University of Belgrad – School of Electrical Engineering

Depratment of Signals and Systems

TUTORIAL FOR USING EEGLAB

Dositej Cvetković 2020/0224
Nina Đerić 2020/0449

Mentor
dr Milica Janković, vanredni profesor

Belgrade, *august* 2024. godine

ACKNOWLEDGEMENTS

We wish to express our deep gratitude to our mentor, Professor Milica Janković, whose expertise, patience, and support were crucial for the successful completion of this work. Her willingness to always be available, provide guidance, and share her knowledge has been invaluable throughout our research.
We also want to thank Assistant Marija Novičić and the entire laboratory exercises team, who selflessly shared their time and effort to help us master complex topics in the processing of evoked potentials. Their support and hands-on training enabled us to acquire the necessary skills and knowledge that were critical for carrying out this research.

Dositej Cvetković
Nina Đerić
In Belgrade, August 2024

CONTENT

1 INTRODUCTION

Evoked potentials (EP) represent an extremely useful tool in studying the functioning of the human brain. This neurophysiological technique allows us to examine the electrical signals generated in the brain in response to stimulation, providing a deeper understanding of cognitive processes, sensory functions, and the processes of perception and attention. The processing of evoked potentials encompasses a wide range of methods for analyzing these signals, including filtering, segmentation, and statistical analysis, to extract relevant information and interpret results.

In this paper, we explore various aspects of evoked potentials processing, focusing on methods for identifying specific components of EPs. The aim of this work is to contribute to a better understanding of the mechanisms behind the generation of evoked potentials. We expect that the results of this study will provide new insights that will enhance our understanding of the functioning of the human brain and provide a foundation for future work in EEG analysis.

2 METHODOLOGY

Writing MATLAB scripts for EEGLAB requires a certain understanding of data structures in EEGLAB. Below, we will present the data structures in EEGLAB.

EEG: current EEG dataset

ALLEEG: array of all loaded EEG datasets

CURRENTSET: index of the current dataset

LASTCOM: last command issued from the EEGLAB menu

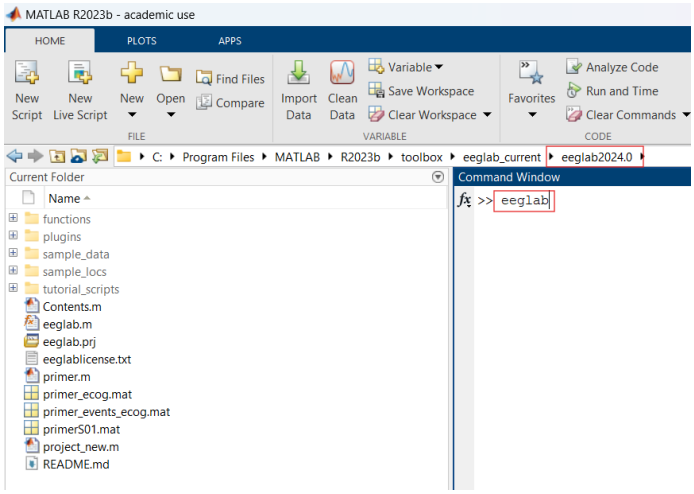
ALLCOM: all commands issued from the EEGLAB menu

STUDY: structure of group analysis in EEGLAB

CURRENTSTUDY: 1 if EEGLAB is performing group analysis, 0 otherwise

Running EEGLAB

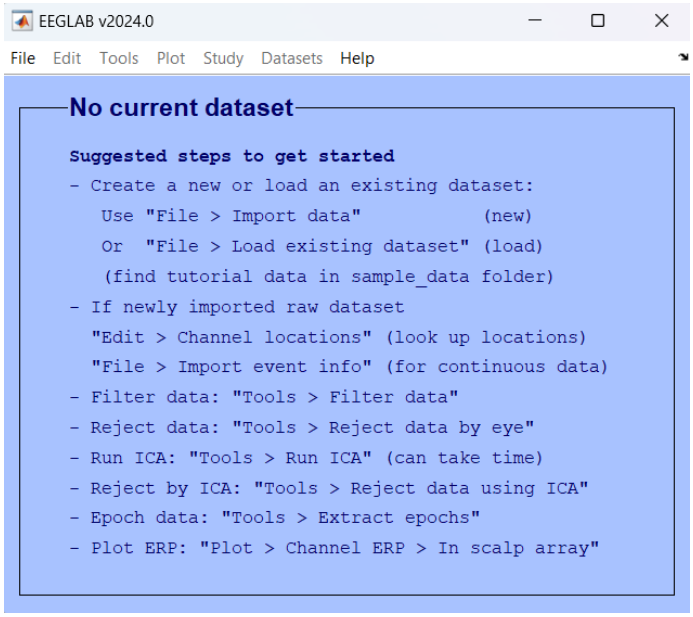
By typing the command ‘eeglab’ in the MATLAB command window, the current folder must be the folder where EEGLAB is installed.



Command to start EEGLAB from a script:

```
[ALLEEG, EEG, CURRENTSET, ALLCOM] = eeglab;
```

EEGLAB Working Environment



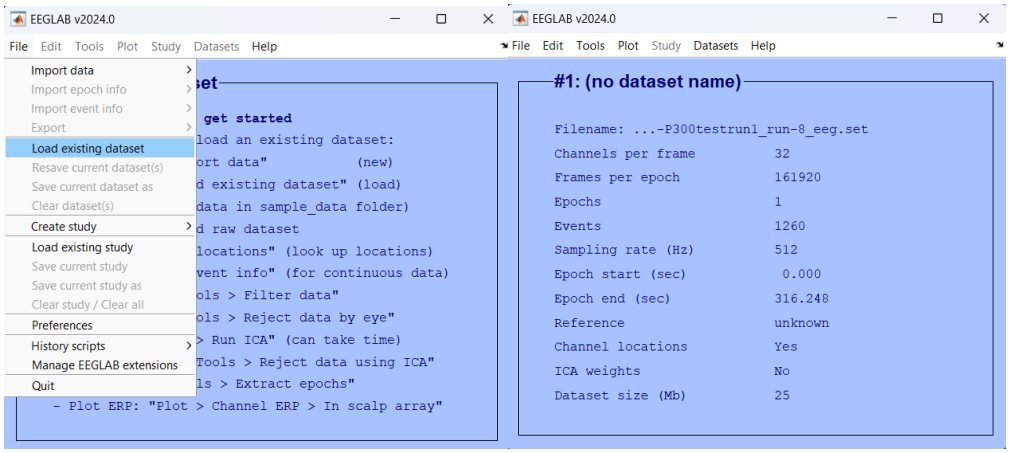
1. Loading EEG Data into EEGLAB

EEGLAB datasets are saved in .set files. .set files are MATLAB files. You can save the EEG structure using the command: save -mat myfile.set EEG.

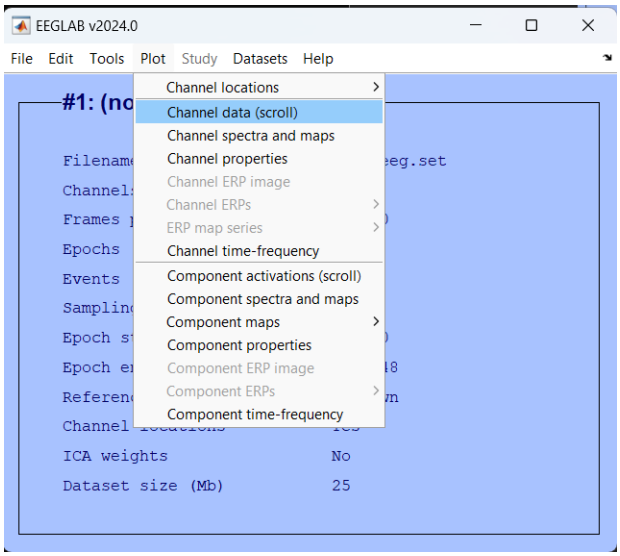
The example we will use is sub-001_task-P300testrun1_run-8_eeg.set, which can be downloaded from this link:

https://springernature.figshare.com/ndownloader/files/35134714

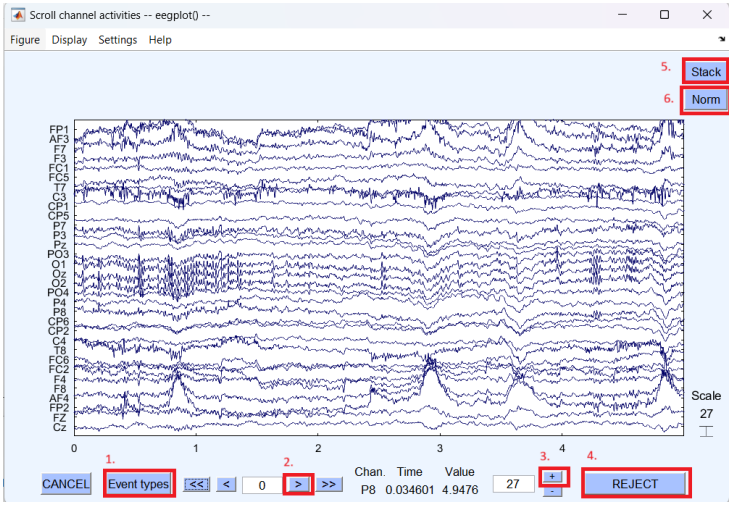
Loading .set files: File -> Load existing dataset



