Case Study: How Does a Bike-Share Navigate Speedy Success?

Adusumalli Jayakrishna

27/01/2022

## Scenario

You are a junior data analyst working in the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company’s future success depends on maximizing the number of annual memberships. Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations.

### Objective

Find out how casual riders and annual members use Cyclistic bikes differently so that Your team can use these insights to create recommendations on how to convert casual riders into annual members.

### Data Source

We use the [Cyclistic’s historical trip data](https://divvy-tripdata.s3.amazonaws.com/index.html) to analyze and identify trends.

(Note: The datasets have a different name because Cyclistic is a fictional company. For the purposes of this case study, the datasets are appropriate and will enable you to answer the business questions. The data has been made available by Motivate International Inc. under this [license](https://ride.divvybikes.com/data-license-agreement).)

## Importing required packages

library(tidyverse) # Makes analyzing data easier  
library(dplyr) # Used for data manipulation  
library(here) # Helps finding files easier  
library(skimr) # Used for finding Summary Statistics  
library(janitor) # Helps clean data  
library(data.table)  
library(lubridate) # Used for manipulating Date attributes  
library(hms)

## Reading data

Reading all the data into a list of data frames and checking each file sizes :-

file\_names <- list.files(path="./all\_data", pattern=".csv", all.files=TRUE,full.names=TRUE)  
print("printing each file size in MB")  
data <- list()  
n <- 1  
for (item in file\_names){  
 cat("File name ", item)  
 data[[n]] = read.csv(item,na.strings=c("","NA")) # reading all the data where empty strings are recorded as NA  
 cat("\n File Size = ")  
 print(object.size(data[[n]]),units = "MB")  
 cat("Number of rows = ",nrow(data[[n]]))  
 cat(" , Number of columns = ",ncol(data[[n]]))  
 cat("\n")  
 n <- n+1  
}

## [1] "printing each file size in MB"  
## File name ./all\_data/202101-divvy-tripdata.csv  
## File Size = 21.7 Mb  
## Number of rows = 96834 , Number of columns = 13  
## File name ./all\_data/202102-divvy-tripdata.csv  
## File Size = 11.9 Mb  
## Number of rows = 49622 , Number of columns = 13  
## File name ./all\_data/202103-divvy-tripdata.csv  
## File Size = 45.6 Mb  
## Number of rows = 228496 , Number of columns = 13  
## File name ./all\_data/202104-divvy-tripdata.csv  
## File Size = 65 Mb  
## Number of rows = 337230 , Number of columns = 13  
## File name ./all\_data/202105-divvy-tripdata.csv  
## File Size = 99.6 Mb  
## Number of rows = 531633 , Number of columns = 13  
## File name ./all\_data/202106-divvy-tripdata.csv  
## File Size = 134.5 Mb  
## Number of rows = 729595 , Number of columns = 13  
## File name ./all\_data/202107-divvy-tripdata.csv  
## File Size = 151.1 Mb  
## Number of rows = 822410 , Number of columns = 13  
## File name ./all\_data/202108-divvy-tripdata.csv  
## File Size = 147.9 Mb  
## Number of rows = 804352 , Number of columns = 13  
## File name ./all\_data/202109-divvy-tripdata.csv  
## File Size = 139.1 Mb  
## Number of rows = 756147 , Number of columns = 13  
## File name ./all\_data/202110-divvy-tripdata.csv  
## File Size = 117.3 Mb  
## Number of rows = 631226 , Number of columns = 13  
## File name ./all\_data/202111-divvy-tripdata.csv  
## File Size = 69.2 Mb  
## Number of rows = 359978 , Number of columns = 13  
## File name ./all\_data/202112-divvy-tripdata.csv  
## File Size = 49.4 Mb  
## Number of rows = 247540 , Number of columns = 13

By above results we can observe that all files have same columns and after further analysis we have found that the columns have similar names and also are arranged in the same order.

