Cyclistic bike share analysis

Carlos Vasquez

09/20/2021

===================== # STEP 1: Set up my environment # =====================

Load library packages and upload the previous 12 months (from time of date,9/20/2021) divvy-tripdata sets.

```
library(tidyverse)
library(janitor)
library(lubridate)
library(scales)
```

```
q9_2020 <- read_csv("202009-divvy-tripdata.csv")
q10_2020 <- read_csv("202010-divvy-tripdata.csv")
q11_2020 <- read_csv("202011-divvy-tripdata.csv")
q12_2020 <- read_csv("202012-divvy-tripdata.csv")
q1_2021 <- read_csv("202101-divvy-tripdata.csv")
q2_2021 <- read_csv("202102-divvy-tripdata.csv")
q3_2021 <- read_csv("202103-divvy-tripdata.csv")
q4_2021 <- read_csv("202104-divvy-tripdata.csv")
q5_2021 <- read_csv("202105-divvy-tripdata.csv")
q6_2021 <- read_csv("202106-divvy-tripdata.csv")
q7_2021 <- read_csv("202107-divvy-tripdata.csv")
q8_2021 <- read_csv("202108-divvy-tripdata.csv")
```

===================== # STEP 2. Make columns consistent and merge them into a single dataframe. # =====================

Use colnames function to compare the column names of each data set

```
#Note all column names were the same but I was unable to merge.
colnames(q9_2020)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```
#colnames(q10_2020)
#colnames(q11_2020)
#colnames(q12_2020)
#colnames(q1_2021)
#colnames(q2_2021)
```

```r
#colnames(q3_2021)
#colnames(q4_2021)
#colnames(q5_2021)
#colnames(q6_2021)
#colnames(q7_2021)
#colnames(q8_2021)
```

Look for inconsistent data types

```r
#inconsistent data type
sapply(q9_2020,class)
```

```
## $ride_id
## [1] "character"
##
## $rideable_type
## [1] "character"
##
## $started_at
## [1] "POSIXct" "POSIXt"
##
## $ended_at
## [1] "POSIXct" "POSIXt"
##
## $start_station_name
## [1] "character"
##
## $start_station_id
## [1] "numeric"
##
## $end_station_name
## [1] "character"
##
## $end_station_id
## [1] "numeric"
##
## $start_lat
## [1] "numeric"
##
## $start_lng
## [1] "numeric"
##
## $end_lat
## [1] "numeric"
##
## $end_lng
## [1] "numeric"
##
## $member_casual
## [1] "character"
```

```r
#inconsistent data type
sapply(q10_2020,class)
```

```
## $ride_id
## [1] "character"
##
## $rideable_type
## [1] "character"
##
## $started_at
## [1] "POSIXct" "POSIXt"
##
## $ended_at
## [1] "POSIXct" "POSIXt"
##
## $start_station_name
## [1] "character"
##
## $start_station_id
## [1] "numeric"
##
## $end_station_name
## [1] "character"
##
## $end_station_id
## [1] "numeric"
##
## $start_lat
## [1] "numeric"
##
## $start_lng
## [1] "numeric"
##
## $end_lat
## [1] "numeric"
##
## $end_lng
## [1] "numeric"
##
## $member_casual
## [1] "character"
```

```r
#inconsistent data type
sapply(q11_2020,class)
```

```
## $ride_id
## [1] "character"
##
## $rideable_type
## [1] "character"
##
## $started_at
## [1] "POSIXct" "POSIXt"
```

```
##
## $ended_at
## [1] "POSIXct" "POSIXt"
##
## $start_station_name
## [1] "character"
##
## $start_station_id
## [1] "numeric"
##
## $end_station_name
## [1] "character"
##
## $end_station_id
## [1] "numeric"
##
## $start_lat
## [1] "numeric"
##
## $start_lng
## [1] "numeric"
##
## $end_lat
## [1] "numeric"
##
## $end_lng
## [1] "numeric"
##
## $member_casual
## [1] "character"
```

```
#Observe start_station and end_station data type in a consistent data set
sapply(q12_2020,class)
```

```
## $ride_id
## [1] "character"
##
## $rideable_type
## [1] "character"
##
## $started_at
## [1] "POSIXct" "POSIXt"
##
## $ended_at
## [1] "POSIXct" "POSIXt"
##
## $start_station_name
## [1] "character"
##
## $start_station_id
## [1] "character"
##
## $end_station_name
## [1] "character"
```

```
##
## $end_station_id
## [1] "character"
##
## $start_lat
## [1] "numeric"
##
## $start_lng
## [1] "numeric"
##
## $end_lat
## [1] "numeric"
##
## $end_lng
## [1] "numeric"
##
## $member_casual
## [1] "character"
```

```
#consistent data sets
#sapply(q1_2021,class)
#sapply(q2_2021,class)
#sapply(q3_2021,class)
#sapply(q4_2021,class)
#sapply(q5_2021,class)
#sapply(q6_2021,class)
#sapply(q7_2021,class)
#sapply(q8_2021,class)
```

Mutate data type to make all columns consistent for merging

```
q9_2020 <- mutate(q9_2020, start_station_id = as.character(start_station_id))
q10_2020 <- mutate(q10_2020, start_station_id = as.character(start_station_id))
q11_2020 <- mutate(q11_2020, start_station_id = as.character(start_station_id))
q9_2020 <- mutate(q9_2020, end_station_id = as.character(end_station_id))
q10_2020 <- mutate(q10_2020, end_station_id = as.character(end_station_id))
q11_2020 <- mutate(q11_2020, end_station_id = as.character(end_station_id))
```

Merge into one data frame

```
bike_rides <- bind_rows(q9_2020, q10_2020, q11_2020, q12_2020, q1_2021, q2_2021, q3_2021, q4_2021, q5_20
```

===================== # STEP 3. Prepare data for analysis # =====================

Inspect the new data frame

```
dim(bike_rides)
```

```
## [1] 4913072      13
```

Create minutes (ride length) column by subtracting ended_at column from started_at column.

```
bike_rides$minutes <- difftime(bike_rides$ended_at,bike_rides$started_at,units = c("min"))
bike_rides$minutes <-  as.numeric(as.character(bike_rides$minutes))
bike_rides$minutes <- round(bike_rides$minutes, digits = 1)#round to tenth decimal place
```

Create columns for: month, day, year, day of week, and hour.

```
bike_rides$date <- as.Date(bike_rides$started_at)
bike_rides$month <- format(as.Date(bike_rides$date), "%m")
bike_rides$day <- format(as.Date(bike_rides$date), "%d")
bike_rides$year <- format(as.Date(bike_rides$date), "%Y")
bike_rides$day_of_week <- format(as.Date(bike_rides$date), "%A")
bike_rides$hour <- lubridate::hour(bike_rides$started_at)
```

Double check newly converted data types

```
is.numeric(bike_rides$minutes)
```

```
## [1] TRUE
```

```
is.Date(bike_rides$date)
```

```
## [1] TRUE
```

