**eda\_intervalrelated.py**

Designed for analyzing longer, continuous stretches of data (e.g., a 5-minute meditation).

* How many total responses (SCR) occurred during this time?
* On average, how strong were those responses?
* How much did the person's baseline arousal level fluctuate?
* Are there any deeper, more complex patterns in the signal (like sympathetic activity or rhythmicity)?

eda\_autocorr: Calculates the autocorrelation of the signal (a measure of its rhythmicity).

eda\_sympathetic: Estimates an index of sympathetic nervous system activity from the EDA signal.

e.g., A dog starts barking loudly -> heart beats faster, breath is faster, pupils get bigger.

Function Definition

def eda\_intervalrelated(data, sampling\_rate=1000, \*\*kwargs):

* **\*\*kwargs**: A way to pass extra settings down to the specialized functions (eda\_autocorr, eda\_sympathetic).

output["SCR\_Peaks\_N"] = np.nansum(data["SCR\_Peaks"].values)

* First, it checks if the SCR\_Peaks column exists (created by eda\_process).
* **np.nansum(data["SCR\_Peaks"].values)**: This simply counts all the 1s in the SCR\_Peaks column. Since peaks are marked with a 1 and non-peaks with a 0, this gives the total number of SCRs in the interval.

if len(data) > sampling\_rate \* 64:

output.update(eda\_sympathetic(...))

* **if len(data) > sampling\_rate \* 64:**: This is a crucial check. This code only attempts to calculate sympathetic activity if the interval is longer than 64 seconds.
* If the condition is met, it calls the sympathetic specialist function.

if len(data) > sampling\_rate \* 30:

output["EDA\_Autocorrelation"] = eda\_autocorr(...)```

\* \*\*`if len(data) > sampling\_rate \* 30:`\*\*: It checks if the interval is longer than 30 seconds.

\* If the condition is met, it calls the `eda\_autocorr` specialist function.

\* \*\*Autocorrelation\*\* measures how much a signal repeats itself over time. It can give clues about underlying rhythms in physiological activity.