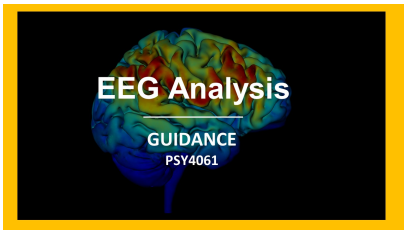
EEG Analysis

Jon Silas & Alex Jones

2023-10-04

On this page

Introduction



This is a companion guide to the module PSY4061 Practical Cognitive Neuroscience and is aimed specifically at helping you with the component of the portfolio linked to 'EEG Analysis'. Your portfolio is worth 80% of your grade and EEG Analysis and the write up is worth 40% of the portfolio.

> ! Portfolio Deadline
>
> You should submit your final functioning EEG analysis write up as part of your portfolio. **Portfolio submission deadline - 15.04.24 10am**

For this part of your assessment you will have to analyse some EEG data and then answer some general questions about EEG analysis and interpret some EEG data. The EEG analysis is taught in a program called **EEGLAB**.

The resources and support in this online guide should be used in conjunction with the other forms of support available to you; in-class, via email and in one-to-ones with the module leaders. This whole guide can be downloaded as a PDF or word document by clicking the download button in the menu and selecting your preferred format.

How to use this book/website

For the most part you should go through this book in a linear fashion; page-by-page. You can skip forward one page at a time by using the arrow key at the bottom right of each page. If you need to come back to some of the information here, use this as you would any website; the search bar - on the left - should help you find anything you are looking for. This website works best on a computer not on a phone or tablet.

You might also find the icons in the navigation bar helpful:

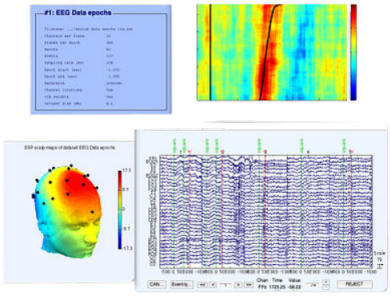
Link to github code for this book.

Click to download this book as a .pdf, .docx, or .epub file.

Click to share to the link to the book with others.

Click to switch between light and dark mode.

Using EEGLAB



Background

This guide serves as a specific set of instructions to talk you through analysing data from Jones & Ward (2019). As you know the experiment explores the differences in the ability to recognise objects when they've been encoded under two different conditions. Specifically, items to be remembered were either presented rhythmically or arrhythmically. The data we will examine in this tutorial was recorded at the recognition phase – so there is no perceptual difference between the items we are comparing, the only difference between them is that when the items were presented to be remembered they were presented rhythmically or arrhythmically. We will use a well-established technique that will allow us to generate an Event Related Potential (ERP). This will allow us to examine the electrophysiological response generated by the brain during the presentation of stimuli that were encoded under two different conditions.

Analysis

This tutorial will take you through the analysis procedure for data from one participant and how to generate a 'grand average' – across multiple participants. You can apply the techniques you learn here to analyse the dataset that has been made available to you for portfolio. We implement the analysis in a software called EEGLAB. The primary benefit of using EEGLAB is that it is, for the most part free and extremely well supported by the online community. Whilst the initial distribution of EEGLAB required a programme called Matlab to run – EEGLAB can now be downloaded as a 'compiled' version. Whilst the compiled version doesn't have all the same functions as the full Matlab version – it is easy to get up and running and operates via a Graphic User Interface (GUI), so you won't need to code.

If you would prefer to install Matlab and use the EEGLAB that is attached to it you are welcome to do so, but we will be providing instruction and support on the stand alone compiled version of EEGLAB.

Teaching and Learning

As in all the software skills we hope you will develop in this programme, we hope that learning how to use EEGLAB is in itself a good skill to have. However, our primary goal is for you to become familiar with the analysis process of EEG, understand how data is cleaned, and manipulated in experimental settings. The idea is that you get a greater insight into the methods used in analysing EEG and how this effects the conclusions we draw about how the brain, and how cognition, works.

You can run through this guide on your own but there will be dedicated in class sessions where you will be able to go through the analysis and get in-person support and feedback from module staff.

Where to Get Help?

- You will be given time in class to explore EEGLAB - speak to tutors and get in person support.
- Use the Resources section for online help.
- Book an online or in-person one-to-one session with module leaders.

Getting Started

Install EEGLAB

> ! Download and install EEGLAb
>
> EEGLAB - download the compiled version

Run the installation software and open EEGLAB on your machine.

If you can't get it running on your machine, you can use machines in class and speak to staff about support on getting EEGLAB installed on loan machines.

The Matlab version of EEGLAB is somewhat more stable and there are more features that are available. We haven't used this as it's not completely free and we want what you learn to be available to you after you leave. However, if you want to install Matlab, for now Middlesex provides a liscence, create an account and download Matlab here:

- Matlab Middlesex Licence Version

Once installed and up and running, you can follow the instructions here on how to get EEGLAB running from within Matlab:

- EEGLAB for Matlab

Download the data for this tutorial

> ! Download the data
>
> Click here to download the data for this tutorial

The data for this tutorial includes data from 6 participants - this isn't the whole data analysed in the Jones & Ward (2019) paper but a subset - we think 6 participants is enough for you to get used to analysis and produce a good ERP by the end of the tutorial.

The file is about 3Gb and may take a while to download.

Once you've downloaded the file unzip it and extract the containing data.

Data format

Each file is named with a participant number e.g., `Participant_03`. There are 6 participant datasets in total. Each participant has 3 separate files each named the same thing with different file formats:

- `.eeg` - the actual EEG data recorded digitally.
- `.vhdr` - the 'header' information, this gives us the names and spatial location of the electrodes.
- `.vmrk` - this file stores the 'trigger information'; information sent from the experimental computer containing information about when a participant saw a particular stimuli or gave a specific response.

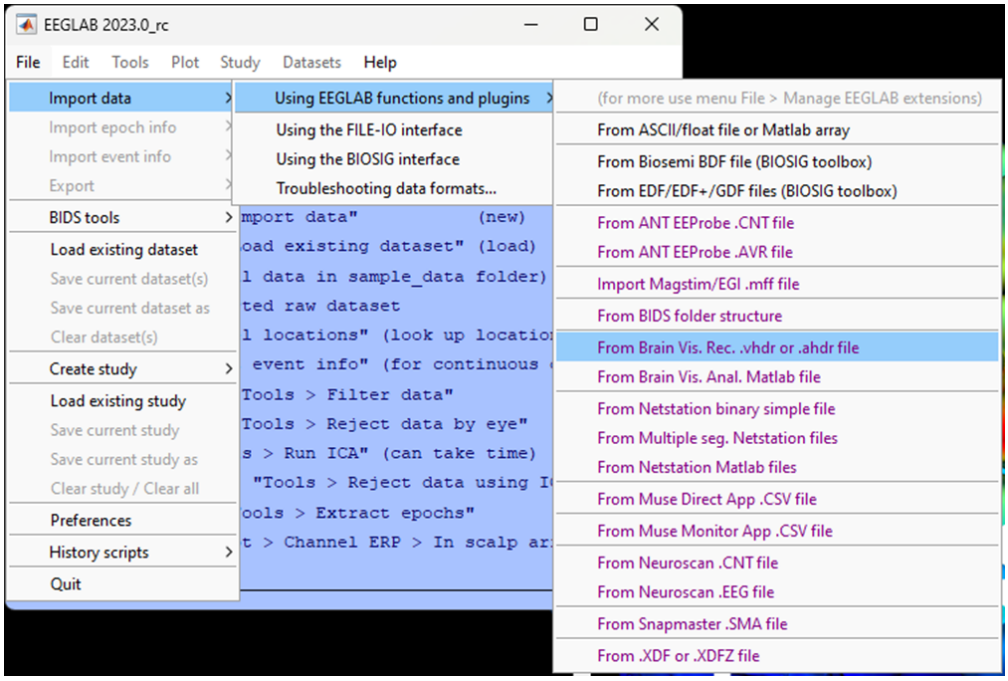
Import Data

Before we can do anything to the data, we need to import it into EEGLAB. EEG data can take many different formats depending on how it has been recorded and if it has been processed or not. We are going to import in 'raw' data that is to say, data that has not been processed at all since it's recording. The data we will take in was recorded on hardware (the actual EEG recording system) made by Brain Products, the specific amplifier is called an actiCHamp plus and we recorded data from 64 electrodes placed on scalp in a 10-20 format using an actiCAP snap headset.

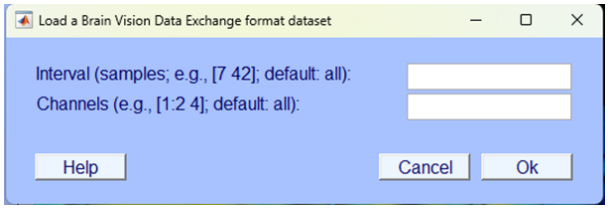
How

In EEGLAB click the following menus:

```
-> File
  -> Import data
    -> Using EEGLAB functions and plugins
      -> From Brain Vis. Rev. .vhdr or .ahdr file
```

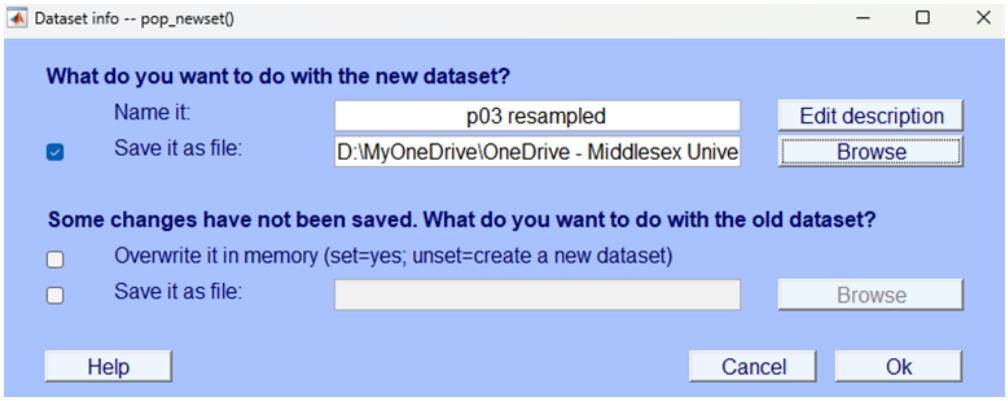


A popup menu will appear, leave the boxes blank and click `Ok`. The defaults here are telling EEGLAB you want to import all the data (intervals) from all of the electrodes (channels).



> **Terminology**
>
> It's easy to get confused with a lot of different terms often used for the same thing but we'll try to help. The terms "**electrodes**" and "**channels**" are often used interchangeably and refer to a single source of electrical activity measured from one specific location.

Next a pop will appear asking you what you want to do with the new dataset - make sure you `Name it` and select the option to `Save it to file` the click `browse` to select an appropriate directory. Use a simple naming convention, this data is from participant 03, so we've called it `p03` and saved it on a local directory, as shown in the figure below. When we save our data after further processing steps, we will amend our file names in a way that will allow us to identify what we've done to the data.



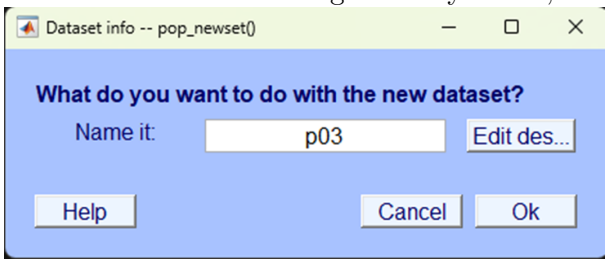
EEGLAb does not automatically save your analysis steps

Make sure you select options that will allow you to save your work.
We strongly recommend saving your data after every step of the analysis - this will mean that if you later encounter an issue with your analysis, it will be easy enough for you to pick up from an earlier point and not have to restart all over again. You should use your own labeling system so that you can easily identify each file and the processing that has been done to it.

After you've clicked `Ok` navigate back to the EEGLAB home window - this window contains useful information about you current dataset. Look through what information is displayed on the current home window. You won't understand all of what is currently displayed here but as you progress through this tutorial you should be able to learn more about what this information means. Furthermore, the home window will be a useful way to check that the processing steps have been correctly applied to the data.

Importing after the first time

You will see a different dialogue box if this isn't the first time you've imported the dataset, like below. This is nothing to worry about, name the file and click `Ok` to continue.



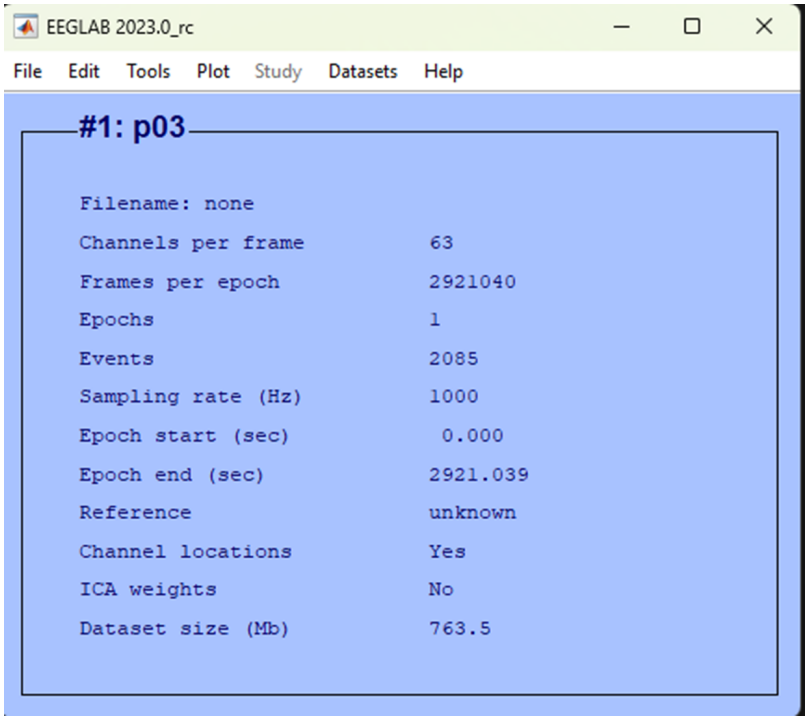


Figure 1: EEGLAB home window

Test yourself

Question 1 | What is the sample rate for the data you've just imported? _____

Question 2 | What is the unit of the sample rate?