Use mutate function to create: season (Spring, Summer, Fall, Winter) column

```
bike_rides <-bike_rides %>% mutate(season =
                                            case_when(month == "03" ~ "Spring",
                                                      month == "04" ~ "Spring",
                                                      month == "05" ~ "Spring",
                                                      month == "06"  ~ "Summer",
                                                      month == "07"  ~ "Summer",
                                                      month == "08"  ~ "Summer",
                                                      month == "09" ~ "Fall",
                                                      month == "10" ~ "Fall",
                                                      month == "11" ~ "Fall",
                                                      month == "12" ~ "Winter",
                                                      month == "01" ~ "Winter",
                                                      month == "02" ~ "Winter"))
```

time_of_day (Night, Morning, Afternoon, Evening,) and

```
bike_rides <-bike_rides %>% mutate(time_of_day =
                                            case_when(hour == "0" ~ "Night",
                                                      hour == "1" ~ "Night",
                                                      hour == "2" ~ "Night",
                                                      hour == "3" ~ "Night",
                                                      hour == "4" ~ "Night",
                                                      hour == "5" ~ "Night",
                                                      hour == "6" ~ "Morning",
                                                      hour == "7" ~ "Morning",
```

```
                                                    hour == "8" ~ "Morning",
                                                    hour == "9" ~ "Morning",
                                                    hour == "10" ~ "Morning",
                                                    hour == "11" ~ "Morning",
                                                    hour == "12" ~ "Afternoon",
                                                    hour == "13" ~ "Afternoon",
                                                    hour == "14" ~ "Afternoon",
                                                    hour == "15" ~ "Afternoon",
                                                    hour == "16" ~ "Afternoon",
                                                    hour == "17" ~ "Afternoon",
                                                    hour == "18" ~ "Evening",
                                                    hour == "19" ~ "Evening",
                                                    hour == "20" ~ "Evening",
                                                    hour == "21" ~ "Evening",
                                                    hour == "22" ~ "Evening",
                                                    hour == "23" ~ "Evening"))
```

to mutate the month column to display the full month name.

```
bike_rides <-bike_rides %>% mutate(month = case_when(month == "01" ~ "January",
                                                    month == "02" ~ "February",
                                                    month == "03" ~ "March",
                                                    month == "04" ~ "April",
                                                    month == "05" ~ "May",
                                                    month == "06" ~ "June",
                                                    month == "07" ~ "July",
                                                    month == "08" ~ "August",
                                                    month == "09" ~ "September",
                                                    month == "10" ~ "October",
                                                    month == "11" ~ "November",
                                                    month == "12" ~ "December"))
```

==================== # STEP 5. Clean the data # ====================

Note: Business task: How do annual members and casual riders use Cyclistic bikes differently? Since our analyses is focusing on casual vs member riders let ensure our data reflects this.

```
 unique(bike_rides$member_casual)
```

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

Remove empty columns, rows and remove NA values all into a new data frame

```
df <- janitor::remove_empty(bike_rides, which = c("cols"))
df <- janitor::remove_empty(bike_rides, which = c("rows"))
df <- distinct(bike_rides)
df<- na.omit(bike_rides)
```

View the dimension

```
dim(df)
```

```
## [1] 4233298      22
```

Note: Number of observations is now 4,233,298 (679,774 rows were removed). Now filter the data frame to remove where ride_length is 0 or negative and filter out unnecessary columns.

```
df <- df %>%
  filter(minutes>0) %>%
    select(-c(ride_id,started_at,ended_at,start_station_id,end_station_name,end_station_id,start_lat,sta
```

Note: New data frame is 4,221,509 observations (11,789 additional observations were removed). View the final data frame.

```
View(df)
dim(df)
```

```
## [1] 4221509      12
```

===================== # STEP 5. Conduct descriptive analysis # =====================

Business task: How do annual members and casual riders use Cyclistic bikes differently?

Casual = customers who purchase single-ride or full-day passes

Members = customers who purchase annual memberships

What date range does our data cover?

```
## [1] "2020-09-01"
```

to

```
## [1] "2021-08-31"
```

How many total rides?