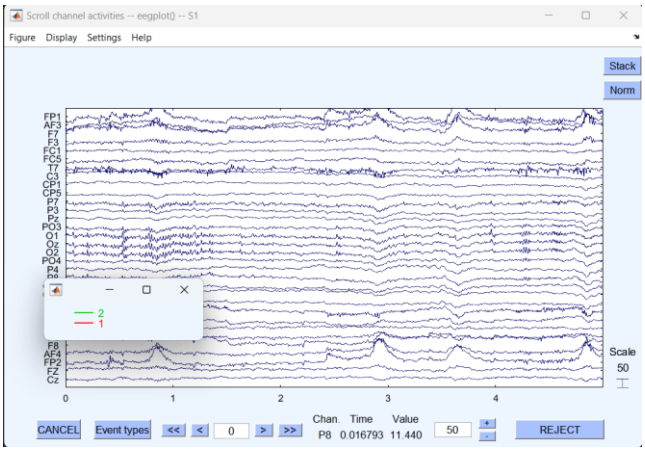
Plotting Loaded Data: Plot -> Channel data (scroll)



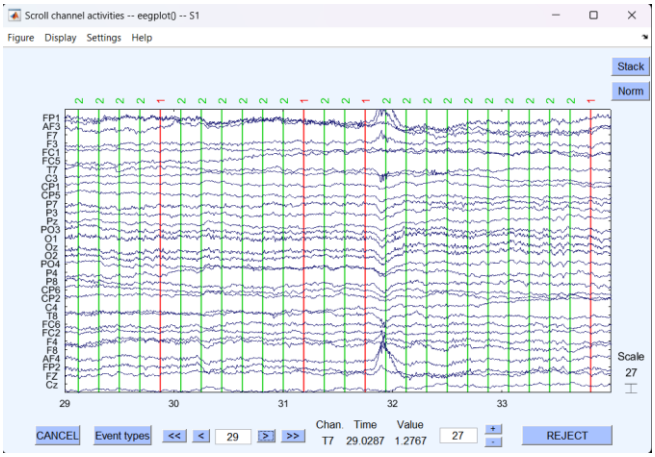
You should get the following window:



1. The "Event types" button lists all types of stimuli we have

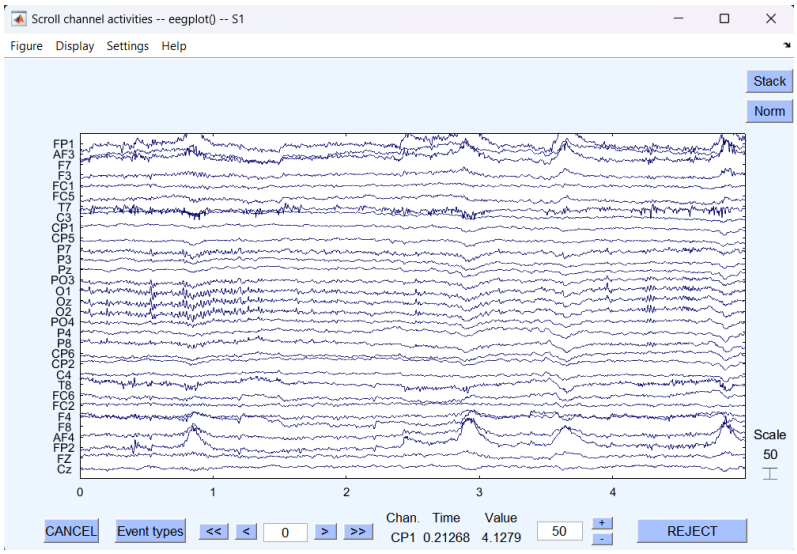


2. The button with one arrow allows navigation through the data in time, while the button with two arrows does so with larger steps. For example, data from the 29th second:



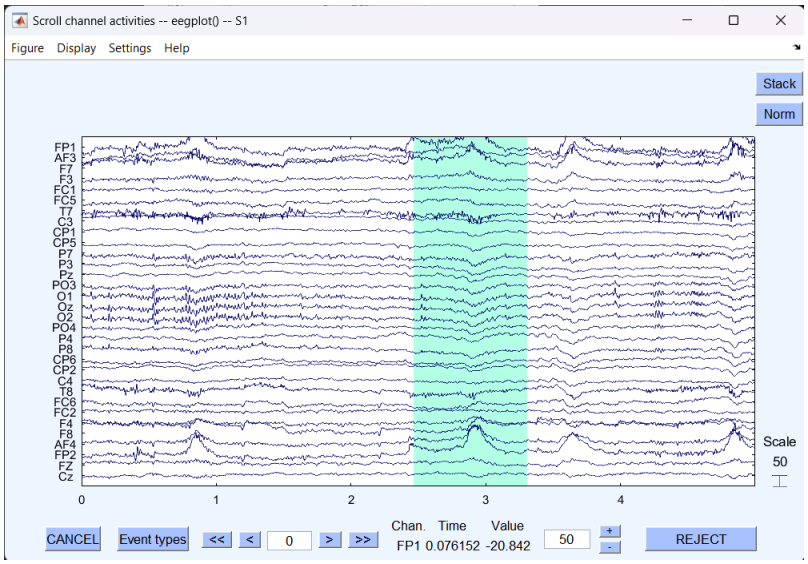
Now we can see stimuli of types 1 and 2.

3. The plus button increases the scale of the graph to a larger range of microvolts, and minus does the opposite.

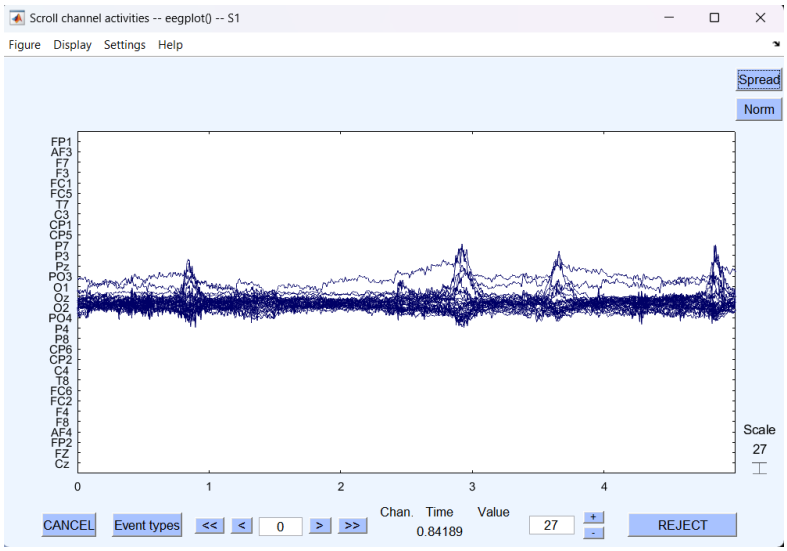


4. The "Reject" button is used to discard data that is visually identified as invalid. Data is selected by clicking and dragging over the graph, and then after marking, the "Reject" button is clicked to exclude them.

