Python Programming Example

Please complete the below programming example. This should take not more than a couple of hours and if you get stuck, document your code, your problems and approach to solve the issues you come across, then move ahead to the next task.

In the end, please export the cleaned dataset as well as the quality metrics you calculated (task 3, 6 and 7). Some later tasks might rely on results of tasks 1 and 2 for a full solution.

In case you do not know how to start implementing the solution to a task, please describe how you would process the data and document all your thoughts towards the solution in a short text.

Installation Python 3 Environment

If you know how to set up an Python 3 environment and you have a preferred editor/IDE, feel free to use what ever works for you. Otherwise please follow these steps and reach out if you need assistance:

1. install Visual Studio Code with the python extension (https://code.visualstudio.com/docs/languages/python)
2. install python using Anaconda (https://docs.anaconda.com/anaconda/install/index.html)
3. create an Anaconda Environment with Python 3.7 (or newer) and install Pandas using the Anaconda Navigator (https://docs.anaconda.com/anaconda/navigator/tutorials/pandas/)

If you cannot find the Pandas package, click "update index" at the top of the Anaconda Navigator. Use the most current version of Pandas.

Once your environment is installed, click on the environment and “Open Terminal”, type “code .” and hit “enter”. Now you run Visual Studio Code with the proper interpreter, create a test.py file and solve the following tasks.

Your input file is "eyetracker.log"

HINT: Use Pandas DataFrame

Customer Study – Data Analysis

You ran a driver distraction study, in which the participants were asked to use a phone app while driving the simulator. The participants were monitored using an eye tracking device which classifies fixations onto the areas of interest (AOI) "Forward Road View (FRV)" and "Phone". In order to check if the customer's technology complies with the NHTSA driver distraction guidelines the data first needs to be cleaned up and quality tests need to be performed.

1. AOI cleanup

Make sure that the AOI column contains only allowed value.

Allow values are:

- -1 = invalid
- 0 = no AOI detected
- 2 = FRV
- 3 = Phone

There should be only allowed values in this column (change to invalid if necessary)

2. Validity check

Check for the validity of each data entry using the following conditions.

Add a column for validity (boolean)
For invalid data points, overwrite the AOI value with -1

Conditions:

- Validity_left = Valid
- Validity_right = Valid
- Type = Fixation or Saccade
- AOI = 0 or 2 or 3

3. Durations

Calculate the total duration for each of the following conditions.

Conditions:

- Type = Fixation
- Type = Saccade
- Validity = Valid (according to column created during task 2)
- Validity = Invalid (according to column created during task 2)
- AOI = 2 (FRV)
- AOI = 3 (Phone)

4. Data quality check

Find rows with incorrect durations.
Flag these invalid in the column created during task 2.
Calculate and store the correct duration value for these rows.

Hint: look at the timestamps to identify the correct duration

5. Reduction

Merge consecutive invalid data points into a single data point, set the “Type” of this row to "Merged" and recalculate the duration for the merged data point.

6. Recalculated durations

Calculate the new total duration for each of the following conditions (with the new duration values generated in task 5).

Conditions

Type = Fixation
Type = Saccade
Validity = Valid (according to column created during task 2)
Validity = Invalid (according to column created during task 2)
AOI = 2 (FRV)
AOI = 3 (Phone)

7. Validity

To check the overall data quality calculate the percentage of valid data:

- Compute the percentage of valid data points in relation to the total number of data points
- Compute the percentage of valid data points in relation to the total duration

8. Export

Export your results into separate files.

- Save the results of task 3, task 6 and task 7 in a text-file (.txt)
- Export the final Pandas DataFrame into a single comma-separated values (.csv) file
- Save your entire code and send as a single Python file called *YourName_test.py*
- Document your though process and problem solving approach as a text file (preferrably Word-Document file)