- (A) Hz
- (B) mV
- (C) mA
- (D) ms

Question 3 | What is the sample rate?

- (A) The frequency of the EEG signal.
- (B) The number of times the EEG signal is digitized per second.
- (C) A small 'sample' of the data at a specific frequency.
- (D) The number of oscilations at a specific frequency.

Eye-balling Data

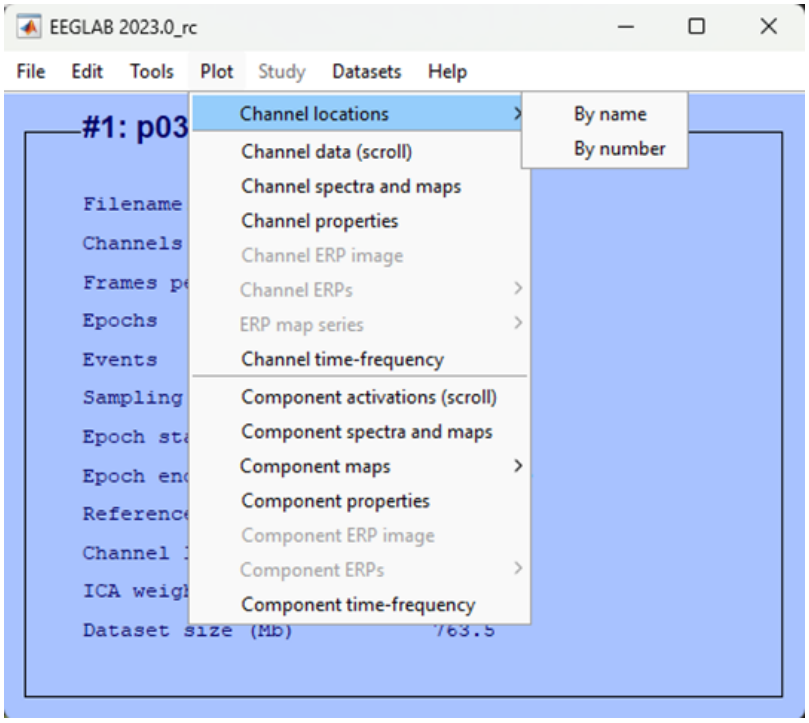
'Eye-balling' refers to visually inspecting your data, in the first instance this is usually to sense check the data you are reading in - are all the electrodes there, is the data visible and as expected. In this section we'll show you how to use EEGLAB to quickly visually examine a few key aspects of the data - but you should feel free to do this whenever you want to check your data and see what it looks like.

How

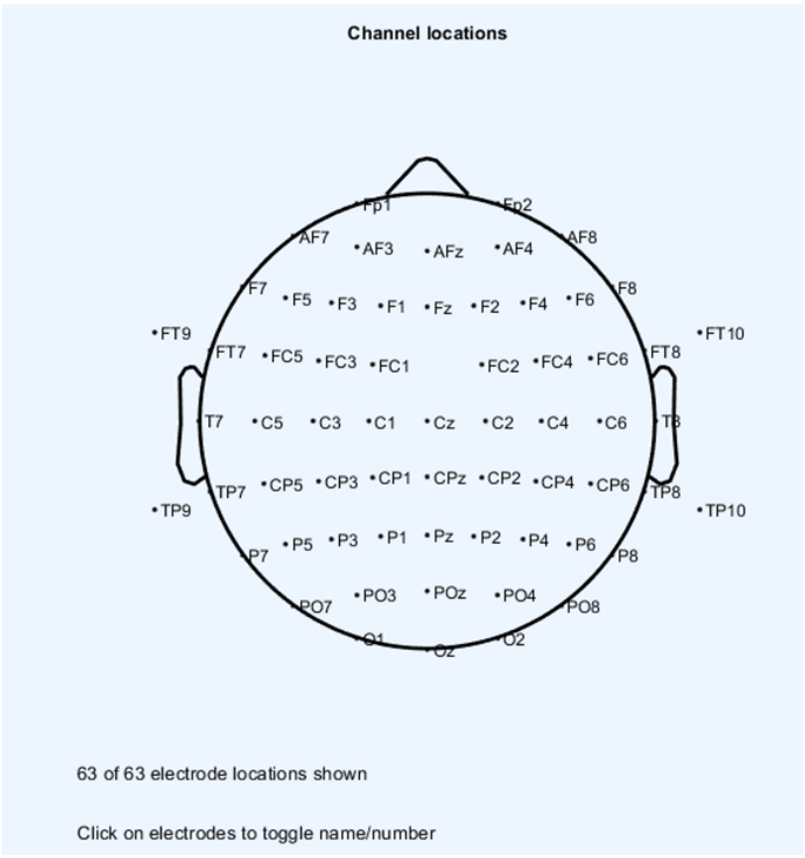
Plot channel locations

Let's start by making sure that the header information was read in correctly and that the electrodes look like they are in the correct spatial location. Click:

```
-> Plot
  -> Channnel locations
    -> By name
```



You should see a schematic of a head, with labeled dots all over it. The triangle at the top represents a nose so you know the facing direction. The labels are the name of electrodes used in the international 10-20 system of EEG scalp electrode placement. Your plot should like something like what is shown below.



We can sense check this schematic representation of the spatial locations of our electrodes based on what we know about the 10-20 system. All electrodes with an odd number should be on the left of the head, those with an even number on the right and those ending in a `z` down the mid-line. Those electrodes starting with certain letters also indicate that they are over a specific brain region:

- `F` = Frontal
- `C` = Central
- `P` = Parietal
- `O` = Occipital

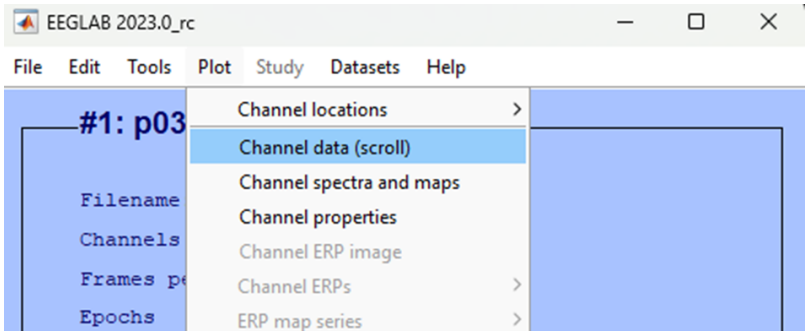
Close the plots that are open.

Plot continous data

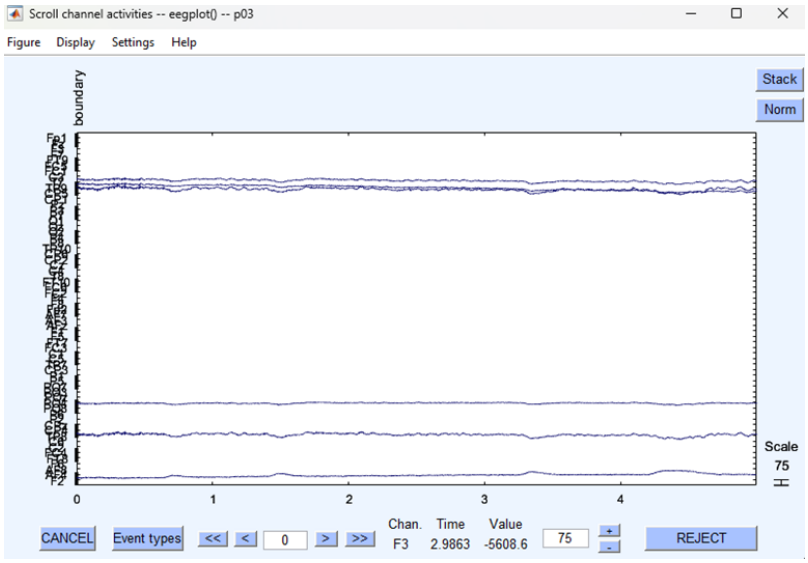
We can now visually examine the actual EEG data, although at this stage it is unlikely to be particularly useful given that we have done very little processing. We are going to ask

EEGLAB to plot the 'continuous' data - that means all of the EEG data in one long chunk in one graph. To do that click:

```
-> plot
  -> Channel data (scroll)
```



You should now see a plot akin to the image below.



In the first instance - this image isn't particularly useful. the actual EEG data isn't really visible that's due to an ongoing direct current present in the data - the EEG amplitude is 'drifting' off the visible area. We will correct this later in the processing steps. However, take some time to play with the settings and see what you notice about how to navigate in this view - it will be one we return to.

Test yourself

Question 1 | When plotting continuos data what do the labels plotted on the y-axis represent?

_____

Question 2 | What is the name of the international system for electrode placement?

- (A) 10-10
- (B) 20-10
- (C) 10-20
- (D) 5-5

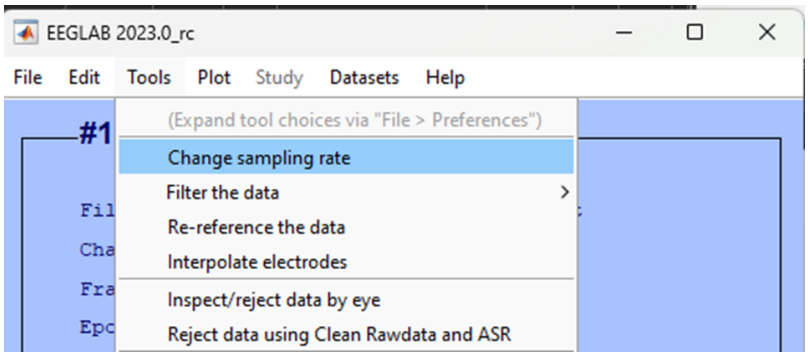
Down Sample

As explained, the sample rate, in Hz, is the number of times a second the analogue EEG signal is stored on a digital computer in a second. During recording we sampled at 1000 Hz - that means for each second 1000 samples were taken for each electrode. In experimental sessions - the EEG recording can often be at least an hour long. So, for 64 electrodes for an hour long we have taken 1000 digital samples a second - all in all that's a lot of data! For us to be able to manipulate this data for this tutorial we want to be able to do so quickly so we are going to 'down sample' and reduce the amount of data we hold. Instead of 1000 samples a second we are going to take one quarter of those data points and down sample to 250 Hz.

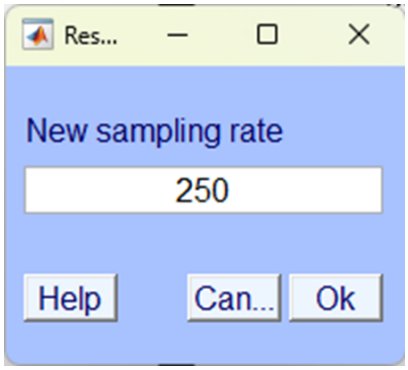
How

To down sample the data for this participant click:

```
-> Tools
  -> Change sampling rate
```



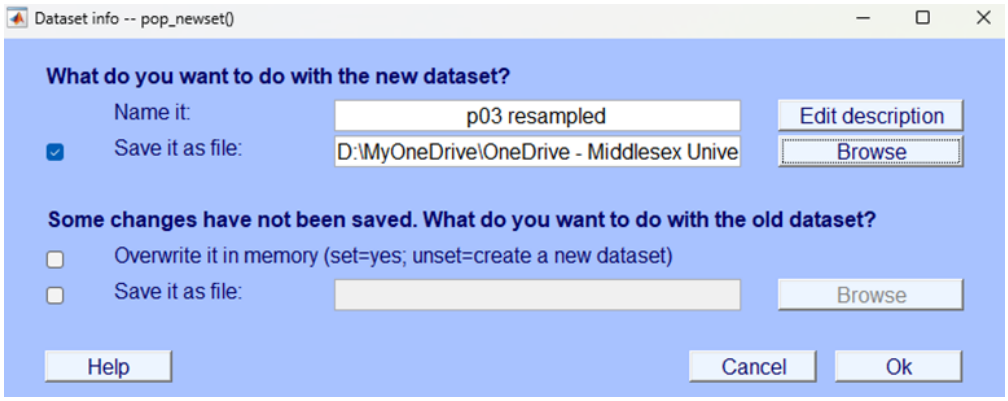
A pop-up will appear, enter `250` as the new sample rate and click `Ok`.



⚠ EEGLAb does **not** automatically save your analysis steps

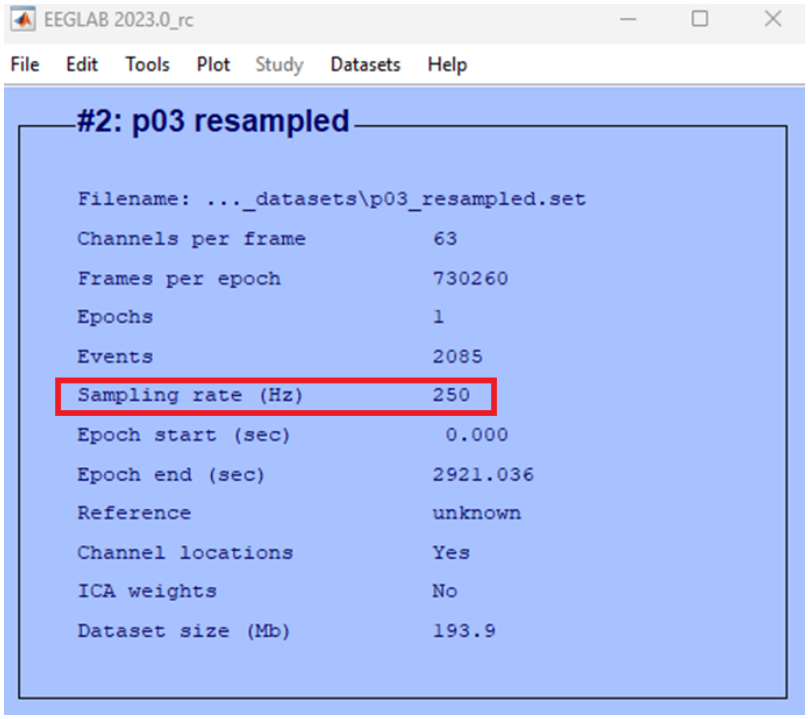
Make sure you select options that will allow you to save your work.
We strongly recommend saving your data after every step of the analysis - this will mean that if you later encounter an issue with your analysis, it will be easy enough for you to pick up from an earlier point and not have to restart all over again. You should use your own labeling system so that you can easily identify each file and the processing that has been done to it.