Now we are going to combine the above collected data into a single data frame for further analysis.

df <- rbindlist(data) # binding all the data frame into a single data frame  
cat("Final dataframe details \n")  
glimpse((df))  
cat("\n")  
cat("column names of final data \n")  
colnames(df)

## Final dataframe details   
## Rows: 5,595,063  
## Columns: 13  
## $ ride\_id <chr> "E19E6F1B8D4C42ED", "DC88F20C2C55F27F", "EC45C94683~  
## $ rideable\_type <chr> "electric\_bike", "electric\_bike", "electric\_bike", ~  
## $ started\_at <chr> "23-01-2021 16:14", "27-01-2021 18:43", "21-01-2021~  
## $ ended\_at <chr> "23-01-2021 16:24", "27-01-2021 18:47", "21-01-2021~  
## $ start\_station\_name <chr> "California Ave & Cortez St", "California Ave & Cor~  
## $ start\_station\_id <chr> "17660", "17660", "17660", "17660", "17660", "17660~  
## $ end\_station\_name <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "Wood St & Augu~  
## $ end\_station\_id <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "657", "13258",~  
## $ start\_lat <dbl> 41.90034, 41.90033, 41.90031, 41.90040, 41.90033, 4~  
## $ start\_lng <dbl> -87.69674, -87.69671, -87.69664, -87.69666, -87.696~  
## $ end\_lat <dbl> 41.89000, 41.90000, 41.90000, 41.92000, 41.90000, 4~  
## $ end\_lng <dbl> -87.72000, -87.69000, -87.70000, -87.69000, -87.700~  
## $ member\_casual <chr> "member", "member", "member", "member", "casual", "~  
##   
## column names of final data   
## [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"

So we have 5,595,063 rows 13 columns Now lets explore the columns in detail

cat("Here are the number of unique values and na values in each column of the final dataframe \n")  
column <- colnames(df)  
no\_of\_unique <- c()  
no\_of\_na <- c()  
  
for(idx in 1:13){  
 no\_of\_unique <- append(no\_of\_unique,length(unique(df[[idx]])))  
 no\_of\_na <- append(no\_of\_na, sum(is.na(df[[idx]])))  
}  
  
print(data.frame(column,no\_of\_unique,no\_of\_na))

## Here are the number of unique values and na values in each column of the final dataframe   
## column no\_of\_unique no\_of\_na  
## 1 ride\_id 5595041 0  
## 2 rideable\_type 3 0  
## 3 started\_at 446087 0  
## 4 ended\_at 446864 0  
## 5 start\_station\_name 848 690809  
## 6 start\_station\_id 834 690806  
## 7 end\_station\_name 845 739170  
## 8 end\_station\_id 832 739170  
## 9 start\_lat 393593 0  
## 10 start\_lng 373296 0  
## 11 end\_lat 444888 4771  
## 12 end\_lng 405092 4771  
## 13 member\_casual 2 0

From the above results we can say about the data that:-  
1. ride\_id column has almost similar number of rows compared to total number of rows which says that ride\_id acts as a primary key in the data but there are some ids that are repeating which should be removed in the cleaning process.  
2. There are only 3 rideable\_type.  
3. started\_at and ended\_at are date data type and has no Na values.  
4. start\_station\_name, end\_station name has more unique values than their respective ids so few station must have similar ids and there are a lot of missing values in these columns.  
5. There are 4 latitude and longitude columns that describe the start position and end position of each ride.  
6. member\_casual column has only 2 unique values with 0 NA.