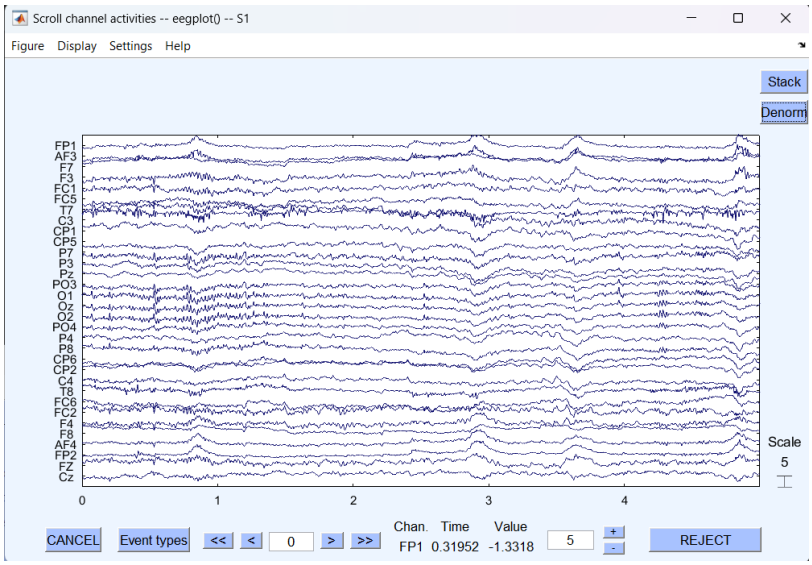
   Example of selected data:



5. *The "Stack" button displays values from all electrodes on one line, one above the other.*



6. The "Norm" button normalizes values by dividing all values by their maximum value.



Command to load .set file from a script:

EEG = pop_loadset('filename', 'filepath');

Command to plot from a script:

EEG = pop_eegplot(EEG, 1, 1, 1);

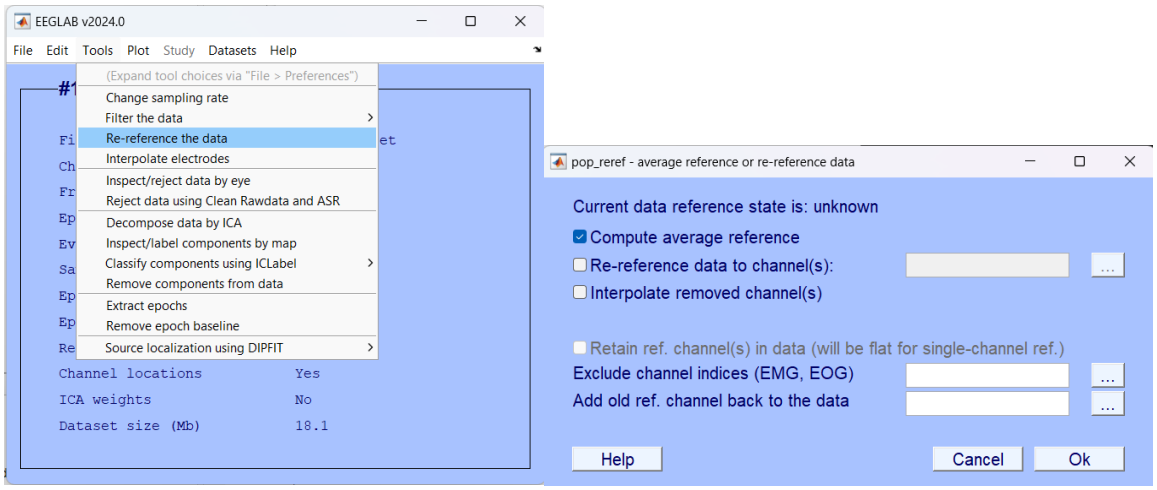
2. Data Processing

I. Referencing Method

Data referencing is performed to remove or reduce unwanted noise and artifacts that may occur during EEG signal recording. By referencing signals to a specific electrode or group of electrodes, we adjust signal values relative to that reference.
This way, we can improve the signal-to-noise ratio and ensure that the signals being analyzed represent actual brain activity rather than environmental noise. Referencing also allows comparability of signals between different electrodes and increases the accuracy of data analysis.

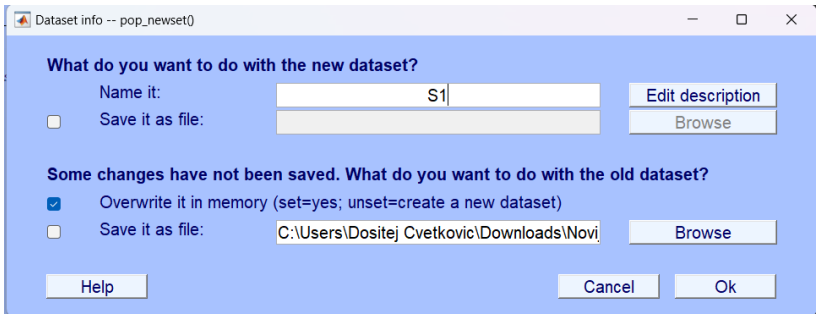
a. Average value of all channels as reference:

Tools -> Re-reference the data



The "Compute average reference" option is selected to re-reference each EEG signal to the average value of all electrodes, removing potential biases that may arise from selecting a specific reference electrode, which can improve signal quality and enable more accurate analysis. This method is especially useful when a uniform reference representing global brain activity is desired rather than local activity specific to one electrode.

A window will appear where you can name the new file; you also have the option to create a new file or overwrite the loaded one by checking the option "Overwrite it in memory," which we will do:



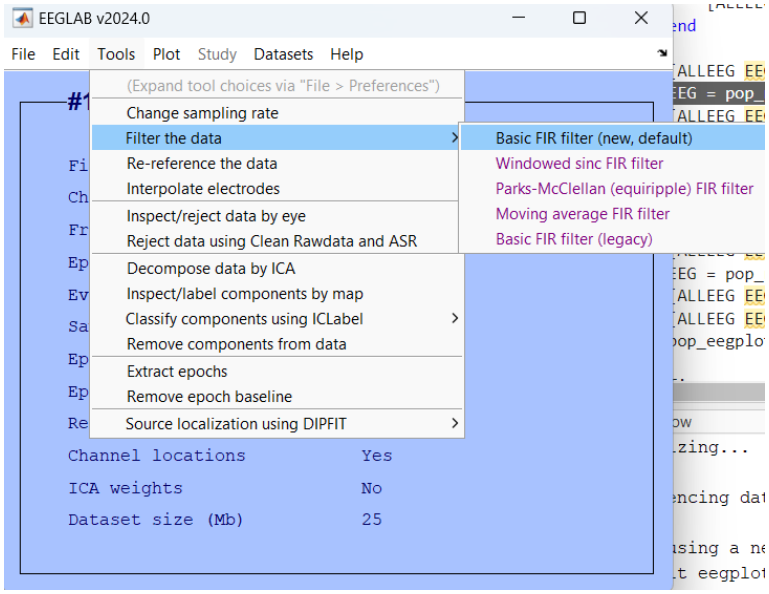
Command for re-referencing from a script:

```
EEG = pop_reref( EEG,[]);
```

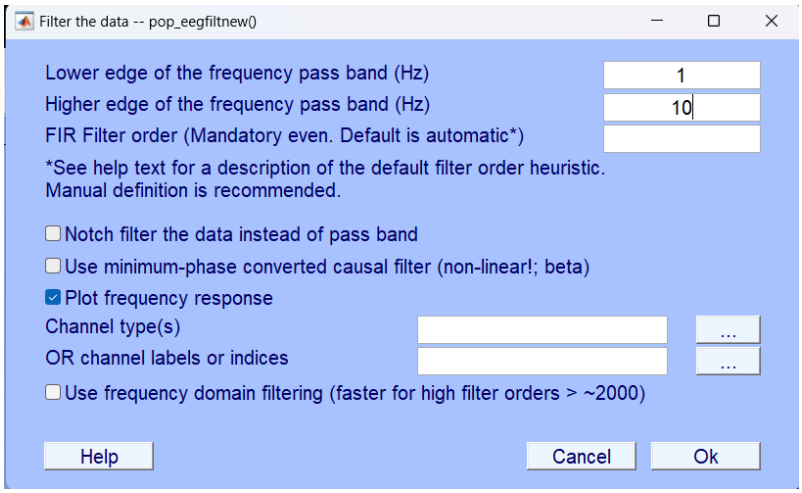
II. Filtering data

Filtering evoked potentials is crucial as it removes noise and artifacts from EEG signals (e.g., movement artifacts, muscle artifacts, eye blink and movement artifacts, electromagnetic noise, heart disturbances, electrode contact issues). The range of interest is 1-10Hz.