Next a pop will appear asking you what you want to do with the new dataset - make sure you `Name it` and select the option to `Save it to file` the click `browse` to select an appropriate directory. Use a simple naming convention, this data is from participant 03 and has been resampled, so we've called it `p03 resampled` and saved it on a local directory, like so:

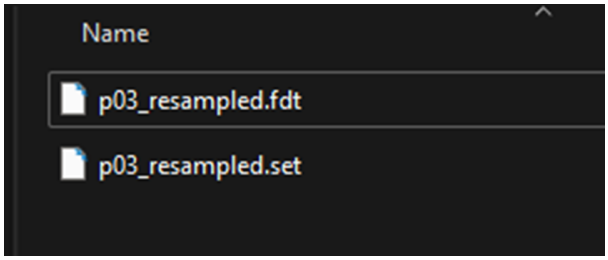


Once you've clicked `Ok` double check everything is as expected.

1. Your EEGLAB home window should state that the `Sample rate (Hz)` is `250`.



2. Use your computers file explorer to check that the file is where you saved it - you should see two files as you named them one ending in `.fdt` and another ending in `.set`. Like this:



EEGLAb file types

The two file types saved by EEGLAB contain different information. `.set` files contain the EEG structure but not the data, while a `.fdt` files contain the EEG data but not the rest of the structure. Data structure and how data is held in memory is actually quite a complicated topic, if you are interested in learning more see information here

Test yourself

Question 1 | Imagine recording from two electrodes with a sample rate of 1000 Hz for 10 seconds. How many bits of data would you have_____

Question 2 | If the same data as in question 1 was downsampled to 500 Hz, how many bits of data would it contain? _____

Question 3 | What's the main reason for down sampling data?

- (A) To remove non-brain artefacts from the data.
- (B) To clean the data from high frequency noise.
- (C) To be able to identify the electrodes.
- (D) To save space and in turn process data more quickly.

Filtering

Raw EEG data contains a lot of 'noise' from many different sources that contaminate the ongoing data. The simplest way to remove a large section of noise is to exclude waveforms that are oscillating at a given frequency that we know cannot be generated by the brain or we know is highly likely to be generated by a source outside the brain. This process is called filtering. We know small currents generated by sweat or electrode movements will cause low frequency changes or 'drift' in the data. We also know the electrical field generated by power outlets and electrical equipment in the room oscillates at approximately 50 Hz (in the UK). We can therefore apply a filter that will remove low frequency changes to remove drift and high frequency oscillations to remove a 'mains-hum'.

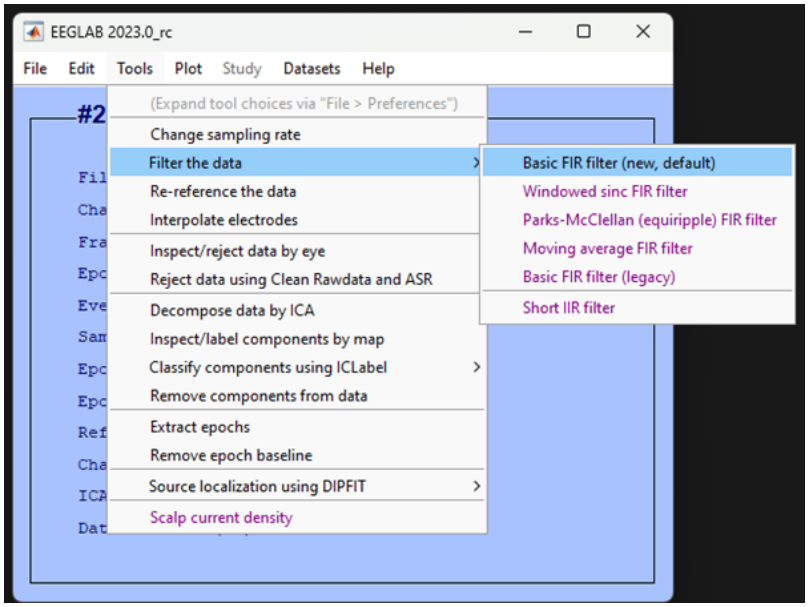
Although conceptually, relatively straightfroward,

> **Terminology**
>
> There are some terms used in filtering that are useful to know:
>
> - Low pass (aka high cut off) = A filter at which all frequencies **below** the specified frequency are **preserved** in the data.
> - High pass (aka low cut off) = A filter at which all frequencies **above** the specified frequency are **preserved** in the data.
> - Band pass (aka pass band) = A filter which **preserves** frequencies within a specific range.
> - Band stop (aka stop band) = A filter which **removes** frequencies within a specific range.
> - Notch filter = A form of band stop filter that usually operates within a more narrow frequency range.

How

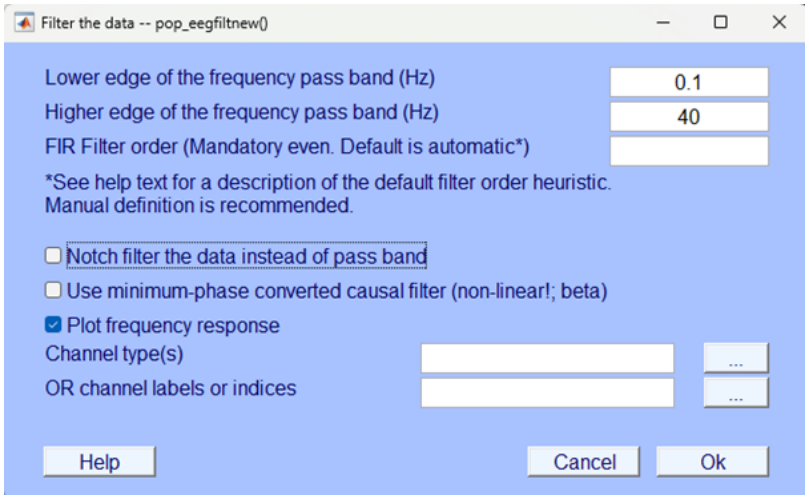
We are now going to filter the dataset that you have already down sampled. To do this in EEGLAB click:

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

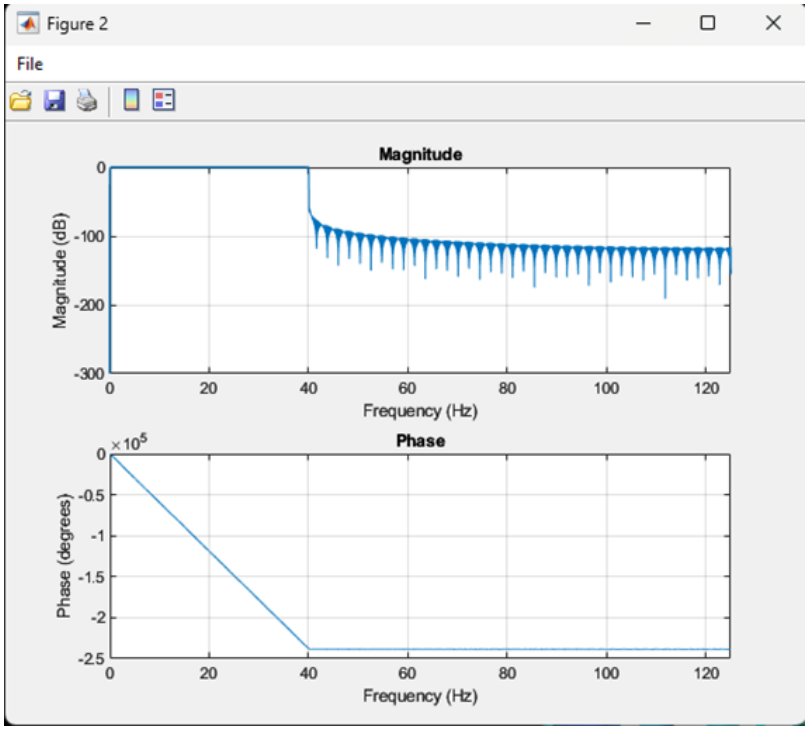


In the window that pops up enter `0.1` for the `Lower edge of the frequency pass band (Hz)` and `40` for the `Higher edge of the frequency pass band (Hz)`.

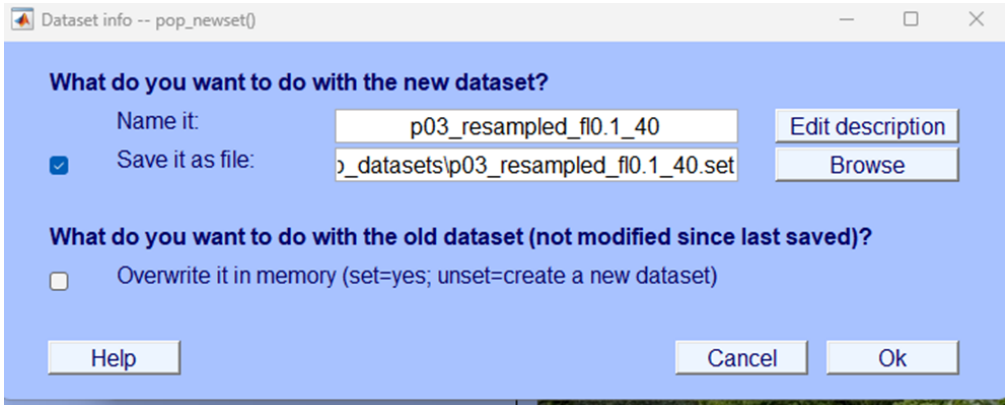
Accept the defaults (i.e., leave blank) other sections and click `Ok`:



Before the filtering is completed you should see a graph pop up that represents the magnitude of the filter as a function of frequency and a second graph showing it's phase. The top graph is a representation of the strength of the effect of the filter for the frequency distribution of the EEG - you should see little effect between 0.1 and 40 Hz (line at zero) and then a sharp increase at 40 Hz. The bottom graph depicts the effect the phase shifting effect of the filter on the EEG signal. In simpler terms, phase describes the relative timing of the waveforms that make up the signal. Effectively, the bottom graph tells you how "shifted" each frequency component is in comparison to a reference. This isn't too important to understand for our practical purposes but it is important to note that filtering does have an effect our signal of interest - albeit a small one. The graphs should look like this:



Your filtering isn't yet finished. Wait till the pop-up giving you the opportunity to save the dataset, give the file a name and save it in a location with the previous files, here we've labeled the file `p03_resampled_fl0.1_40` - this should allow us to easily identify what processing steps have been applied to this data