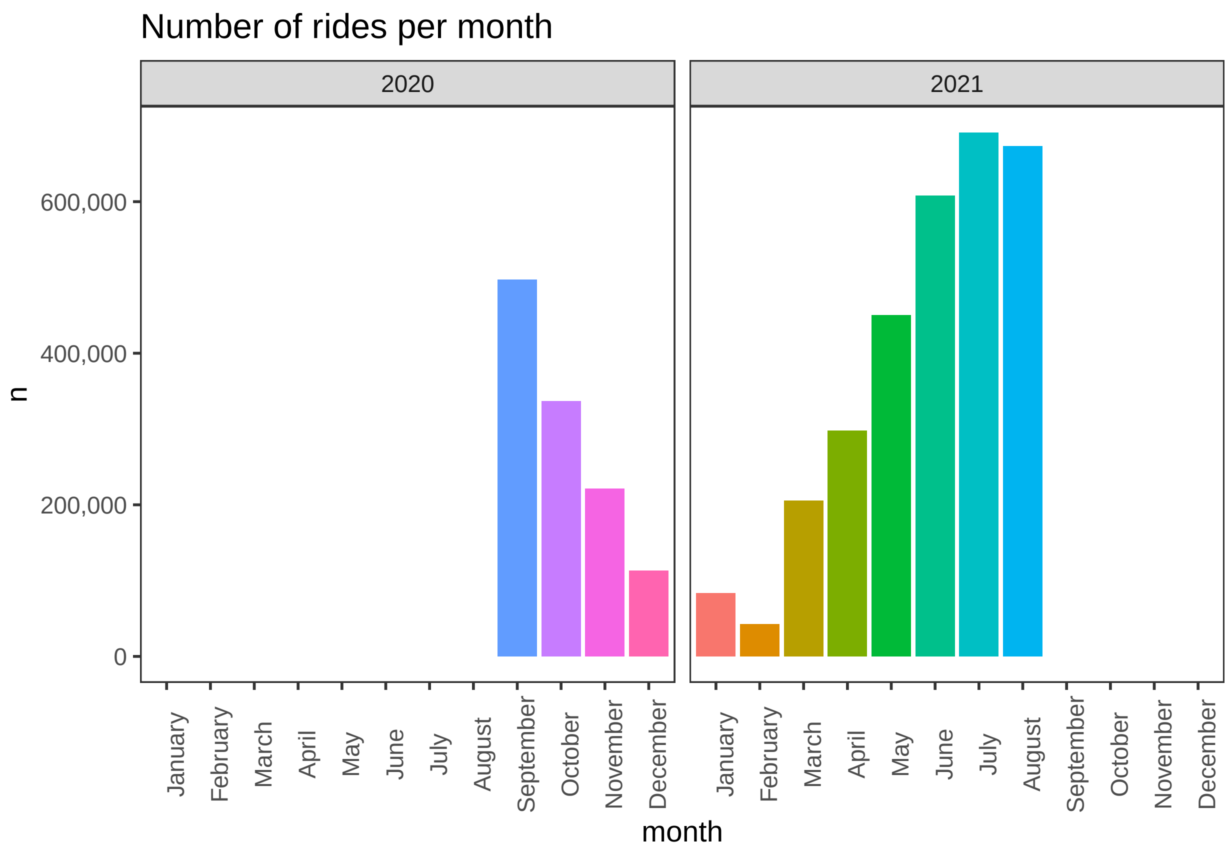
```
## [1] 4221509
```

Find the number of rides per month

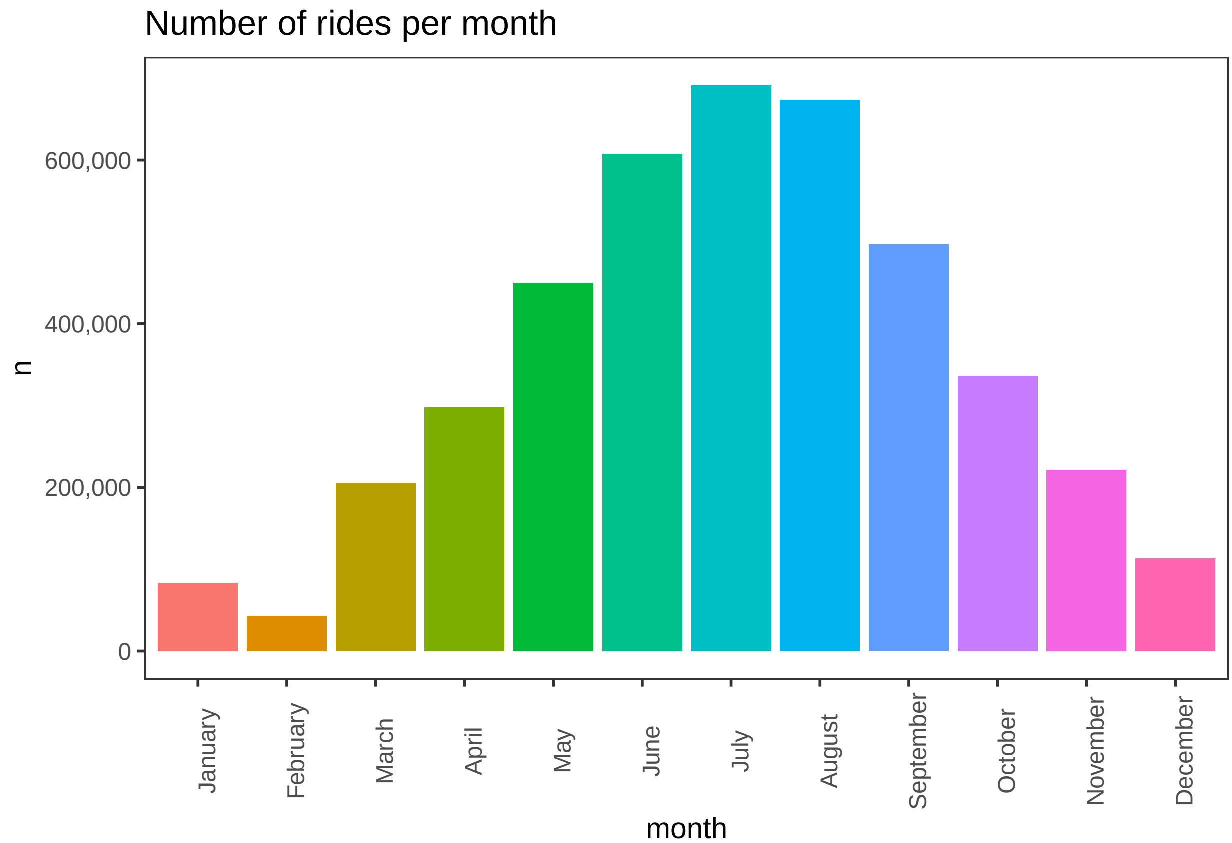
```
## # A tibble: 12 x 3
## # Groups:   month [12]
##    month     year       n
##    <fct>     <chr>  <int>
##  1 September 2020  497294
##  2 October   2020  336698
##  3 November  2020  221591
##  4 December  2020  113371
##  5 January   2021   83366
##  6 February  2021   42840
##  7 March     2021  205454
```

```
##   8 April     2021  297801
##   9 May       2021  450278
## 10 June      2021  607945
## 11 July      2021  691376
## 12 August    2021  673495
```

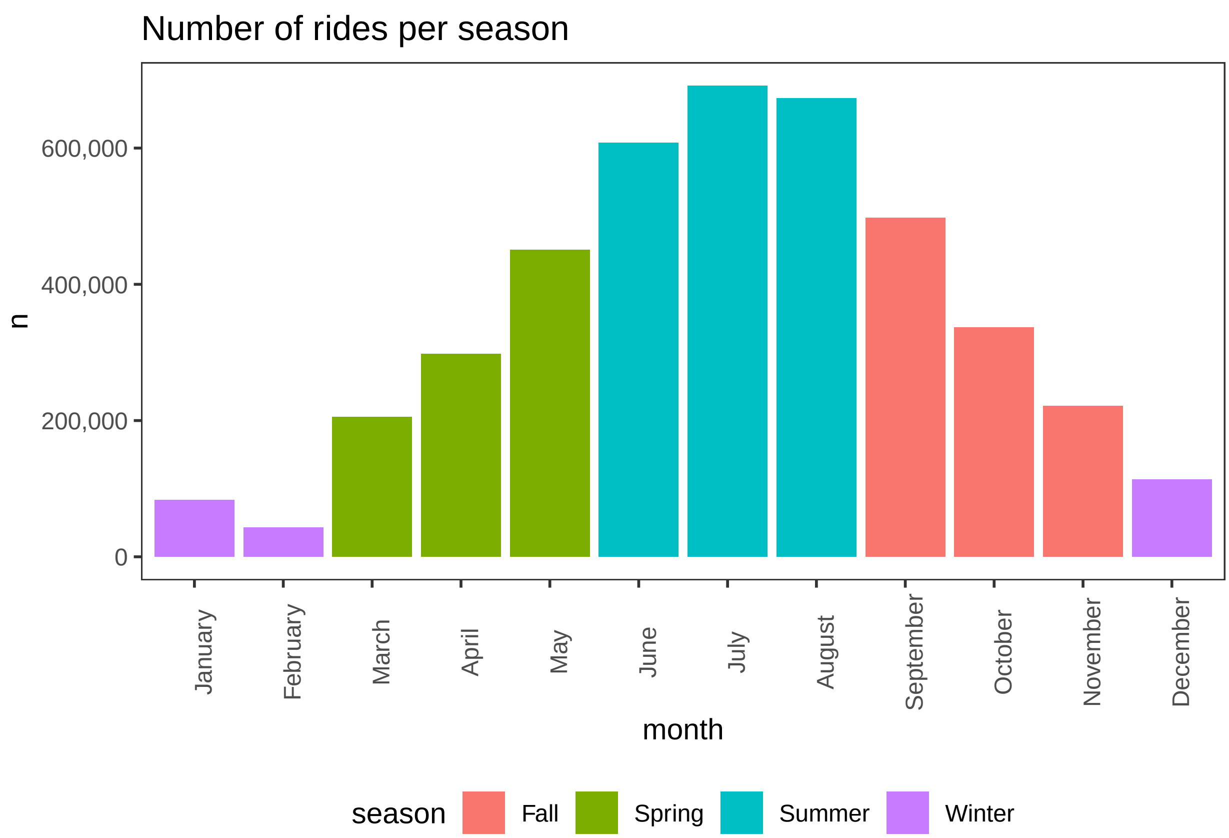
Lets visualize the data.



