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

**Introduction:**

This is about the case study performed as a part of Google Data Analytics Certifications to demonstrate my knowledge and skill learned through the course. Considering myself as a junior data analyst at Cyclistic, a bike-share company going to perform analysis to support the hypothesis that the company’s future success depends on maximizing the number of annual memberships.

**Key highlights about the company:**

A bike-share program that features more than 5,800 bicycles and 600 docking stations.

Types of bikes offered:

* reclining bikes
* hand tricycles
* cargo bikes

The majority of riders opt for traditional bikes; about 8% of riders use the assistive options.

Cyclistic users are more likely to ride for leisure, but about 30% use them to commute to work each day.

In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago.

The customers categorised based on the passes purchased:

* Casual riders: customers who purchases single-ride and full-day passes.
* Cyclistic Members: customers who purchases annual memberships.

Company’s finance analyst found that annual members are more profitable than casual riders.

Moreno, the director of marketing set a goal:

* To convert casual riders into annual members.
* How annual members and casual riders differ and why casual riders would buy a membership.

Implementing the different phases of data analysis process:

**1. ASK:** Ask the questions to identify, understand and define the problem.

Key tasks –

* Identify the **business task**: Design marketing strategies aimed at converting casual riders into annual members. Questions to consider -
  1. How annual members and casual riders differ?
  2. Why casual riders would buy a membership?
  3. How digital media could affect their marketing tactics?
* Consider **key stakeholders**:
  1. LilyMoreno - director of marketing and manager: who sets the goal.
  2. Cyclistic marketing analystics team: responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy.
  3. Cyclistic executive team: will decide whether to approve the recommended marketing program.

**2. PREPARE:** Collecting the data, understanding the metrics of data and storing data.

* Using the Cyclistic’s previous 12 months historical trip data.
* Instead of using the latest months data considered 2021 data i.e., from 2021 January to 2021 December to analyse the customers data annually.
* The data has been downloaded from the AWS data storage and saved into my local computer folders following naming conventions with all safety measures.
* The data is organized according to the year-month wise in .CSV/Excel format.
* Each file contains 13 columns/attributes.
* Looking at the size of data, seems it includes the day-to-day data inputs which makes us to consider that the data is Reliable, Original, Comprehensive, Current and cited.
* The data has been made available by Motivate International Inc. under license. Allowed to explore different customers using the bikes data but data-privacy issues prohibit us from using riders’ personally identifiable information.

**Limitations found in the data**: After going through different files the following are the observation made.

1. Each trip data is recorded with ride\_id as unique id, but no attribute for customer unique id at least annual members. This customer unique Id data would help us to find count of individual or repeated customers etc.
2. Bike station Ids strings are inconsistent which impacts the data integrity.
3. Found station names, Ids details are missing in few records.

**3. PROCESS:** Cleaning the data to maintain the data integrity before we start analyzing it.

Data can be cleaned and manipulated as individual .CSV file but when the files are merged during data processing the size of all datasets together is 995MB. So, handling such huge data in excel would be impossible. So, R is being used for cleaning, managing, manipulation and analyzing data using the RStudio IDE.

#=========================

# STEP 1: LOADING PACKAGES

#=========================

library(tidyverse #helps wrangle data

library(lubridate) #helps wrangle date attributes

library(ggplot2) #helps visualize data

library(janitor)#for cleaning data

library(skimr) #for summarizing data easy

library(dplyr) # for manipulating datasets

library(ggmap) # to retrieve raster map tiles from online mapping services

getwd() #displays your working directory

setwd("C:/Users/Sowjanya KAKE/OneDrive/Desktop/DAC/tripdata-CSVfiles")

#=====================

# STEP 2: COLLECT DATA

#=====================

*#Upload Divvy datasets (csv files) here*

df\_202101 <- read.csv("202101-divvy-tripdata.csv")

df\_202102 <- read.csv("202102-divvy-tripdata.csv")

df\_202103 <- read.csv("202103-divvy-tripdata.csv")

df\_202104 <- read.csv("202104-divvy-tripdata.csv")

df\_202105 <- read.csv("202105-divvy-tripdata.csv")

df\_202106 <- read.csv("202106-divvy-tripdata.csv")

df\_202107 <- read.csv("202107-divvy-tripdata.csv")

df\_202108 <- read.csv("202108-divvy-tripdata.csv")

df\_202109 <- read.csv("202109-divvy-tripdata.csv")

df\_202110 <- read.csv("202110-divvy-tripdata.csv")

df\_202111 <- read.csv("202111-divvy-tripdata.csv")

df\_202112 <- read.csv("202112-divvy-tripdata.csv")

#====================================================

# STEP 3: WRANGLE DATA AND COMBINE INTO A SINGLE FILE

#====================================================

*3.1: Compare column names each of the files*

While the names don't have to be in the same order, they DO need to match perfectly before we can use a command to join them into one file

compare\_df\_cols(df\_202101, df\_202102, df\_202103, df\_202104, df\_202105, df\_202106, df\_202107, df\_202108, df\_202109, df\_202110, df\_202111, df\_202112)

#(OR)

compare\_df\_cols(df\_202101, df\_202102, df\_202103, df\_202104, df\_202105, df\_202106, df\_202107, df\_202108, df\_202109, df\_202110, df\_202111, df\_202112, return = "mismatch")

[1] column\_name df\_202101 df\_202102 df\_202103 df\_202104 df\_202105 df\_202106 df\_202107 df\_202108 df\_202109 df\_202110 df\_202111 df\_202112

<0 rows> (or 0-length row.names)

*3.2: Stack individual data frames into one big data frame*

all\_trips\_df <- bind\_rows(df\_202101, df\_202102, df\_202103, df\_202104, df\_202105, df\_202106, df\_202107, df\_202108, df\_202109, df\_202110, df\_202111, df\_202112)

*3.3: Rename columns to make them consistent*

all\_trips\_df <- rename(all\_trips\_df,

trip\_id = ride\_id,

bike\_type = rideable\_type,

start\_time = started\_at,

end\_time = ended\_at,

from\_station\_name = start\_station\_name,

from\_station\_id = start\_station\_id,

to\_station\_name = end\_station\_name,

to\_station\_id = end\_station\_id,

user\_type = member\_casual)

#======================================================

# STEP 4: CLEAN UP AND ADD DATA TO PREPARE FOR ANALYSIS

#======================================================

*4.1: List of column names*

colnames(all\_trips\_df)

[1] "trip\_id" "bike\_type" "start\_time" "end\_time" "from\_station\_name"

[6] "from\_station\_id" "to\_station\_name" "to\_station\_id" "start\_lat" "start\_lng"

[11] "end\_lat"

*4.2: How many rows are in data frame?*

nrow(all\_trips\_df)

[1] 5595063

*4.3: Dimensions of the data frame?*

dim(all\_trips\_df)

[1] 5595063 13

*4.4: See the first and last 6 rows of data frame.*

head(all\_trips\_df)

trip\_id bike\_type start\_time end\_time from\_station\_name

1 E19E6F1B8D4C42ED electric\_bike 2021-01-23 16:14:19 2021-01-23 16:24:44 California Ave & Cortez St

2 DC88F20C2C55F27F electric\_bike 2021-01-27 18:43:08 2021-01-27 18:47:12 California Ave & Cortez St

3 EC45C94683FE3F27 electric\_bike 2021-01-21 22:35:54 2021-01-21 22:37:14 California Ave & Cortez St

4 4FA453A75AE377DB electric\_bike 2021-01-07 13:31:13 2021-01-07 13:42:55 California Ave & Cortez St

5 BE5E8EB4E7263A0B electric\_bike 2021-01-23 02:24:02 2021-01-23 02:24:45 California Ave & Cortez St

6 5D8969F88C773979 electric\_bike 2021-01-09 14:24:07 2021-01-09 15:17:54 California Ave & Cortez St

from\_station\_id to\_station\_name to\_station\_id start\_lat start\_lng end\_lat end\_lng user\_type

1 17660 41.90034 -87.69674 41.89 -87.72 member

2 17660 41.90033 -87.69671 41.90 -87.69 member

3 17660 41.90031 -87.69664 41.90 -87.70 member

4 17660 41.90040 -87.69666 41.92 -87.69 member

5 17660 41.90033 -87.69670 41.90 -87.70 casual

6 17660 41.90041 -87.69676 41.94 -87.71 casual

tail(all\_trips\_df)

trip\_id bike\_type start\_time end\_time from\_station\_name

5595058 92BBAB97D1683D69 electric\_bike 2021-12-24 15:42:09 2021-12-24 19:29:35 Canal St & Madison St

5595059 847431F3D5353AB7 electric\_bike 2021-12-12 13:36:55 2021-12-12 13:56:08 Canal St & Madison St

5595060 CF407BBC3B9FAD63 electric\_bike 2021-12-06 19:37:50 2021-12-06 19:44:51 Canal St & Madison St

5595061 60BB69EBF5440E92 electric\_bike 2021-12-02 08:57:04 2021-12-02 09:05:21 Canal St & Madison St

5595062 C414F654A28635B8 electric\_bike 2021-12-13 09:00:26 2021-12-13 09:14:39 Lawndale Ave & 16th St

5595063 37AC57E34B2E7E97 classic\_bike 2021-12-13 08:45:32 2021-12-13 08:49:09 Michigan Ave & Jackson Blvd

from\_station\_id to\_station\_name to\_station\_id start\_lat start\_lng end\_lat end\_lng user\_type

5595058 13341 41.88180 -87.63997 41.88000 -87.64000 casual

5595059 13341 41.88229 -87.63975 41.89000 -87.61000 casual

5595060 13341 Kingsbury St & Kinzie St KA1503000043 41.88212 -87.64005 41.88911 -87.63886 member

5595061 13341 Dearborn St & Monroe St TA1305000006 41.88196 -87.63995 41.88025 -87.62960 member

5595062 362.0 41.86000 -87.72000 41.85000 -87.71000 member

5595063 TA1309000002 Dearborn St & Monroe St TA1305000006 41.87785 -87.62408 41.88132 -87.62952 member

*4.5: See list of columns and data types*

str(all\_trips\_df)

'data.frame': 5595063 obs. of 13 variables:

$ trip\_id : chr "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA453A75AE377DB" ...