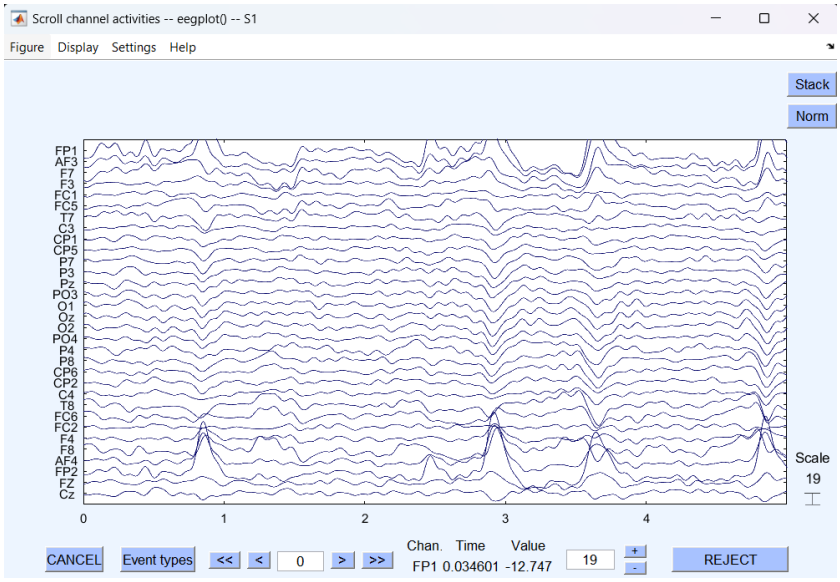
Tools -> Filter the data -> Basic FIR filter (new, default)



We select a bandpass of 1-10Hz:



The signal after filtering should look like this (Plot -> Channel data (scroll)):

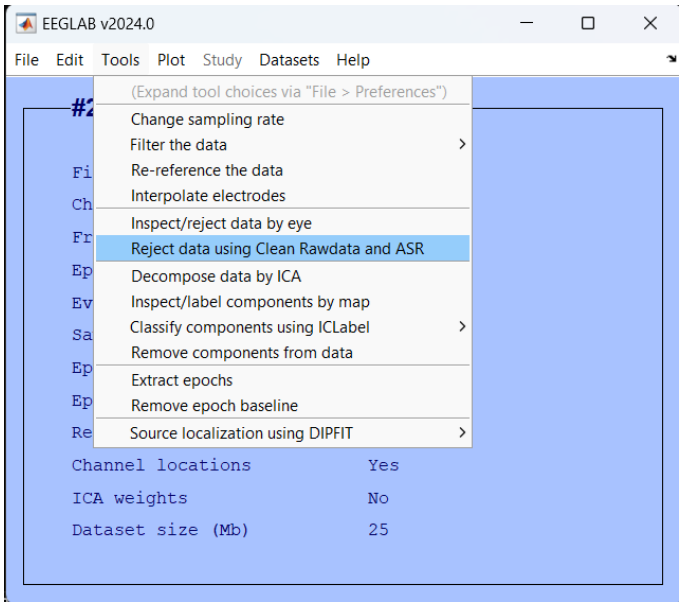


Command for filtering from a script:

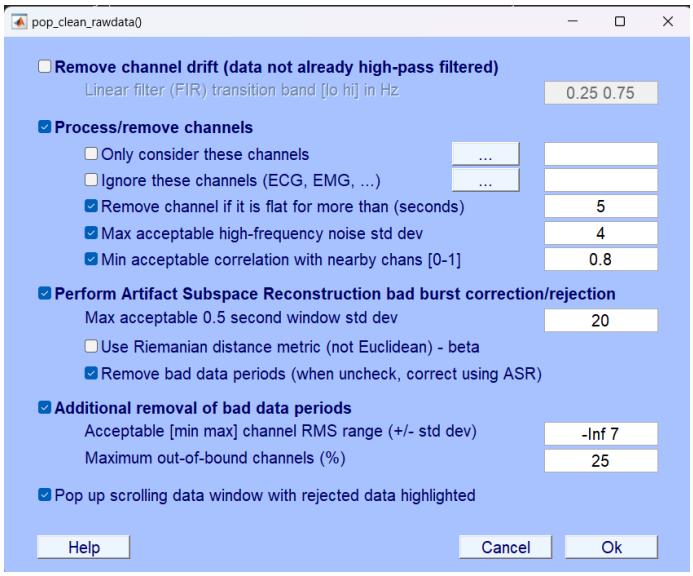
```
EEG = pop_eegfiltnew( EEG,'locutoff',1,'hicutoff',10,'plotfreqz',0);
```

III. Data Cleaning using Clean Rawdata

Tools -> Reject data using Clean Rawdata and ASR



The following window opens:

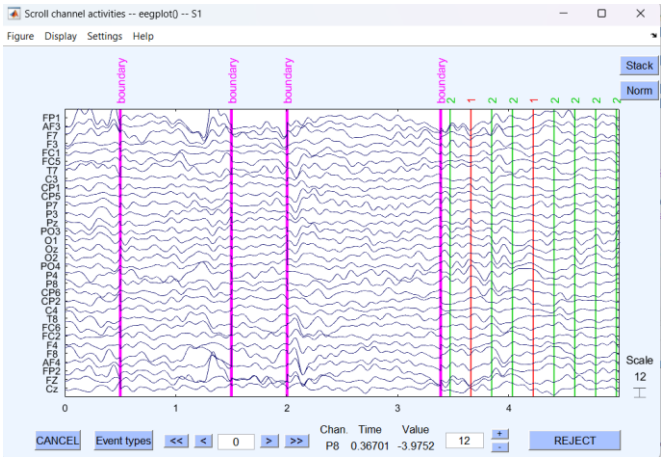


We have the following options:

- Remove channel if it is flat for more than (seconds) - Removes a channel if the signal is flat for longer than the specified time. If the signal is flat for more than 5 seconds, it usually indicates that the channel no longer records valid data, making it considered faulty or unusable.

- Max acceptable high-frequency standard deviation - Sets the maximum allowable standard deviation for high frequencies. This option helps identify channels with excessive noise in high frequencies. A threshold of 4 standard deviations is high enough to filter most natural variations in the signal, but low enough to detect significant anomalies.
- Min acceptable correlation with nearby channels [0-1] - Defines the minimum acceptable degree of correlation between signals on neighboring channels (values between 0 and 1). A threshold of 0.8 for correlation is often used as it represents a relatively high degree of similarity between signals on neighboring channels. A correlation less than 0.8 may indicate that one channel is recording signals significantly different from those on neighboring channels, which may be a sign that the channel is problematic or recording noise instead of actual signals.
-

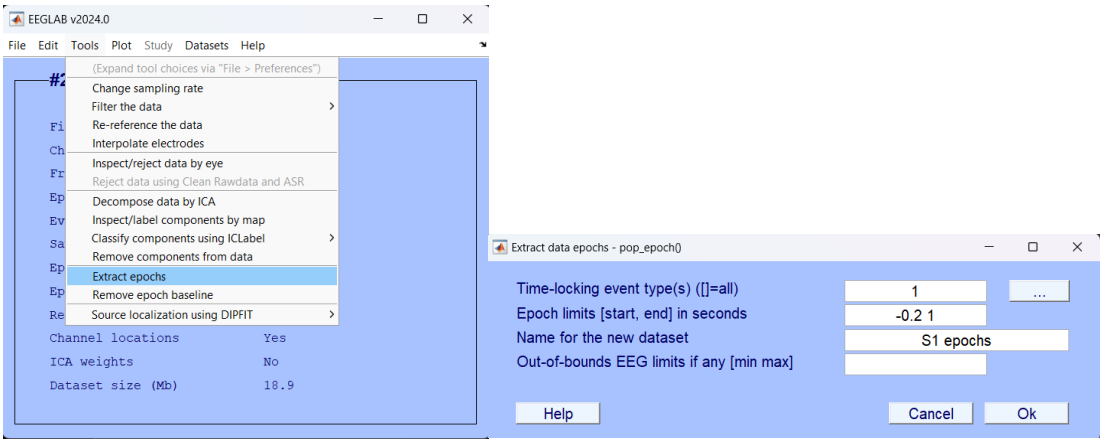
  Appearance of data after cleaning:



We can see boundary purple lines indicating breaks where the program removed faulty data.

3. Extracting Evoked Potentials

Tools -> Extract epochs



We select the stimulus named 1 as it is the trigger for the evoked potential we are interested in. The Epoch Limits option defines how much time before and after the stimulus we want to extract from the data, expressed in seconds; we are interested in 200ms before and 1s after the stimulus.