Our data covers 12 months, 2020-09-01 to 2021-08-31, that is the end of 2020 to the beginning of 2021. Lets visualize our graph chronologically. Image 2

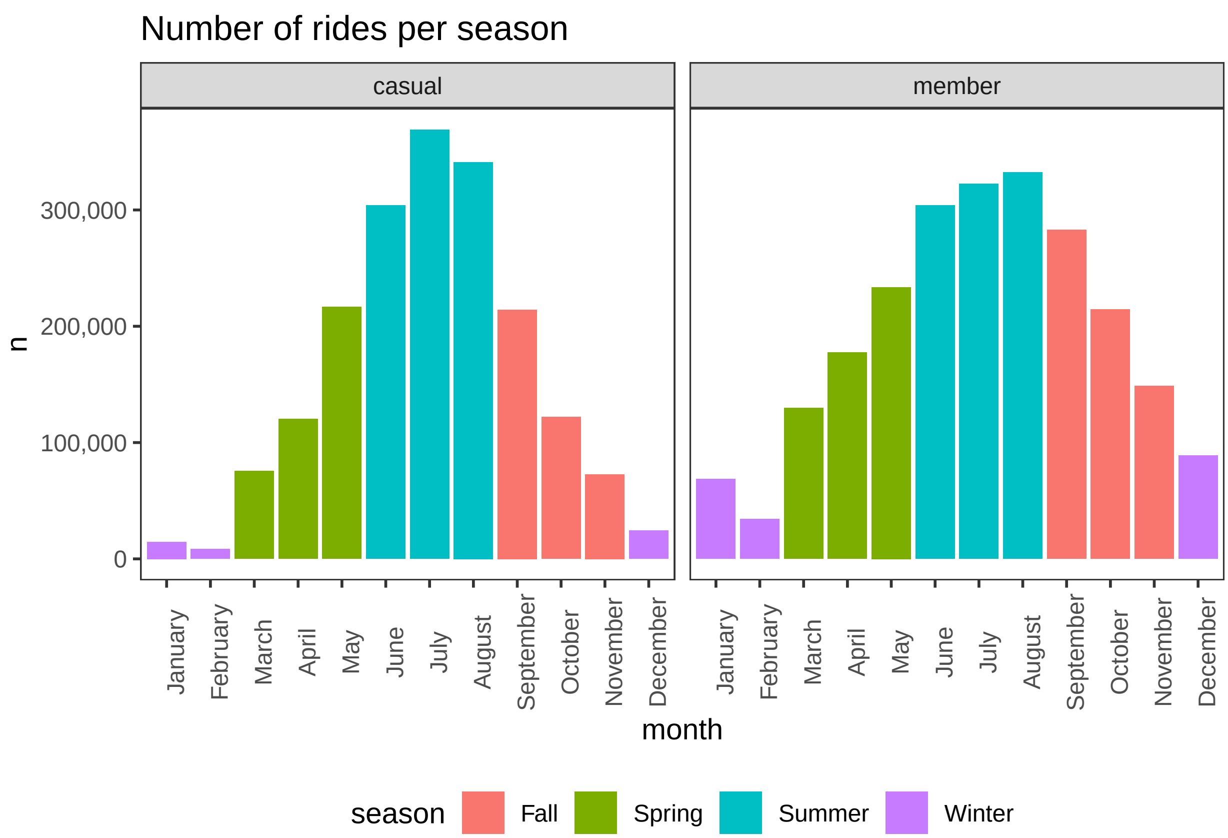


Viewing the data in chronological order by month makes the data into a bell shape distribution. We can see that the peak of bike rides takes place in the month of July. For sake of this analysis, the season will be as fol-



lows.

The peak months of number of bike rides are in the months of June-August,summer time. We will come back to this time frame. Is there a difference between type of riders and number of rides in the overall data?



At hindsight we can see the number of bike rides for both member and casual riders are at its highest levels during the summer time (June-August). The total number of rides during summer time is

```
## # A tibble: 1 x 2
##         n  prop
##     <int> <dbl>
## 1 1972816 0.467
```

Around 47 percent of all rides take place during the summer time. Let's focus and continue our analysis in this time frame (June-August). First lets find the total number of riders by type of rider.

```
## # A tibble: 2 x 3
##   member_casual       n  prop
##   <chr>           <int> <dbl>
## 1 casual        1014122 0.514
## 2 member         958694 0.486
```

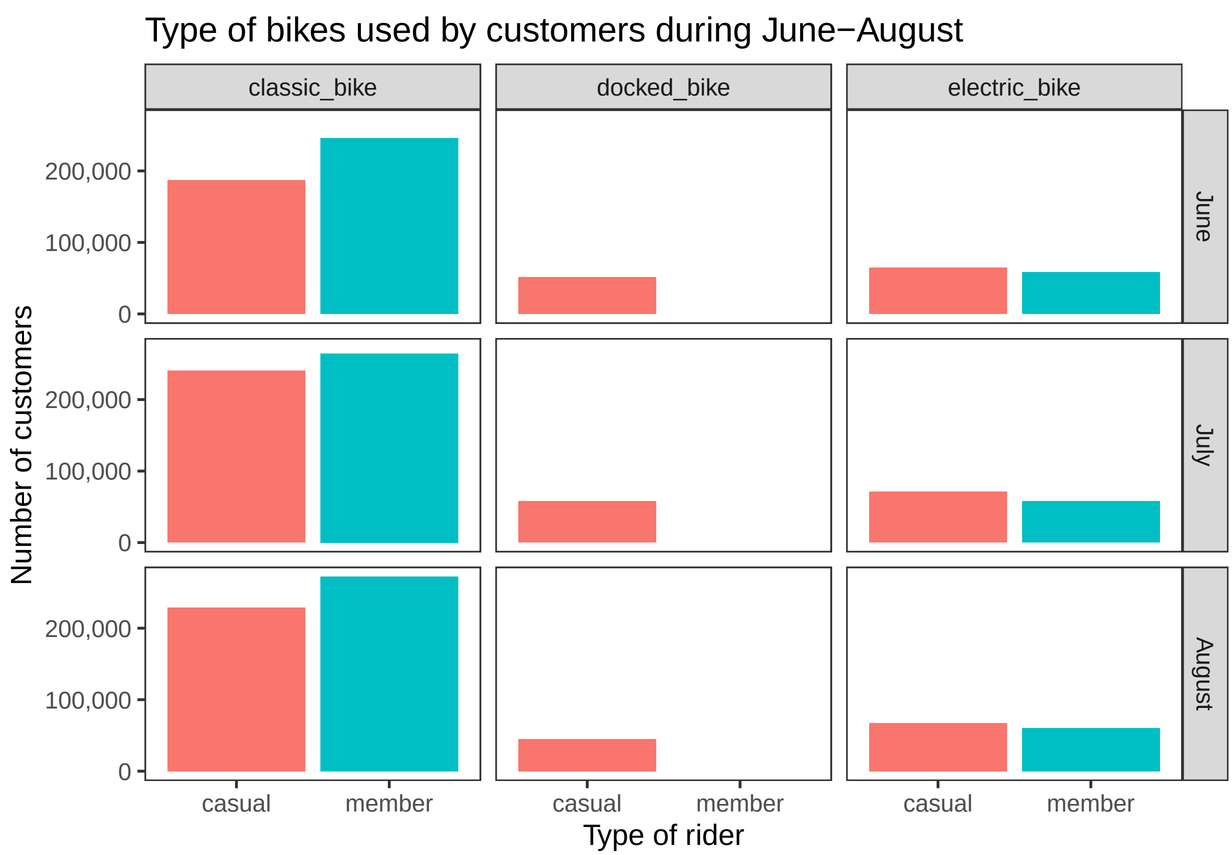
During summer time, casual riders tend to out number the member riders. As shown above (Image 2), July was the busiest month with casual riders outnumbering members during July. What are the figures of the type of bicycle used during June-August?

```
## # A tibble: 3 x 3
##   rideable_type       n   prop
##   <chr>           <int>  <dbl>
## 1 classic_bike  1439012 0.729
## 2 docked_bike    154390 0.0783
## 3 electric_bike  379414 0.192
```

