DATASET

For future make loading from web database.

# Path to dataset of ECG   
path = 'D:/SCIENCE/Datasets/autonomic-aging-a-dataset-to-quantify-changes-of-cardiovascular-autonomic-function-during-healthy-aging-1.0.0'  
  
csv\_info\_file = 'subject-info.csv'

####################################################################################### OPENING RECORDS######################  
################################################################

FUNCTIONS

**def read\_ECGs\_annotation\_data():**

***""" Open csv info file, print header and information for each record. Then fill ECG DATABASE. """***

**def open\_record(id, min\_point, max\_point)**

*Open each record with ECGs by Id.*

*Input parameters:  
 - Id - Id of record (for example: 0001, 0002 …)  
 - min\_point - minimum point, at which starts ECG (including this point)  
 - max\_point - maximum point, at which ends ECG (not including this point)  
  
 Output parameters:  
 - [sequence\_1, sequence\_2] - list with sequence\_1 for first ECG and sequence\_2 for second ECG"""  
  
  
Describing:  
 wfdb.rdrecord(path + '/' + id, min\_point, max\_point, [0, 1])  
  
 min\_point = 0 - The starting sample number to read for all channels  
 (point from what graphic starts (min\_point)).  
  
 max\_point = None - The sample number at which to stop reading for all  
 channels (max\_point). Reads the entire duration by default.  
  
 [0, 1] - first two channels (ECG 1, ECG 2); [0] - only first ECG.*

**valid\_indices = np.where(total\_time <= duration)[0]**

**Програма ChatGPT сказала:**

**The line of code:**

**python**

**КопіюватиРедагувати**

**valid\_indices = np.where(total\_time <= duration)[0]**

**means:**

* **total\_time is likely a NumPy array (or something NumPy can compare element-wise).**
* **duration is a scalar value.**
* **total\_time <= duration creates a boolean array: True where the condition is satisfied.**
* **np.where(...) returns a tuple of arrays with the indices where the condition is True.**
* **Adding [0] gets the first array from that tuple, which in this case contains the indices of elements in total\_time that are less than or equal to duration.**

**So, this line selects all the indices where total\_time[i] <= duration.**

**Example:**

**python**

**КопіюватиРедагувати**

**import numpy as np**

**total\_time = np.array([3, 7, 2, 9, 5])**

**duration = 5**

**valid\_indices = np.where(total\_time <= duration)[0]**

**print(valid\_indices)**

**# Output: [0 2 4]**

**Let me know if you want help using valid\_indices or applying it to filter another array.**