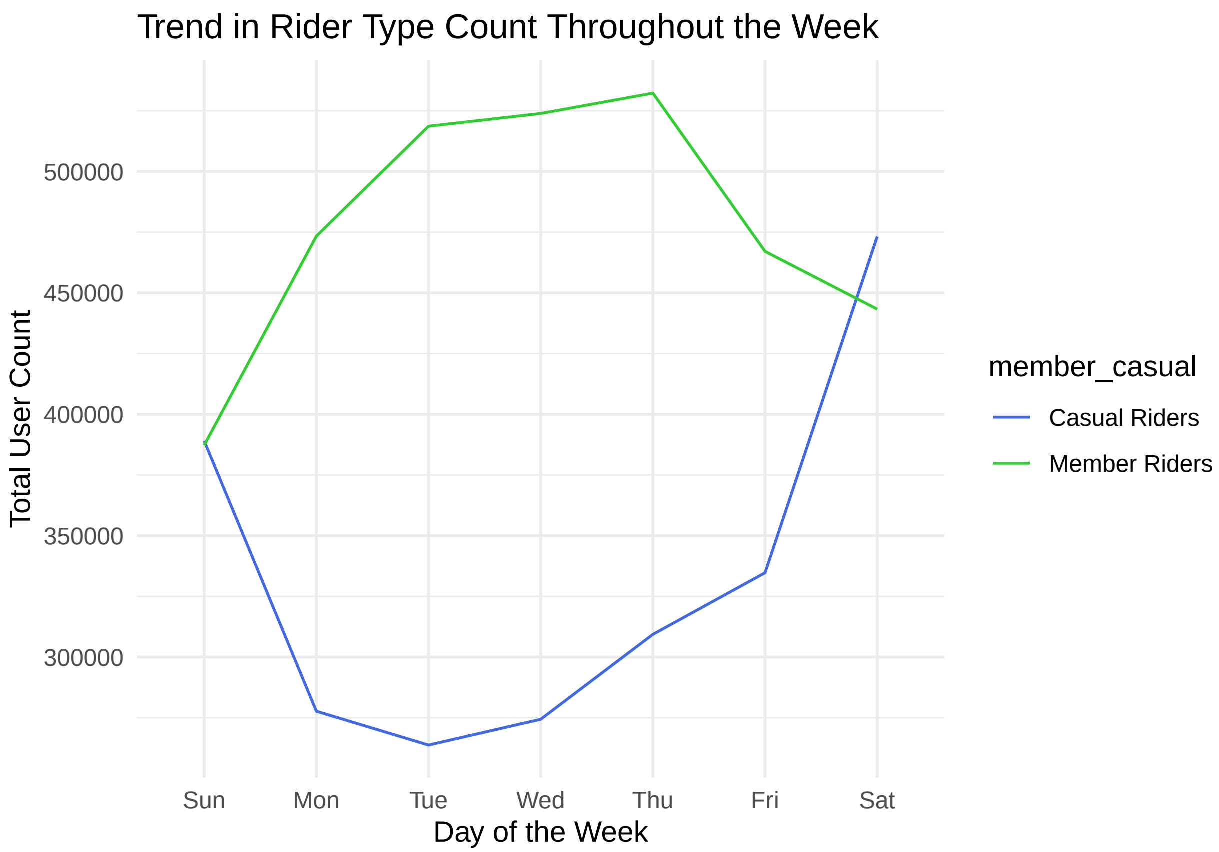
```r
ggplot(subset(tripdatav3, member_casual == "casual"), aes(x = starting_date)) +
  geom_bar(position = "dodge", fill = "royalblue") +
  labs(x = "Date", y = "Casual Rider Count", title = "Daily Count of Casual Riders") +
  theme_minimal()
```



- Bike Usage peaks in Summer

Visualizing the **trend in total rider count per type throughout the week** by plotting a line graph total users per day of the week against day of the week