The most popular bikes during June-August was classic bikes.Users used classic bikes 9.3 more times than docked bikes and 3.8 more times than electric bikes. The individual numbers by month and type of bike are as follows:

```
## # A tibble: 9 x 4
##   month  rideable_type      n   prop
##   <fct>  <chr>          <int>  <dbl>
## 1 June   classic_bike  433145 0.220
## 2 June   docked_bike    51694 0.0262
## 3 June   electric_bike 123106 0.0624
## 4 July   classic_bike  504791 0.256
## 5 July   docked_bike    57664 0.0292
## 6 July   electric_bike 128921 0.0653
## 7 August classic_bike  501076 0.254
## 8 August docked_bike    45032 0.0228
## 9 August electric_bike 127387 0.0646
```

Lets visualize and lets also consider the type of member utilizing these bikes during the summer.



As mentioned earlier, users use classic bikes 9.3 more times than docked bikes and 3.8 more times than electric bikes. Classic bikes are favorable regardless of type of rider and summer month. Individual number of graphs are below:

```
## # A tibble: 15 x 4
##    month  rideable_type member_casual      n
##    <fct>  <chr>         <chr>          <int>
##  1 June   classic_bike  casual        187234
```

```
##  2 June   docked_bike   casual         51694
##  3 June   electric_bike casual         64976
##  4 July   classic_bike  casual        240315
##  5 July   docked_bike   casual         57664
##  6 July   electric_bike casual         71073
##  7 August classic_bike  casual        228931
##  8 August docked_bike   casual         45032
##  9 August electric_bike casual         67203
## 10 June   classic_bike  member        245911
## 11 June   electric_bike member         58130
## 12 July   classic_bike  member        264476
## 13 July   electric_bike member         57848
## 14 August classic_bike  member        272145
## 15 August electric_bike member         60184
```

Lets find the mean, median, max, and min for the ride length (minutes) for customers during summer time.

```
## # A tibble: 1 x 4
##   Average_ride_length   min   med    max
##                 <dbl> <dbl> <dbl>  <dbl>
## 1                23.8   0.1  13.3 55944.
```

Between casual riders and members.

```
## # A tibble: 2 x 5
##   member_casual Average_duration   min   med    max
##   <chr>                    <dbl> <dbl> <dbl>  <dbl>
## 1 casual                    33.3   0.1  17.2 55944.
## 2 member                    13.8   0.1  10.4  1496.
```

Not only do casual riders outnumber members they also on average spend longer time riding bicycles than members. What are the average ride length between casual rider and members in a a given day ? (Note: Order the days of the week to make it easy to analyse.)

```
df$day_of_week <- ordered(df$day_of_week, levels=c("Sunday", "Monday","Tuesday", "Wednesday", "Thursday
```

Find the average minutes spend riding bikes by day of the week between casual riders and members.

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

```
## # A tibble: 14 x 3
## # Groups:   member_casual [2]
##    member_casual day_of_week average_duration
##    <chr>         <ord>                  <dbl>
##  1 casual        Sunday                  37.2
##  2 casual        Monday                  32.4
##  3 casual        Tuesday                 29.6
##  4 casual        Wednesday               30.4
##  5 casual        Thursday                30.9
##  6 casual        Friday                  31.8
##  7 casual        Saturday                35.9
##  8 member        Sunday                  15.8
```

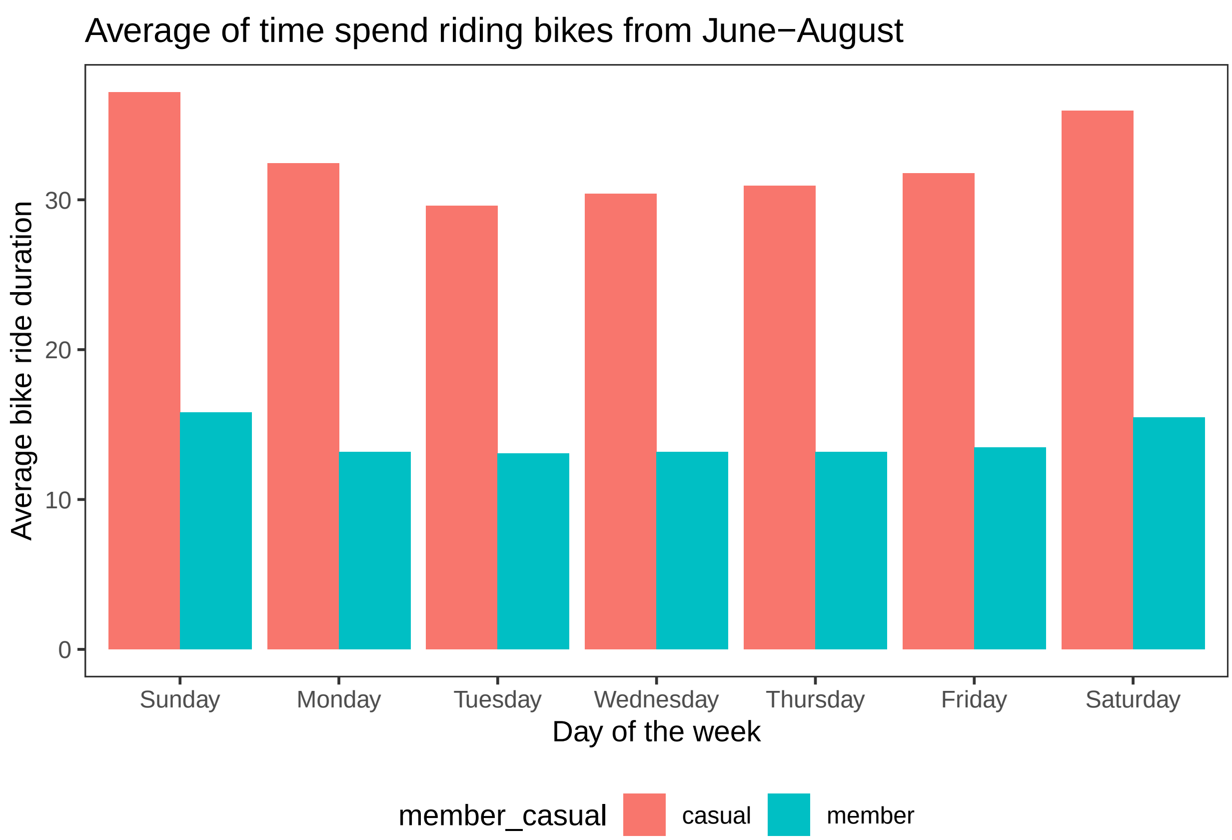
```
##  9 member        Monday                  13.2
## 10 member        Tuesday                 13.1
## 11 member        Wednesday               13.2
## 12 member        Thursday                13.2
## 13 member        Friday                  13.5
## 14 member        Saturday                15.5
```

Lets visualize (Note: Visualization is comparing casual riders vs members).