$ bike\_type : chr "electric\_bike" "electric\_bike" "electric\_bike" "electric\_bike" ...

$ start\_time : chr "2021-01-23 16:14:19" "2021-01-27 18:43:08" "2021-01-21 22:35:54" "2021-01-07 13:31:13" ...

$ end\_time : chr "2021-01-23 16:24:44" "2021-01-27 18:47:12" "2021-01-21 22:37:14" "2021-01-07 13:42:55" ...

$ from\_station\_name: chr "California Ave & Cortez St" "California Ave & Cortez St" "California Ave & Cortez St" "California Ave & Cortez St" ...

$ from\_station\_id : chr "17660" "17660" "17660" "17660" ...

$ to\_station\_name : chr "" "" "" "" ...

$ to\_station\_id : chr "" "" "" "" ...

$ start\_lat : num 41.9 41.9 41.9 41.9 41.9 ...

$ start\_lng : num -87.7 -87.7 -87.7 -87.7 -87.7 ...

$ end\_lat : num 41.9 41.9 41.9 41.9 41.9 ...

$ end\_lng : num -87.7 -87.7 -87.7 -87.7 -87.7 ...

$ user\_type : chr "member" "member" "member" "member" ...

*4.6: Statistical summary of data. Mainly for numerics*

summary(all\_trips\_df)

trip\_id bike\_type start\_time end\_time from\_station\_name

Length:5595063 Length:5595063 Length:5595063 Length:5595063 Length:5595063

Class :character Class :character Class :character Class :character Class :character

Mode :character Mode :character Mode :character Mode :character Mode :character

from\_station\_id to\_station\_name to\_station\_id start\_lat start\_lng end\_lat

Length:5595063 Length:5595063 Length:5595063 Min. :41.64 Min. :-87.84 Min. :41.39

Class :character Class :character Class :character 1st Qu.:41.88 1st Qu.:-87.66 1st Qu.:41.88

Mode :character Mode :character Mode :character Median :41.90 Median :-87.64 Median :41.90

Mean :41.90 Mean :-87.65 Mean :41.90

3rd Qu.:41.93 3rd Qu.:-87.63 3rd Qu.:41.93

Max. :42.07 Max. :-87.52 Max. :42.17

NA's :4771

end\_lng user\_type

Min. :-88.97 Length:5595063

1st Qu.:-87.66 Class :character

Median :-87.64 Mode :character

Mean :-87.65

3rd Qu.:-87.63

Max. :-87.49

NA's :4771

*4.7: There are a few problems we will need to fix:*

(a). In the "user\_type" column, checking for distinct values to modify in case found any other name other than member and casual.

unique(all\_trips\_df$user\_type)

[1] "member" "casual"

(b). Add columns that list the date, month, day, and year of each ride

This will allow us to aggregate ride data for each month, day, or year ... before completing these operations we could only aggregate at the ride level.

*4.8: Add columns that list the date, month, day, and year of each ride*

all\_trips\_df$date <- as.Date(all\_trips\_df$start\_time)

View(all\_trips\_df)

all\_trips\_df$month <- format(as.Date(all\_trips\_df$date), "%m")

all\_trips\_df$day <- format(as.Date(all\_trips\_df$date), "%d")

all\_trips\_df$year <- format(as.Date(all\_trips\_df$date), "%Y")

all\_trips\_df$day\_of\_week <- format(as.Date(all\_trips\_df$date), "%A")

(c) We will want to add a calculated field for length of ride since the 2020Q1 data did not have the "tripduration" column. We will add "ride\_length" to the entire dataframe for consistency.

*4.9: Add a "ride\_length" calculation to all\_trips (in seconds)*

all\_trips\_df$ride\_length <- difftime(all\_trips\_df$end\_time, all\_trips\_df$start\_time, units = "mins")

*4.10: Convert "ride\_length" from Factor to numeric so we can run calculations on the data*

is.numeric(all\_trips\_df$ride\_length)

[1] FALSE

all\_trips\_df$ride\_length <- as.numeric(as.character(all\_trips\_df$ride\_length))

is.numeric(all\_trips\_df$ride\_length)

[1] TRUE

*4.11: Finding duplicate records:*

get\_dupes(all\_trips\_df, trip\_id)

No duplicate combinations found of: trip\_id

[1] trip\_id dupe\_count bike\_type start\_time end\_time from\_station\_name

[7] from\_station\_id to\_station\_name to\_station\_id start\_lat start\_lng end\_lat

[13] end\_lng user\_type date month day year

[19] day\_of\_week ride\_length

<0 rows> (or 0-length row.names)

*4.12: Summary of the data frame:*

skim\_without\_charts(all\_trips\_df)

── Data Summary ────────────────────────

Values

Name all\_trips\_df

Number of rows 5595063

Number of columns 19

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Column type frequency:

character 13

Date 1

difftime 1

numeric 4

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Group variables None

── Variable type: character ──────────────────────────────────────────────────────────────────────────────────────────

skim\_variable n\_missing complete\_rate min max empty n\_unique whitespace

1 trip\_id 0 1 16 16 0 5595063 0

2 bike\_type 0 1 11 13 0 3 0

3 start\_time 0 1 19 19 0 4677998 0

4 end\_time 0 1 19 19 0 4671372 0

5 from\_station\_name 0 1 0 53 690809 848 0

6 from\_station\_id 0 1 0 36 690806 835 0

7 to\_station\_name 0 1 0 53 739170 845 0

8 to\_station\_id 0 1 0 36 739170 833 0

9 user\_type 0 1 6 6 0 2 0

10 month 0 1 2 2 0 12 0

11 day 0 1 2 2 0 31 0

12 year 0 1 4 4 0 1 0

13 day\_of\_week 0 1 6 9 0 7 0

── Variable type: Date ───────────────────────────────────────────────────────────────────────────────────────────────

skim\_variable n\_missing complete\_rate min max median n\_unique

1 date 0 1 2021-01-01 2021-12-31 2021-08-01 365

── Variable type: difftime ───────────────────────────────────────────────────────────────────────────────────────────

skim\_variable n\_missing complete\_rate min max median n\_unique

1 ride\_length 0 1 -58.03333 mins 55944.15 mins 12 mins 25645

── Variable type: numeric ────────────────────────────────────────────────────────────────────────────────────────────

skim\_variable n\_missing complete\_rate mean sd p0 p25 p50 p75 p100

1 start\_lat 0 1 41.9 0.0461 41.6 41.9 41.9 41.9 42.1

2 start\_lng 0 1 -87.6 0.0287 -87.8 -87.7 -87.6 -87.6 -87.5

3 end\_lat 4771 0.999 41.9 0.0462 41.4 41.9 41.9 41.9 42.2

4 end\_lng 4771 0.999 -87.6 0.0289 -89.0 -87.7 -87.6 -87.6 -87.5

*4.13: Remove "bad" data*

percentiles <- quantile(all\_trips\_df$ride\_length, probs = seq(0, 1, 0.01))

all\_trips\_v2 <- all\_trips\_df[!(all\_trips\_df$ride\_length < 0|

all\_trips\_df$ride\_length >= percentiles[100]|

all\_trips\_df$from\_station\_name == "" |

all\_trips\_df$from\_station\_id == "" |

all\_trips\_df$to\_station\_name == "" |

all\_trips\_df$to\_station\_id == ""),]

*4.14: Checking summary of final dataset:*

skim(all\_trips\_v2)

── Data Summary ────────────────────────

Values

Name all\_trips\_v2

Number of rows 4541990

Number of columns 19

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Column type frequency:

character 13

Date 1

numeric 5

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Group variables None

── Variable type: character ──────────────────────────────────────────────────────────────────────────────────────────

skim\_variable n\_missing complete\_rate min max empty n\_unique whitespace

1 trip\_id 0 1 16 16 0 4541990 0

2 bike\_type 0 1 11 13 0 3 0

3 start\_time 0 1 19 19 0 3900322 0

4 end\_time 0 1 19 19 0 3892829 0

5 from\_station\_name 0 1 3 53 0 842 0

6 from\_station\_id 0 1 3 36 0 829 0

7 to\_station\_name 0 1 10 53 0 839 0

8 to\_station\_id 0 1 3 36 0 827 0

9 user\_type 0 1 6 6 0 2 0

10 month 0 1 2 2 0 12 0

11 day 0 1 2 2 0 31 0

12 year 0 1 4 4 0 1 0

13 day\_of\_week 0 1 6 9 0 7 0

── Variable type: Date ───────────────────────────────────────────────────────────────────────────────────────────────

skim\_variable n\_missing complete\_rate min max median n\_unique

1 date 0 1 2021-01-01 2021-12-31 2021-07-28 365

── Variable type: numeric ────────────────────────────────────────────────────────────────────────────────────────────

skim\_variable n\_missing complete\_rate mean sd p0 p25 p50 p75 p100 hist

1 start\_lat 0 1 41.9 0.0402 41.6 41.9 41.9 41.9 42.1 ▁▁▇▇▁

2 start\_lng 0 1 -87.6 0.0233 -87.8 -87.7 -87.6 -87.6 -87.5 ▁▁▅▇▁

3 end\_lat 0 1 41.9 0.0404 41.6 41.9 41.9 41.9 42.1 ▁▁▇▅▁

4 end\_lng 0 1 -87.6 0.0235 -87.8 -87.7 -87.6 -87.6 -87.5 ▁▁▅▇▁

5 ride\_length 0 1 17.6 17.6 0 6.9 12.1 21.6 130. ▇▂▁▁▁

*4.15: Exporting the data:*

write.csv(all\_trips\_v2, "all\_trips.csv")

**4. ANALYSE:**

#=====================================

# STEP 5: CONDUCT DESCRIPTIVE ANALYSIS

#=====================================

*#* *Attributes to consider for analysis:*

Here we are trying to answer the question:

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

User types are members and casual riders.

