How Does a Bike-Share Navigate Speedy Success ?

Mahmoud Essam

2022-09-21

Introduction :

Welcome to the Cyclistic bike-share analysis! in this scenario will discuess casual riders and annual members use Cyclistic bikes differently till we can convert casual riders into annual members.

Scenario

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

Installing packages

first of all we will now import the important packages we need for now .

 # Install required packages

 # tidyverse for data import and wrangling

 # lubridate for date functions

 # ggplot for visualization

Importing Packages

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.4
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.0      ✔ stringr 1.4.1
## ✔ readr   2.1.2      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

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

```
library(janitor)
```

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

```
library(skimr)
library(hydroTSM)
```

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

Let's involve into the project

► *download the data we need from this link as it's available by Motivate International Inc. under this license*

► after this step now it's time to put all data into R

```
d1 <- read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202109-divvy-tripdata.csv")
d2 <-  read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202110-divvy-tripdata.csv")
d3 <- read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202111-divvy-tripdata.csv")
d4 <- read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202112-divvy-tripdata.csv")
d5 <-  read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202201-divvy-tripdata.csv")
d6 <- read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202202-divvy-tripdata.csv")
d7 <-  read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202203-divvy-tripdata.csv")
d8 <- read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202204-divvy-tripdata.csv")
d9 <- read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202205-divvy-tripdata.csv")
d10 <- read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202206-divvy-tripdata.csv")
d11 <-  read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202207-divvy-tripdata.csv")
d12 <- read.csv("C:/Users/BLU-RAY/Desktop/Divvy-TripData/Work_Data_Sheets_CSV/Trip_Travles_1_To_12_From_2020_To_2022/Trips_of _last_12_months/202208-divvy-tripdata.csv")
```

► **Now we need to bind all this data into one**

```
Ddata <- rbind(d1,d2,d3,d4,d5,d6,d7,d8,d9,d10,d11,d12)
```

► **make a great view for our current data :**

```
summary(Ddata)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:5883043     Length:5883043     Length:5883043     Length:5883043
##  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:5883043     Length:5883043     Length:5883043     Length:5883043
##  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   :5727    NA's   :5727
##  member_casual
##  Length:5883043
##  Class :character
##  Mode  :character
##
##
##
##
```

```
glimpse(Ddata)
```

```
## Rows: 5,883,043
## Columns: 13
## $ ride_id            <chr> "9DC7B962304CBFD8", "F930E2C6872D6B32", "6EF7213790…
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", …
## $ started_at         <chr> "2021-09-28 16:07:10", "2021-09-28 14:24:51", "2021…
## $ ended_at           <chr> "2021-09-28 16:09:54", "2021-09-28 14:40:05", "2021…
## $ start_station_name <chr> "", "", "", "", "", "", "", "", "", "", "Clark St &…
## $ start_station_id   <chr> "", "", "", "", "", "", "", "", "", "", "TA13070001…
## $ end_station_name   <chr> "", "", "", "", "", "", "", "", "", "", "", "", "",…
## $ end_station_id     <chr> "", "", "", "", "", "", "", "", "", "", "", "", "",…
## $ start_lat          <dbl> 41.89000, 41.94000, 41.81000, 41.80000, 41.88000, 4…
## $ start_lng          <dbl> -87.68000, -87.64000, -87.72000, -87.72000, -87.740…
## $ end_lat            <dbl> 41.89, 41.98, 41.80, 41.81, 41.88, 41.88, 41.74, 41…
## $ end_lng            <dbl> -87.67, -87.67, -87.72, -87.72, -87.71, -87.74, -87…
## $ member_casual      <chr> "casual", "casual", "casual", "casual", "casual", "…
```

```
skim_without_charts(Ddata)
```

Data summary

Name	Ddata
Number of rows	5883043
Number of columns	13
_______________________	
Column type frequency:	
character	9
numeric	4
________________________	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1	16	16	0	5883043	0
rideable_type	0	1	11	13	0	3	0
started_at	0	1	19	19	0	4912339	0
ended_at	0	1	19	19	0	4919944	0
start_station_name	0	1	0	64	884365	1439	0
start_station_id	0	1	0	44	884363	1273	0
end_station_name	0	1	0	64	946303	1453	0
end_station_id	0	1	0	44	946303	1282	0
member_casual	0	1	6	6	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
start_lat	0	1	41.90	0.05	41.64	41.88	41.90	41.93	45.64
start_lng	0	1	-87.65	0.03	-87.84	-87.66	-87.64	-87.63	-73.80
end_lat	5727	1	41.90	0.05	41.39	41.88	41.90	41.93	42.37
end_lng	5727	1	-87.65	0.03	-88.97	-87.66	-87.64	-87.63	-87.50