Casual = customers who purchase single-ride or full-day passes

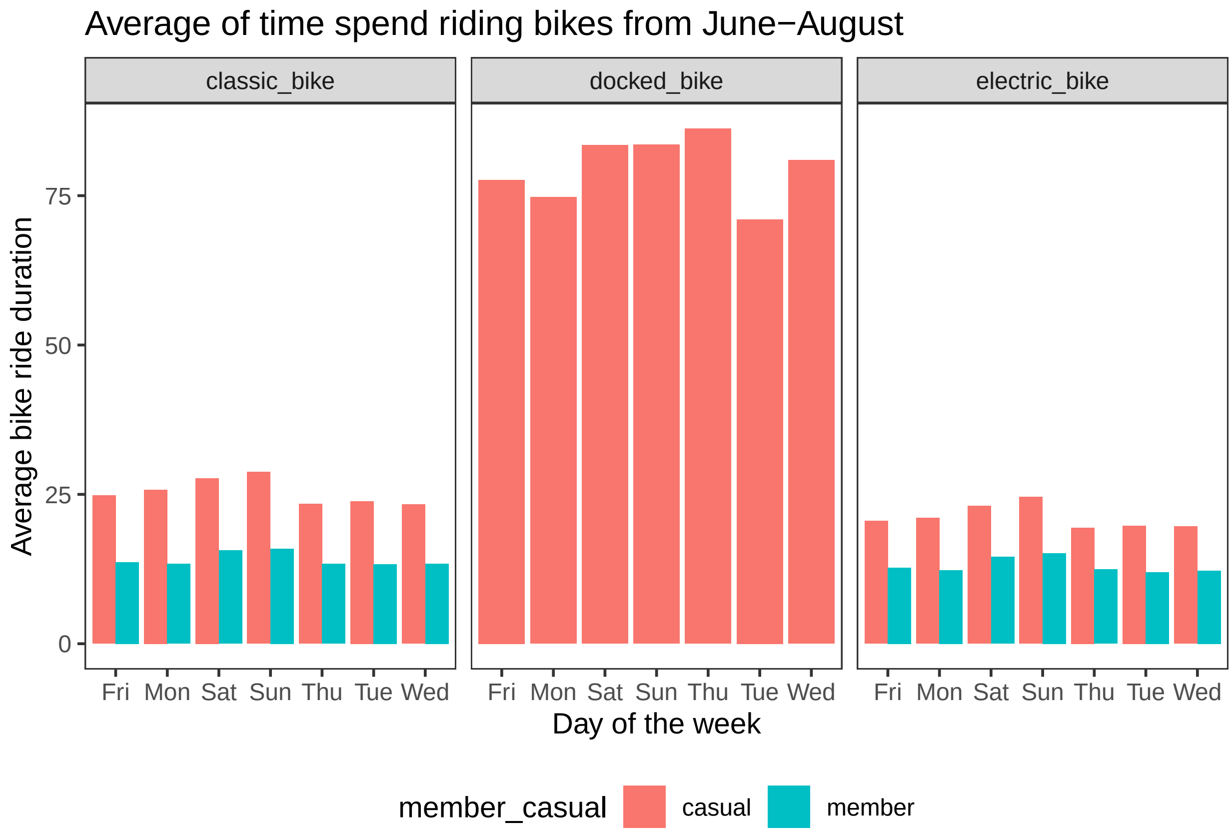
Members = customers who purchase annual memberships

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



Is there a change when we filter for type of bike used?

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



Look at this. Casual riders on average spend more time riding docked bikes on any given day of the week.

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

## # A tibble: 5 x 3
## # Groups:   member_casual [2]
##   member_casual rideable_type average_duration
##   <chr>         <chr>                    <dbl>
## 1 casual        classic_bike              25.9
## 2 casual        docked_bike               80.4
## 3 casual        electric_bike             21.3
## 4 member        classic_bike              14.1
## 5 member        electric_bike             12.9
```

Casual riders spend on average 3.1 times longer riding docked bicycles compared with classic bicycles. We will come back to this. For now lets find the number of rides per day of the week between casual riders and members

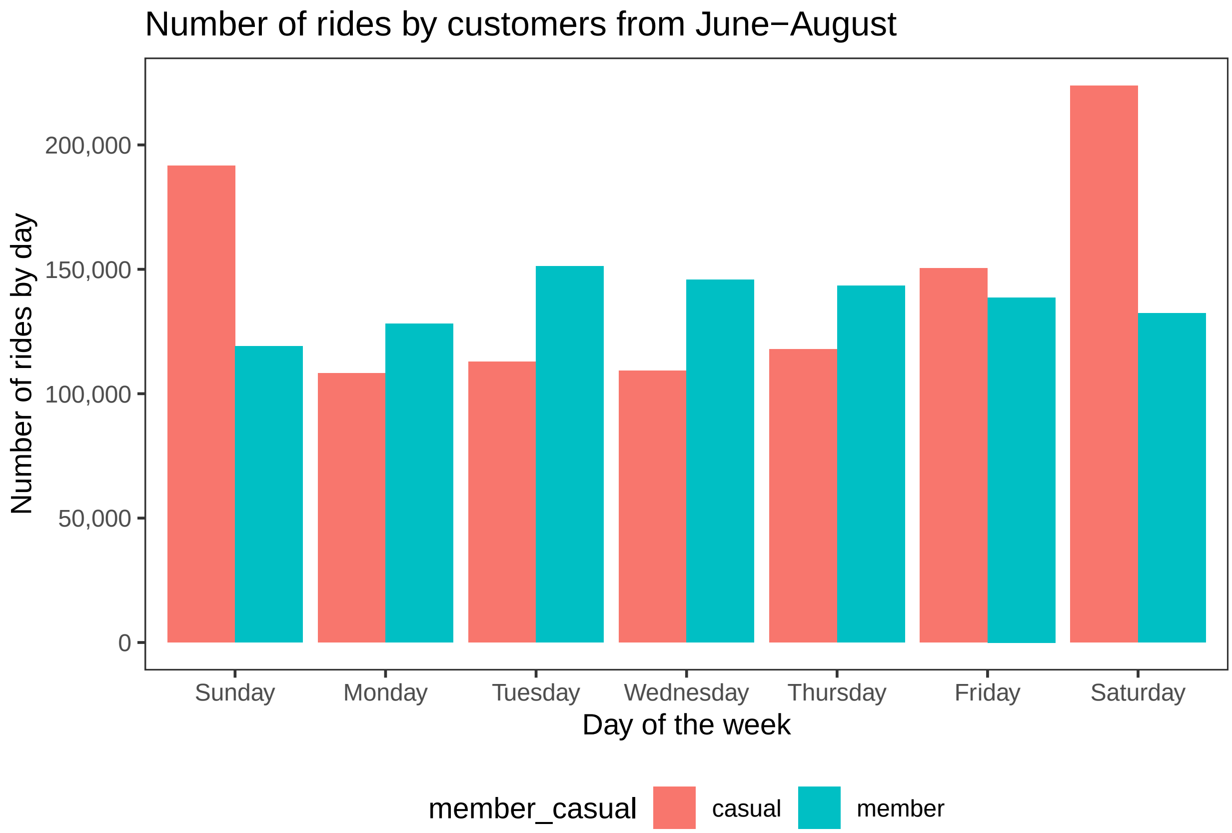
```
## `summarise()` has grouped output by 'member_casual'. You can override using the `.groups` argument.