## Cleaning and Processing data

### 1. Cleaning

Firstly we see the sample of data

head(df)

## ride\_id rideable\_type started\_at ended\_at  
## 1: E19E6F1B8D4C42ED electric\_bike 23-01-2021 16:14 23-01-2021 16:24  
## 2: DC88F20C2C55F27F electric\_bike 27-01-2021 18:43 27-01-2021 18:47  
## 3: EC45C94683FE3F27 electric\_bike 21-01-2021 22:35 21-01-2021 22:37  
## 4: 4FA453A75AE377DB electric\_bike 07-01-2021 13:31 07-01-2021 13:42  
## 5: BE5E8EB4E7263A0B electric\_bike 23-01-2021 02:24 23-01-2021 02:24  
## 6: 5D8969F88C773979 electric\_bike 09-01-2021 14:24 09-01-2021 15:17  
## start\_station\_name start\_station\_id end\_station\_name end\_station\_id  
## 1: California Ave & Cortez St 17660 <NA> <NA>  
## 2: California Ave & Cortez St 17660 <NA> <NA>  
## 3: California Ave & Cortez St 17660 <NA> <NA>  
## 4: California Ave & Cortez St 17660 <NA> <NA>  
## 5: California Ave & Cortez St 17660 <NA> <NA>  
## 6: California Ave & Cortez St 17660 <NA> <NA>  
## start\_lat start\_lng end\_lat end\_lng member\_casual  
## 1: 41.90034 -87.69674 41.89 -87.72 member  
## 2: 41.90033 -87.69671 41.90 -87.69 member  
## 3: 41.90031 -87.69664 41.90 -87.70 member  
## 4: 41.90040 -87.69666 41.92 -87.69 member  
## 5: 41.90033 -87.69670 41.90 -87.70 casual  
## 6: 41.90041 -87.69676 41.94 -87.71 casual

Now lets convert the date columns in date datatype (POSIXct) using lubridate package

df <- df %>%  
 mutate(started\_at = dmy\_hms(started\_at))  
df <- df %>%  
 mutate(ended\_at = dmy\_hms(ended\_at))

Now we remove the rows which contain duplicate ride\_id

cat("old number of rows :- ",nrow(df))

## old number of rows :- 5595063

df <- df %>% distinct(ride\_id, .keep\_all= TRUE)  
cat("\n")

cat("New number of rows :- ",nrow(df))

## New number of rows :- 5595041

### 2. Processing data

Firstly we will create a new column calculating the ride length (time taken for each ride) which would be of huge help in our analysis.

df <- df %>%  
 mutate(ride\_length = as\_hms(difftime(ended\_at,started\_at)))   
 # difftime is a function in lubridate package used for calculating time differences # as\_hms is a function in hms package used for converting time itno hh:mm:ss

Now lets find out the weekday of each ride and store it in a new column in numerical format so that it would be helpful for easier analysis.

df <- df %>%  
 mutate(day\_of\_week = recode(weekdays(started\_at),"Sunday"=1,"Monday"=2,"Tuesday"=3,"Wednesday"=4,"Thursday"=5,"Friday"=6,"Saturday"=7))

Here we have marked the days of a week form Sunday to Saturday with values 1 to 7 respectively. Now lets do the same for months.

df <- df %>%  
 mutate(day\_of\_month = month(started\_at)) #month function directly converts the months into numerical values

Lets check the newly created data for any errors.

cat("Minimum ride length :- ",min(df$ride\_length))  
cat("\n")  
cat("Maximum ride length :- ",max(df$ride\_length))  
cat("\n")  
cat("no of NA values in ride\_length :- ",sum(is.na(df$ride\_length)))  
cat("\n")  
cat("Unique values in day\_of\_week :- ",unique(df$day\_of\_week))  
cat("\n")  
cat("no of NA values in day\_of\_week :- ",sum(is.na(df$day\_of\_week)))  
cat("\n")  
cat("Unique values in day\_of\_month :- ",unique(df$day\_of\_month))  
cat("\n")  
cat("no of NA values in day\_of\_month :- ",sum(is.na(df$day\_of\_month)))  
cat("\n")

## Minimum ride length :- -31533835  
## Maximum ride length :- 3369132  
## no of NA values in ride\_length :- 0  
## Unique values in day\_of\_week :- 5 2 3 7 6 4 1  
## no of NA values in day\_of\_week :- 0  
## Unique values in day\_of\_month :- 1 2 3 4 5 6 7 8 9 10 11 12  
## no of NA values in day\_of\_month :- 0