► **Cleaning our data :**

```
old_NOR <- nrow(Ddata) #to see the number of old data
New_Data <- distinct(Ddata) # to return data without any duplication
New_NOR <- nrow(New_Data) #to see the number after get duplicated
if (old_NOR == New_NOR) {
  print("This data doesn't have any duplicate numbers")
}else{
  print("This data have duplicated number = " + old_NOR-New_NOR)
}
```

```
## [1] "This data doesn't have any duplicate numbers"
```

► **Now we need to return all data we need to use :**

Now will start our time with analyze :

```
unique(Ddata$rideable_type) # to check the unique kinds of rideables
```

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

```
unique(Ddata$member_casual) # to check the unique kinds of riders
```

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

```
Ddata <- Ddata %>% select(2, 3, 4, 13) # the current data required
Ddata$date <- as.Date(Ddata$started_at)#converting it into Date type
Ddata <- Ddata %>% mutate(
      start_year = year(started_at),
      start_month = month(started_at),
      start_day = day(started_at))
Ddata <- Ddata %>%arrange(date) #sorting the data by date
Ddata <- Ddata %>% mutate(season = time2season(date,out.fmt = "seasons")) # Convert dates to seasons
Ddata$day_of_week <- format(as.Date(Ddata$date), "%A") #adding day column
Ddata$day_of_week <- ordered(Ddata$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")) # ordering day of the week
```

But what about if our end date lower than start one ? let's check it !

```
Ddata %>% filter(ended_at < started_at) %>% count() #here will know about the number of negative date
```

```
##     n
## 1 135
```

```
Ddata <- Ddata %>% filter(ended_at>started_at) #assigned the new data without any negative inputs
str(Ddata)
```

```
## 'data.frame':    5882437 obs. of  10 variables:
##  $ rideable_type: chr  "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at   : chr  "2021-09-01 16:32:56" "2021-09-01 15:12:59" "2021-09-01 15:35:38" "2021-09-01 02:16:04" ...
##  $ ended_at     : chr  "2021-09-01 16:49:38" "2021-09-01 15:28:53" "2021-09-01 15:48:42" "2021-09-01 02:23:49" ...
##  $ member_casual: chr  "member" "casual" "casual" "casual" ...
##  $ date         : Date, format: "2021-09-01" "2021-09-01" ...
##  $ start_year   : num  2021 2021 2021 2021 2021 ...
##  $ start_month  : num  9 9 9 9 9 9 9 9 9 9 ...
##  $ start_day    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ season       : chr  "autumm" "autumm" "autumm" "autumm" ...
##  $ day_of_week  : Ord.factor w/ 7 levels "Sunday"<"Monday"<..: 4 4 4 4 4 4 4 4 4 4 ...
```

Now back to our analyze by getting the ride length

```
Ddata <- Ddata %>% mutate(Ride_Length = as.numeric(difftime(Ddata$ended_at,Ddata$started_at)))  # to return the ride length of the bikers in seconds
Ddata <- Ddata %>% mutate(Ride_Length_in_mins = as.numeric(difftime(Ddata$ended_at,Ddata$started_at,units = "mins"))) # to return the ride length of the bikers in minutes
colSums(is.na(Ddata)) #to check the na data
```

```
##       rideable_type          started_at            ended_at       member_casual
##                   0                   0                   0                   0
##                date          start_year         start_month           start_day
##                   0                   0                   0                   0
##              season         day_of_week         Ride_Length Ride_Length_in_mins
##                   0                   0                   0                   0
```

Give a look to our current data to make sense of it :

```
summary(Ddata$Ride_Length) #to know what's the maximum and minimum and other details of our ride
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1     363     643    1185    1160 2442301
```

```
aggregate(Ddata$Ride_Length ~ Ddata$member_casual, FUN = max)# Comparing members and
```

```
##   Ddata$member_casual Ddata$Ride_Length
## 1              casual           2442301
## 2              member             93594
```

```
aggregate(Ddata$Ride_Length ~ Ddata$member_casual, FUN = min)# Comparing members and casual users min
```

```
##   Ddata$member_casual Ddata$Ride_Length
## 1              casual                 1
## 2              member                 1
```

```
aggregate(Ddata$Ride_Length ~ Ddata$member_casual, FUN = median)# Comparing members and casual users median
```

```
##   Ddata$member_casual Ddata$Ride_Length
## 1              casual               835
## 2              member               538
```

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

```
##   Ddata$member_casual Ddata$Ride_Length
## 1              casual         1758.0725
## 2              member          771.3586
```

Now we got a great summary of our current data , now let's pivot it ! :

```
#Step 1 let's know number of rides per day of member and casual
number_of_rides_per_day <- Ddata%>%group_by(member_casual,Ddata$start_day)%>%summarise(number_of_rides_general=n())
```