## # 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               191607             37.2
```

```
##   2 casual        Monday             108241            32.4
##   3 casual        Tuesday            112901            29.6
##   4 casual        Wednesday          109301            30.4
##   5 casual        Thursday           117835            30.9
##   6 casual        Friday             150376            31.8
##   7 casual        Saturday           223861            35.9
##   8 member        Sunday             119107            15.8
##   9 member        Monday             128107            13.2
##  10 member        Tuesday            151194            13.1
##  11 member        Wednesday          145784            13.2
##  12 member        Thursday           143466            13.2
##  13 member        Friday             138681            13.5
##  14 member        Saturday           132355            15.5
```

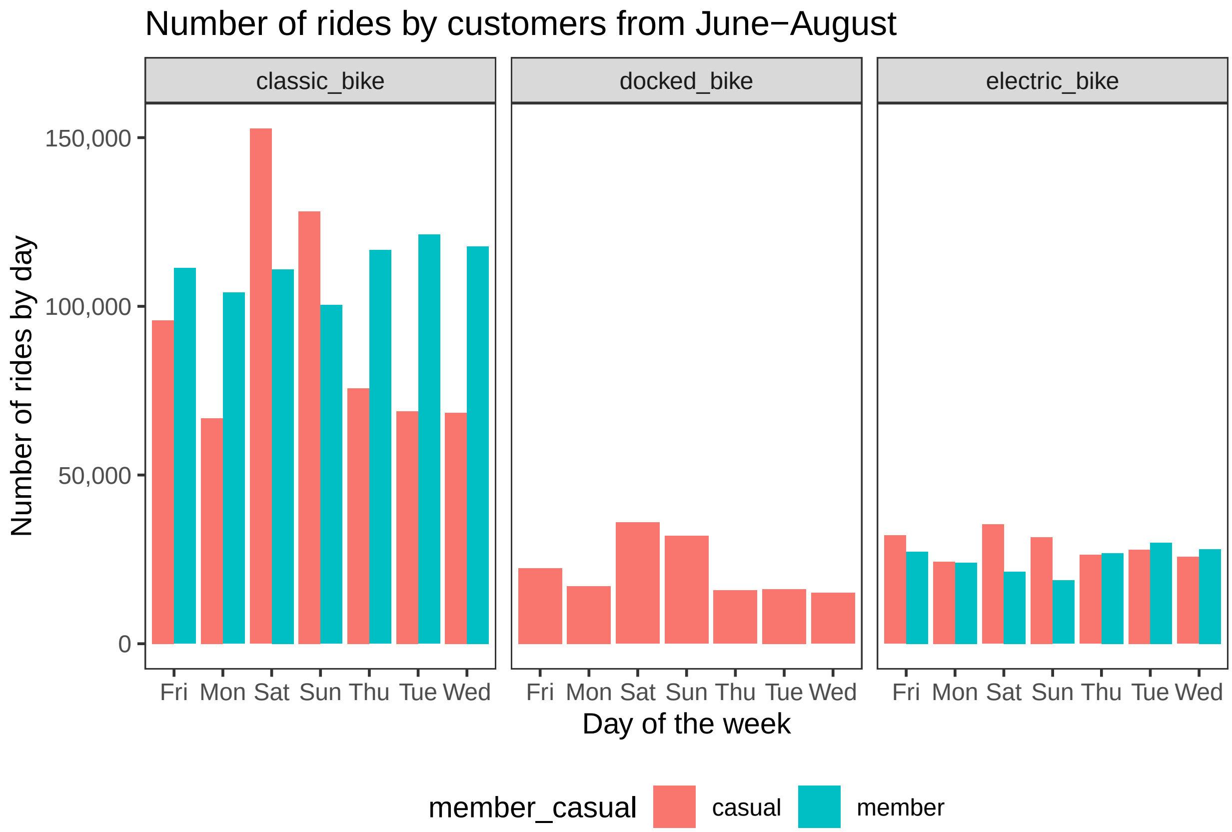
Visualize the number of rides by rider type

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



Lets see the difference between the number of rider per day by analyzing by type of bike

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



Even though casual riders on average spend more time riding docked bikes on any given day of the week, docked bicycles are not used as frequently compared to classic and electric bicycles.

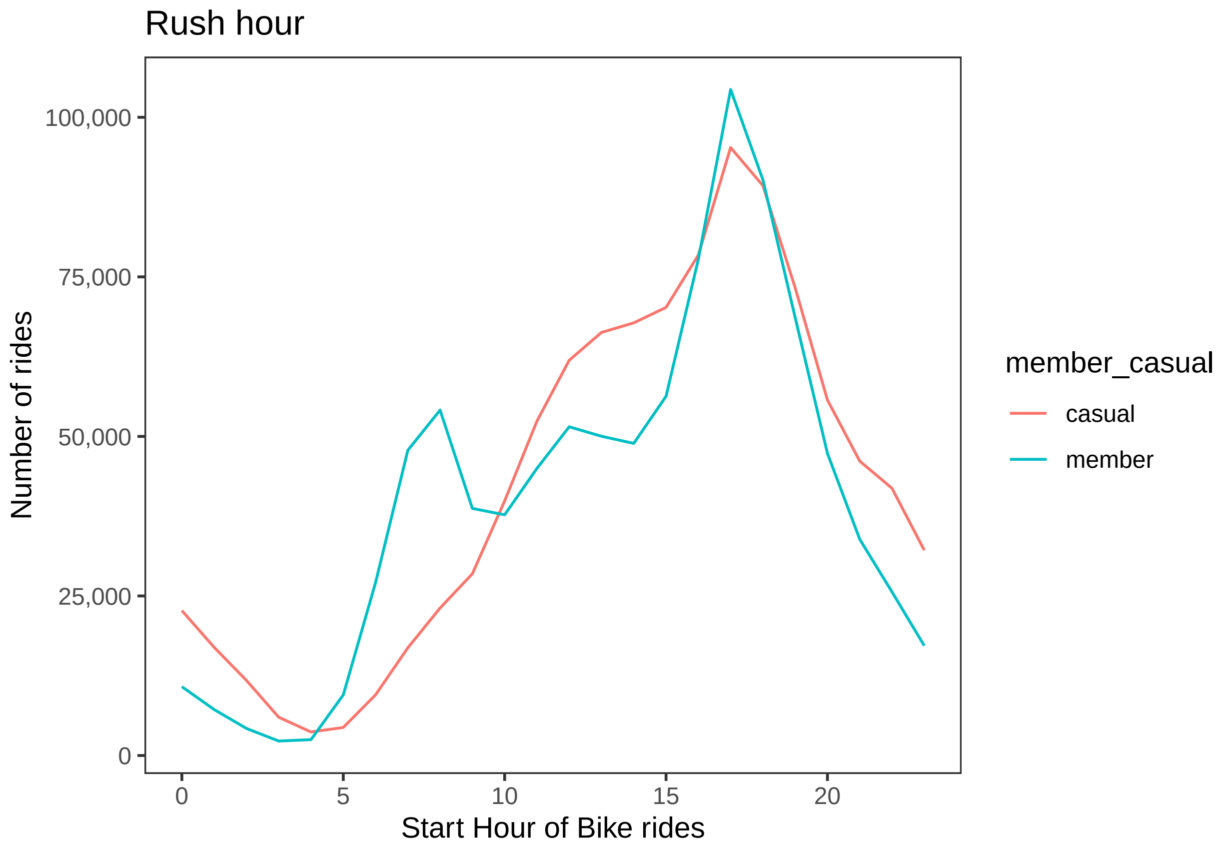
```
## # A tibble: 5 x 4
##   rideable_type member_casual      n   prop
##   <chr>         <chr>          <int>  <dbl>
## 1 classic_bike  casual        656480 0.333
## 2 classic_bike  member        782532 0.397
## 3 docked_bike   casual        154390 0.0783
## 4 electric_bike casual        203252 0.103
## 5 electric_bike member        176162 0.0893
```

Casual riders use classic bicycles 4.3 more times than docked bicycles. What time during the day do we see the most riders?

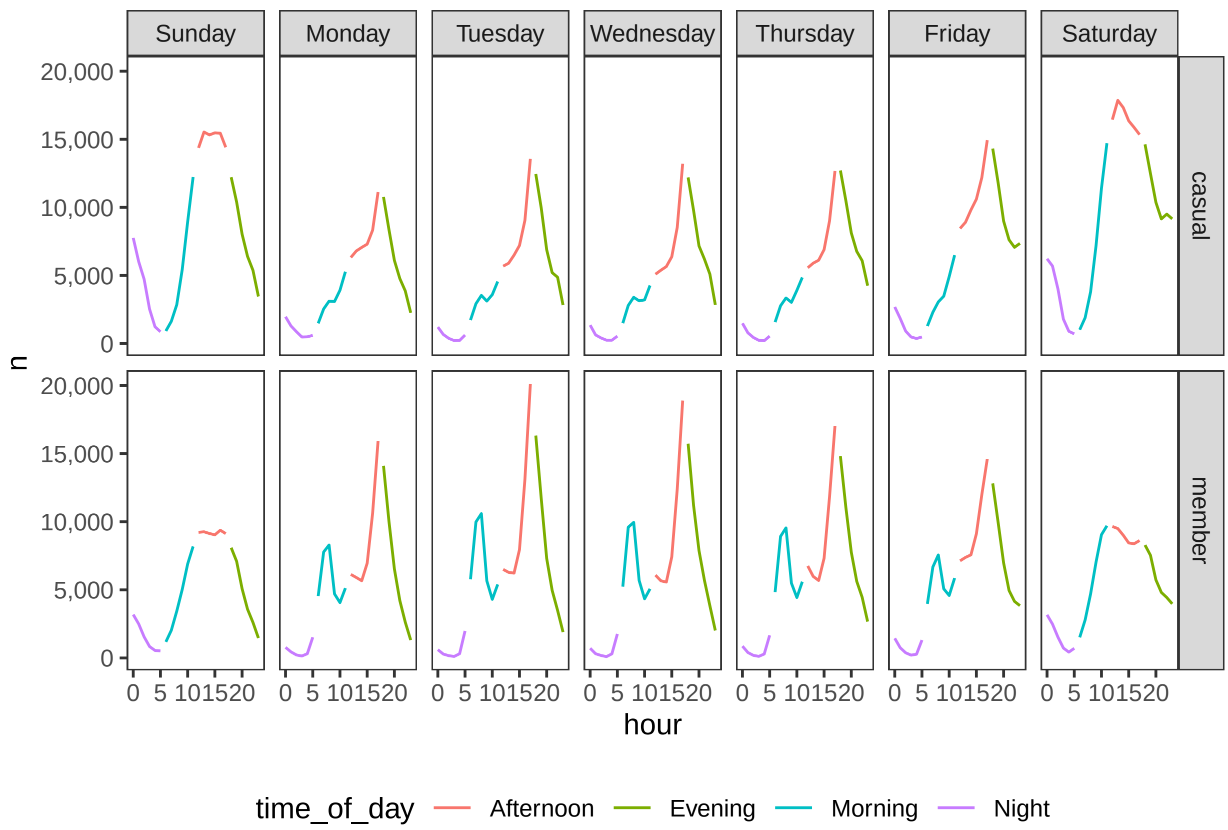
```
## # A tibble: 48 x 3
## # Groups:   member_casual [2]
##    member_casual  hour      n
##    <chr>         <int>  <int>
##  1 member           17 104359
##  2 casual           17  95257
##  3 member           18  90221
##  4 casual           18  89295
##  5 casual           16  78423
##  6 member           16  77755
##  7 casual           19  73276
```