As you can see above we cannot have ride length less than 0 so we will remove the rides with ride length less than 0.

cat("Number of rows before deletion :- ",nrow(df))  
cat("\n")  
cat("Minimum ride length before deletion :- ",min(df$ride\_length))  
cat("\n")  
df <- df %>%  
 filter(ride\_length > 0)  
cat("Number of rows after deletion :- ",nrow(df))  
cat("\n")  
cat("Minimum ride length after deletion :- ",min(df$ride\_length))  
cat("\n")

## Number of rows before deletion :- 5595041  
## Minimum ride length before deletion :- -31533835  
## Number of rows after deletion :- 5543742  
## Minimum ride length after deletion :- 1

New data frame summary :-

summary(df)

## ride\_id rideable\_type started\_at   
## Length:5543742 Length:5543742 Min. :2020-01-01 21:00:02   
## Class :character Class :character 1st Qu.:2020-06-07 21:00:18   
## Mode :character Mode :character Median :2020-08-01 21:02:30   
## Mean :2020-07-29 13:26:13   
## 3rd Qu.:2020-09-24 21:16:25   
## Max. :2020-12-31 21:23:53   
##   
## ended\_at start\_station\_name start\_station\_id   
## Min. :2020-01-01 21:00:08 Length:5543742 Length:5543742   
## 1st Qu.:2020-06-07 21:01:17 Class :character Class :character   
## Median :2020-08-01 21:03:02 Mode :character Mode :character   
## Mean :2020-07-29 13:39:01   
## 3rd Qu.:2020-09-24 21:16:43   
## Max. :2020-12-31 21:23:59   
##   
## end\_station\_name end\_station\_id start\_lat start\_lng   
## Length:5543742 Length:5543742 Min. :41.64 Min. :-87.84   
## Class :character Class :character 1st Qu.:41.88 1st Qu.:-87.66   
## Mode :character Mode :character Median :41.90 Median :-87.64   
## Mean :41.90 Mean :-87.65   
## 3rd Qu.:41.93 3rd Qu.:-87.63   
## Max. :42.07 Max. :-87.52   
##   
## end\_lat end\_lng member\_casual ride\_length   
## Min. :41.39 Min. :-88.97 Length:5543742 Length:5543742   
## 1st Qu.:41.88 1st Qu.:-87.66 Class :character Class1:hms   
## Median :41.90 Median :-87.64 Mode :character Class2:difftime   
## Mean :41.90 Mean :-87.65 Mode :numeric   
## 3rd Qu.:41.93 3rd Qu.:-87.63   
## Max. :42.17 Max. :-87.49   
## NA's :4741 NA's :4741   
## day\_of\_week day\_of\_month   
## Min. :1.000 Min. : 1.000   
## 1st Qu.:2.000 1st Qu.: 6.000   
## Median :4.000 Median : 8.000   
## Mean :4.178 Mean : 7.418   
## 3rd Qu.:6.000 3rd Qu.: 9.000   
## Max. :7.000 Max. :12.000   
##

Removing the data which we are not going to use any further

We are going to remove **ride\_id, started\_at, ended\_at, start\_station\_id, end\_station\_id, start\_lat, start\_lng, end\_lat, end\_lng** columns from the data frame.

df <- df %>%  
 select(-ride\_id, -started\_at, -ended\_at, -start\_station\_id, -end\_station\_id, -start\_lat, -start\_lng, -end\_lat, -end\_lng)  
glimpse(df)