Variables to consider for analyzing user types:

* 1. Rides count
  2. Types of bikes
  3. Most visited stations
  4. Geospatial locations
  5. Ride duration (Ride length)

Based on the above attributes, we can perform descriptive analysis along with the visualizations using R.

**I. Analysis based on rides count:**

(A) **Compare user type by ride count**:

*Dataset:*

all\_trips\_v2 %>%

group\_by(user\_type) %>%

summarise(number\_of\_rides = n())

# A tibble: 2 × 2

user\_type number\_of\_rides

<chr> <int>

1 casual 2005230

2 member 2536760

*Visualization using column chart:*

all\_trips\_v2 %>%

group\_by(user\_type) %>%

summarise(number\_of\_rides = n()) %>%

ggplot(aes(user\_type, number\_of\_rides, fill = user\_type))+

geom\_col(position = "dodge")+

labs(title = "Number of rides taken by casual and member riders", x = "User types", y = "Number of rides")

Chart, bar chart

Description automatically generated

(B). **Riders percentage as a whole**:

*Dataset:*

all\_trips\_v2 %>%

group\_by(user\_type) %>%

summarise(number\_of\_rides = n()) %>%

mutate(perc = number\_of\_rides / sum(number\_of\_rides), labels = scales::percent(perc))

# A tibble: 2 × 4

user\_type number\_of\_rides perc labels

<chr> <int> <dbl> <chr>

1 casual 2005230 0.441 44%

2 member 2536760 0.559 56%

*Visualization using pie chart:*

all\_trips\_v2 %>%

group\_by(user\_type) %>%

summarise(number\_of\_rides = n()) %>%

mutate(perc = number\_of\_rides / sum(number\_of\_rides), labels = scales::percent(perc)) %>%

ggplot(aes("", number\_of\_rides, fill = user\_type))+

geom\_col()+

geom\_label(aes(label = labels),

position = position\_stack(vjust = 0.5),

show.legend = FALSE) +

coord\_polar("y") +

labs(title = "Rides by users", x = "", y = "") +

theme(axis.text.x = element\_blank())

Chart, pie chart

Description automatically generated

(C.1). **Rides count by hour of the day**

*Dataset:*

all\_trips\_v2 %>%

mutate(hour = hour(start\_time)) %>%

group\_by(user\_type, hour) %>%

summarise(number\_of\_rides = n(), .groups = "drop")

# A tibble: 48 × 3

user\_type hour number\_of\_rides

<chr> <int> <int>

1 casual 0 41489

2 casual 1 30061

3 casual 2 19161

4 casual 3 9954

5 casual 4 6531

6 casual 5 8678

7 casual 6 19137

8 casual 7 35557

9 casual 8 48791

10 casual 9 59264

# … with 38 more rows

*Visualization using line chart:*

all\_trips\_v2 %>%

mutate(hour = hour(start\_time)) %>%

group\_by(user\_type, hour) %>%

summarise(number\_of\_rides = n(), .groups = "drop") %>%

ggplot(aes(hour,number\_of\_rides, color = user\_type))+

geom\_line()+

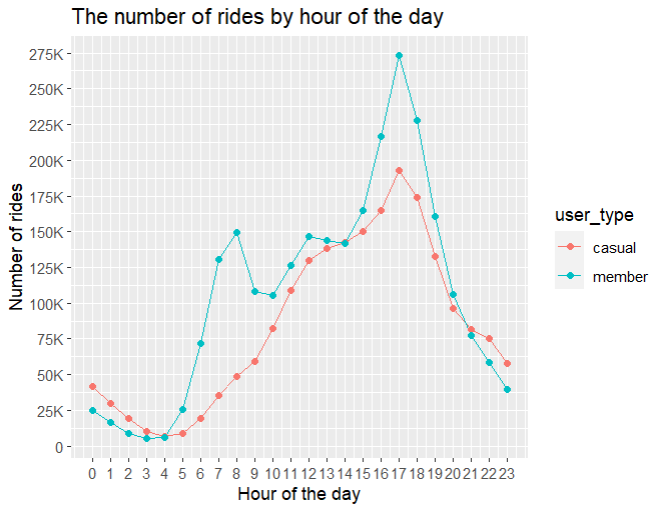
geom\_point()+

scale\_x\_continuous(breaks = seq(0,23,1))+

scale\_y\_continuous(labels = scales::label\_number\_si(),

breaks = seq(0,300000,25000))+

labs(title = "Number of rides by hour", x="Hour of the day", y="Number of rides")

**

(C.2). **Rides count by hour of the day throughout the week**

all\_trips\_v2 %>%

mutate(hour = hour(start\_time)) %>%

group\_by(user\_type, hour, day\_of\_week) %>%

summarise(number\_of\_rides = n(), .groups = "drop") %>%

ggplot(aes(hour,number\_of\_rides, color = user\_type))+

geom\_line()+

geom\_point()+

scale\_x\_continuous(breaks = seq(0,23,1))+

scale\_y\_continuous(labels = scales::label\_number\_si(), breaks = seq(0,300000,25000))+

labs(title = "Number of rides by hour", x="Hour of the day", y="Number of rides")+

facet\_wrap(~ day\_of\_week)

*Visualization using line chart using facet\_wrap:*

Histogram

Description automatically generated

(D). **Rides count by days of the week**

Fetching the weekdays from date field:

all\_trips\_v2$day\_of\_week <- format(as.Date(all\_trips\_v2$date), "%A")

Setting the order of the week:

all\_trips\_v2$day\_of\_week <- ordered(all\_trips\_v2$day\_of\_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))

*Dataset:*

all\_trips\_v2 %>%

group\_by(user\_type, day\_of\_week) %>%

summarise(number\_of\_rides = n(),.groups = "drop")

# A tibble: 14 × 3

user\_type day\_of\_week number\_of\_rides

<chr> <ord> <int>

1 casual Sunday 392600

2 casual Monday 224034

3 casual Tuesday 211201

4 casual Wednesday 214701

5 casual Thursday 220660

6 casual Friday 284967

7 casual Saturday 457067

8 member Sunday 310627

9 member Monday 346127

10 member Tuesday 387780

11 member Wednesday 397307

12 member Thursday 373125

13 member Friday 365359

14 member Saturday 356435

*Visualization using column chart:*

all\_trips\_v2 %>%

group\_by(user\_type, day\_of\_week) %>%

summarise(number\_of\_rides = n(),.groups = "drop") %>%

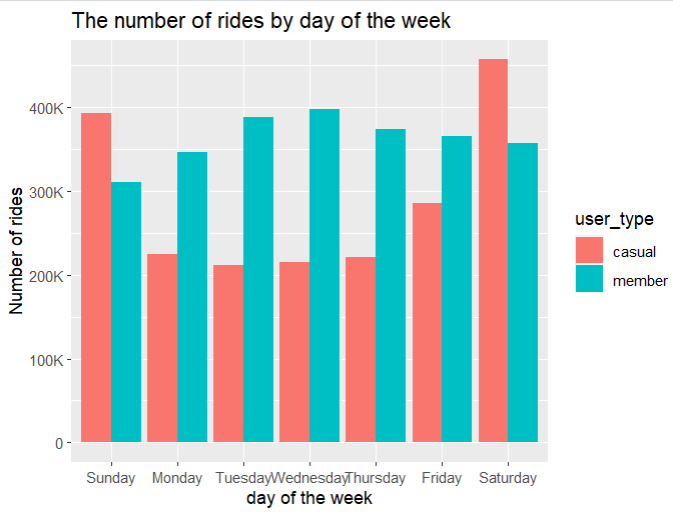
ggplot(aes(day\_of\_week, number\_of\_rides, fill = user\_type)) +

geom\_col(position = "dodge") +

scale\_y\_continuous(labels = scales::label\_number\_si(),

breaks = seq(0, 500000, 100000)) +

labs(title = "The number of rides by day of the week", x = "day of the week", y = "Number of rides")



**(E). Rides count by day of the month**

*Dataset:*

all\_trips\_v2 %>%

group\_by(user\_type, day = as.numeric(day)) %>%

summarise(number\_of\_ride = n(),.groups = "drop")

