Google Analytics Capstone Project

Victor Okanda

11/17l/2022

Introduction

In this documentation, I give a detailed description of my process of analysis, from data collection to visualization. I am required to help **Cyclistic**, a bike-share company make an appropriate marketing strategy. In my outcome, I am to showcase the differentials in the bike use between casual riders and manual members. The datasets contain trip information for the last 12 months (NOV, 2021 to OCT, 2022. Find the link to the datasets here.

Loading of packages

Let us load the important packages that will be used for this analysis. The following packages will be installed and loaded:

- tidyverse
- ggplot2
- janitor
- lubridate
- geosphere
- webr

Loading the Dataset

Tidyverse package will be used to combine and read all the CSV files in the folder. The following function will be used.

ride_data <- list.files(path = "C:/Users/User/Desktop/trip_data", pattern = ".csv")%>% map_df(~read_csv(.))

EDA

Let's explore the data to learn of its structure and composition. The exploration will look at:

- Column names
- column data types
- Check for inconsistencies
- Check for null values, and
- Any other issue that needs tweaking

```
summary(ride_data)
```

```
##    ride_id          rideable_type        started_at
##  Length:5755694     Length:5755694     Min.   :2021-11-01 00:00:14
##  Class :character   Class :character   1st Qu.:2022-04-27 16:40:09
##  Mode  :character   Mode  :character   Median :2022-06-30 18:31:03
##                                        Mean   :2022-06-13 23:04:32
##                                        3rd Qu.:2022-08-24 19:52:19
##                                        Max.   :2022-10-31 23:59:33
##
##     ended_at                   start_station_name start_station_id
##  Min.   :2021-11-01 00:04:06   Length:5755694     Length:5755694
##  1st Qu.:2022-04-27 16:51:40   Class :character   Class :character
##  Median :2022-06-30 18:49:28   Mode  :character   Mode  :character
##  Mean   :2022-06-13 23:23:58
##  3rd Qu.:2022-08-24 20:10:05
##  Max.   :2022-11-07 04:53:58
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:5755694     Length:5755694     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.   :45.64   Max.   :-73.80
##
##     end_lat         end_lng       member_casual
##  Min.   :41.39   Min.   :-88.97   Length:5755694
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character
##  Median :41.90   Median :-87.64   Mode  :character
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.37   Max.   :-87.30
##  NA's   :5835    NA's   :5835
```

```
sum(is.na(ride_data))
```

```
## [1] 3648044
```

From the exploration, it can quickly be deduced that there are 5 million rows of data with 13 columns. We can also see some null values.

Removing Null Values

From the exploration, there are null values in the "*start_station_name*","*end_station_name*", "*start_station_id*", and "*end_station_id*" columns. This is not ideal for a cyclist must commence a trip from a station and end at a given station. Therefore, we will remove records that have no start and end station names.

```
df<- ride_data %>%
  filter(! is.na(start_station_name)) %>%
  filter(! is.na(end_station_name))
```

Adding Columns

Other columns will be added to the dataset to calculate *trip_duration*, *trip_distance* (using the geosphere package), *trip_year*, *trip_month*, *day_of_week*, and *hour_of_day*. Mutate function of the dplry package will be used.

```
df_1 <- df %>%
  mutate(trip_duration = as.numeric(difftime(ended_at, started_at, units = "mins"))) %>%
  mutate(trip_distance= distHaversine(cbind(start_lng, start_lat), cbind(end_lng, end_lat))) %>%
  mutate(trip_year = year(started_at)) %>%
  mutate(trip_month = month(started_at, label = TRUE)) %>%
  mutate(day_of_week = weekdays(started_at)) %>%
  mutate(hour_of_day = hour(started_at))
```

Further Cleaning

It is further realized during the exploration that some records have either zero, or negative numbers representing trip_duration and trip_distance. This is impossible since these variables must have positive values. This calls for further cleaning.

```
df_2 <- df_1 %>%
  filter(trip_duration > 0 & trip_distance > 0)
```

Analysis

The analysis will discover trends on how the casual and annual members use bikes. Only few columns will be chosen for the analysis. The analysis tries to answer the following questions:

- What is the average ride length for annual and casual members?
- What is the maximum, and minimum ride lengths for the category of the customers?
- What is the average ride length by days of week?
- What is the number of rides for users by *day_of_week*?
- What is the most preferred ride type?