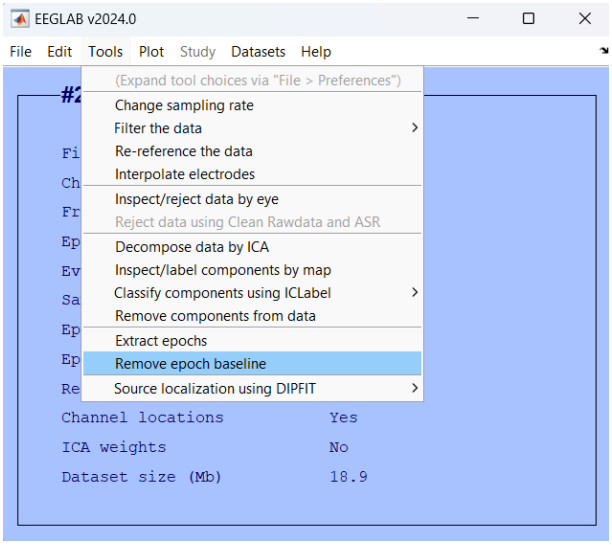
Command for epoch extraction from a script:

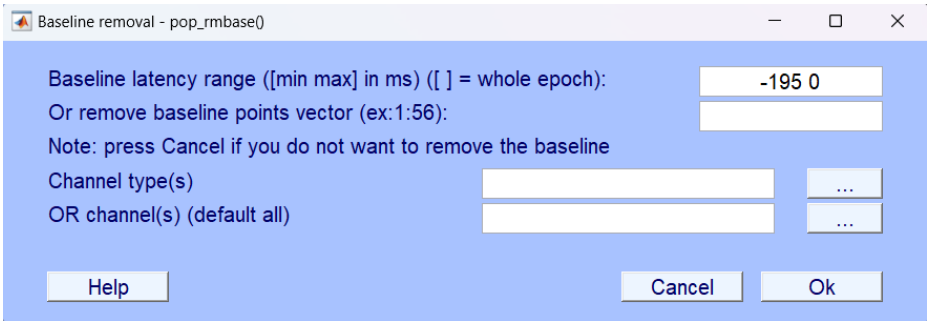
```
EEG = pop_epoch( EEG,{'1'},[-0.2 1] ,'epochinfo','yes');
```

4. Removing the Baseline from Evoked Potentials

Removing the baseline from the extracted epochs is essential as it helps eliminate low-frequency noise and unwanted variations not related to the stimulus.

Tools -> Remove epoch baseline





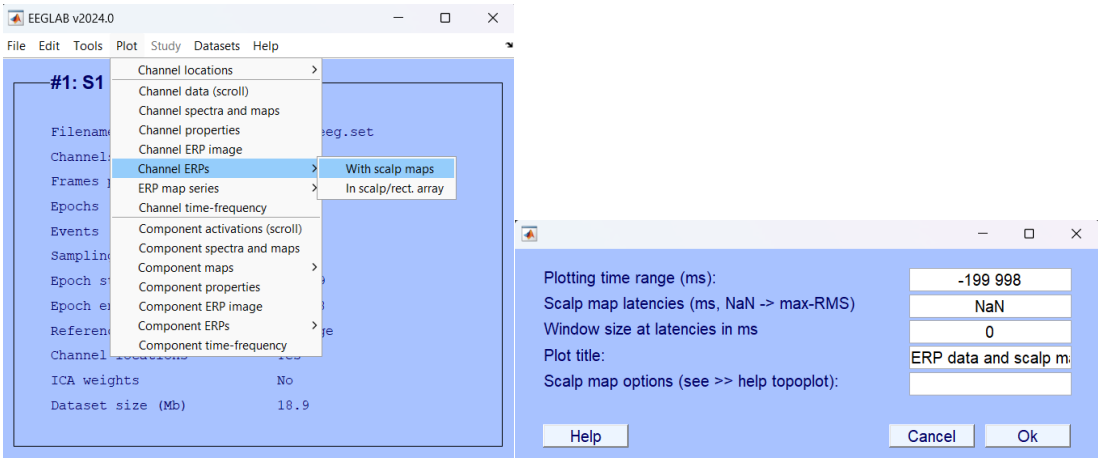
We gather as much information as possible about the baseline (195ms before the stimulus) so that we can accurately remove low-frequency variations (drift) from the epochs.

Komanda za uklanjanje drift-a iz skripte:

```
EEG = pop_rmbase( EEG,[-195 0] ,[]);
```

5. Plotting Evoked Potentials

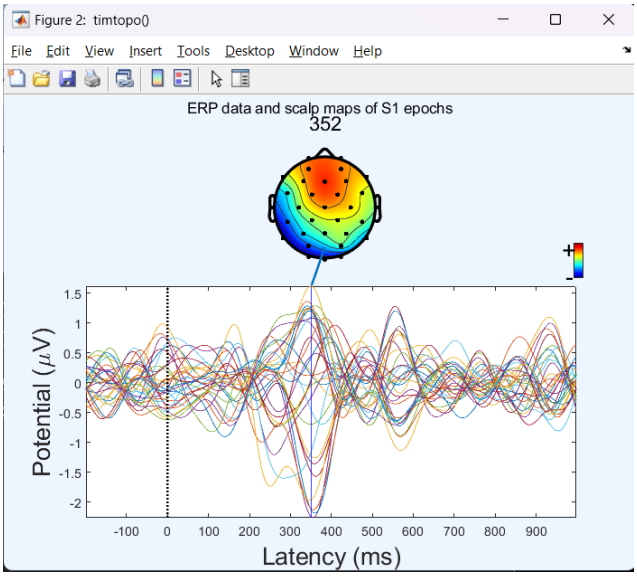
a. To display the average ERP of all epochs and the scalp activity map at selected latencies, **Plot → Channel ERPs → With scalp maps**



***Plotting time range**: Plots the graph in the specified time range; we will use the entire extracted range.*

Scalp map latencies: Determines the latency (time after the stimulus) for drawing the scalp activity map. If we set it to NaN, the maps will be displayed at the latency with the maximum standard deviation of the extracted signal.

The following graph should be obtained:



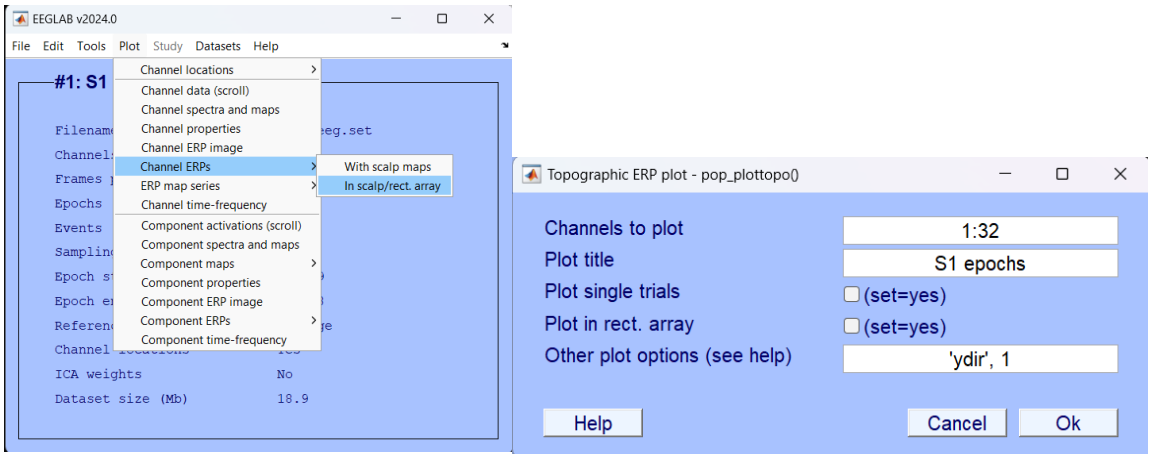
In the obtained graph, each line represents the average ERP for one channel, while the scalp map shows the distribution of the average potential at a specific latency.

We see that the latency with the maximum standard deviation is 352 ms, which is expected since we are examining the P300 evoked potential.