## Rows: 5,543,742  
## Columns: 7  
## $ rideable\_type <chr> "electric\_bike", "electric\_bike", "electric\_bike", ~  
## $ start\_station\_name <chr> "California Ave & Cortez St", "California Ave & Cor~  
## $ end\_station\_name <chr> NA, NA, NA, NA, NA, NA, NA, NA, "Wood St & Augusta ~  
## $ member\_casual <chr> "member", "member", "member", "member", "casual", "~  
## $ ride\_length <time> 00:00:10, 00:00:04, 00:00:02, 00:00:11, 00:00:53, ~  
## $ day\_of\_week <dbl> 5, 2, 3, 3, 5, 7, 3, 5, 6, 5, 5, 5, 6, 4, 1, 5, 2, ~  
## $ day\_of\_month <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~

Now lets create an another data frame which is completely removed of null values in start\_station\_name, end\_station\_name columns for any anlysis based on station details.

df2 <- df%>%  
 filter(!is.na(start\_station\_name),!is.na(end\_station\_name))  
glimpse(df2)

## Rows: 4,553,365  
## Columns: 7  
## $ rideable\_type <chr> "classic\_bike", "electric\_bike", "electric\_bike", "~  
## $ start\_station\_name <chr> "California Ave & Cortez St", "California Ave & Cor~  
## $ end\_station\_name <chr> "Wood St & Augusta Blvd", "California Ave & North A~  
## $ member\_casual <chr> "member", "member", "casual", "casual", "casual", "~  
## $ ride\_length <time> 00:00:07, 00:00:05, 00:00:09, 00:00:09, 00:00:11, ~  
## $ day\_of\_week <dbl> 6, 5, 5, 5, 6, 4, 1, 5, 2, 4, 6, 7, 3, 7, 4, 5, 3, ~  
## $ day\_of\_month <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~

Since there is more than a million rows deleted from the data frame we choose to create another data frame for the analysis related to station details.

## Analysis and Visualization

### 1. Analysis

Lets do some descriptive analysis before visualizing.  
Finding total number of casual riders and annual members in the data frame from which we removed all the station names with null values.

member\_counts <- df2%>%  
 count(member\_casual)  
member\_counts

## member\_casual n  
## 1: casual 2037062  
## 2: member 2516303

total\_casual <- member\_counts[1]$n  
total\_members <- member\_counts[2]$n

Now lets calculate the percentage of casual riders in the top 100 stations with high casual riders.

# top 100 start stations with high casual riders  
top\_start <- tabyl(df2,start\_station\_name,member\_casual)  
#top\_start[order(top\_start$casual,decreasing = TRUE),]  
top\_start<- top\_start %>% slice\_max(casual,n=100)  
per\_start <- (sum(top\_start$casual)/total\_casual)\*100  
cat("Percentage of casual riders in 100 start stations with high casual riders :- ",per\_start)  
cat("\n")  
# top 100 end stations with high casual riders  
top\_end <- tabyl(df2,end\_station\_name,member\_casual)  
#top\_end[order(top\_end$casual,decreasing = TRUE),]  
top\_end<- top\_end %>% slice\_max(casual,n=100)  
per\_end <- (sum(top\_end$casual)/total\_casual)\*100  
cat("Percentage of casual riders in 100 end stations with high casual riders :- ",per\_end)  
cat("\n")  
# lets see the total unique stations from both start and end stations  
uni100 <- length(union(top\_start$start\_station\_name,top\_end$end\_station\_name))  
# total number of stations  
total\_station <- length(union(df2$start\_station\_name,df2$end\_station\_name))  
per\_station <- (uni100/total\_station)\*100  
cat("Percentage of stations with more than 50 % casual riders :- ",per\_station)  
cat("\n")

## Percentage of casual riders in 100 start stations with high casual riders :- 54.28676  
## Percentage of casual riders in 100 end stations with high casual riders :- 54.48155  
## Percentage of stations with more than 50 % casual riders :- 12.44076

**Observation:-**  
From the above three statements we can say that **only 12 % of stations are having more than 50% of casual riders**.

