Laboratory Session : April 1, 2025
Exercises due on : April 20, 2025

Exercise 1 - NYC bike-sharing data

The repository `https://drive.google.com/drive/folders/1NESuaJ5yGIrAli1TgrpnK5hnoxGsMi3h?usp=sharing` contains bike-sharing data provided by New York City, Citi Bike[1] sharing system. The data (in `csv` format) is structured as follows

- `Trip duration` (in seconds)
- `Start Time` and `date`
- `Stop Time` and `date`
- `Start Station ID`, `name`, `latitude` and `longitude`
- `End Station ID`, `name`, `latitude` and `longitude`
- `Bike ID`
- `User Type` (*Customer* or *Subscriber*)
- `Birth's Year`
- `Gender` (0=unknown; 1=male; 2=female)

1) read the data and import in a `data.frame` or `tibble` structure

2) merge the five data frames in an unique structure[2]

3) check for missing data and remove it, if any

4.1) compute the average and the median trip duration in minutes

4.2) evaluate the minimum and maximum trip duration; does that seem like a plausible value?

4.3) repeat the calculation of the average (and the median) trip duration by excluding trips longer than 3 hours. Next, evaluate the number of skimmed entries

4.4) plot the distribution of trip duration after the skimming of the previous point

5) plot the monthly average trip duration

6.1) plot the average number of rides per day of the week

6.2) plot the hourly distribution on weekdays and on weekends

6.3) plot again the average hourly distribution on weekdays but separating *customer* and *subscriber* users

[1] The official page of the service is `https://citibikenyc.com/` and the open data can be retrieved from `https://s3.amazonaws.com/tripdata/index.html`

[2] If the data is too heavy for your computing resources, you can work with a sufficiently large subsample of it.

7.1) using the latitude and longitude information[3], evaluate the average speed (in *km/h*) of a user, discarding the trip lasting longer than 1 hour

7.2) plot the average speed as a function of route length for the following group of distances d $<$ 500 m, 500 m $<$ d $<$ 1000 m, 1000 m $<$ d $<$ 2000 m, 2000 m $<$ d $<$ 3000 m, d $>$ 3000 m and discarding trips longer than 1 hour

8.1) find the most common start station and the least popular end station

8.2) find the three most common routes (start and end station) and the three least popular ones

Exercise 2 - Parallel pixelated-sensors

A detector designed for charge identification of incoming particles consists of two parallel planes, each composed of an 8×8 array of pixelated sensors. The collected data is stored in the file available in the repository `https://drive.google.com/file/d/1dYPF5tL3qnBmTVbawyKbPOQooi_CCElV/view?usp=sharing`, where:

- The first 64 columns correspond to the response of the pixels in the upstream matrix.
- The next 64 columns correspond to the response of the pixels in the downstream matrix.

Each pixel is indexed according to the following formula:

$$\texttt{Pixel Index} = \texttt{Row} + \texttt{Column} \times N_{\texttt{cols}}, \tag{1}$$

where $N_{\texttt{cols}} = 8$ is the total number of columns in the matrix.

The analysis consists of the following tasks:

- For each event[4] and for each 8×8 matrix, perform the following steps:
  1. Check for missing values in the dataset and handle them appropriately.
  2. Determine the *maximum* and *second maximum* pixel values.
  3. Identify the corresponding pixel indices.
- Plot the distributions of the indices corresponding to the maximum and second maximum pixel values.
- Repeat the same plot but excluding events where the maximum signal is less than 10.
- Compute event-by-event, the ratio between the second and first maximum values.
- Only consider events where both values are greater than zero. Plot the distribution of this ratio in four different signal ranges:
  1. $10 \leq \texttt{Max} < 300$
  2. $300 \leq \texttt{Max} < 1200$
  3. $1200 \leq \texttt{Max} < 30000$
  4. $\texttt{Max} \geq 30000$
- Generate heatmaps illustrating the spatial distribution (in terms of row and column) of:
  - The maximum pixel indices.
  - The second maximum pixel indices.

[3] Hint: in the `geosphere` `R` package, you can find the function `distHaversine` that gives you the shortest distance between two points according to the "haversine" method, which makes the assumption of spherical Earth.

[4] one row of the `csv` file corresponds to one event.

- Count the number of times the pixel index of the maximum signal in the upstream matrix matches that of the downstream matrix. Create a bar plot, considering only events where the maximum value exceeds 10.
- Identify the four most frequently occurring pairs of maximum pixel indices (Max Index Upstream , Max Index Downstream)

[Hint:] use the `tidyverse` packages to manipulate the data frame and produce the visualization plots (i.e. `dplyr`, `ggplot2`, ...)