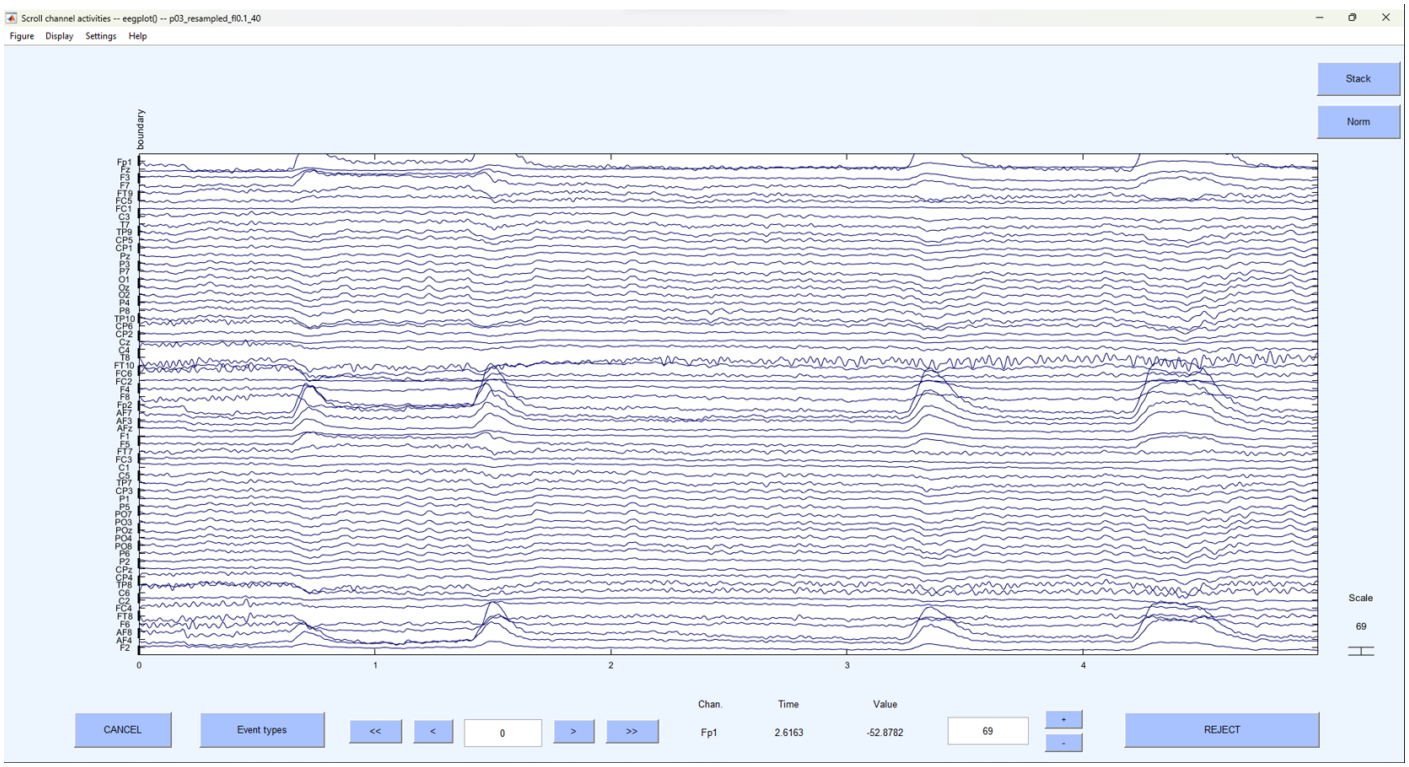


More eye-balling

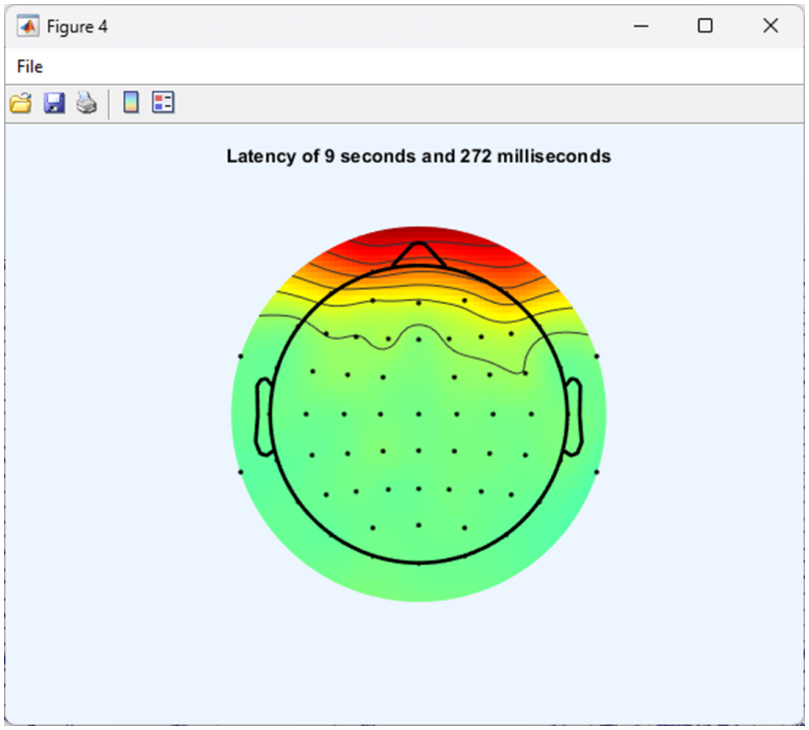
Now, let's visually inspect the data to see what it looks like after filtering. You want to plot the continuous data and scroll through it, if you don't remember how to do that you can check

back to Eye-balling chapter, specifically: **?@sec-ebdata**.

Your newly filtered data should look a little like the image below.

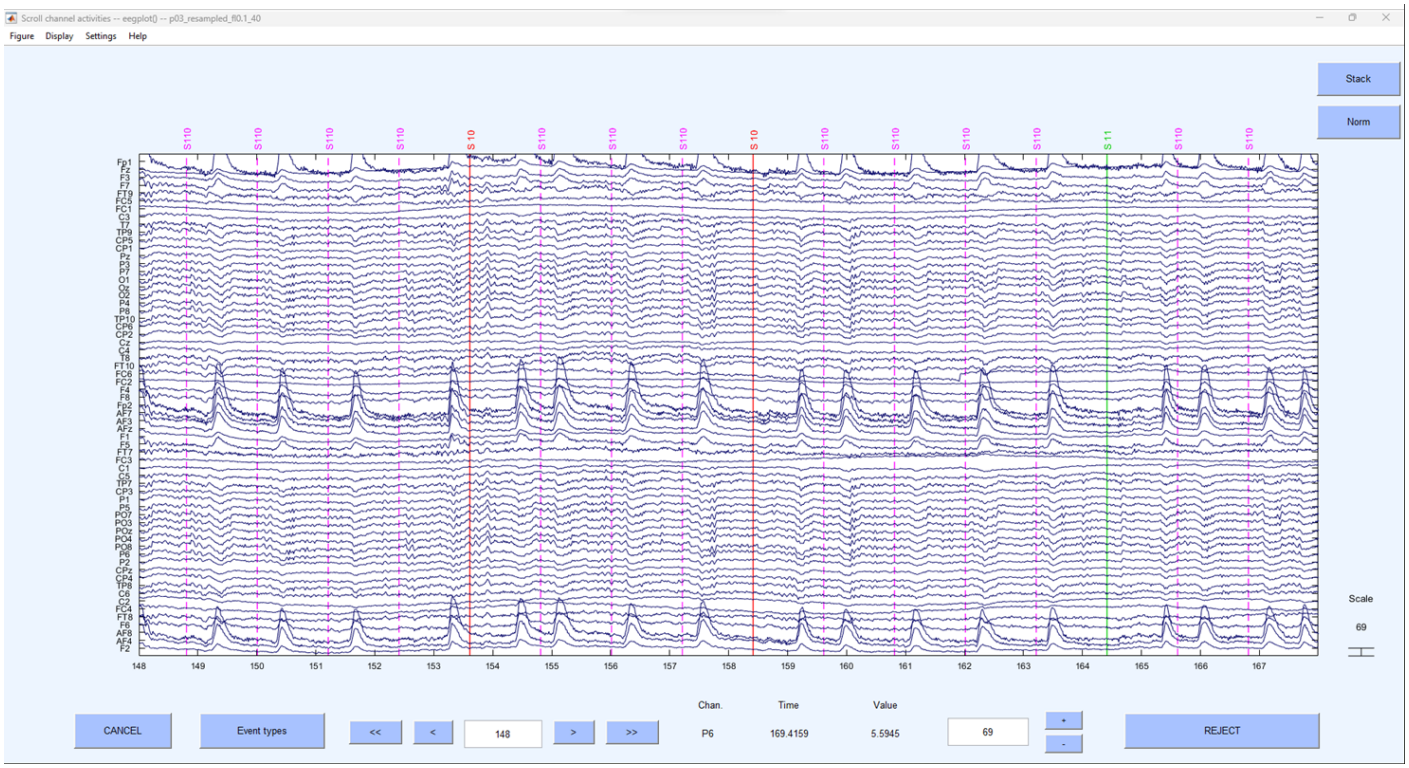


As you did when initially visually examining the data play around with the settings and explore further along the continuous data. You should now be able to identify individual electrodes and the EEG activity associated with them. If you right click on the data EEGLAB will plot the data 'topographically' - meaning the data at a specific time point (wherever you clicked on the plot) will be displayed in colour and mapped onto the scalp surface. A little like this:



Events

If you scroll along far enough in the EEG data you may some some coloured vertical lines labeled with letters and numbers - a little like the figure shown below. These are commonly referred to as 'events' or 'markers'. They donate a specific point in time when something happened during the experimental protocol. So, for example, an event might be when a participant was presented with a stimuli or when the participant made a response on the keyboard. Given that our data is from the Jones & Ward (2019) paper - events might be when a participants was shown object in a rhythmic condition, or it could represent stimuli presented arhythmically.



Test yourself

Question 1 | What effect will a 10 Hz high pass filter have on the data?

- (A) Allow frequencies below 10 Hz in the data.
- (B) Allow frequencies at 10 Hz only in the data.
- (C) Allow frequencies above 10 Hz in the data.
- (D) Remove frequencies at 10 Hz only in the data.

Question 2 | What is the name of the plot that graphically represents EEG activity for a specific point in time by mapping it on the scalp?_____

Interpolation

Now that the data has been down sampled and filtered it is clean enough to visually inspect. This takes some training but isn't hard to learn. In the first instance we are looking for electrodes that display unusual variance. Sometimes this can be obvious, sometimes a little more subtle. To do this we want to carefully skim through the data by moving through time and looking at all the electrodes. We can play with the settings after we've plotted the continuous data to change the scale on the x and y axes to better be able to view the data quickly.

Spotting these 'noisy' or 'bad' electrodes takes practice and you'll improve with time. To be identified as 'bad' electrodes need to meet a few criteria:

1. There voltage is visibly different and, usually, greater than their neighbours.
2. This deviation shouldn't be temporary but effect large and ongoing sections of the data.
3. The electrode should be visibly noisy where 'events' have been marked.

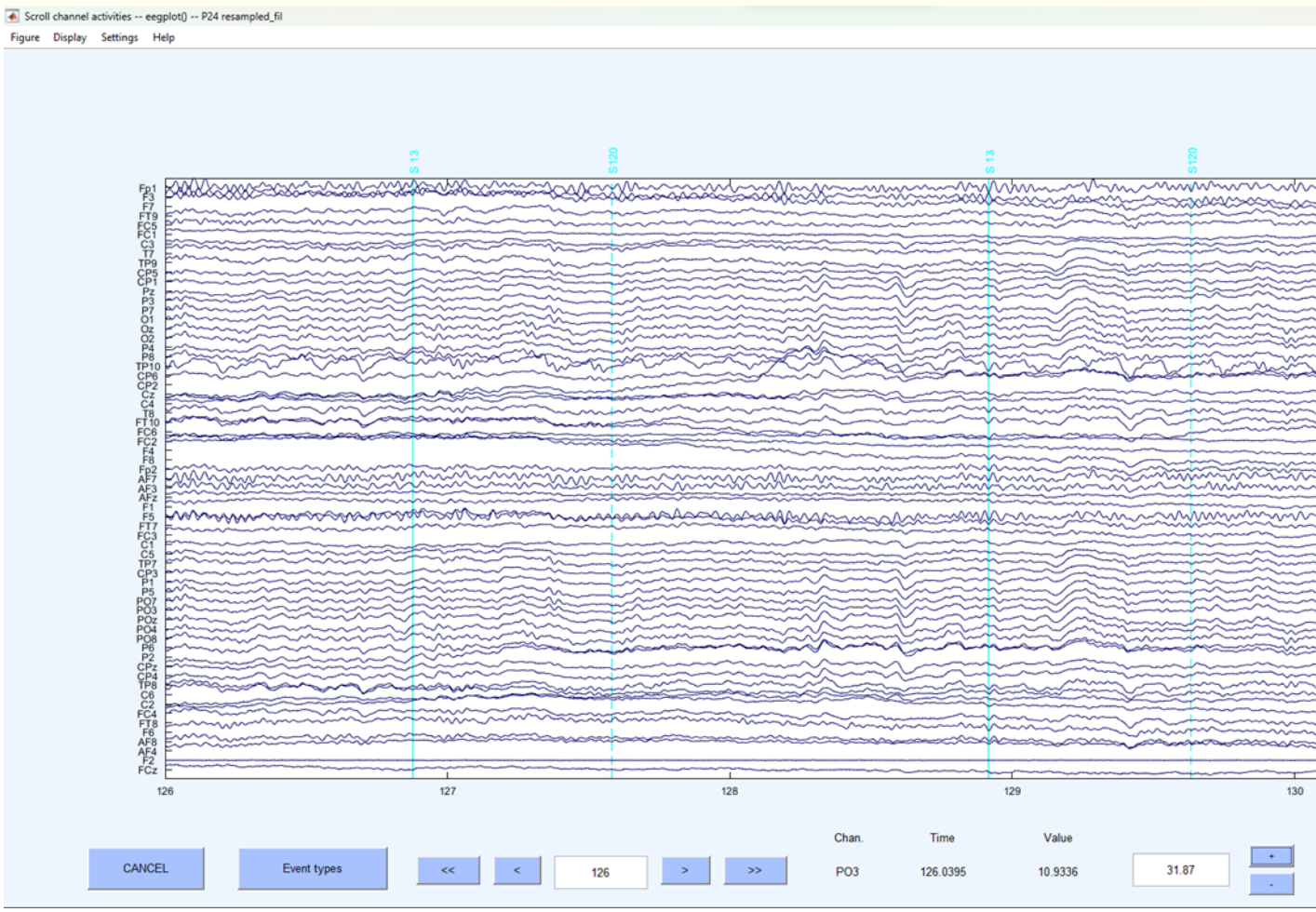
Once identified we want to remove these electrodes from any further analysis, to do so we don't want to delete them completely but interpolate the data at that electrode on the basis of data from other surrounding electrodes and its location on the scalp. This allows us to continue to use the electrode in later calculations, but you cannot do this to an electrode whose data you wish to statistically analyse. If, when looking at the data, you aren't sure about which electrode is bad - speak to an instructor in class. In this particular dataset - most of the files don't have any identifiable 'bad' electrodes, so don't worry if you can't spot any.

How

Identification of bad electrodes

There aren't many bad electrodes in the data from the participants that we've given you, the specific example we are using comes from Participant 24 - you may not yet be able to see any bad electrodes in the specific file you are looking at.

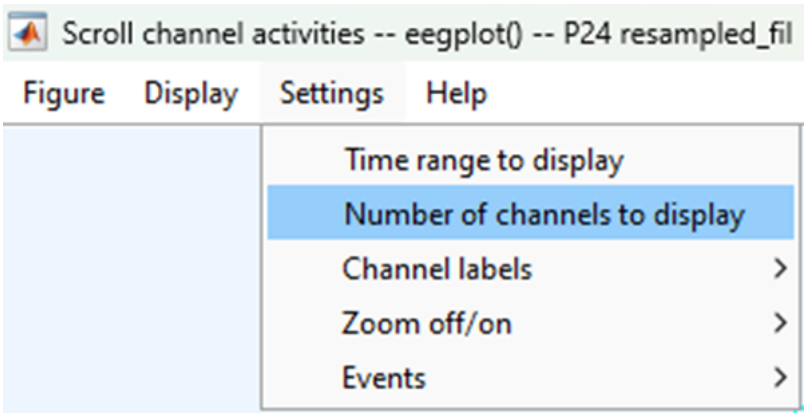
Initial plotting of the continuous data does indeed show unusual fluctuations in voltage, greater than neighbouring electrodes that seems to continue whilst event markers are visible.