# top 100 start stations with high annual members  
top\_start <- tabyl(df2,start\_station\_name,member\_casual)  
#top\_start[order(top\_start$member,decreasing = TRUE),]  
top\_start<- top\_start %>% slice\_max(member,n=100)  
per\_start <- (sum(top\_start$member)/total\_members)\*100  
cat("Percentage of annual members in 100 start stations with high annual members :- ",per\_start)  
cat("\n")  
# top 100 end stations with high annual members  
top\_end <- tabyl(df2,end\_station\_name,member\_casual)  
#top\_end[order(top\_end$member,decreasing = TRUE),]  
top\_end<- top\_end %>% slice\_max(member,n=100)  
per\_end <- (sum(top\_end$member)/total\_members)\*100  
cat("Percentage of annual members in 100 end stations with high annual members :- ",per\_end)  
cat("\n")  
# lets see the total unique stations from both start and end stations  
uni100 <- length(union(top\_start$start\_station\_name,top\_end$end\_station\_name))  
# total number of stations  
total\_station <- length(union(df2$start\_station\_name,df2$end\_station\_name))  
per\_station <- (uni100/total\_station)\*100  
cat("Percentage of stations with more than 50 % annual members :- ",per\_station)  
cat("\n")

## Percentage of annual members in 100 start stations with high annual members :- 48.05657  
## Percentage of annual members in 100 end stations with high annual members :- 48.18259  
## Percentage of stations with more than 50 % annual members :- 12.08531

**Observation:-**  
Annual members are more spread compared to casual members.

Top 10 most chosen ride path by casual riders :-

se <- df2 %>%  
 unite("start\_end",start\_station\_name,end\_station\_name,sep = ' - ')%>%  
 tabyl(start\_end,member\_casual)%>%  
 slice\_max(casual,n=10)  
se$start\_end

## [1] "Streeter Dr & Grand Ave - Streeter Dr & Grand Ave"   
## [2] "Millennium Park - Millennium Park"   
## [3] "Michigan Ave & Oak St - Michigan Ave & Oak St"   
## [4] "Lake Shore Dr & Monroe St - Lake Shore Dr & Monroe St"   
## [5] "Buckingham Fountain - Buckingham Fountain"   
## [6] "Streeter Dr & Grand Ave - Millennium Park"   
## [7] "Theater on the Lake - Theater on the Lake"   
## [8] "DuSable Lake Shore Dr & Monroe St - DuSable Lake Shore Dr & Monroe St"  
## [9] "Montrose Harbor - Montrose Harbor"   
## [10] "Millennium Park - Streeter Dr & Grand Ave"

**Observation:-**  
In the above output we can see most of the start station and end station are similar that means most casual riders take round trips and reach the same station.  
Now lets count the number of round trips taken by casual riders and annual members.

round\_trip <- df2%>%  
 filter(start\_station\_name==end\_station\_name)%>%  
 group\_by(member\_casual)%>%  
 count(member\_casual)  
cat("Percentage of casual riders taking round trips :- ",((round\_trip$n[1])/total\_casual)\*100)  
cat("\n")  
cat("Percentage of annual members taking round trips :- ",((round\_trip$n[2])/total\_members)\*100)  
cat("\n")

## Percentage of casual riders taking round trips :- 10.57798  
## Percentage of annual members taking round trips :- 3.493141

cat("Maximum ride lengths")  
cat("\n")  
print(df%>%group\_by(member\_casual)%>%summarise(minimum\_ride\_length = min(ride\_length)))  
cat("\nAverage ride lengths")  
cat("\n")  
print(df%>%group\_by(member\_casual)%>%summarise(Average\_ride\_length = mean(ride\_length)))  
cat("\nMaximum ride lengths")  
cat("\n")  
print(df%>%group\_by(member\_casual)%>%summarise(maximum\_ride\_length = max(ride\_length)))