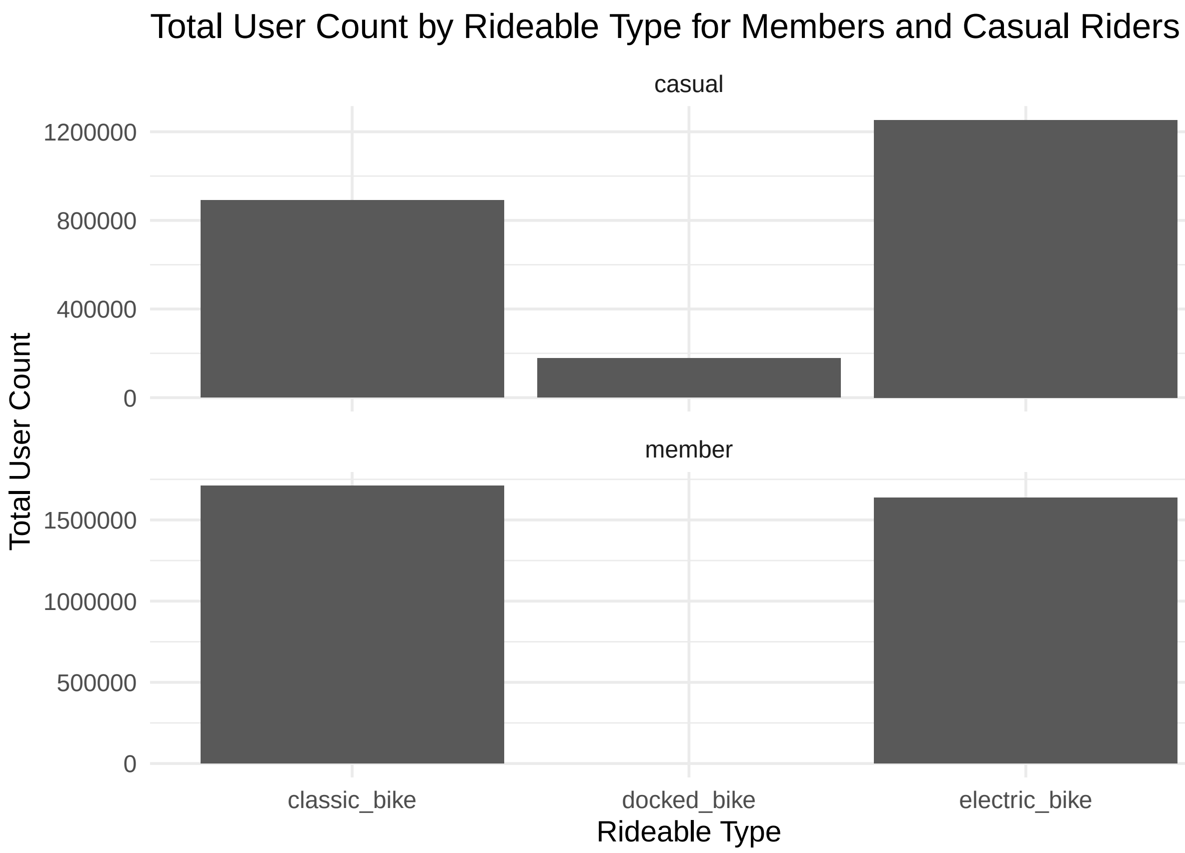
```
ggplot(tripdatav3, aes(x = day_of_week, group = member_casual, color = member_casual)) +
  geom_line(stat = "count") +
  labs(x = "Day of the Week", y = "Total User Count",
       title = "Trend in Rider Type Count Throughout the Week") +
  scale_x_discrete(labels = c("Sun", "Mon", "Tue", "Wed", "Thu", "Fri", "Sat")) +
  scale_color_manual(values = c("royalblue", "limegreen"), labels = c("Casual Riders", "Member Riders")
  theme_minimal()
```



- Casual riders are more frequent on weekends

Visualizing the **count of riders of every bike type for each rider type** by plotting a bar graph of rider count against rideable type by members and casuals.

```
ggplot(tripdatav3, aes(x = rideable_type)) +
  geom_bar() +
  labs(x = "Rideable Type", y = "Total User Count",
       title = "Total User Count by Rideable Type for Members and Casual Riders") +
  facet_wrap(~member_casual, scales = "free_y", ncol = 1) +
  theme_minimal()
```



- Docked bikes are only used by casual riders

Visit my Tableau Page to view an interactive dashboard of this data and more vizzes.