### Project Report: EEG Data Analysis

#### 1. Introduction

The objective of this project was to analyze EEG data from a task-related pain study using various preprocessing steps and analytical methods. The analysis aimed to extract meaningful neural patterns, compute event-related potentials (ERPs), and assess spectral power and connectivity measures. This project utilized MATLAB and the EEGLAB toolbox for processing and analyzing the EEG data.

#### 2. Data Acquisition

The dataset was obtained from a CSV file located at:

*/MATLAB Drive/THE Model/eeglab\_current/eeglab2024.2/sub-esgpilot02\_task-pain\_run-01\_channels.tsv*

This dataset contained EEG recordings from participants during a pain-related task.

#### 3. Methodology

The analysis followed several key preprocessing steps:

* ***Import Data-***The EEG data was imported from a TSV file using the readtable function. The data was checked for structure and integrity.
* ***Filtering- A*** bandpass Butterworth filter was applied to the data with a low cutoff frequency of 1 Hz and a high cutoff frequency of 40 Hz to remove noise and artifacts.
* ***Re-referencing-*** The data was re-referenced using the average reference method to ensure consistent reference across channels.
* ***Epoching-*The continuous EEG data was segmented into epochs from -200 ms to 800 ms around defined event onsets, allowing analysis of brain responses associated with specific stimuli.**
* ***Baseline Correction-***The mean amplitude during the baseline period (0 to 200 ms) was subtracted from each epoch to correct for baseline shifts.
* ***Artifact Detection and Removal-***Outliers exceeding a defined threshold of 100 microvolts were detected and removed to ensure data quality.
* ***Rejection of Bad Channels-***Channels identified as problematic were excluded from further analysis.
* ***Interpolation-***Missing data points in the cleaned dataset were interpolated to maintain data integrity.

#### 4. Data Analysis

The following analyses were performed on the preprocessed data

* ***Event-Related Potentials (ERPs)-*** The average response across all epochs was computed to visualize the ERP waveform.
* ***Time Series Analysis-***The cleaned data served as the basis for time series analysis, examining the temporal dynamics of EEG signals.
* ***Spectral Power Analysis-*** The power spectral density was estimated using the Welch method to assess frequency-specific brain activity.
* ***Connectivity Measures-*** Correlation coefficients between different channels were computed to analyze functional connectivity.
* ***Group Averages-*** Group averages were calculated to understand the overall neural response across participants.

#### 5. Statistical Testing

A one-sample t-test was conducted on the cleaned data to assess significant differences in the EEG responses. The results provided insights into whether the observed changes in brain activity were statistically significant.

#### 6. Results

* **ERPs-** The group average ERP waveform indicated the presence of specific neural responses to the pain-related task.
* **Spectral Power-** Analysis revealed frequency bands with significant power changes during the task.
* **Connectivity-** Connectivity measures indicated potential relationships between brain regions during the pain stimulus.

#### 7. Conclusion

This project successfully demonstrated the application of EEG data preprocessing and analysis techniques to study neural responses associated with pain. The findings contribute to our understanding of how the brain processes pain stimuli and highlight the importance of robust preprocessing methods in EEG research. Future work could involve expanding the analysis to include more subjects or different experimental conditions to enhance the findings further.

#### 8. Future Work

Future studies could explore:

1. The influence of individual differences (e.g., pain sensitivity, psychological factors) on EEG responses.
2. The application of machine learning techniques to classify EEG patterns associated with different types of pain stimuli.
3. Longitudinal studies to assess changes in brain activity over time in response to pain treatment interventions.