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

```
number_of_rides_per_day
```

```
## # A tibble: 62 × 3
## # Groups:   member_casual [2]
##    member_casual `Ddata$start_day` number_of_rides_general
##    <chr>                     <int>                   <int>
##  1 casual                        1                   71452
##  2 casual                        2                   79798
##  3 casual                        3                   78103
##  4 casual                        4                   80508
##  5 casual                        5                   90814
##  6 casual                        6                   76515
##  7 casual                        7                   65522
##  8 casual                        8                   59206
##  9 casual                        9                   95279
## 10 casual                       10                   97610
## # … with 52 more rows
```

```
#Step 2 let's know number of rides per month of member and casual
number_of_rides_per_month <- Ddata%>%group_by(member_casual,Ddata$start_month)%>%summarise(number_of_rides_general=n())
```

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

```
number_of_rides_per_month
```

```
## # A tibble: 24 × 3
## # Groups:   member_casual [2]
##    member_casual `Ddata$start_month` number_of_rides_general
##    <chr>                       <dbl>                   <int>
##  1 casual                          1                   18517
##  2 casual                          2                   21414
##  3 casual                          3                   89874
##  4 casual                          4                  126398
##  5 casual                          5                  280387
##  6 casual                          6                  369022
##  7 casual                          7                  406013
##  8 casual                          8                  358886
##  9 casual                          9                  363840
## 10 casual                         10                  257203
## # … with 14 more rows
```

```
#step 3 let's know number of rides with kind of the ridable
number_of_rides_per_ridable <- Ddata %>%group_by(member_casual,Ddata$rideable_type)%>%summarise(number_of_rides_general=n())
```

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

```
number_of_rides_per_ridable
```

```
## # A tibble: 5 × 3
## # Groups:   member_casual [2]
##   member_casual `Ddata$rideable_type` number_of_rides_general
##   <chr>         <chr>                                   <int>
## 1 casual        classic_bike                          1031461
## 2 casual        docked_bike                            207975
## 3 casual        electric_bike                         1228731
## 4 member        classic_bike                          1865008
## 5 member        electric_bike                         1549262
```

```
number_or_rider_per_season <- Ddata%>%group_by(member_casual,Ddata$season)%>%summarise(number_of_rides_general=n())
```

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

```
number_or_rider_per_season
```

```
## # A tibble: 8 × 3
## # Groups:   member_casual [2]
##   member_casual `Ddata$season` number_of_rides_general
##   <chr>         <chr>                            <int>
## 1 casual        autumm                          727927
## 2 casual        spring                          496659
## 3 casual        summer                         1133921
## 4 casual        winter                          109660
## 5 member        autumm                         1019161
## 6 member        spring                          793393
## 7 member        summer                         1244488
## 8 member        winter                          357228
```

