/\*

Data is sourced from first party (https://divvy-tripdata.s3.amazonaws.com/index.html).

Data is public and has been made available by Motivate International Inc. under this license (https://www.divvybikes.com/data-license-agreement).

Data taken is for period Nov 2020 - Oct 2021.

\*/

/\*

1.  Data Preparation

     •  The data owner changed its structure for data collected < Dec2020, thus cannot be directly combined with the newer data.

        These dataset will need additional processing.

     •  There are some NULL data but the majority of data seemed complete and consistent thus the dataset deemed acceptable.

     •  Initial glance of the data seemed to favour casual riders, i.e. casual riders seemed to rent the bicycle for much longer than annual members.

\*/

#2. Preparing Nov 2020 data

#    a)  Combine monthly data to 1 table using UNION ALL.

#        Save the result as new table tripdata2020\_12\_2021\_10.

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2021\_10`

          UNION ALL

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2021\_09`

          UNION ALL

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2021\_08`

          UNION ALL

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2021\_07`

          UNION ALL

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2021\_06`

          UNION ALL

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2021\_05`

          UNION ALL

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2021\_04`

          UNION ALL

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2021\_03`

          UNION ALL

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2021\_02`

          UNION ALL

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2021\_09`

          UNION ALL

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_12`;

#   b)  use table in a) to extract the new station id --> save the result as new table station\_name

       # step a) is necessary in order to have cemplete station list.

          SELECT DISTINCT start\_station\_name,

                          start\_station\_id,

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_12\_2021\_10`;

 #  c) CAST the start\_station\_id and end\_station\_id from integer to string --> save the result as new table tripdata2020\_11string

          SELECT \*,

          CAST(start\_station\_id AS STRING) AS start\_station\_id\_1,  # cannot override the original table, that’s why its given different name

                      CAST(end\_station\_id AS STRING) AS end\_station\_id\_1

              FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_11`

              WHERE ride\_id IS NOT NULL AND

                  rideable\_type IS NOT NULL AND

                  started\_at IS NOT NULL AND

                  ended\_at IS NOT NULL AND

                  start\_station\_name IS NOT NULL AND

                  start\_station\_id IS NOT NULL AND

                  end\_station\_name IS NOT NULL AND

                  end\_station\_id IS NOT NULL AND

                  start\_lat IS NOT NULL AND

                  start\_lng IS NOT NULL AND

                  end\_lat IS NOT NULL AND

                  end\_lng IS NOT NULL AND

                  member\_casual IS NOT NULL;

#   d) Then UPDATE the new string column with the new id convention.

          UPDATE `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_11string` AS NOV20

          SET NOV20.end\_station\_id\_1 = name.start\_station\_id

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.station\_name` AS name

          WHERE NOV20.end\_station\_name = name.start\_station\_name;

          #AND

          UPDATE `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_11string` AS NOV20

          SET NOV20.start\_station\_id\_1 = name.start\_station\_id

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.station\_name` AS name

          WHERE NOV20.start\_station\_name = name.start\_station\_name;

#   e) Remove the old station id columns --> save the result as new table tripdata2020\_11clean

          SELECT         ride\_id,

                          rideable\_type ,

                          member\_casual ,

                          started\_at,

                          ended\_at,

                          start\_station\_name ,

                          start\_station\_id\_1 AS start\_station\_id ,

                          end\_station\_name ,

                          end\_station\_id\_1 AS start\_station\_id ,

                          start\_lat,

                          start\_lng,

                          end\_lat,

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_11string` ;

#3. Now we can combine Nov 2020 data with the others.

#   Save the result as new table tripdata2020\_11\_2021\_10\_clean

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_12\_2021\_10`

          UNION ALL

          SELECT  \*

          FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_11clean` ;

#4. Preliminary investigation:

#     i. Ride\_length returned negatives number.

#        Using this query, I found there are 358 rows with negative ride\_length

               SELECT COUNT (\*)

               FROM (

               SELECT  \*,

                               DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) AS ride\_length

                       FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_11\_2021\_10`

                       WHERE ride\_id IS NOT NULL AND

                               rideable\_type IS NOT NULL AND

                               started\_at IS NOT NULL AND

                               ended\_at IS NOT NULL AND

                                         start\_station\_name IS NOT NULL AND

                start\_station\_id IS NOT NULL AND

                               end\_station\_name IS NOT NULL AND

                               end\_station\_id IS NOT NULL AND

                               start\_lat IS NOT NULL AND

                               start\_lng IS NOT NULL AND

                               end\_lat IS NOT NULL AND

                               end\_lng IS NOT NULL AND

                               member\_casual IS NOT NULL AND

                               DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) > 0

                         )

               WHERE ride\_length < 0 ;

#     ii. There are ~ >1200 entry with ride\_length > 24 hours (1,440 minutes). Most of which came from casual customer.

#         In fact member customer only logged maximum 1500 hours.

               SELECT \*

               FROM (

                       SELECT  ride\_id,

                               rideable\_type ,

                               member\_casual ,

                               FORMAT\_DATE('%A', started\_at) AS day\_of\_week,

                               started\_at,

                               ended\_at,

                               DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) AS ride\_length,

                               start\_station\_name ,

                               start\_station\_id ,

                               end\_station\_name ,

                               end\_station\_id ,

                               start\_lat,

                               start\_lng,

                               end\_lat,

                               end\_lng

                       FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_11\_2021\_10`

                       WHERE ride\_id IS NOT NULL AND

                               rideable\_type IS NOT NULL AND

                               started\_at IS NOT NULL AND

                               ended\_at IS NOT NULL AND

                               start\_station\_name IS NOT NULL AND

                               start\_station\_id IS NOT NULL AND

                               end\_station\_name IS NOT NULL AND

                               end\_station\_id IS NOT NULL AND

                               start\_lat IS NOT NULL AND

                               start\_lng IS NOT NULL AND

                               end\_lat IS NOT NULL AND

                               end\_lng IS NOT NULL AND

                               member\_casual IS NOT NULL AND

                               DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) > 0 AND

                               end\_station\_name <> "Base - 2132 W Hubbard Warehouse" AND

                               start\_station\_name <> "Base - 2132 W Hubbard Warehouse"

               )

               WHERE member\_casual = "member"

               ORDER BY ride\_length DESC limit 10 ;

#5. Since the negatives result are small compared to the sample size, we exclude the negatives ride\_length data from the dataset.