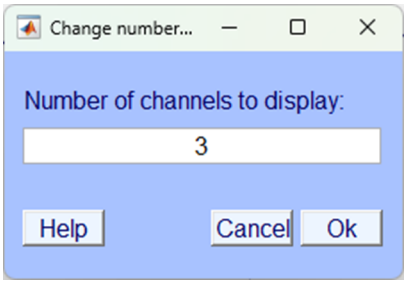
A provisional examination suggests that electrode `TP10` is unusual. Let's examine it a little closer by just displaying that electrode and it's closest neighbours.

On the plot window click:

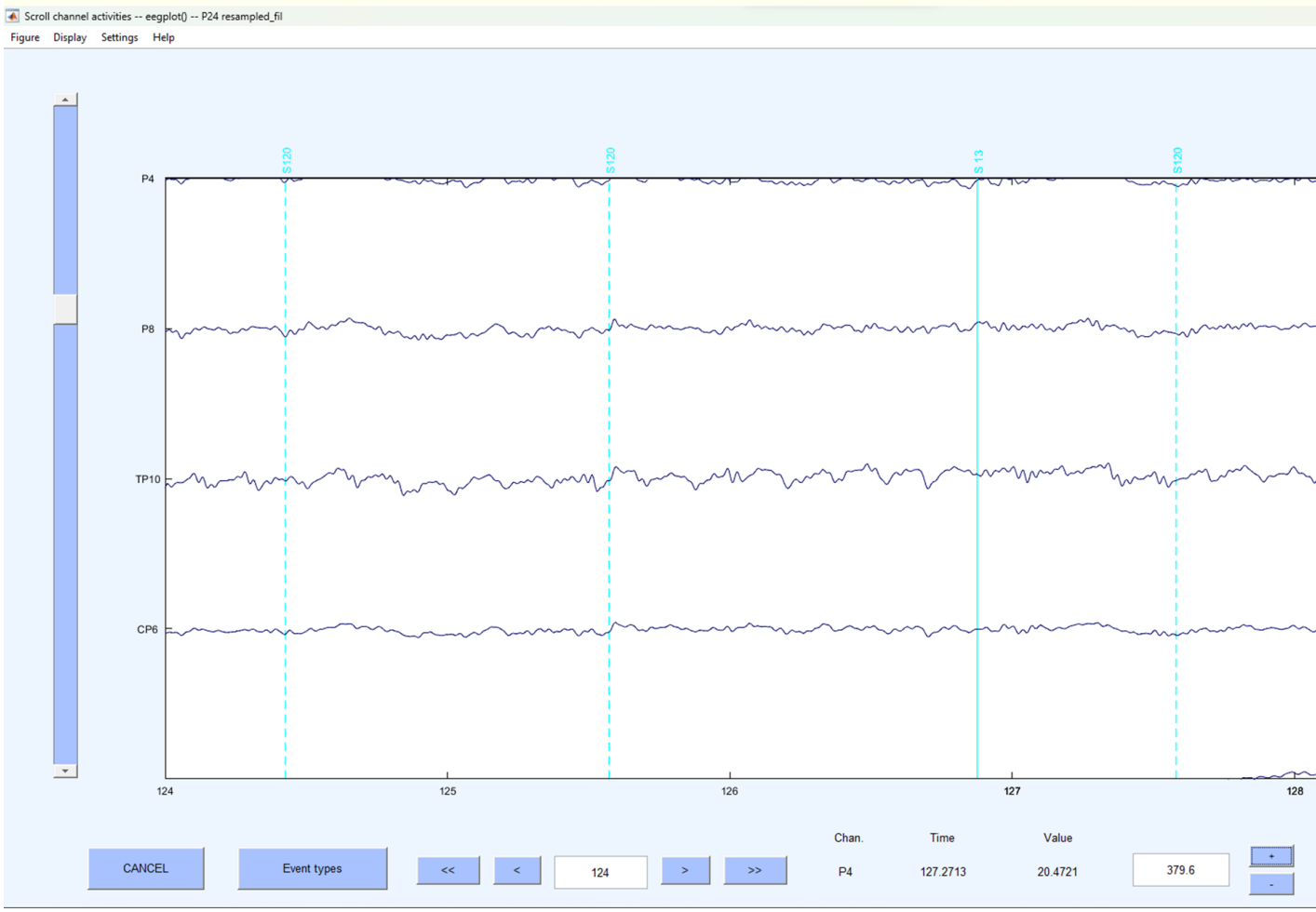
```
-> Settings
  -> Number of channels to display
```



And enter `3` in the pop up window.



Then scroll along the horizontal scroll bar until electrode `TP10`is displayed centrally.

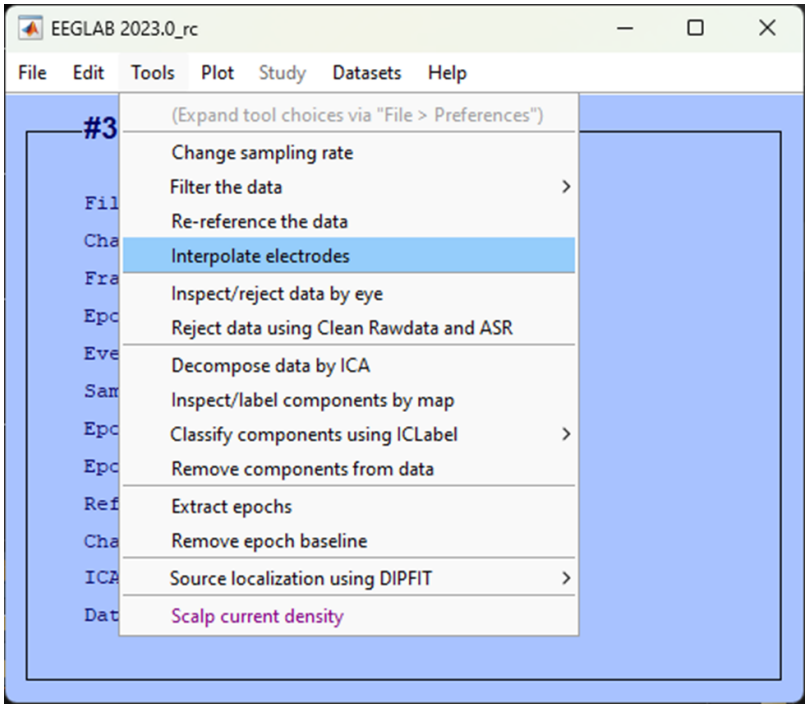


Here - it should now be quite visible, that electrode `TP10` is displaying unusual continuous large fluctuations in voltage compared to other electrodes. Let's interpolate it.

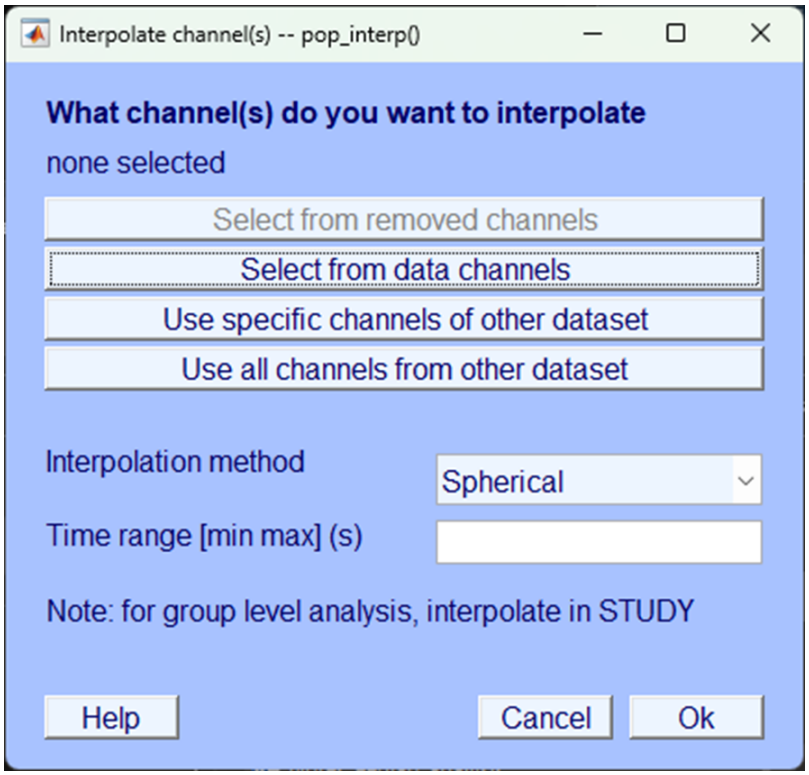
Interpolation

Close the plot window and on the EEGLAB home screen click:

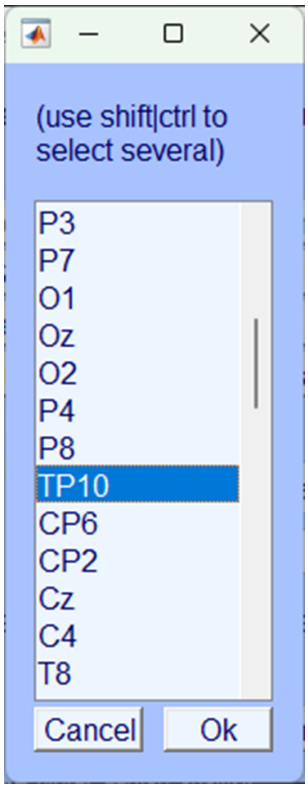
```
-> Tools
  -> Interpolate electrodes
```



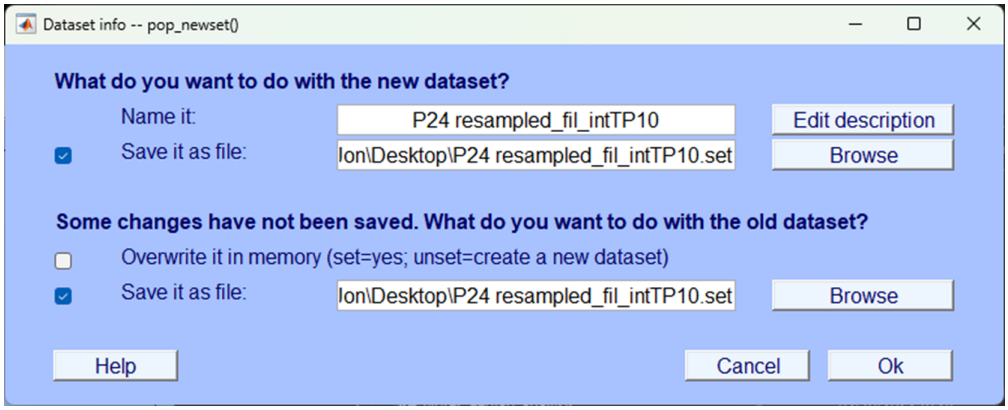
In the resultant pop up window select `Select from data channels`



Then, select the electrode you want to interpolate, in this case `TP10` and click `Ok`.

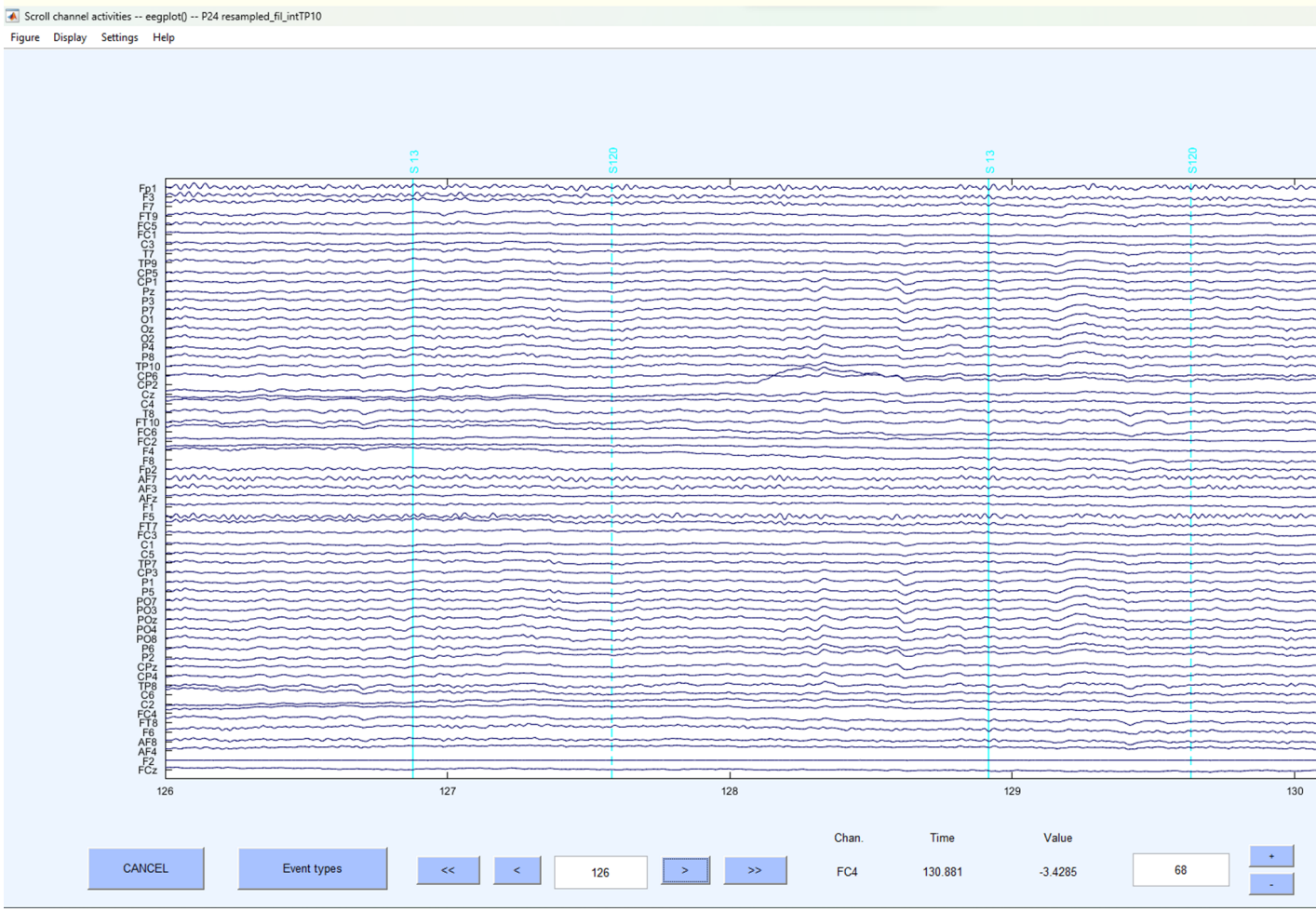


Make sure you save the dataset.