# A tibble: 62 × 3

user\_type day number\_of\_ride

<chr> <dbl> <int>

1 casual 1 67763

2 casual 2 67594

3 casual 3 72215

4 casual 4 75181

5 casual 5 79445

6 casual 6 74461

7 casual 7 63551

8 casual 8 59363

9 casual 9 67140

10 casual 10 64774

# … with 52 more rows

*Visualization using line chart:*

all\_trips\_v2 %>%

group\_by(user\_type, day = as.numeric(day)) %>%

summarise(number\_of\_ride = n(),.groups = "drop") %>%

ggplot(aes(day, number\_of\_ride, color = user\_type)) +

geom\_line() +

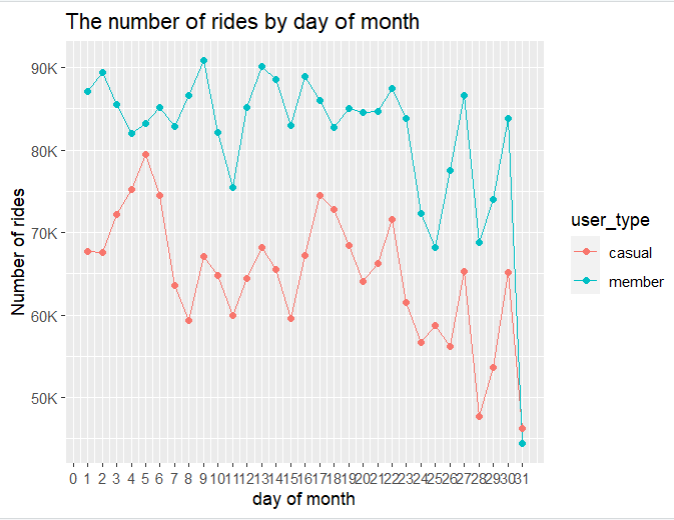
geom\_point() +

scale\_y\_continuous(labels = scales::label\_number\_si(),

breaks = seq(0, 100000, 10000)) +

scale\_x\_continuous(breaks = seq(0, 31, 1)) +

labs(title = "The number of rides by day of month", x = "day of month", y = "Number of rides")



**(F). Rides count by month**

*Dataset:*

all\_trips\_v2 %>%

mutate(month = month(start\_time, label = TRUE)) %>%

group\_by(user\_type, month) %>%

summarise(number\_of\_rides = n(),.groups = "drop") %>%

arrange(month)

# A tibble: 24 × 3

user\_type month number\_of\_rides

<chr> <ord> <int>

1 casual Jan 14484

2 member Jan 68742

3 casual Feb 8364

4 member Feb 34275

5 casual Mar 73891

6 member Mar 129899

7 casual Apr 117424

8 member Apr 177499

9 casual May 210689

10 member May 233828

# … with 14 more rows

*Visualization using column chart:*

all\_trips\_v2 %>%

mutate(month = month(start\_time, label = TRUE)) %>%

group\_by(user\_type, month) %>%

summarise(number\_of\_rides = n(),.groups = "drop") %>%

arrange(month) %>%

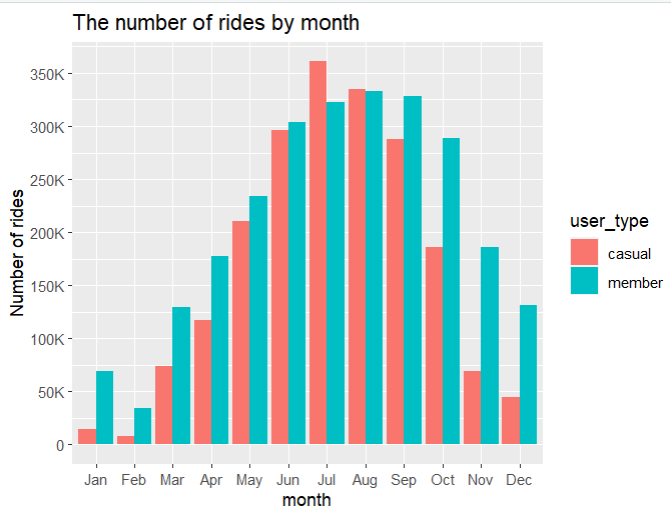
ggplot(aes(month, number\_of\_rides, fill = user\_type)) +

geom\_col(position = "dodge") +

scale\_y\_continuous(labels = scales::label\_number\_si(),

breaks = seq(0, 450000, 50000)) +

labs(title = "The number of rides by month", x = "month", y = "Number of rides")



**II. Analysis based on type of bikes:**

**(A) Number of rides taken by casual and member riders:**

*Dataset:*

all\_trips\_v2 %>%

group\_by(user\_type, bike\_type) %>%

summarise(number\_of\_rides = n())

# A tibble: 6 × 3

# Groups: user\_type [2]

user\_type bike\_type number\_of\_rides

<chr> <chr> <int>

1 casual classic\_bike 1242017

2 casual docked\_bike 290811

3 casual electric\_bike 472402

4 member classic\_bike 1978118

5 member docked\_bike 1

6 member electric\_bike 558641

*Visualization using Grouped column chart:*

all\_trips\_v2 %>%

group\_by(user\_type, bike\_type) %>%

summarise(number\_of\_rides = n()) %>%

ggplot(aes(user\_type, number\_of\_rides, fill = bike\_type))+

geom\_col(position = "dodge")+

labs(title = "Number of rides taken by casual and member riders", x = "User types", y = "Number of rides")

Chart, bar chart

Description automatically generated

*Visualization using stacked column chart:*

all\_trips\_v2 %>%

group\_by(user\_type, bike\_type) %>%

summarise(number\_of\_rides = n()) %>%

ggplot(aes(bike\_type, number\_of\_rides, fill = user\_type ))+

geom\_col(position = "stack")+

#geom\_bar()+

labs(title = "Rides count based on bike types ", x = "Bike types", y = "Number of rides")

Chart, bar chart

Description automatically generated

**(B) Rides count percentage by bike types:**

*Dataset:*

all\_trips\_v2 %>%

group\_by(bike\_type) %>%

summarise(number\_of\_ride = n()) %>%

mutate(perc = number\_of\_ride / sum(number\_of\_ride),

labels = scales::percent(perc))

# A tibble: 3 × 4

bike\_type number\_of\_ride perc labels

<chr> <int> <dbl> <chr>

1 classic\_bike 3220135 0.709 71%

2 docked\_bike 290812 0.0640 6%

3 electric\_bike 1031043 0.227 23%

*Visualization using pie chart:*

all\_trips\_v2 %>%

group\_by(bike\_type) %>%

summarise(number\_of\_ride = n()) %>%

mutate(perc = number\_of\_ride / sum(number\_of\_ride), labels = scales::percent(perc)) %>%

ggplot(aes("", perc, fill = bike\_type, )) +

geom\_col() +

geom\_label(aes(label = labels), position = position\_stack(vjust = 0.5), show.legend = FALSE)

+ coord\_polar(theta = "y") +

labs(title = "Bike type usage (percentage)", x = "", y = "") +

theme(axis.text.x=element\_blank())

Chart, pie chart

Description automatically generated

**(C) Percentage of rides based on user and bike types:**

*Dataset:*

all\_trips\_v2 %>%

group\_by(user\_type, bike\_type) %>%

summarise(number\_of\_rides = n()) %>%

mutate(perc = number\_of\_rides / sum(number\_of\_rides), labels = scales::percent(perc))

# A tibble: 6 × 5

# Groups: user\_type [2]

user\_type bike\_type number\_of\_rides perc labels

<chr> <chr> <int> <dbl> <chr>

1 casual classic\_bike 1242017 0.619 61.9%

2 casual docked\_bike 290811 0.145 14.5%

3 casual electric\_bike 472402 0.236 23.6%

4 member classic\_bike 1978118 0.780 78%

5 member docked\_bike 1 0.000000394 0%

6 member electric\_bike 558641 0.220 22%

*Visualization using Heatmap:*

all\_trips\_v2 %>%

group\_by(user\_type, bike\_type) %>%

summarise(number\_of\_rides = n()) %>%

mutate(perc = number\_of\_rides / sum(number\_of\_rides), labels = scales::percent(perc)) %>%

ggplot(aes(user\_type, bike\_type, fill = number\_of\_rides ))+

geom\_tile() +

geom\_label(aes(label = labels), #position = position\_stack(vjust = 0.5),

show.legend = FALSE)+

labs(title = "Rides count based on bike types ", x = "User types", y = "Bike types")

Chart

Description automatically generated

**III. Most Visited stations:**

**(A) Total number of stations:**

length(unique(as.vector(as.matrix(data.frame(all\_trips\_v2$from\_station\_name, all\_trips\_v2$to\_station\_name)))))

[1] 845

**(OR)**

all\_trips\_v2 %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

distinct(station\_name) %>%

summarise(count\_station = n())

count\_station

1. 845

**(B) Top most visited stations:**

*Dataset:*

all\_trips\_v2 %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_of\_rides = n()) %>%

arrange(desc(number\_of\_rides)) %>%

slice(1:10)