## Maximum ride lengths  
## # A tibble: 2 x 2  
## member\_casual minimum\_ride\_length  
## <chr> <drtn>   
## 1 casual 1 secs   
## 2 member 1 secs   
##   
## Average ride lengths  
## # A tibble: 2 x 2  
## member\_casual Average\_ride\_length  
## <chr> <drtn>   
## 1 casual 1328.2519 secs   
## 2 member 306.0473 secs   
##   
## Maximum ride lengths  
## # A tibble: 2 x 2  
## member\_casual maximum\_ride\_length  
## <chr> <drtn>   
## 1 casual 3369132 secs   
## 2 member 171480 secs

cat("Week day with more rides\n")  
getmode <- function(v) {  
 uni <- unique(v)  
 uni[which.max(tabulate(match(v, uni)))]  
}  
getmode(df$day\_of\_week)

## Week day with more rides  
## [1] 6

Since we have marked weekdays from Sunday to Saturday **the week day with more rides is Friday**.

cat("Month with more rides\n")  
getmode(df$day\_of\_month)

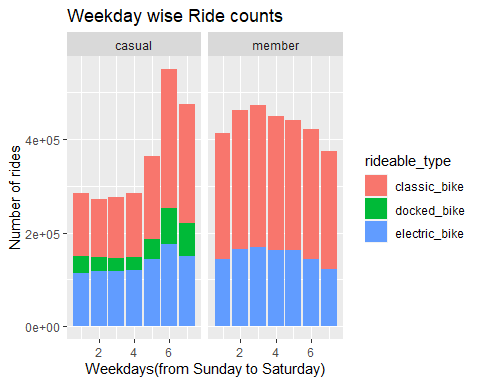
## Month with more rides  
## [1] 7

**July has the more number of rides month wise**.

### 2. Visualization

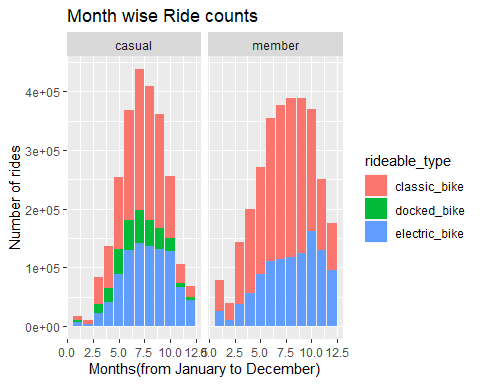
Firstly lets do weekday wise analysis of number of rides taken my members and casuals

ggplot(data = df) +  
 geom\_bar(mapping = aes(x=day\_of\_week,fill=rideable\_type))+  
 facet\_wrap(~member\_casual)+  
 labs(title = "Weekday wise Ride counts", x = "Weekdays(from Sunday to Saturday)", y = "Number of rides")



**Observation:-**  
As we can see from above graph **members usually take more rides on working days (Monday-Friday)** compared to casual riders also **casual riders take more rides on weekends (Saturday, Sunday)**.  
Also we can see that **casual riders use docked bikes more than annual members**.

ggplot(data = df) +  
 geom\_bar(mapping = aes(x=day\_of\_month,fill=rideable\_type))+  
 facet\_wrap(~member\_casual)+  
 labs(title = "Month wise Ride counts", x = "Months(from January to December)", y = "Number of rides")

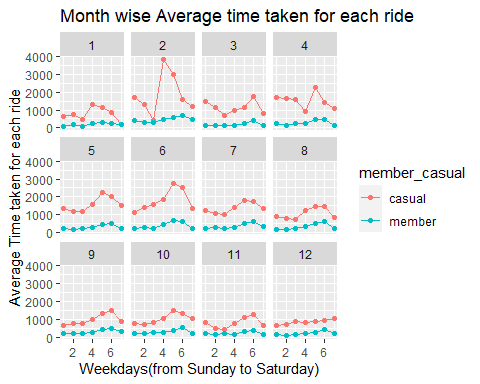


**Observation :-**  
From the above graph we can say that **casual riders use rides mostly in 3rd quarter of the year**.

Now lets plot the difference between the average ride lengths of each ride taken by annual members and casual riders in the days of the week of each month.