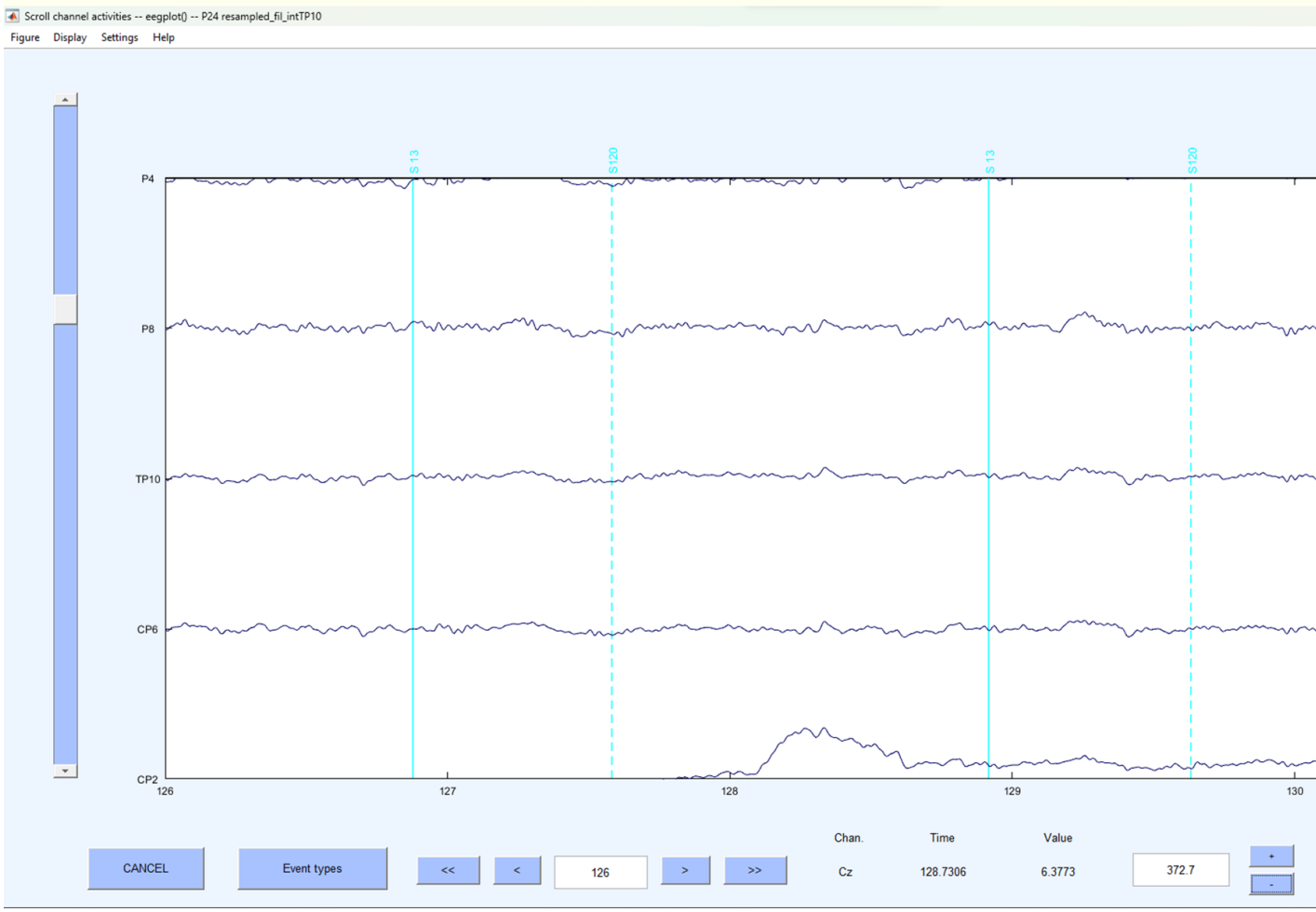


Check the interpolation

Now you've interpolated the channel let's plot the data as we did before and look at the same time to see what the electrode looks like.



As you can see `TP10` now looks far more like the other electrodes around it. Let's take a closer look:



Clearly, the artefactual noise that was previously present in the electrode is now no longer visible.

Test yourself

Question 1 | Should you interpolate an electrode with large artefactual fluctuations only present when no triggers are in the data?

- (A) Yes
- (B) No

Question 2 | Should you interpolate an electrode that shows no voltage fluctuations at all?

- (A) Yes
- (B) No

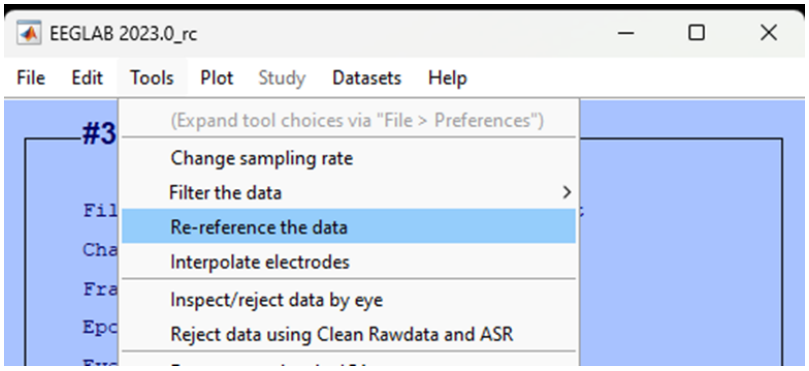
Re-reference

Voltage is the measure of potential difference between two measurement locations. Therefore, any voltage fluctuation we see at any given electrode is relative to measurements taken from another location. There is a lengthy debate in the literature about the appropriate reference to use for EEG analysis and it will depend greatly on the type of experiment you are running and previous research. However, increasingly, with high numbers of electrodes placed all over the scalp many researchers are choosing to use an 'average reference' – that is, the average voltage measurement taken across all electrodes is used as the reference point, rather than measurements taken from any one electrode.

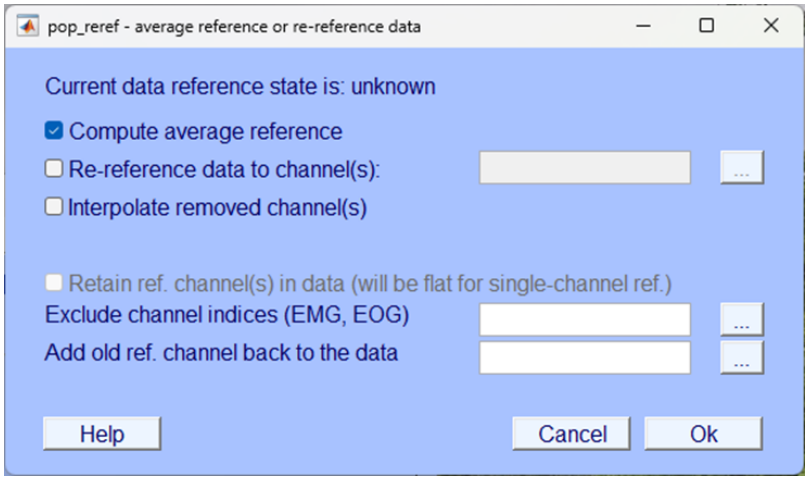
How

To re-reference the data to the average reference in EEGLAB click:

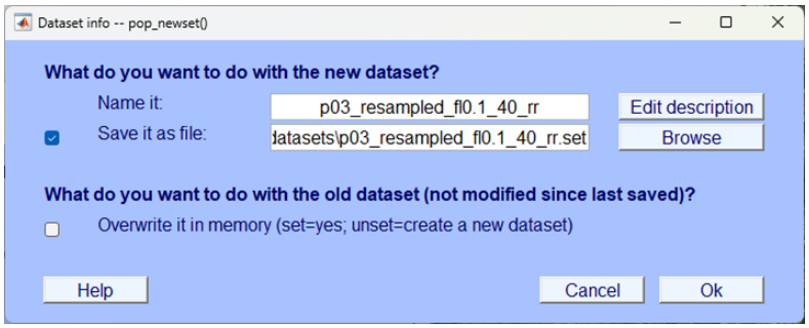
```
-> Tools
  -> Re-reference the data
```



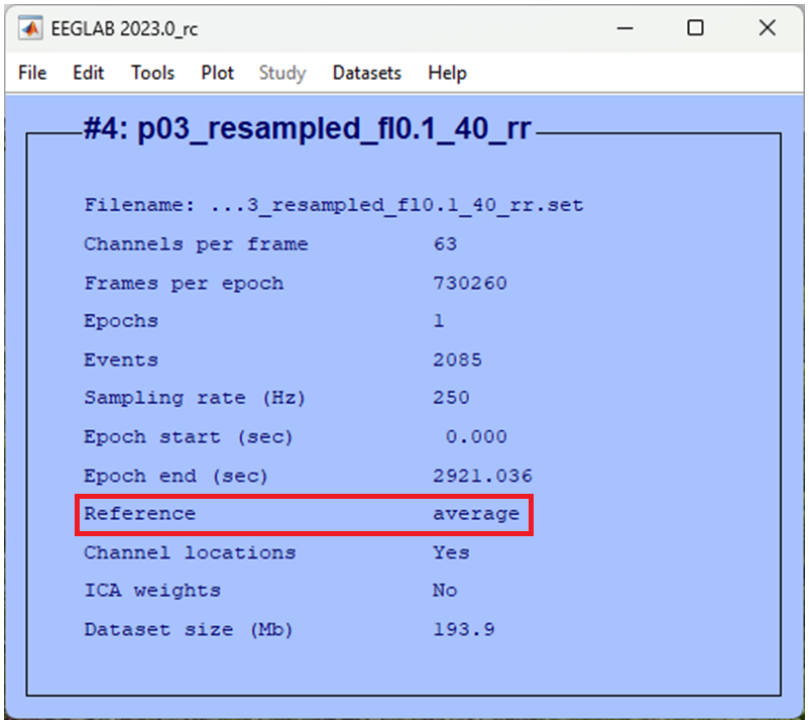
In the pop up that appears, click `Compute average reference` and select `Ok`.



You will then be prompted to name the dataset, make sure you also select the option to `Save it as a file` and append the name with an appropriate suffix, we used `_rr` to denote re-reference.



Your modification to the dataset should now be visible in the EEGLAB home window:



Test yourself

Question 1 | In EEG analysis, what does ''voltage" measure? _____

Question 2 | Why is re-referencing important in EEG analysis?

- (A) To enhance signal quality
- (B) To calculate power spectrum
- (C) To measure brainwave frequencies
- (D) To normalize voltage fluctuations

ICA

There are multiple sources of non-brain data in the EEG recording and much of pre-processing is aimed at limiting the contribution of these artefacts to the data that is used in any statistical analysis. Whilst filtering is a useful approach for removing any data the falls outside of the frequency spectrum typically associated with brain activity, certain artefactual contributions do sit within the same frequency bandwidth as brain data.

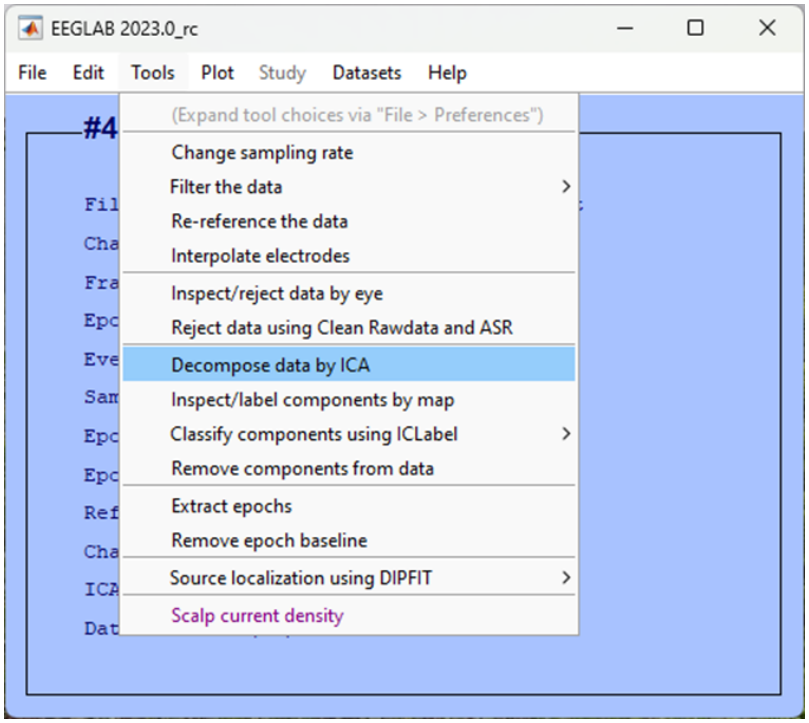
One common source of such artefacts in EEG is the result of eye movements. Eye movements during EEG recordings cause electro-muscular potentials that are not generated by the brain and swamp the EEG recording. Traditionally, when an eye movement was identified (such as a blink) this section of the data was removed and excluded from further analysis. More modern techniques use some form of regression to model-out the artefact caused by ocular movements and effectively clean the data. We will use one such approach in EEGLAB. Specifically, we will use EEGLAB to separate the ongoing EEG activity into 'components' using an approach known as Independent Component Analysis (ICA). ICA is a method that separates mixed signals into distinct, uncorrelated sources, aiming to reveal the original underlying contributing factors. One can apply ICA to EEG signals and, because eye-movements are so distinct from brain activity in the data they generate, they are typically identifiable in one or two components. Once identified, those components can be removed from the data. Hopefully, these conceptual explanations will become more clear as you engage practically with the data.