# A tibble: 10 × 2

station\_name number\_of\_rides

<chr> <int>

1 Streeter Dr & Grand Ave 158311

2 Michigan Ave & Oak St 84488

3 Wells St & Concord Ln 83292

4 Millennium Park 79648

5 Clark St & Elm St 78002

6 Wells St & Elm St 71668

7 Theater on the Lake 71041

8 Kingsbury St & Kinzie St 63749

9 Clark St & Lincoln Ave 63520

10 Clark St & Armitage Ave 62029

*Visualization using column chart:*

all\_trips\_v2 %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_of\_rides = n()) %>%

arrange(desc(number\_of\_rides)) %>%

slice(1:10) %>%

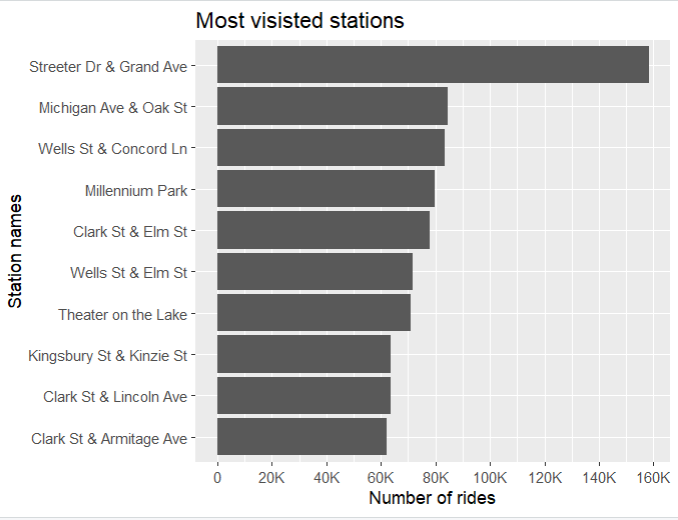
ggplot(aes(number\_of\_rides, reorder(station\_name,number\_of\_rides)))+

geom\_col()+

scale\_x\_continuous(labels = scales::label\_number\_si(),

breaks = seq(0, 200000, 20000))+

labs(title = "Most visisted stations", x="Number of rides", y="Station names")

****

**(C) Top 10 most visited stations by members riders:**

*Dataset:*

all\_trips\_v2 %>%

filter(user\_type == "member") %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n()) %>%

arrange(desc(number\_rides)) %>%

slice(1:10)

# A tibble: 10 × 2

station\_name number\_rides

<chr> <int>

1 Clark St & Elm St 47809

2 Wells St & Concord Ln 46131

3 Kingsbury St & Kinzie St 45488

4 Wells St & Elm St 41008

5 Dearborn St & Erie St 37765

6 St. Clair St & Erie St 36374

7 Wells St & Huron St 36087

8 Broadway & Barry Ave 34467

9 Clinton St & Madison St 33173

10 Clark St & Armitage Ave 31699

*Visualization using column chart:*

all\_trips\_v2 %>%

filter(user\_type == "member") %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n()) %>%

arrange(desc(number\_rides)) %>%

slice(1:10) %>%

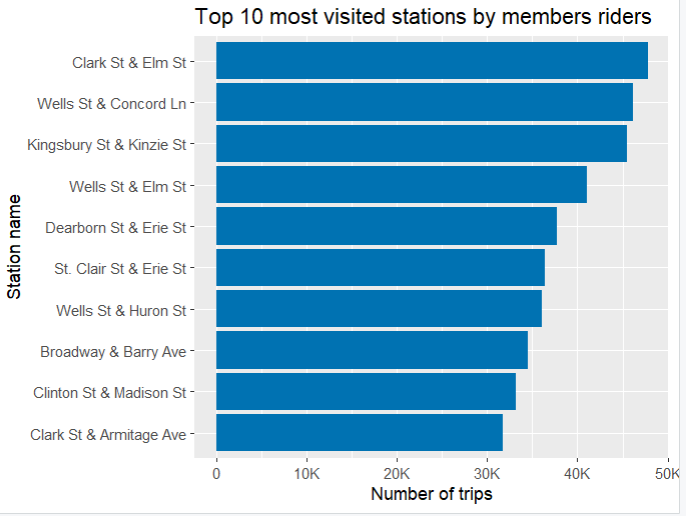
ggplot(aes(number\_rides, reorder(station\_name, number\_rides)))+

geom\_col(fill = "#0072B2")+

scale\_x\_continuous(labels = scales::label\_number\_si(),

breaks = seq(0, 130000, 10000))+

labs(title = "Top 10 most visited stations by members riders", x = "Number of trips", y = "Station name")



**(D) Top 10 most visited stations by casual riders:**

*Dataset:*

all\_trips\_v2 %>%

filter(user\_type == "casual") %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n()) %>%

arrange(desc(number\_rides)) %>%

slice(1:10)

# A tibble: 10 × 2

station\_name number\_rides

<chr> <int>

1 Streeter Dr & Grand Ave 128231

2 Millennium Park 63367

3 Michigan Ave & Oak St 57293

4 Shedd Aquarium 42868

5 Theater on the Lake 41920

6 Wells St & Concord Ln 37161

7 Lake Shore Dr & Monroe St 35812

8 Clark St & Lincoln Ave 32353

9 Wabash Ave & Grand Ave 30911

10 Lake Shore Dr & North Blvd 30773

*Visualization using column chart:*

all\_trips\_v2 %>%

filter(user\_type == "casual") %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n()) %>%

arrange(desc(number\_rides)) %>%

slice(1:10) %>%

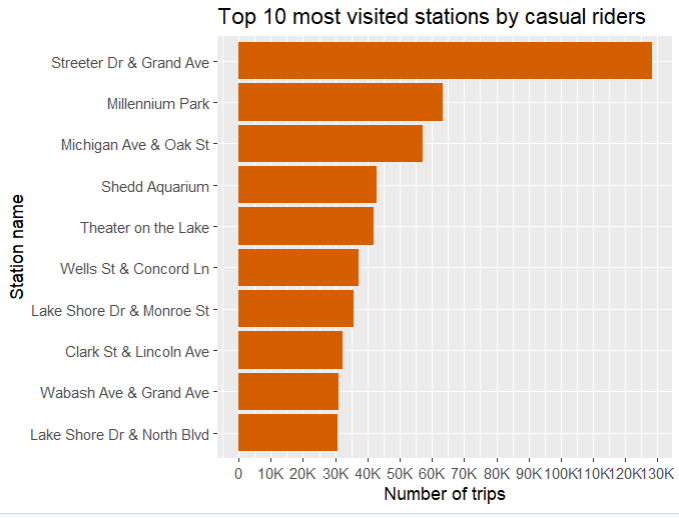
ggplot(aes(number\_rides, reorder(station\_name, number\_rides)))+

geom\_col(fill = "#D55E00")+

scale\_x\_continuous(labels = scales::label\_number\_si(),

breaks = seq(0, 130000, 10000))+

labs(title = "Top 10 most visited stations by casual riders", x = "Number of trips", y = "Station name")



**IV. Geospatial locations**

**(A) Geographical representation of all stations:**

*Dataset:*

all\_trips\_v2 %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n(),

latitude = first(start\_lat),

longitude = first(start\_lng))

# A tibble: 845 × 4

station\_name number\_rides latitude longitude

<chr> <int> <dbl> <dbl>

1 2112 W Peterson Ave 1835 42.0 -87.7

2 351 1 41.9 -87.8

3 63rd St Beach 3861 41.8 -87.6

4 900 W Harrison St 16192 41.9 -87.6

5 Aberdeen St & Jackson Blvd 22853 41.9 -87.7

6 Aberdeen St & Monroe St 21461 41.9 -87.7

7 Aberdeen St & Randolph St 19492 41.9 -87.7

8 Ada St & 113th St 97 41.7 -87.7

9 Ada St & Washington Blvd 17495 41.9 -87.7

10 Adler Planetarium 29110 41.9 -87.6

# … with 835 more rows

*Spatial visualization using Stamen map with ggmap:*

stations\_map <- all\_trips\_v2 %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n(), latitude = first(start\_lat), longitude = first(start\_lng))

height <- max(stations\_map$latitude) - min(stations\_map$latitude)

width <- max(stations\_map$longitude) - min(stations\_map$longitude)

borders <- c(bottom = min(stations\_map$latitude) - 0.1 \* height,

top = max(stations\_map$latitude) + 0.1 \* height,

left = min(stations\_map$longitude) - 0.1 \* width,

right = max(stations\_map$longitude) + 0.1 \* width)

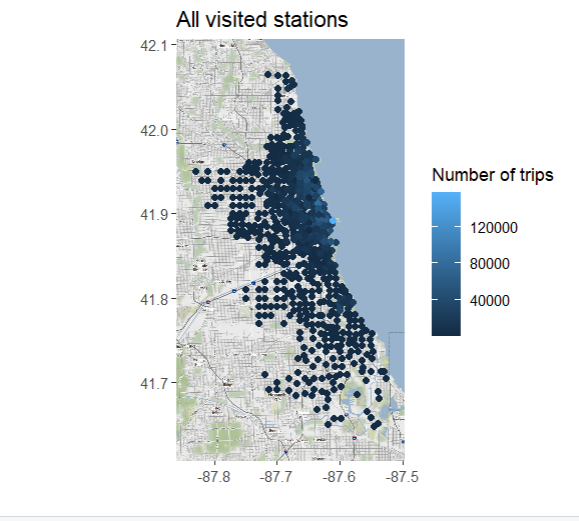
map <- get\_stamenmap(borders, zoom = 12, maptype = "terrain")

ggmap(map) +

geom\_point(data = stations\_map, mapping = aes(x = longitude, y = latitude,

col = number\_rides)) +

labs(title = "All visited stations", x= "", y = "", color = "Number of trips")



Now considering approximately 10% of the stations.