**THE CODE**

|  |
| --- |
| % Clear the workspace and command window  clear; clc;  % Load the necessary EEGLAB library  cd('/MATLAB Drive/THE Model/eeglab\_current');  addpath(genpath('/MATLAB Drive/THE Model/eeglab\_current'));  savepath; % Save the changes  % Initialize EEGLAB  eeglab;  % Import EEG data from CSV file  data = readtable('/MATLAB Drive/THE Model/eeglab\_current/eeglab2024.2/sub-esgpilot02\_task-pain\_run-01\_channels.csv');  % Check the structure of the data to identify the columns  disp(data);  % Extract relevant columns for EEG data and convert types if needed  eeg\_data = data{:, 3:end}; % Adjust as necessary based on your dataset structure  eeg\_data = cell2mat(eeg\_data); % Convert cell array to numeric array if applicable  % Filtering parameters  low\_cutoff = 1; % Low cutoff frequency (Hz)  high\_cutoff = 40; % High cutoff frequency (Hz)  fs = 250; % Sampling frequency (Hz)  % Bandpass filter  [b, a] = butter(2, [low\_cutoff high\_cutoff] / (fs / 2), 'bandpass');  filtered\_data = filtfilt(b, a, eeg\_data);  % Re-referencing (average reference)  re\_reference\_data = filtered\_data - mean(filtered\_data, 2);  % Epoching (e.g., from -200 ms to 800 ms around events)  event\_onset = 1; % Example onset (you'll need to adjust based on your data)  epoch\_length = [0.2, 0.8]; % Start and end in seconds  num\_samples = size(re\_reference\_data, 1);  epoch\_start = round(event\_onset \* fs) + round(epoch\_length(1) \* fs);  epoch\_end = round(event\_onset \* fs) + round(epoch\_length(2) \* fs);  epochs = re\_reference\_data(epoch\_start:epoch\_end, :);  % Baseline correction  baseline\_period = [0, 0.2]; % 0 to 200 ms baseline  baseline\_start = round(event\_onset \* fs) + round(baseline\_period(1) \* fs);  baseline\_end = round(event\_onset \* fs) + round(baseline\_period(2) \* fs);  baseline\_mean = mean(re\_reference\_data(baseline\_start:baseline\_end, :), 1);  baseline\_corrected\_data = epochs - baseline\_mean;  % Artifact detection and removal  % Define thresholds for detection  threshold = 100; % microvolts  artifacts = abs(baseline\_corrected\_data) > threshold;  cleaned\_data = baseline\_corrected\_data(~any(artifacts, 2), :);  % Rejection of bad channels  % Example of rejecting channels (you can specify which channels to exclude)  bad\_channels = []; % Specify bad channels here  cleaned\_data(:, bad\_channels) = [];  % Interpolation if needed  % Assuming cleaned\_data has gaps that need interpolation  if any(isnan(cleaned\_data), 'all')  for i = 1:size(cleaned\_data, 2)  if any(isnan(cleaned\_data(:, i)))  cleaned\_data(:, i) = fillmissing(cleaned\_data(:, i), 'linear');  end  end  end  % Extracting ERP, time series, spectral power, and connectivity measures  erp = mean(cleaned\_data, 1);  time\_series = cleaned\_data; % Using cleaned\_data as time series  spectral\_power = pwelch(cleaned\_data, [], [], [], fs);  connectivity\_measure = corr(cleaned\_data);  % Compute group averages (if you have multiple subjects, average across them)  % Assuming group\_data is a 3D array where the first dimension is subjects  group\_average = mean(cleaned\_data, 1);  % Plotting group averages  time\_vector = (1:size(group\_average, 2)) / fs; % Time vector in seconds  figure;  plot(time\_vector, group\_average);  xlabel('Time (s)');  ylabel('Amplitude (uV)');  title('Group Average ERP');  % Statistical test (e.g., t-test between conditions)  % Adjust based on your conditions  [~, p\_value] = ttest(cleaned\_data); % Example: one-sample t-test  % Save the results  save('EEG\_Analysis\_Results.mat', 'erp', 'time\_series', 'spectral\_power', 'connectivity\_measure', 'group\_average', 'p\_value'); |

**\*\*\*\*THE END \*\*\*\***