```
## 8 casual          15  70212
## 9 member          19  68617
## 10 casual         14  67790
## # ... with 38 more rows
```

Lets visualize



Visualize for time of day and during the day of the week between casual riders and members.



The afternoon is the peak time the most riders come on any given day of the week. Casual drivers come most on Saturday and Sunday. Popular Start Stations for Casual riders are:

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

```
## Adding missing grouping variables: `member_casual`
```

```
## # A tibble: 30 x 3
## # Groups:   member_casual [1]
##    member_casual start_station_name                 number_of_ride
##    <chr>         <chr>                                       <int>
##  1 casual        Streeter Dr & Grand Ave                     36421
##  2 casual        Michigan Ave & Oak St                       16113
##  3 casual        Millennium Park                             15963
##  4 casual        Theater on the Lake                         11798
##  5 casual        Shedd Aquarium                              11218
##  6 casual        Wells St & Concord Ln                        9804
##  7 casual        Lake Shore Dr & North Blvd                   9546
##  8 casual        Lake Shore Dr & Monroe St                    9383
##  9 casual        Clark St & Lincoln Ave                       8697
## 10 casual        DuSable Lake Shore Dr & North Blvd           8273
## # ... with 20 more rows
```

Popular Start Stations for Member riders:

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

```
## Adding missing grouping variables: `member_casual`

## # A tibble: 30 x 3
## # Groups:   member_casual [1]
##    member_casual start_station_name       number_of_ride
##    <chr>         <chr>                             <int>
##  1 member        Wells St & Concord Ln              9337
##  2 member        Clark St & Elm St                  9097
##  3 member        Kingsbury St & Kinzie St           8197
##  4 member        Streeter Dr & Grand Ave            7864
##  5 member        Wells St & Elm St                  7858
##  6 member        Theater on the Lake                7465
##  7 member        Clark St & Lincoln Ave             7044
##  8 member        Michigan Ave & Oak St              6782
##  9 member        Broadway & Barry Ave               6739
## 10 member        Wells St & Huron St                6727
## # ... with 20 more rows
```

End of analysis.

Summary:

-I learned that docked bicycle type is on average ridden longer by casual riders. However, casual riders use classic bicycles 4.3 more than docked bicycles.

-Saturday and Sunday afternoons are the most popular riding days for casual riders.

-November through February have the least number of casual riders while June, July, and August have a particularly high number of Casual riders.

-The most popular stations for Casual riders in descending order are Streeter Dr & Grand Ave, Michigan Ave & Oak St, Millennium Park, Theater on the Lake, Shedd Aquarium.

Recommendations

-Based on the data analyzed I would recommend we focus our marketing efforts for Casual riders with these parameters

1: Increase marketing for docket bicycles 2. Heavier marketing from June through August 3. Focus marketing on afternoon weekends 4. Invest in marketing at the top 5 stations as noted above.