**(B) Top 80 most visited stations**

*Dataset:*

all\_trips\_v2 %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n(),

latitude = first(start\_lat),

longitude = first(start\_lng)) %>%

arrange(desc(number\_rides)) %>%

slice(1:80)

# A tibble: 80 × 4

station\_name number\_rides latitude longitude

<chr> <int> <dbl> <dbl>

1 Streeter Dr & Grand Ave 158311 41.9 -87.6

2 Michigan Ave & Oak St 84488 41.9 -87.6

3 Wells St & Concord Ln 83292 41.9 -87.6

4 Millennium Park 79648 41.9 -87.6

5 Clark St & Elm St 78002 41.9 -87.6

6 Wells St & Elm St 71668 41.9 -87.6

7 Theater on the Lake 71041 41.9 -87.6

8 Kingsbury St & Kinzie St 63749 41.9 -87.6

9 Clark St & Lincoln Ave 63520 41.9 -87.6

10 Clark St & Armitage Ave 62029 41.9 -87.6

# … with 70 more rows

*Spatial visualization using Stamen map with ggmap:*

stations\_map <- all\_trips\_v2 %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n(),

latitude = first(start\_lat),

longitude = first(start\_lng)) %>%

arrange(desc(number\_rides)) %>%

slice(1:80)

height <- max(stations\_map$latitude) - min(stations\_map$latitude)

width <- max(stations\_map$longitude) - min(stations\_map$longitude)

borders <- c(bottom = min(stations\_map$latitude) - 0.1 \* height,

top = max(stations\_map$latitude) + 0.1 \* height,

left = min(stations\_map$longitude) - 0.1 \* width,

right = max(stations\_map$longitude) + 0.1 \* width)

map <- get\_stamenmap(borders, zoom = 15, maptype = "terrain")

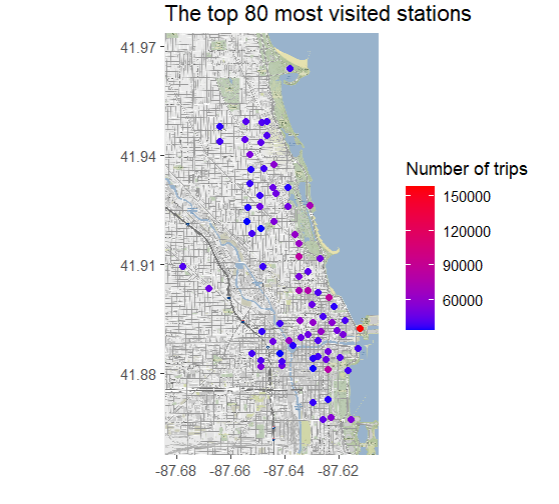
ggmap(map) +

geom\_point(data = stations\_map, mapping = aes(x = longitude, y = latitude,

col = number\_rides)) +

scale\_color\_gradient(low = "blue", high = "red") +

labs(title = "The top 80 most visited stations", x= "", y = "", color = "Number of trips")

****

**(C) Top 10 stations visited by casual riders**

*Dataset:*

all\_trips\_v2 %>%

filter(user\_type == "casual") %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n(),

latitude = first(start\_lat),

longitude = first(start\_lng)) %>%

arrange(desc(number\_rides)) %>%

slice(1:10)

# A tibble: 80 × 4

station\_name number\_rides latitude longitude

<chr> <int> <dbl> <dbl>

1 Streeter Dr & Grand Ave 128231 41.9 -87.6

2 Millennium Park 63367 41.9 -87.6

3 Michigan Ave & Oak St 57293 41.9 -87.6

4 Shedd Aquarium 42868 41.9 -87.6

5 Theater on the Lake 41920 41.9 -87.6

6 Wells St & Concord Ln 37161 41.9 -87.6

7 Lake Shore Dr & Monroe St 35812 41.9 -87.6

8 Clark St & Lincoln Ave 32353 41.9 -87.6

9 Wabash Ave & Grand Ave 30911 41.9 -87.6

10 Lake Shore Dr & North Blvd 30773 41.9 -87.6

# … with 70 more rows

*Spatial visualization using Stamen map with ggmap:*

stations\_map <- all\_trips\_v2 %>%

filter(user\_type == "casual") %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n(),

latitude = first(start\_lat),

longitude = first(start\_lng)) %>%

arrange(desc(number\_rides)) %>%

slice(1:10)

height <- max(stations\_map$latitude) - min(stations\_map$latitude)

width <- max(stations\_map$longitude) - min(stations\_map$longitude)

borders <- c(bottom = min(stations\_map$latitude) - 0.1 \* height,

top = max(stations\_map$latitude) + 0.1 \* height,

left = min(stations\_map$longitude) - 0.1 \* width,

right = max(stations\_map$longitude) + 0.1 \* width)

map <- get\_stamenmap(borders, zoom = 15, maptype = "terrain")

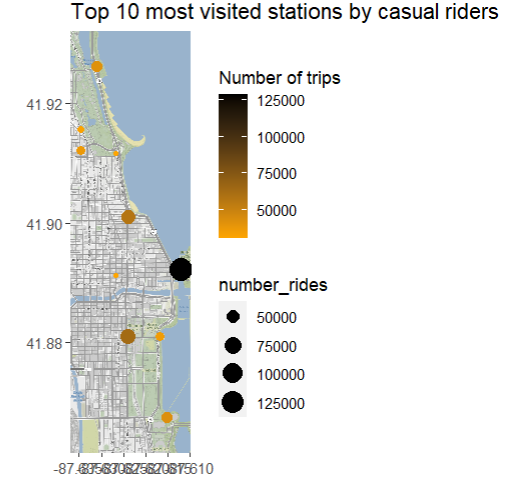
ggmap(map) +

geom\_point(data = stations\_map, mapping = aes(x = longitude, y = latitude,

col = number\_rides, size = number\_rides)) +

scale\_color\_gradient(low = "orange", high = "black") +

labs(title = "Top 10 most visited stations by casual riders", x= "", y = "", color = "Number of trips")

****

**(D) Top 10 stations visited by member riders**

*Dataset:*

all\_trips\_v2 %>%

filter(user\_type == "member") %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n(),

latitude = first(start\_lat),

longitude = first(start\_lng)) %>%

arrange(desc(number\_rides)) %>%

slice(1:10)

# A tibble: 80 × 4

station\_name number\_rides latitude longitude

<chr> <int> <dbl> <dbl>

1 Clark St & Elm St 47809 41.9 -87.6

2 Wells St & Concord Ln 46131 41.9 -87.6

3 Kingsbury St & Kinzie St 45488 41.9 -87.6

4 Wells St & Elm St 41008 41.9 -87.6

5 Dearborn St & Erie St 37765 41.9 -87.6

6 St. Clair St & Erie St 36374 41.9 -87.6

7 Wells St & Huron St 36087 41.9 -87.6

8 Broadway & Barry Ave 34467 41.9 -87.6

9 Clinton St & Madison St 33173 41.9 -87.6

10 Clark St & Armitage Ave 31699 41.9 -87.6

# … with 70 more rows

*Spatial visualization using Stamen map with ggmap:*

stations\_map <- all\_trips\_v2 %>%

filter(user\_type == "member") %>%

gather(key, station\_name, from\_station\_name, to\_station\_name) %>%

group\_by(station\_name) %>%

summarise(number\_rides = n(),

latitude = first(start\_lat),

longitude = first(start\_lng)) %>%

arrange(desc(number\_rides)) %>%

slice(1:10)

height <- max(stations\_map$latitude) - min(stations\_map$latitude)

width <- max(stations\_map$longitude) - min(stations\_map$longitude)

borders <- c(bottom = min(stations\_map$latitude) - 0.1 \* height,

top = max(stations\_map$latitude) + 0.1 \* height,

left = min(stations\_map$longitude) - 0.1 \* width,

right = max(stations\_map$longitude) + 0.1 \* width)

map <- get\_stamenmap(borders, zoom = 15, maptype = "terrain")

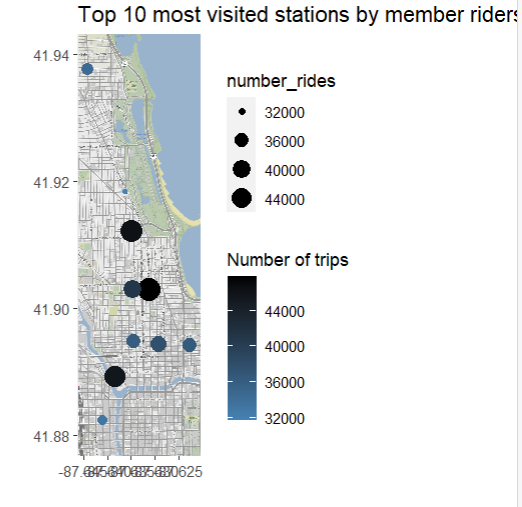
ggmap(map) +

geom\_point(data = stations\_map, mapping = aes(x = longitude, y = latitude,

col = number\_rides, size = number\_rides)) +

scale\_color\_gradient(low = "yellow", high = "red") +

labs(title = "Top 10 most visited stations by member riders", x= "", y = "", color = "Number of trips")



**V. Analysis based on rides length (trip duration):**

**(A) Ride length distribution charts:**

**1.** Distribution of ride length using Histogram:

*Dataset:*

summary(all\_trips\_v2$ride\_length)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.00 6.90 12.08 17.63 21.63 129.68

*Visualization using Histogram:*

m = mean(all\_trips\_v2$ride\_length)

md = median(all\_trips\_v2$ride\_length)