#   Also to exclude all ride\_length > 24 hours (1,440 minutes).

#     Note: From Divvy bikes website, for annual pass member, max free usage is 45minutes, and day pass allowed up to 3 hours free usage. Afterwards $0.15 / min.

#           So it is unlikely someone will continuously using the bike for more than 24 hours.

             SELECT  ride\_id,

                     rideable\_type ,

                     member\_casual ,

                     FORMAT\_DATE('%A', started\_at) AS day\_of\_week,

                     started\_at,

                     ended\_at,

                     DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) AS ride\_length,

                     start\_station\_name ,

                     start\_station\_id ,

                     end\_station\_name ,

                     end\_station\_id ,

                     start\_lat,

                     start\_lng,

                     end\_lat,

                     end\_lng

             FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_11\_2021\_10`

             WHERE ride\_id IS NOT NULL AND

                     rideable\_type IS NOT NULL AND

                     started\_at IS NOT NULL AND

                     ended\_at IS NOT NULL AND

                     start\_station\_name IS NOT NULL AND

                     start\_station\_id IS NOT NULL AND

                     end\_station\_name IS NOT NULL AND

                     end\_station\_id IS NOT NULL AND

                     start\_lat IS NOT NULL AND

                     start\_lng IS NOT NULL AND

                     end\_lat IS NOT NULL AND

                     end\_lng IS NOT NULL AND

                     member\_casual IS NOT NULL AND

                     DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) > 0 AND

                     DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) < 1440 AND

                     end\_station\_name <> "Base - 2132 W Hubbard Warehouse" AND

                     start\_station\_name <> "Base - 2132 W Hubbard Warehouse"

             ORDER BY ride\_length DESC ;

# 6. Add new column day\_of\_week

#    Create a column called “day\_of\_week,” and calculate the day of the week that each ride started using the “WEEKDAY” command (for example, =WEEKDAY(C2,1)) in each file.

#    Format as General or as a number with no decimals, noting that 1 = Sunday and 7 = Saturday.

          #Using SQL BIGQUERY

                  SELECT  \*,

                          DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) AS ride\_length,

                          FORMAT\_DATE('%A', started\_at) AS day\_of\_week

                                        # see bigquery reference guide for other format like ‘%A’

                  FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_11\_2021\_10`

                  WHERE ride\_id IS NOT NULL AND

                          rideable\_type IS NOT NULL AND

                          started\_at IS NOT NULL AND

                          ended\_at IS NOT NULL AND

                          start\_station\_name IS NOT NULL AND

                          start\_station\_id IS NOT NULL AND

                          end\_station\_name IS NOT NULL AND

                          end\_station\_id IS NOT NULL AND

                          start\_lat IS NOT NULL AND

                          start\_lng IS NOT NULL AND

                          end\_lat IS NOT NULL AND

                          end\_lng IS NOT NULL AND

                          member\_casual IS NOT NULL AND

                          DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) > 0 AND

                          DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) < 1440 AND

                          end\_station\_name <> "Base - 2132 W Hubbard Warehouse" AND

                          start\_station\_name <> "Base - 2132 W Hubbard Warehouse" ;

#7.  Rearranging the table by specifying the order in SELECT  function

                  SELECT  ride\_id,

                          rideable\_type ,

                          member\_casual ,

                          FORMAT\_DATE('%A', started\_at) AS day\_of\_week,

                          started\_at,

                          ended\_at,

                          DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) AS ride\_length,

                          start\_station\_name ,

                          start\_station\_id ,

                          end\_station\_name ,

                          end\_station\_id ,

                          start\_lat,

                          start\_lng,

                          end\_lat,

                          end\_lng

                  FROM `bigquery-kaggle-327512.casestudy1\_bikesharing.tripdata2020\_11\_2021\_10`

                  WHERE ride\_id IS NOT NULL AND

                          rideable\_type IS NOT NULL AND

                          started\_at IS NOT NULL AND

                          ended\_at IS NOT NULL AND

                          start\_station\_name IS NOT NULL AND

                          start\_station\_id IS NOT NULL AND

                          end\_station\_name IS NOT NULL AND

                          end\_station\_id IS NOT NULL AND

                          start\_lat IS NOT NULL AND

                          start\_lng IS NOT NULL AND

                          end\_lat IS NOT NULL AND

                          end\_lng IS NOT NULL AND

                          member\_casual IS NOT NULL AND

                          DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) > 0 AND

                          DATETIME\_DIFF(DATETIME(TIMESTAMP(ended\_at)), DATETIME(TIMESTAMP(started\_at)), MINUTE) < 1440 AND

                          end\_station\_name <> "Base - 2132 W Hubbard Warehouse" AND

                          start\_station\_name <> "Base - 2132 W Hubbard Warehouse" ;

#8. Export and further process in RStudio