The command for plotting from the script is:

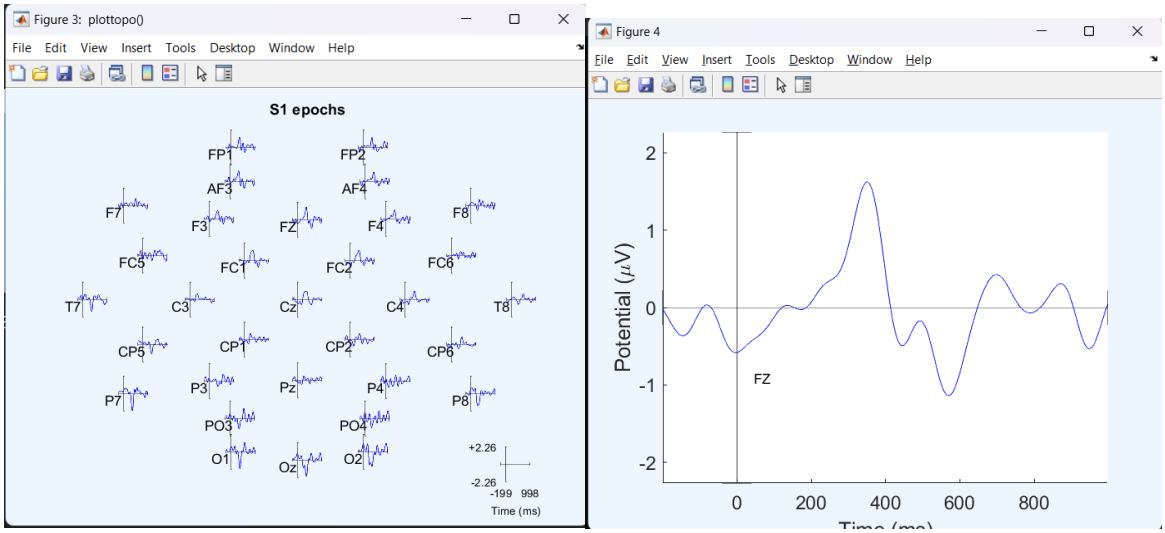
```
figure; pop_timtopo(EEG, [-199.2188     998.0469], [NaN], 'ERP data and scalp maps');
```

b. Plotting the Topographic Map, Evoked Potentials by Channel



We select the channels for which we will plot the evoked potentials; we will plot for all 32 channels (1:32).

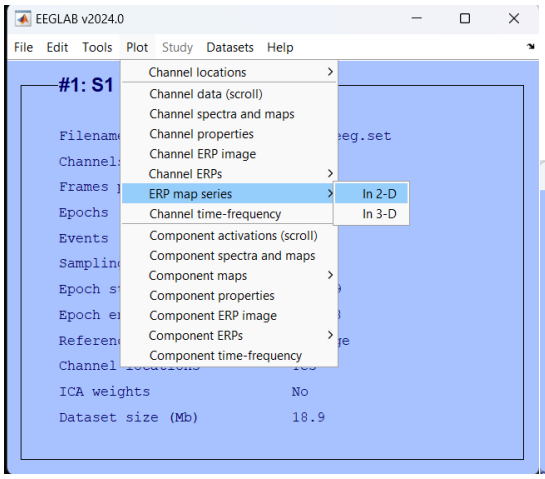
We get the following graph, and by clicking on a specific channel, we obtain an enlarged graph of that channel, e.g., FZ:

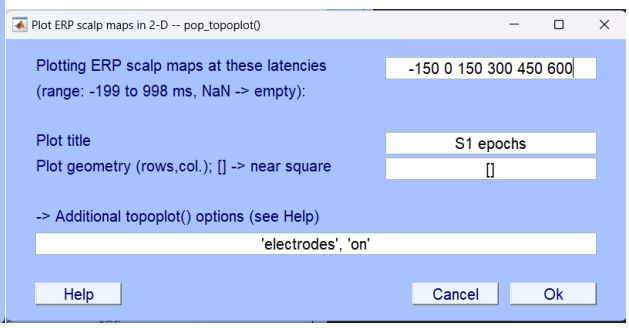


The command for plotting from the script is:

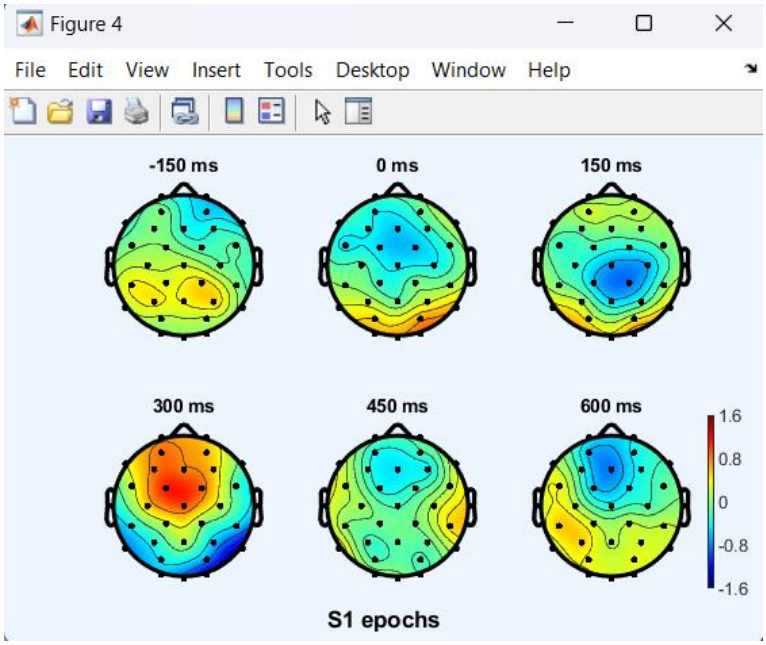
```
figure; pop_plottopo(EEG, [1:32] , '', 0, 'ydir',1);
```

c. Plotting Scalp Activity Maps at Specific Latencies





We choose the latencies for which we want to plot the scalp activity maps.a



We obtained the expected result with the highest activity at 300 ms (P300).

The command to run from the script is:

```
pop_topoplot(EEG, 1, [-150:150:600] ,'',[2 3] ,0,'electrodes','on');
```

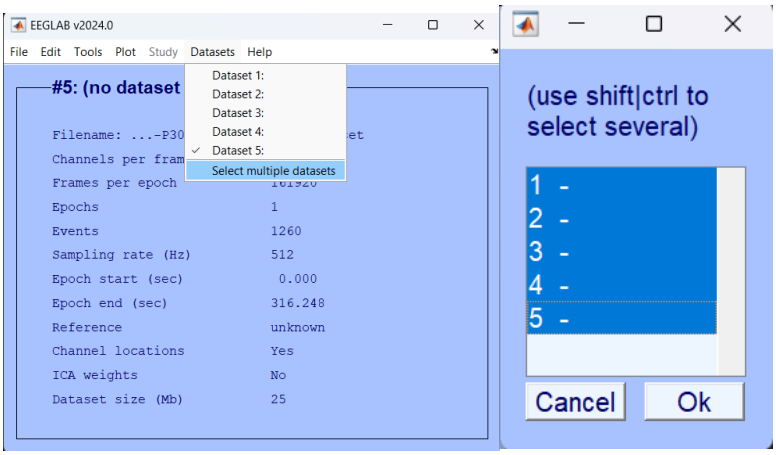
6. Creating a Study

If we have multiple subjects, we can create a study that analyzes evoked potentials averaged across all subjects to obtain consistent and representative results.

The first step in creating a study is loading all the files. For example, we will use P300testrun1_run-8_eeg.set from subjects sub002, sub003, sub004, sub005, and sub006.