# Distribution of ride\_length using Histogram:

all\_trips\_v2 %>%

ggplot(aes(ride\_length))+

#geom\_histogram(color = "white", fill = "steelblue")+

geom\_histogram(bins = 10)+

scale\_x\_continuous(labels = scales::label\_number\_si(),

breaks = seq(0, 200, 10))+

geom\_vline(xintercept = m, col = "green", lwd = 2) +

geom\_vline(xintercept = md, col = "red", lwd = 2) +

annotate("text",

x = m \* 3.5,

y = m \* 100000,

label = paste("Mean =", m),

col = "blue",

size = 4)+

annotate("text",

x = md \* 5,

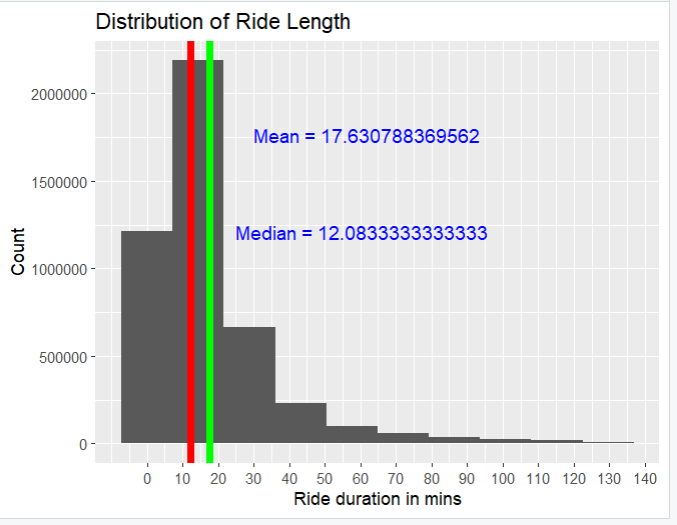
y = md \* 100000,

label = paste("Median =", md),

col = "blue",

size = 4)+

labs(title = "Distribution of Ride Length", x = "Ride duration in mins", y = "Count")



**2.** Distribution of ride length for user type:

*Dataset:*

aggregate(all\_trips\_v2$ride\_length ~ all\_trips\_v2$user\_type, FUN = mean)

all\_trips\_v2$user\_type all\_trips\_v2$ride\_length

1 casual 23.71077

2 member 12.82475

aggregate(all\_trips\_v2$ride\_length ~ all\_trips\_v2$user\_type, FUN = median)

all\_trips\_v2$user\_type all\_trips\_v2$ride\_length

1 casual 16.26667

2 member 9.70000

aggregate(all\_trips\_v2$ride\_length ~ all\_trips\_v2$user\_type, FUN = max)

all\_trips\_v2$user\_type all\_trips\_v2$ride\_length

1 casual 129.6833

2 member 129.6000

aggregate(all\_trips\_v2$ride\_length ~ all\_trips\_v2$user\_type, FUN = min)

all\_trips\_v2$user\_type all\_trips\_v2$ride\_length

1 casual 0

2 member 0

*Visualization* using violin chart*:*

all\_trips\_v2 %>%

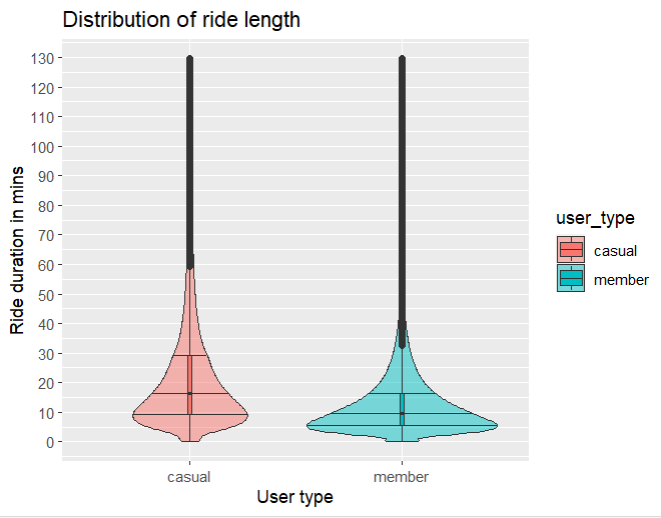
ggplot(aes(x= user\_type, y = ride\_length, fill = user\_type))+

geom\_violin(draw\_quantiles = c(.25, .50, .75), alpha = 0.5) + geom\_boxplot(width = 0.02) +

scale\_y\_continuous(labels = scales::label\_number\_si(),

breaks = seq(0, 200, 10))+

labs(title = "Distribution of ride length", x = "User type", y = "Ride duration in mins")



**(B) Average ride length by hour of the day**

*Dataset:*

all\_trips\_v2 %>%

mutate(hour = hour(start\_time)) %>%

group\_by(user\_type, hour) %>%

summarise(avg\_ride\_duration = mean(ride\_length))

# A tibble: 48 × 3

# Groups: user\_type [2]

user\_type hour avg\_ride\_duration

<chr> <int> <dbl>

1 casual 0 21.2

2 casual 1 20.5

3 casual 2 20.2

4 casual 3 20.2

5 casual 4 18.1

6 casual 5 16.6

7 casual 6 15.3

8 casual 7 16.2

9 casual 8 18.7

10 casual 9 23.4

# … with 38 more rows

*Visualization using line chart:*

all\_trips\_v2 %>%

mutate(hour = hour(start\_time)) %>%

group\_by(user\_type, hour) %>%

summarise(avg\_ride\_duration = mean(ride\_length)) %>%

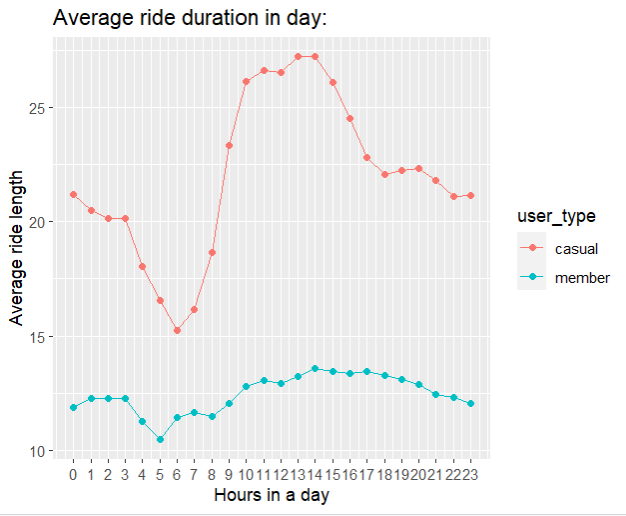
ggplot(aes(hour, avg\_ride\_duration, color = user\_type))+

geom\_line()+

geom\_point()+

scale\_x\_continuous(breaks = seq(0, 23, 1)) +

labs(title = "Average ride duration in day:", x = "Hours in a day", y = "Average ride length")

****

**(C) Average ride length by days of week**

*Dataset:*

all\_trips\_v2 %>%

mutate(day\_week = wday(start\_time, label = TRUE)) %>%

group\_by(user\_type, day\_week) %>%

summarise(avg\_ride\_duration = mean(ride\_length)) %>%

arrange(day\_week)

# A tibble: 14 × 3

# Groups: user\_type [2]

user\_type day\_week avg\_ride\_duration

<chr> <ord> <dbl>

1 casual Sun 26.9

2 casual Mon 24.2

3 casual Tue 21.8

4 casual Wed 21.0

5 casual Thu 20.5

6 casual Fri 22.2

7 casual Sat 25.4

8 member Sun 14.7

9 member Mon 12.4

10 member Tue 12.1

11 member Wed 12.2

12 member Thu 12.1

13 member Fri 12.5

14 member Sat 14.3

*Visualization using column chart:*

all\_trips\_v2 %>%

mutate(day\_week = wday(start\_time, label = TRUE)) %>%

group\_by(user\_type, day\_week) %>%

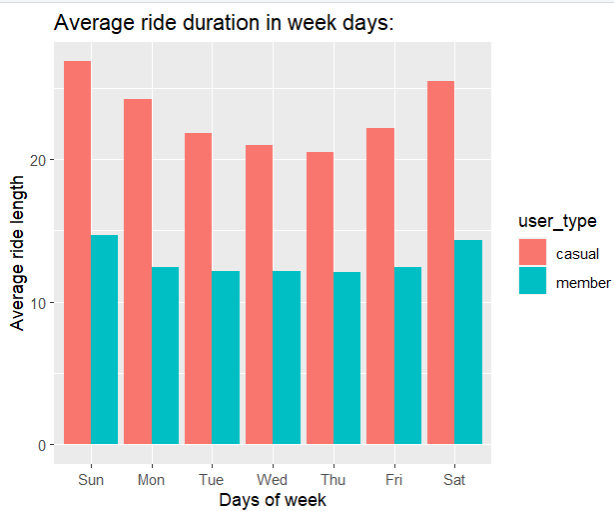
summarise(avg\_ride\_duration = mean(ride\_length)) %>%

arrange(day\_week) %>%

ggplot(aes(day\_week, avg\_ride\_duration, fill = user\_type))+

geom\_col(position = "dodge")+

labs(title = "Average ride duration in week days:", x = "Days of week", y = "Average ride length")



**(D) Average ride length by day of month**

*Dataset:*

all\_trips\_v2 %>%

group\_by(user\_type, day = as.numeric(day)) %>%

summarise(avg\_ride\_duration = mean(ride\_length))