# firstly we group the data based on monthly ,weekly, member type details then find the average ride length  
temp <- df%>%  
 group\_by(day\_of\_month,day\_of\_week,member\_casual)%>%  
 summarise(avg\_ride\_length = mean(ride\_length))

ggplot(data = temp, aes(x = day\_of\_week, y = avg\_ride\_length, colour = member\_casual)) +  
 geom\_point() + geom\_line()+  
 facet\_wrap(~day\_of\_month)+  
 labs(title = "Month wise Average time taken for each ride",x = "Weekdays(from Sunday to Saturday)", y = "Average Time taken for each ride")

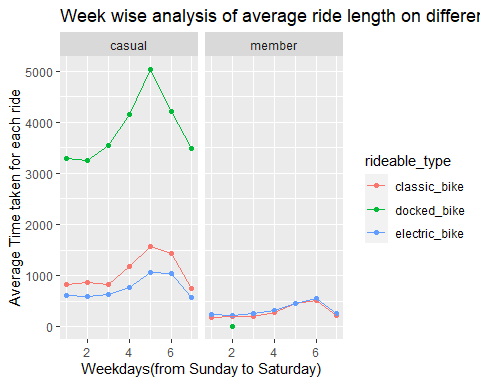


**Observation :-**  
As we can see from the above graph that on an average scale **casual riders take more time for each ride than the annual members**.

Now lets analyze average ride tame taken while using different rideable types.

# firstly we group the data based on week day, member type, rideable type details then find the average ride length  
temp <- df%>%  
 group\_by(day\_of\_week,member\_casual,rideable\_type)%>%  
 summarise(avg\_ride\_length = mean(ride\_length))

ggplot(data = temp, aes(x = day\_of\_week, y = avg\_ride\_length, colour = rideable\_type)) +  
 geom\_point() + geom\_line()+  
 facet\_wrap(~member\_casual)+  
 labs(title = "Week wise analysis of average ride length on different rideable types",x = "Weekdays(from Sunday to Saturday)", y = "Average Time taken for each ride")

 **Observation:-**  
Firstly we see that **annual members usage of docked bikes is very low**.  
On an average casual riders take more time on each rides but **casual riders take the highest time for each ride while using docked bikes**.

## Summary

Difference between casual riders and annual members in the usage of Cyclistic bikes :-

| Casual riders | Annual members |
| --- | --- |
| Average time taken for each ride is larger than annual members. | Average time taken for each ride is low compared to casual riders. |
| Number of rides are more on Friday. | Rides are consistent all over the week. |
| Number of rides are high in 3rd quarter of a year. | Number of rides in 1st,2nd and 4th quarters of the year are high compared to casual riders. |
| More than 50 % of casual riders ride in less than 13 % of the stations. | Annual members are spread more consistently than casual members. |
| Uses all types of rides. | classic and electric bikes are mostly used docked bikes are almost not used at all. |
| Average Time taken on docked bikes is extremely high compared to other ride types. | Average time taken by electric, classic bikes are almost similar. |
| 10% of casual riders take round trips. | Only 3% of annual members take round trips. |

## Recommendations for maximizing annual memberships

1. Reducing ride charges per hour for members can increase the attraction for casual riders.
2. Providing weekend sales on Friday,Saturday and Sunday for members can boost the thought of having a membership in casual riders.
3. Summer sales for members can increase the number of memberships.
4. Instead on annual membership we can provide monthly membership which helps in increasing monthly memberships.
5. Increasing the number advertisements around the top 100 stations with high casual riders can attract more than 50% of the casual riders which also helps in reducing money spent on advertisements.
6. Having special offers in top 100 stations with high casual stations for annual or monthly members also increases the chances of buying a membership.
7. Decreasing hourly charge or cost per ride for annual or monthly members on Docked bikes is also a chance for increasing memberships.
8. Reduce charges for round trips for members.

**Comments :-**  
There are more than a million ride details that does not have start station and end station names or latitude, longitude details.