Once we load all the files, we can select them for simultaneous processing as follows:

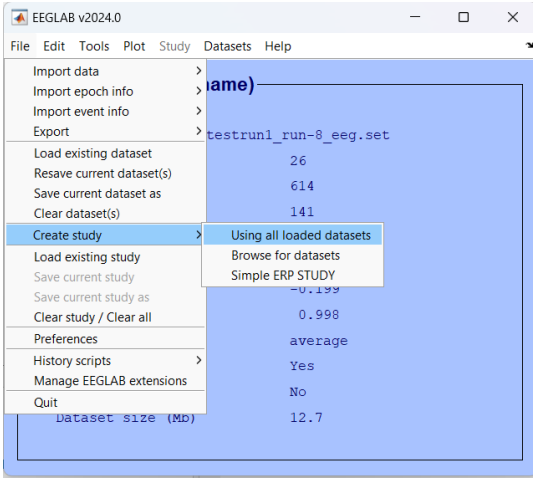
Datasets -> Select multiple datasets



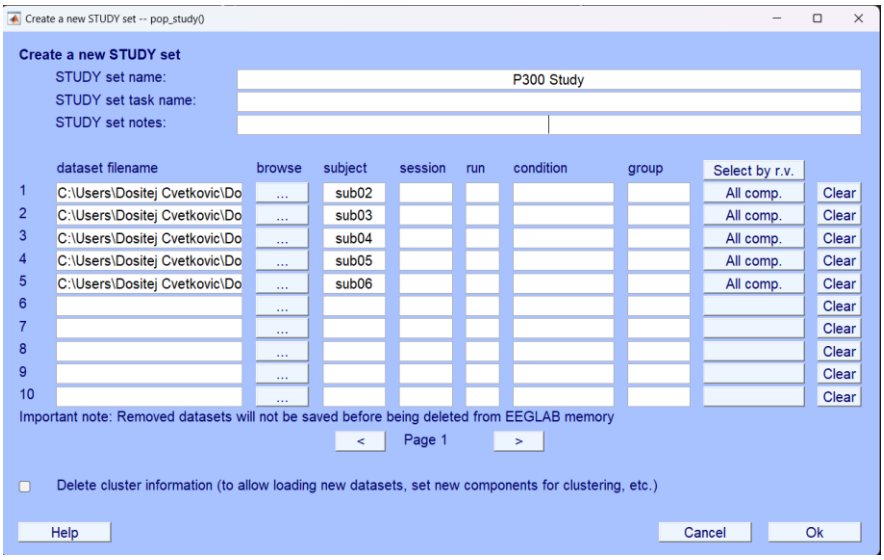
Repeat all steps from the beginning as for sub001

Then select only one file again and go to:

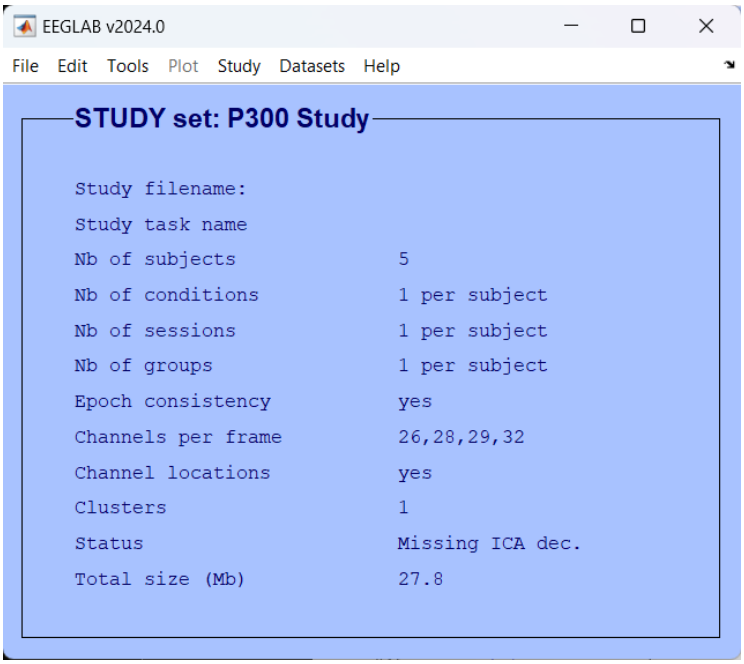
File -> Create study -> Using all loaded datasets



Here we can name the study, add more files, and we have an overview of the loaded files.



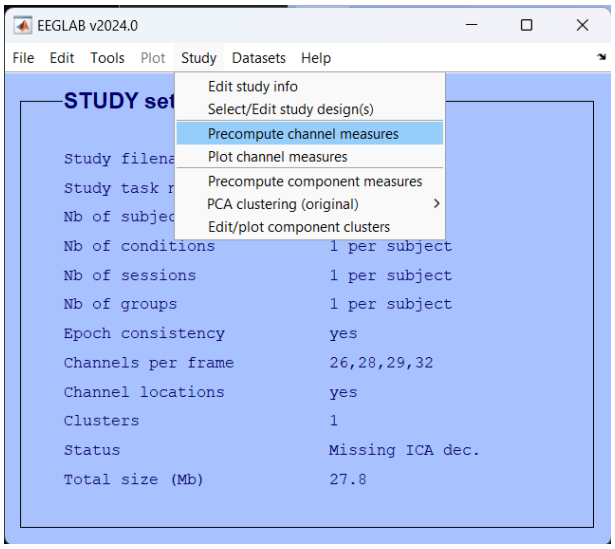
Environment Appearance:



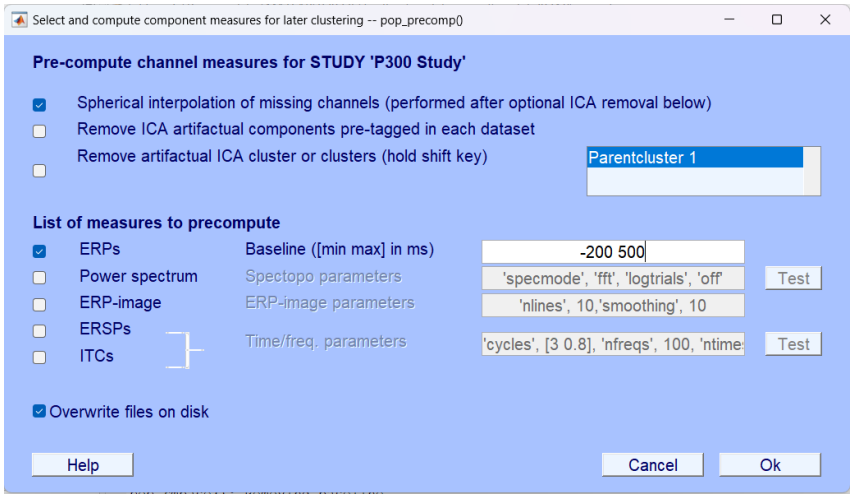
The command for creating the study from the script is:

[STUDY ALLEEG] = std_editset(STUDY, ALLEEG, 'name','P300 Study','updatedat','on','rmclust','on');

Then we go to **Study -> Precompute channel measures**



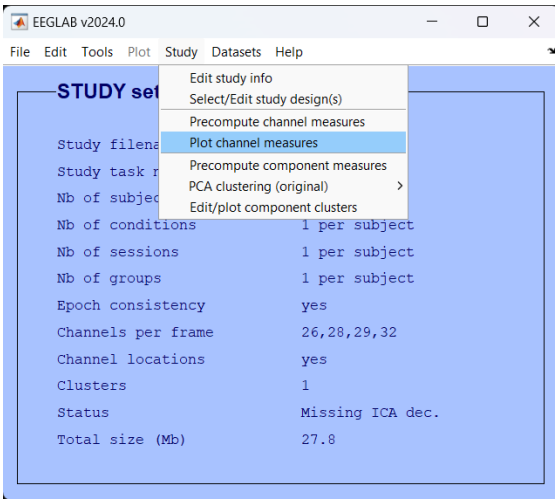
Here we choose to precompute the ERP (event-related potential) as a measure, focusing on the period from 200 ms before the stimulus to 500 ms after the stimulus.

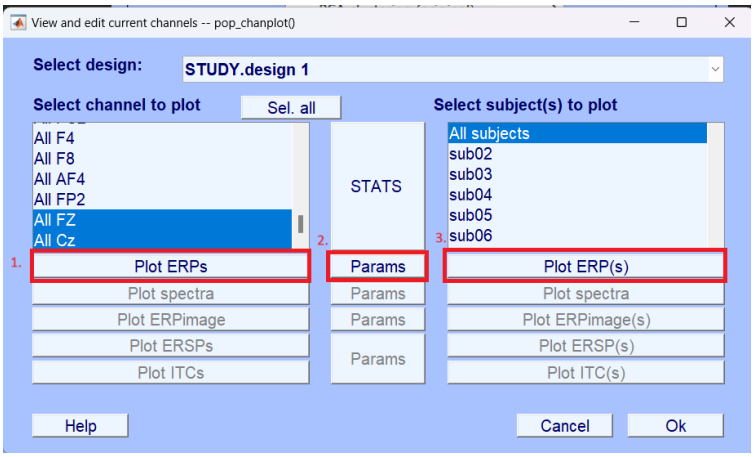


The command for the script is:

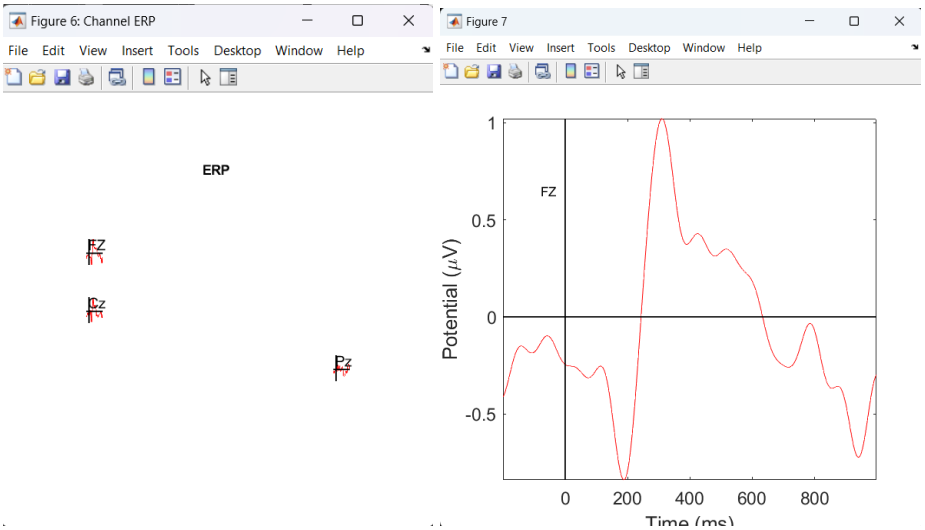
[STUDY,ALLEEG]=std_precomp(STUDY,ALLEEG,{},'savetrials','on','interp','on','recompute','on','erp','on','erpparams',{'rmbase',[-200 500] });

Plotting Evoked Potentials of the Study

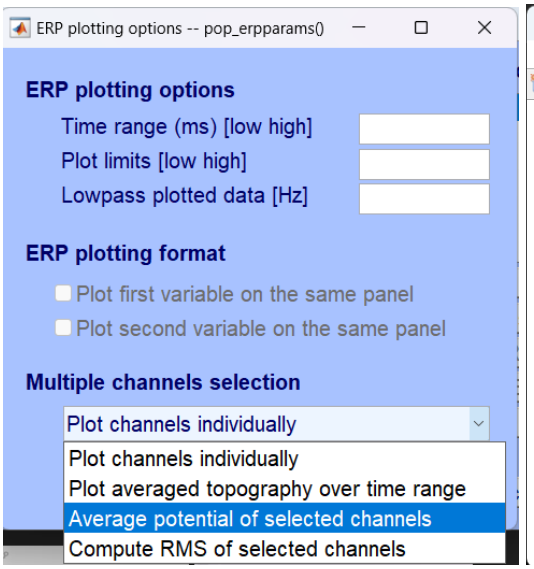


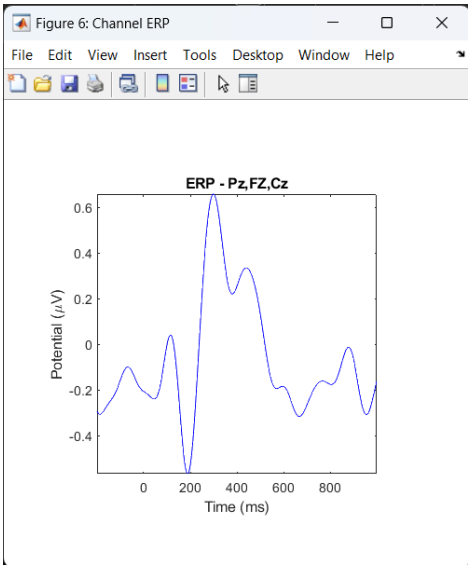


1. Plots the average evoked potential of all subjects for each electrode; by clicking on the electrode, e.g., FZ, we obtain the graph for that electrode.



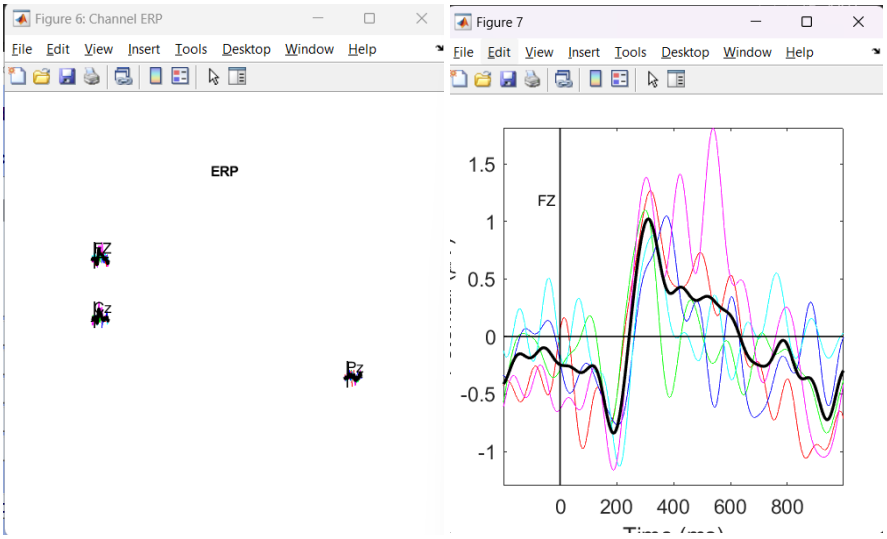
2. By clicking on the Params button and selecting the option Average potential of selected channels, we plot the potentials averaged across all selected electrodes.





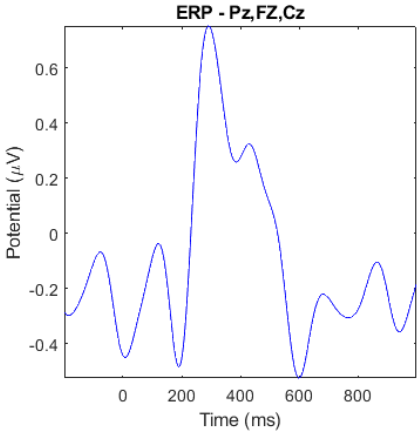
From the script: STUDY = pop_erpparams(STUDY, 'averagechan','on');

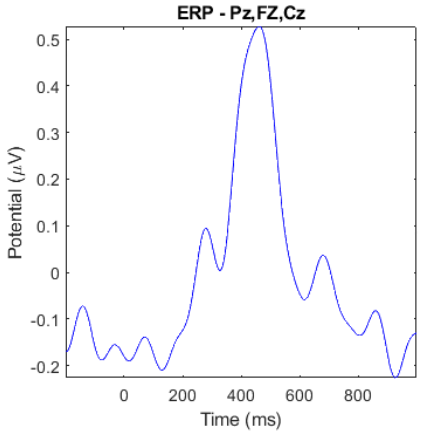
3. Similar to option 1, it additionally plots the evoked potential for each subject over the average.



3 RESULTS

The obtained graphs for averaged P300 and RSVP potentials, respectively:





4 DISCUSSION

This paper presents a template for processing evoked potentials using EEGLAB. The processing of evoked potentials is essential in neurophysiological research, as it allows for an understanding of brain reactions to specific stimuli. Efficient and precise extraction of VEP signals from EEG data is crucial for achieving accurate and reliable results.

The proposed processing template includes key steps such as data loading, filtering, re-referencing, data cleaning, and extracting evoked potentials. Each of these steps is carefully selected and optimized to ensure high signal fidelity and minimize noise and artifacts that may affect the analysis.

The importance of this procedure lies in its ability to enhance the signal-to-noise ratio, which is critical for the extraction and analysis of VEPs. Data cleaning using the ASR method, automatic rejection of poor-quality channels, and removal of noise from the electrical grid are key steps that preserve significant neural information while effectively eliminating unwanted artifacts.

Moreover, using a data processing pipeline not only saves time but also standardizes the analysis process, reducing the likelihood of human error and increasing the consistency of results. This is especially important when dealing with a larger number of datasets and subjects, as it ensures consistency in processing and analysis.

5 CONCLUSION

This paper presents a comprehensive approach to processing evoked potentials (VEP) using the EEGLAB software package. The developed data processing template enables standardized and efficient analysis of EEG signals, ensuring high precision and reliability of results.

6 REFERENCES

[1] *Muñoz Bohollo, L., Porr, B. and Dahiya, R. (2022)* EEG and P300 database to determine the signal to noise ratio during a variety of realistic tasks. *[Data Collection]*

[2] *https://eeglab.org/tutorials/*