# A tibble: 62 × 3

# Groups: user\_type [2]

user\_type day avg\_ride\_duration

<chr> <dbl> <dbl>

1 casual 1 24.0

2 casual 2 24.5

3 casual 3 25.0

4 casual 4 25.8

5 casual 5 25.6

6 casual 6 25.0

7 casual 7 23.5

8 casual 8 22.9

9 casual 9 23.5

10 casual 10 23.0

# … with 52 more rows

*Visualization using line chart:*

all\_trips\_v2 %>%

group\_by(user\_type, day = as.numeric(day)) %>%

summarise(avg\_ride\_duration = mean(ride\_length)) %>%

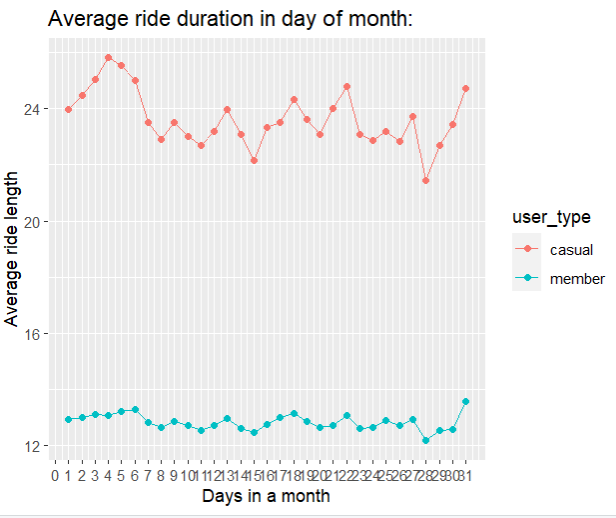
ggplot(aes(day, avg\_ride\_duration, color = user\_type))+

geom\_line()+

geom\_point()+

scale\_x\_continuous(breaks = seq(0, 31, 1)) +

labs(title = "Average ride duration in day of month:", x = "Days in a month", y = "Average ride length")



**(E) Average ride length by month**

*Dataset:*

all\_trips\_v2 %>%

mutate(month = month(start\_time, label = TRUE)) %>%

group\_by(user\_type, month) %>%

summarise(avg\_ride\_duration = mean(ride\_length)) %>%

arrange(month)

# A tibble: 24 × 3

# Groups: user\_type [2]

user\_type month avg\_ride\_duration

<chr> <ord> <dbl>

1 casual Jan 18.5

2 member Jan 11.7

3 casual Feb 23.4

4 member Feb 13.5

5 casual Mar 26.9

6 member Mar 13.3

7 casual Apr 26.6

8 member Apr 13.8

9 casual May 27.2

10 member May 13.9

# … with 14 more rows

*Visualization using column chart:*

all\_trips\_v2 %>%

mutate(month = month(start\_time, label = TRUE)) %>%

group\_by(user\_type, month) %>%

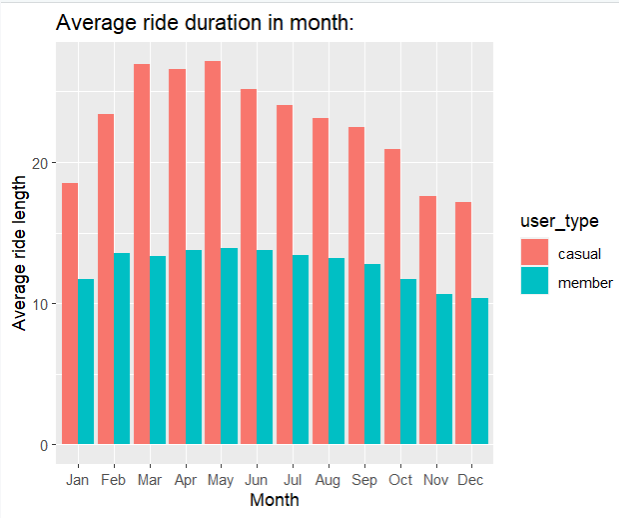
summarise(avg\_ride\_duration = mean(ride\_length)) %>%

arrange(month) %>%

ggplot(aes(month, avg\_ride\_duration, fill = user\_type))+

geom\_col(position = "dodge")+

labs(title = "Average ride duration in month:", x = "Month", y = "Average ride length")



**5. SHARE:**

After performing the required calculations and analysis over different datasets of both riders’ data, below are the observations made:

*a) Ride count:*

1. Member ride count of 2536760 is high –

*casual*: 2005230

*member*: **2536760**

2. Percentage of members ride count of 56% is high –

*casual*: 44%

*member*: **56%**

3. Members rides count by hour is max at evening rush hours–

*casual*: Min - 4:00 Hrs, Max - 17:00 Hrs

*member*: Min - 3:00 Hrs, Max - **17:00 Hrs**

4. Casual riders ride count by day of week is max on Saturday and Sunday

Member rides count seems consistent throughout the week and peak during mid of week–

casual: High – **Saturday**, Low – Tuesday

member: High – Wednesday, Low – Sunday

5. Members rides count is high on 9th day of month –

casual: Min – 31st day, Max – 5th day

member: Min – 31st day, Max – **9th day**

6. Casual riders rides count is max during July month-

casual: Min – Feb, Max - **Jul**

member: Min – Feb, Max - Aug

*b) Bike type usage:*

1. Members with classic bike show rides count of 1978118 as high –

*casual*: High - classic\_bike – 1242017

Low - docked\_bike - 290811

*member*: High - **classic\_bike – 1978118**

Low - docked\_bike - 1

2. Classic bikes rides count percentage of 71% is High -

classic\_bike - **71%**

docked\_bike - 6%

electric\_bike - 23%

*c) Most visited stations:*

1. Total number of stations – **845**

2. Top most visited - **Streeter Dr & Grand Ave - 158311**

2. Most visited stations by users –

casual: Streeter Dr & Grand Ave - **128231**

member: Clark St & Elm St - **47809**

*d) Ride length:*

1. Average ride length of casual riders is high. Member riders count is high in probability in the distribution.

*casual* – Average: **23.71077**

*member* – Average: 12.82475

2. Casual riders average ride length by hour is high during afternoon 13:00-14:00Hrs.

*casual: Min – 6:00 Hrs, Max –* ***13:00-14:00 Hrs***

*member*: Min – 5:00 Hrs, Max – 14:00 Hrs

3. Casual riders average ride length by day of week of is high during weekends and

Members average ride length is consistent throughout the week.

*casual*: Min – Tues & Wed, **Max - Sun**

*member*: Min – Mid week, Max - Sun

4. Casual riders average ride length is high on 4th day of month

*casual*: Min – 28th day, **Max – 4th day**

*member*: Min – 28th day, Max – 31st day

5. Casuals riders average ride length is high between March and May. Wherein, member riders average ride length is consistent throughout the year.

*casual*: Min - Dec, Max - **May**

*member*: Min - Dec, Max – May

**6. ACT:**

**Key findings of analysis (to trace how members differ from casuals)**:

* Members ride count is high overall. And member riders are

high during evening rush hours, across the month, consistent almost throughout week.

So, the purpose could be for daily exercise routine or commuting back from jobs.

* Casual ride count is high on weekend and mid of summer. These rides could be for weekend retreat or visiting holiday places.
* Classic bike is more used by both users.
* Casual riders ride duration is so high during weekends and throughout spring season. Which indicates most of the casual rides are for holidaying.

* Mostly casual riders are visiting beach side places, which indicates they are tourist or visiting holidaying. Wherein, member riders are mostly visiting mid of the city, may be for regular commuting to offices.

* Though the rides count of members is high, but ride duration of casual users is high.

**Recommendations:**

1. *Promotions to capture regular casual riders:*

By promoting during rush hours of the day, company can make casual riders to understand using Cyclistic bikes on daily basis with membership is more beneficial.

1. *Special promotions near holidaying places:*

To encourage leisure casual riders, we can introduce special tourist membership during spring season or on weekends near holidaying places.

1. *New campaign on company’s app:*

A special marketing campaign for promoting company's app. So that the company can maintain customers data. Accordingly customized offers can be notified to the riders.

1. *Fitness bikes for health regime riders:*

By improving the existing bike model or introducing a new bike as "Fitness bikes" and marketing its benefits, we can capture section of casual riders who are riding regularly as part of health regime.

1. *Providing discounts for long duration riders:*

As the duration of rides of casual riders is high than members, by providing discounts to riders to go on long rides which would improve the goodwill of the company.

1. *Promotion places:*
   * Marketing at sightseeing places like famous holiday spots, beaches during weekends and spring season.
   * Sticking the latest offers as stickers on the bikes.
   * Partnership sponsoring of sports or events happening in the city.
   * Promoting during rush hours of the day.
   * Tie up with other health regime companies.
   * Encouraging the existing members to add new member riders to the company by providing the discounts.
   * Payments through apps can be encouraged, to get the customer rides data.
   * Digital promotions on fitness and travel related social networking sites, also could be a viable solution for promoting the company’s annual subscriptions benefits.

**Suggestions for further analysis:**

* More data about members would help us to do analysis regarding what are the common factors of those customers making them to subscribe.
* Pricing strategies of the both riders’ information would help to make detailed profitable analysis.

**Conclusion:**

My special thanks to Google Data Analytics Professional Certificate program provided by Coursera, for helping me to understand the concepts and providing resources to build skill set required to successfully complete this case study. I really enjoyed working with R with all data analysis concepts. Hope you all enjoy my work. Thank you.

KAGGLE: Case study – URL

<https://www.kaggle.com/code/sowjanyakake/case-study-cyclistic-bike-share>