How

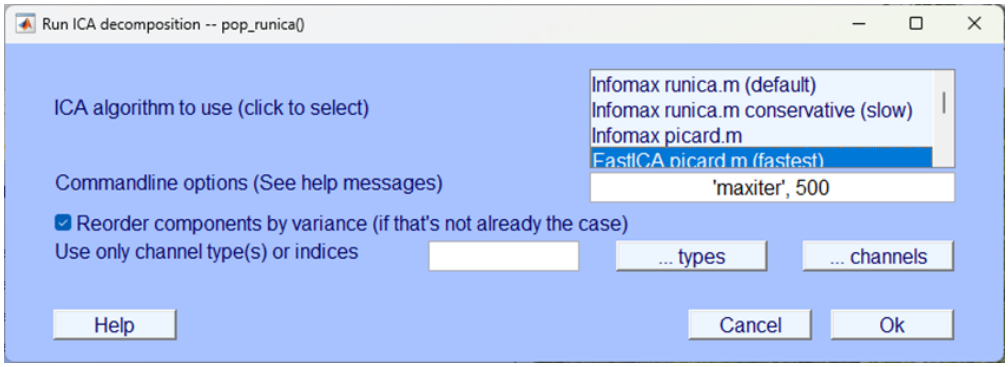
Decompose data into indepentent components

Like interpolation, conducting successful ICA and subsequent removal of components from the data will take some practice. In the first instance do the following in EEGLAB:

```
-> Tools
  -> Decompose data by ICA
```

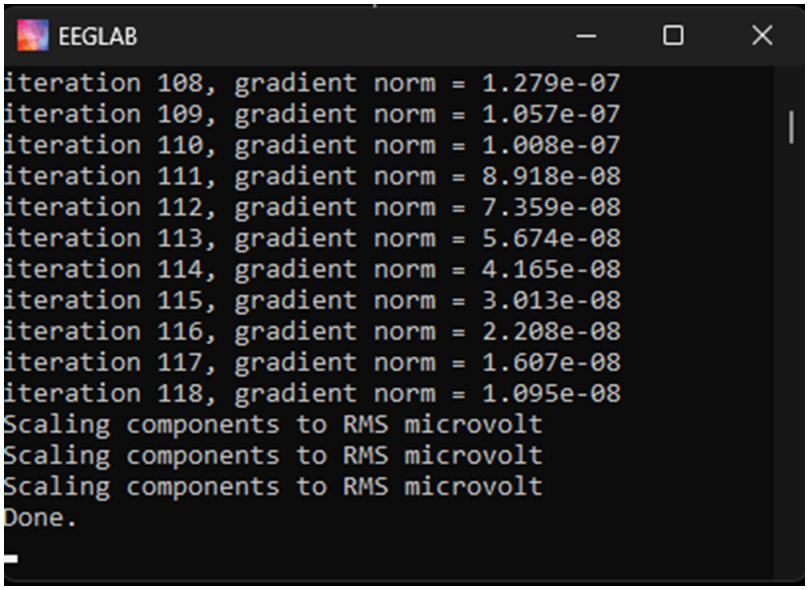


For `ICA algorithm to use (click to select)` make sure you choose `FastICA picard.m (fastest)` and then click `Ok`

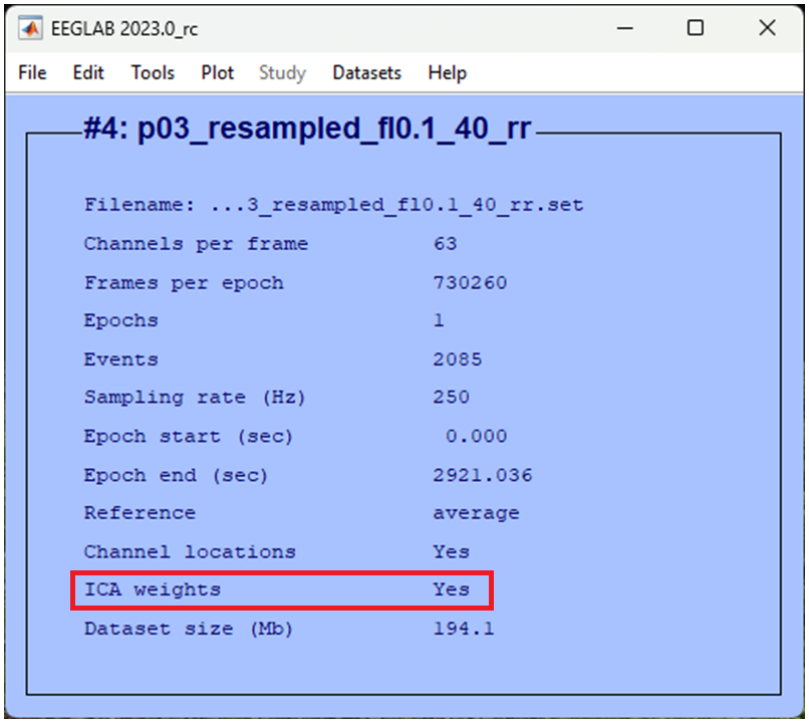


ICA processing time

ICA, even the "fastest" option can take a while to compute - be patient. You can check the progress of the any EEGLAB process by looking at the EEGLAB background terminal, when you see the word `Done.` you know the process is over.



Once completed you should be able to view that your data now has `ICA weights`:

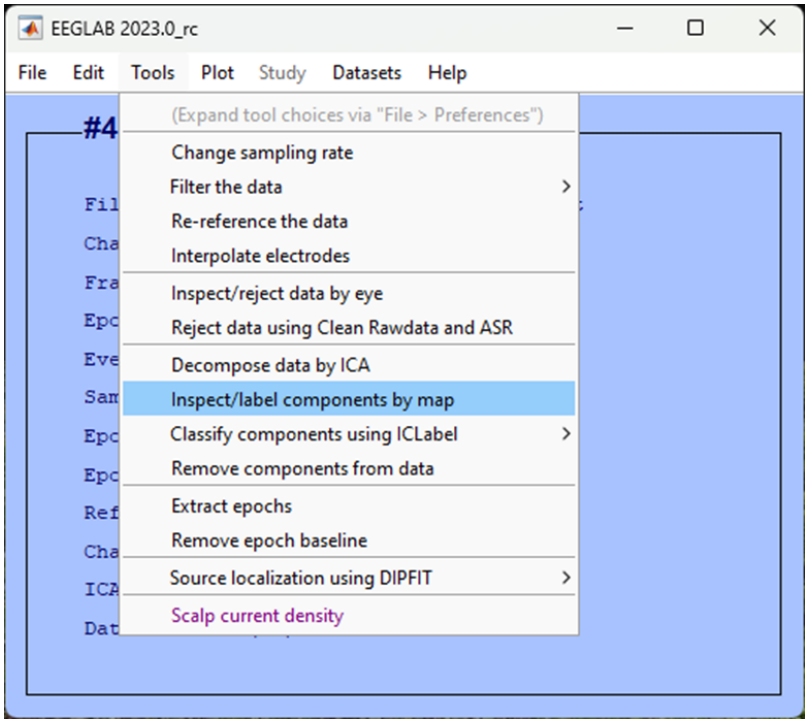


Inspect ICA components

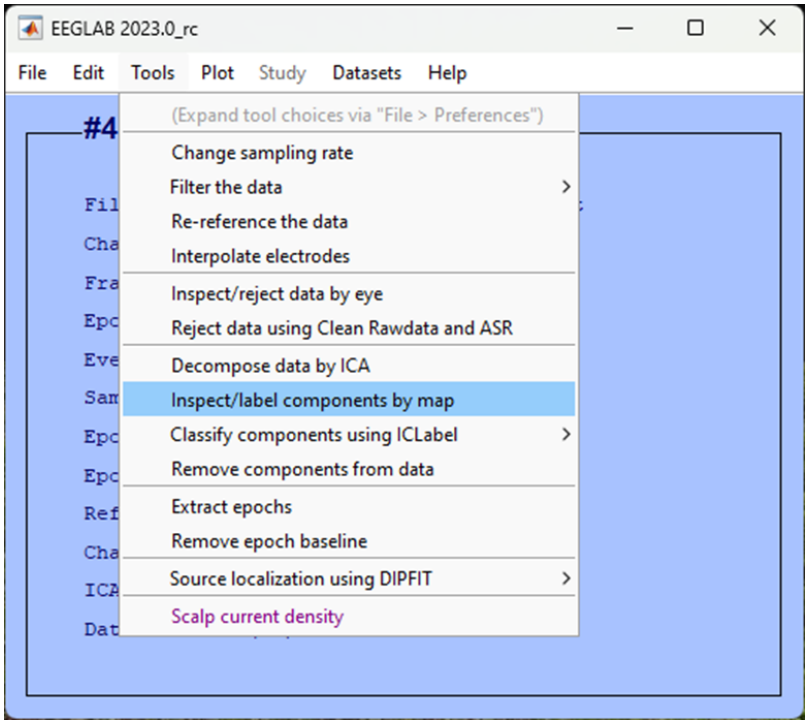
Now that the data has been decomposed into independent components, we need to identify which component is most likely contributing to ocular artefacts so that we can remove it from the data. The most easily identifiable and hence clearest component to remove from the dataset is usually the one that best corresponds to a blink. The topography, temporal distribution and activity power spectrum of a blink are relatively specific and can usually be identified via visual inspection. This will take practice and experience but we'll talk you through it and you can always ask for help in class.

First click:

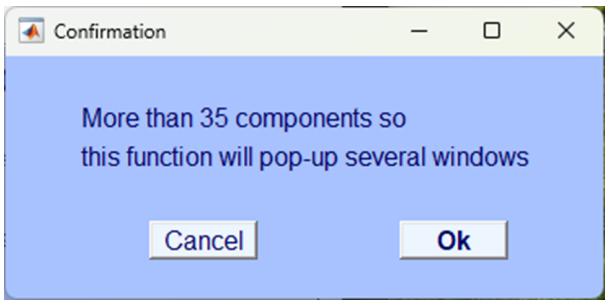
```
-> Tools
  -> Inspect/label components by map
```



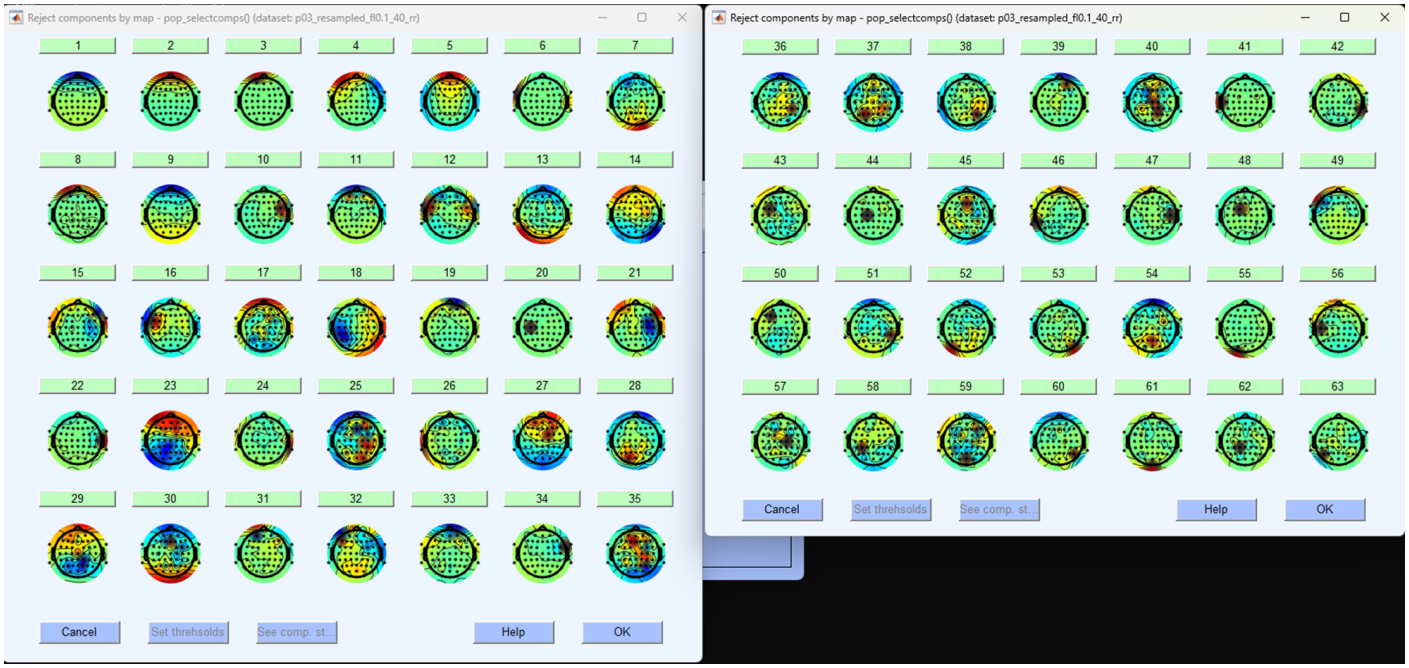
Accept the defaults for the next pop-up window to plot all components, click `Ok`.