Maximum, Minimum,and average ride lengths for users

```
df_2 %>%
  group_by(member_casual) %>%
  summarise(average_ride_length= mean(trip_distance),
            max_ride_length= max(trip_distance),
            min_ride_length= min(trip_distance)
            )
```

```
## # A tibble: 2 x 4
##   member_casual average_ride_length max_ride_length min_ride_length
##   <chr>                       <dbl>           <dbl>           <dbl>
## 1 casual                      2324.        1190855.          0.0332
## 2 member                      2111.          27518.          0.0202
```

Average ride length and number of rides by day of week

From the output below, many cyclists prefer riding on Saturdays.

```
df_2 %>%
  group_by( day_of_week) %>%
  summarize(average_ride_length= mean(trip_distance),
            number_of_rides = n()) %>%
  arrange(-number_of_rides)
```

```
## # A tibble: 7 x 3
##   day_of_week average_ride_length number_of_rides
##   <chr>                     <dbl>           <int>
## 1 Saturday                  2349.          681622
## 2 Thursday                  2151.          614215
## 3 Wednesday                 2134.          594220
## 4 Friday                    2148.          588442
## 5 Tuesday                   2123.          585124
## 6 Sunday                    2299.          571992
## 7 Monday                    2133.          571736
```

The Preferred ride type

Many riders prefer classic bikes

```
df_2 %>%
  group_by(rideable_type) %>%
  summarise(number_of_rides= n()) %>%
  arrange(-number_of_rides)
```

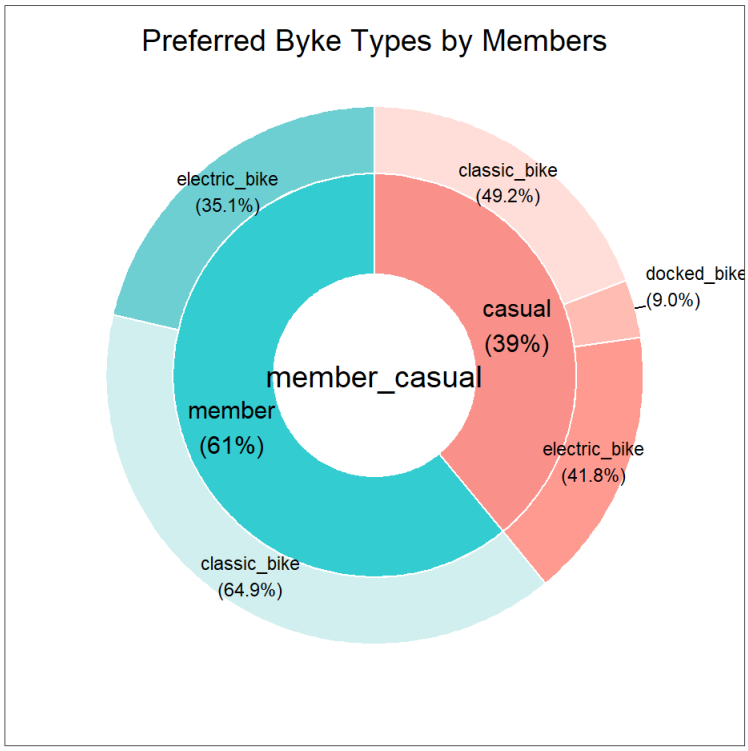
```
## # A tibble: 3 x 2
##   rideable_type number_of_rides
##   <chr>                   <int>
## 1 classic_bike          2473799
## 2 electric_bike         1586079
## 3 docked_bike            147473
```

The Preferred Ride Type Among Members

Here, pie donut chart from the webr package will be used to determine how ride type is distributed among cyclists.

```
df_2 %>%
  PieDonut(aes(member_casual, rideable_type),
           title = "Preferred Byke Types by Members",
           r0 = 0.45, r1 = 0.9)
```

```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## i The deprecated feature was likely used in the webr package.
##   Please report the issue at <https://github.com/cardiomoon/webr/issues>.
```



Creation of a Final Dataset for Further Visualization

A summarized dataset will be created to be used for visualization in Tableau, Power BI, or any other BI tool.

```
final_trip_data<- df_2 %>%
  group_by(member_casual,
           rideable_type,
          trip_year, trip_month,
           day_of_week,
           hour_of_day) %>%
  summarize( number_of_rides= n(),
             avg_ride_duration= mean(trip_duration),
             avg_ride_distance = mean(trip_distance))
```

```
## `summarise()` has grouped output by 'member_casual', 'rideable_type',
## 'trip_year', 'trip_month', 'day_of_week'. You can override using the `.groups`
## argument.
```

Exporting Summarized Data

The clean summarized data is finally exported to the PC using *write_csv* for further analysis and visualization.

```
write_csv(final_trip_data,
          "C:\\Users\\User\\Desktop\\R\\summary_data.csv",
          append = FALSE )
```

```
*--END--*
```