after knowing current requirements of riders to know what's the things affect on them, let's test their time in riding :

```
#Calculate ride length per member and casual in day
Avg_Time_per_day <- aggregate(Ddata$Ride_Length_in_mins ~Ddata$member_casual + Ddata$start_day,FUN = mean)
Avg_Time_per_day
```

```
##    Ddata$member_casual Ddata$start_day Ddata$Ride_Length_in_mins
## 1               casual               1                  27.48500
## 2               member               1                  12.32544
## 3               casual               2                  30.93040
## 4               member               2                  12.90488
## 5               casual               3                  31.00935
## 6               member               3                  12.80193
## 7               casual               4                  28.79256
## 8               member               4                  12.67550
## 9               casual               5                  33.48997
## 10              member               5                  13.38879
## 11              casual               6                  32.31350
## 12              member               6                  12.90608
## 13              casual               7                  26.74881
## 14              member               7                  12.56800
## 15              casual               8                  26.14086
## 16              member               8                  12.19394
## 17              casual               9                  31.98205
## 18              member               9                  13.27059
## 19              casual              10                  33.35144
## 20              member              10                  13.35841
## 21              casual              11                  31.04680
## 22              member              11                  12.97376
## 23              casual              12                  29.92877
## 24              member              12                  12.86569
## 25              casual              13                  29.84805
## 26              member              13                  12.73913
## 27              casual              14                  31.04128
## 28              member              14                  13.10473
## 29              casual              15                  30.06442
## 30              member              15                  12.44426
## 31              casual              16                  28.80507
## 32              member              16                  13.07055
## 33              casual              17                  31.57768
## 34              member              17                  12.94526
## 35              casual              18                  28.02731
## 36              member              18                  12.76420
## 37              casual              19                  29.39967
## 38              member              19                  13.29609
## 39              casual              20                  27.84312
## 40              member              20                  12.82546
## 41              casual              21                  29.23942
## 42              member              21                  12.87108
## 43              casual              22                  26.72870
## 44              member              22                  12.35749
## 45              casual              23                  29.67047
## 46              member              23                  13.24159
## 47              casual              24                  26.20994
## 48              member              24                  12.79854
## 49              casual              25                  25.94940
## 50              member              25                  12.66816
## 51              casual              26                  29.30685
## 52              member              26                  12.92902
## 53              casual              27                  26.17286
## 54              member              27                  12.64487
## 55              casual              28                  27.72289
## 56              member              28                  12.56899
## 57              casual              29                  27.86271
## 58              member              29                  12.60325
## 59              casual              30                  27.94567
## 60              member              30                  13.17438
## 61              casual              31                  25.07403
## 62              member              31                  12.99424
```

```
#Calcualte ride length per member and casual in month
Avg_Time_per_month <- aggregate(Ddata$Ride_Length_in_mins~Ddata$member_casual + Ddata$start_month,FUN = mean)
Avg_Time_per_month
```

```
##    Ddata$member_casual Ddata$start_month Ddata$Ride_Length_in_mins
## 1               casual                 1                  30.38238
## 2               member                 1                  11.98179
## 3               casual                 2                  26.71095
## 4               member                 2                  11.40589
## 5               casual                 3                  32.62536
## 6               member                 3                  11.95894
## 7               casual                 4                  29.53687
## 8               member                 4                  11.49297
## 9               casual                 5                  30.87270
## 10              member                 5                  13.36743
## 11              casual                 6                  32.10152
## 12              member                 6                  14.00039
## 13              casual                 7                  29.28123
## 14              member                 7                  13.71940
## 15              casual                 8                  29.31318
## 16              member                 8                  13.38542
## 17              casual                 9                  27.81882
## 18              member                 9                  13.73658
## 19              casual                10                  28.67832
## 20              member                10                  12.50311
## 21              casual                11                  23.13548
## 22              member                11                  11.31012
## 23              casual                12                  23.49732
## 24              member                12                  11.00568
```

```
#Calculate ride length per member and casual in type of ridable
Avg_Time_per_Ride <- aggregate(Ddata$Ride_Length_in_mins~Ddata$member_casual + Ddata$rideable_type,FUN = mean)
Avg_Time_per_Ride
```

```
##   Ddata$member_casual Ddata$rideable_type Ddata$Ride_Length_in_mins
## 1              casual        classic_bike                  28.66531
## 2              member        classic_bike                  13.79840
## 3              casual         docked_bike                 103.11653
## 4              casual       electric_bike                  17.34103
## 5              member       electric_bike                  11.72148
```

```
#Calculate ride length per member and casual in season
Avg_Time_per_season <- aggregate(Ddata$Ride_Length_in_mins~Ddata$member_casual+Ddata$season,FUN = mean)
Avg_Time_per_season
```

```
##   Ddata$member_casual Ddata$season Ddata$Ride_Length_in_mins
## 1              casual       autumm                  27.43484
## 2              member       autumm                  12.68162
## 3              casual       spring                  30.84989
## 4              member       spring                  12.44435
## 5              casual       summer                  30.20917
## 6              member       summer                  13.69516
## 7              casual       winter                  25.28747
## 8              member       winter                  11.34414
```

in our last destination now it's time for some little visualize so we are in need to export this data now for tableau for some viz with excel !

```
#Exporting our data out
write.csv(Ddata,"C:/Users/BLU-RAY/Desktop/Divvy-TripData/Result Data/Total_Data.csv")
write.csv(number_of_rides_per_day,"C:/Users/BLU-RAY/Desktop/Divvy-TripData/Result Data/number_of_rides_per_day.csv")
write.csv(number_of_rides_per_month,"C:/Users/BLU-RAY/Desktop/Divvy-TripData/Result Data/number_of_rides_per_month.csv")
write.csv(number_of_rides_per_ridable,"C:/Users/BLU-RAY/Desktop/Divvy-TripData/Result Data/number_of_rides_per_ridable.csv")
write.csv(number_or_rider_per_season,"C:/Users/BLU-RAY/Desktop/Divvy-TripData/Result Data/number_of_rides_per_season.csv")
write.csv(Avg_Time_per_day,"C:/Users/BLU-RAY/Desktop/Divvy-TripData/Result Data/Avg_time_per_day.csv")
write.csv(Avg_Time_per_month,"C:/Users/BLU-RAY/Desktop/Divvy-TripData/Result Data/Avg_time_per_month.csv")
write.csv(Avg_Time_per_Ride,"C:/Users/BLU-RAY/Desktop/Divvy-TripData/Result Data/Avg_Time_per_Ride.csv")
write.csv(Avg_Time_per_season,"C:/Users/BLU-RAY/Desktop/Divvy-TripData/Result Data/Avg_Time_per_Season.csv")
```