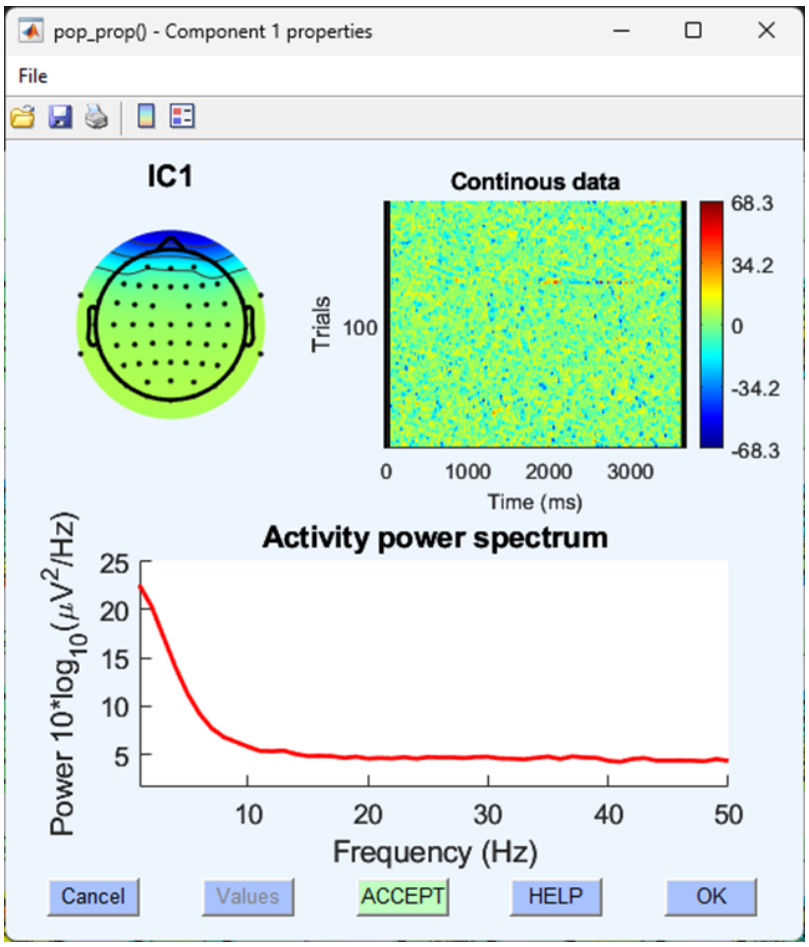
Again, it is likely that this will result in more than 35 components and so EEGLAB asks you to confirm that you are happy with several pop-up windows, click `Ok`.



It will take a minute or so but the resultant pop up windows should plot all components topographically, like so:



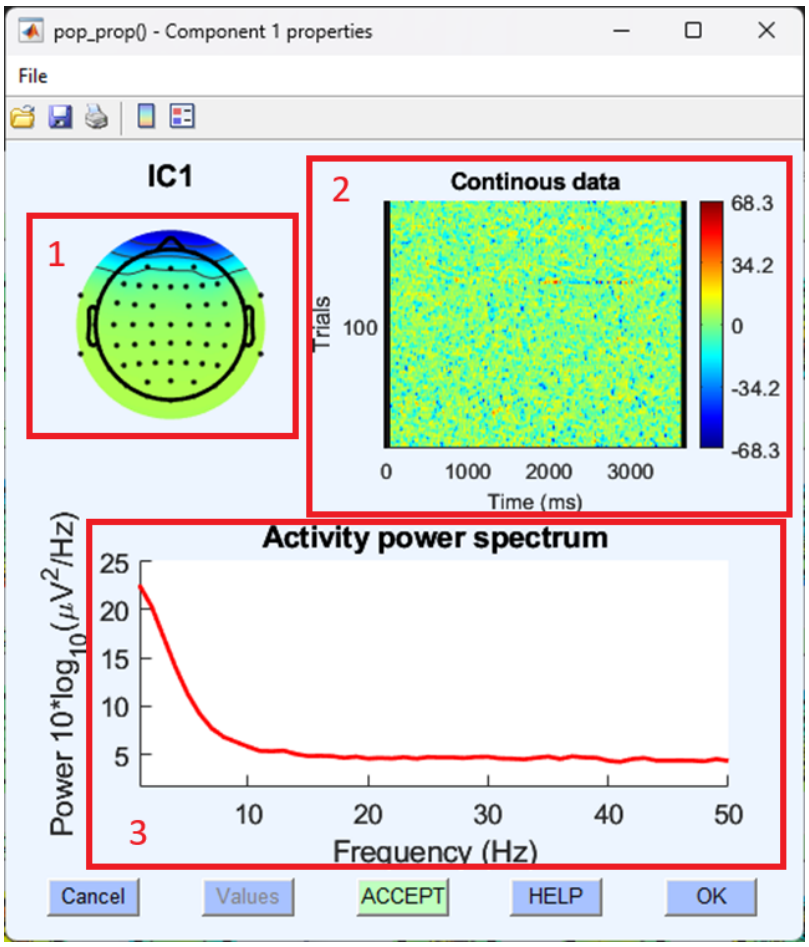
These are spatial distributions of the independent components extracted from the continuous EEG data. They are in order so that the first component explains the most amount of variance in the data. Blinks often contribute a good amount to the ongoing EEG data and are easily separable from other components, as such they are often found in the first five components (but not always). We will have to identify blinks based on a few different aspects. Each component can be examined further by clicking on the number. Your data will look different to mine, but in this example I've clicked on the first component to explore it further, this results in a figure like the one shown below.



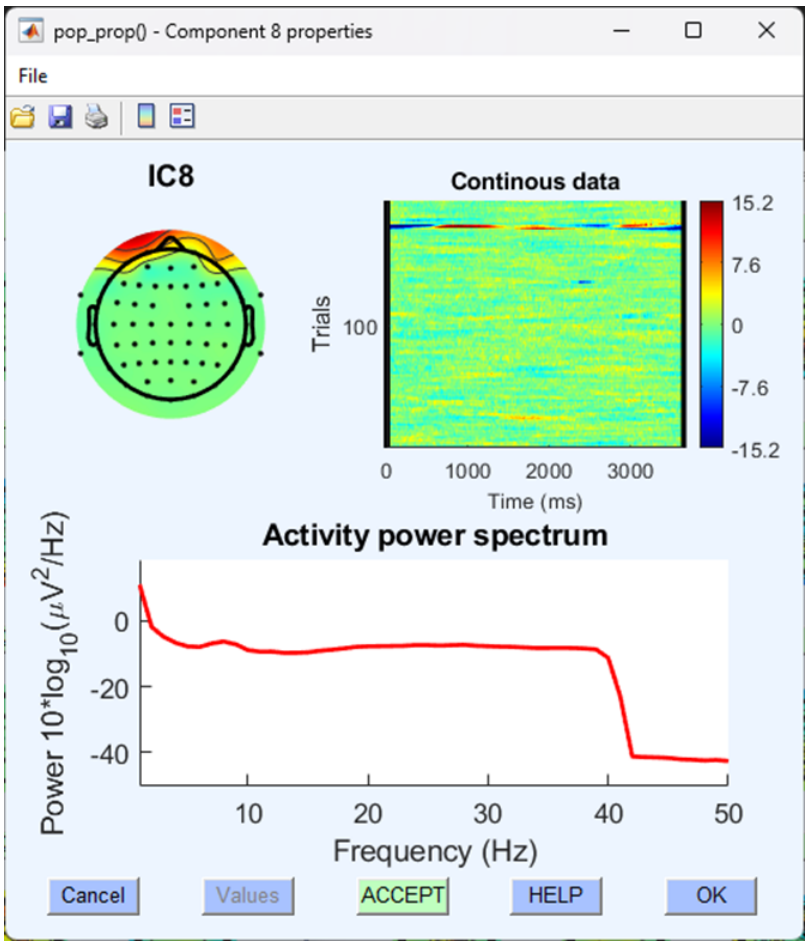
The information here leads us to think that this component is a typical blink for the following reasons. A typical blink has:

1. A spatial distribution of electrical activity focused bilaterally on the very front of the scalp.
2. A sporadic and random distribution of trials throughout the continuous data.
3. An activity power spectrum with little power above the 10Hz range.

numbers correspond to those in the graphical representation below



This might still be difficult to identify, let's look at another component with similar topography to separate it from a blink:



Although this component shares a similar topography to the blink component, it is focused over a specific set of trials and the activity power spectrum includes higher amplitudes all the way up to 40 Hz - this is less likely to be a component indicative of EEG activity caused by a blink.

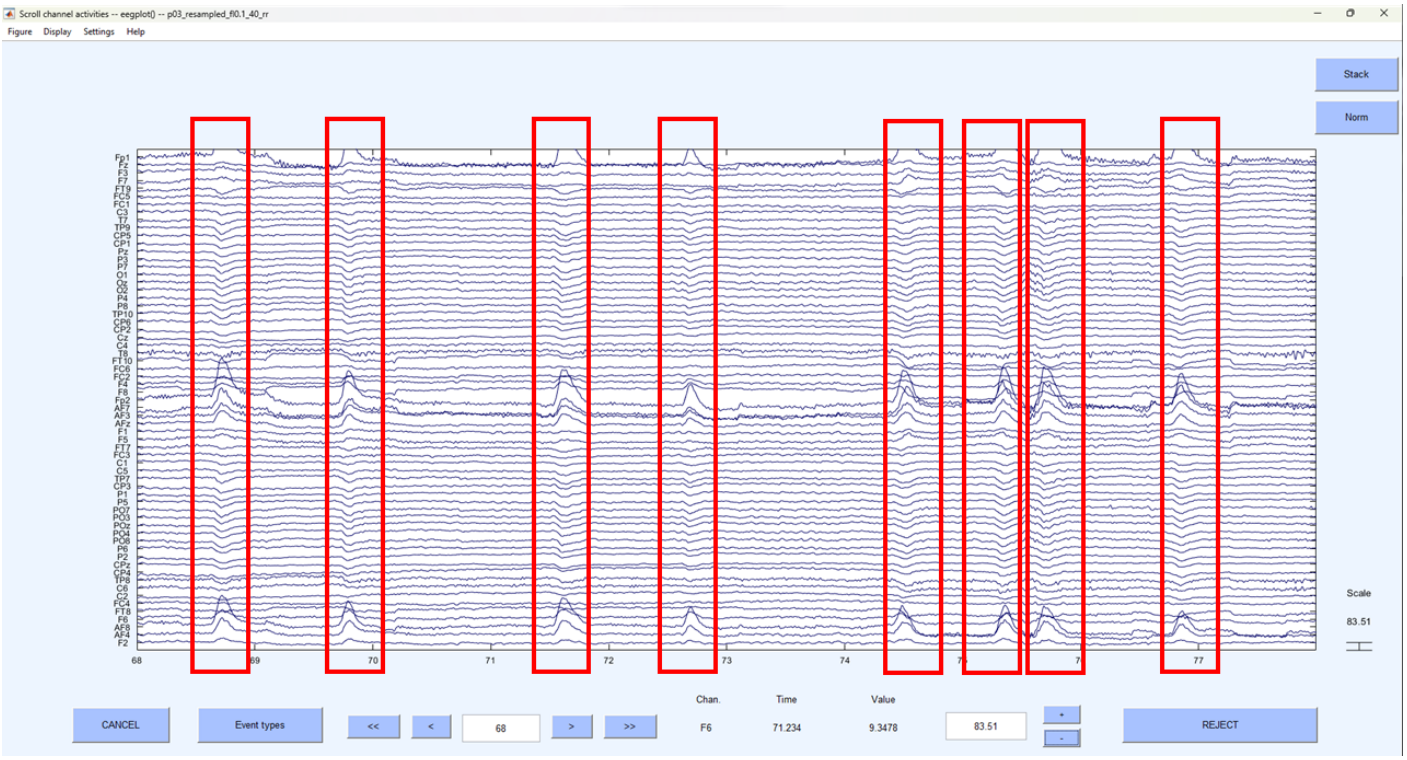
> **Valance**
>
> Whether positive or negative (i.e., blue or red) on the component topographical map isn't relevant to identifying whether or not what you are looking at is a component describing a blink. Often, two 'blink' components can be identified with opposite valences.

Once we have identified which component is most likely to be indicative of a blink, we should note down the number of that component and then move on to the next stage of removing that component from the data.

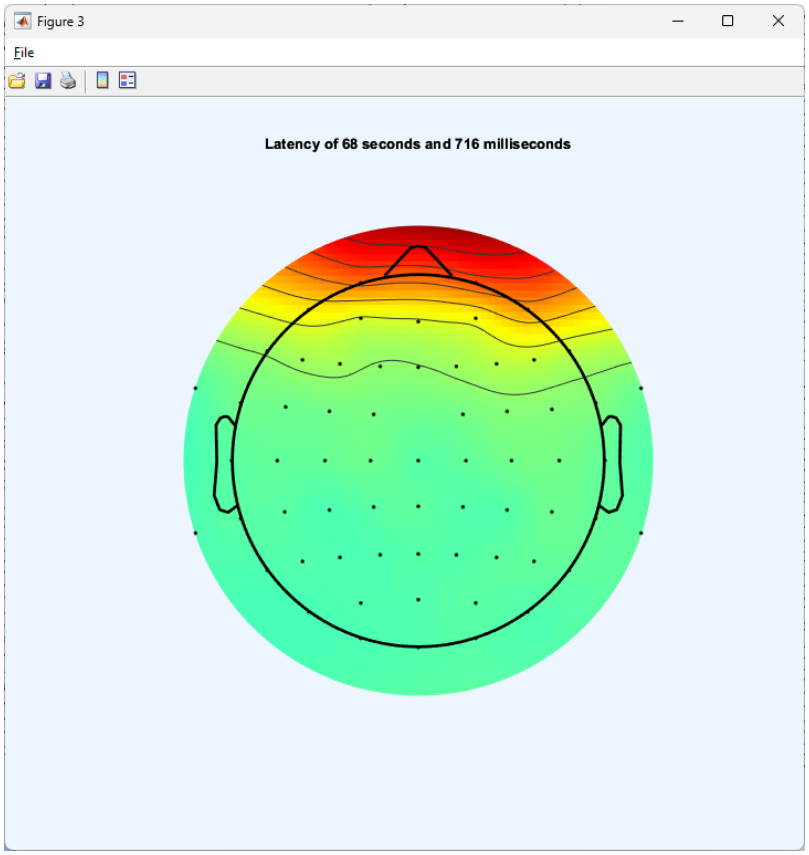
Checking the data before component removal

Now that we have separated the EEG data into independent components and identified which component is most likely to be a blink before we go any further we should visually inspect our continuous EEG data and identify clear blinks. Plot the continuous data(if you've forgotten how see - Eye-balling). Blinks are easily and immediately identifiable in continuous EEG data - there is a large and obvious voltage deflection focused on frontal electrodes that lasts approximately 150 ms.

In the image below, we've highlighted all the identifiable blink in a section of continuous EEG.



If you're not sure if what you are looking at is a blink, right click on the peak of activation and you should see a typical blink topography.

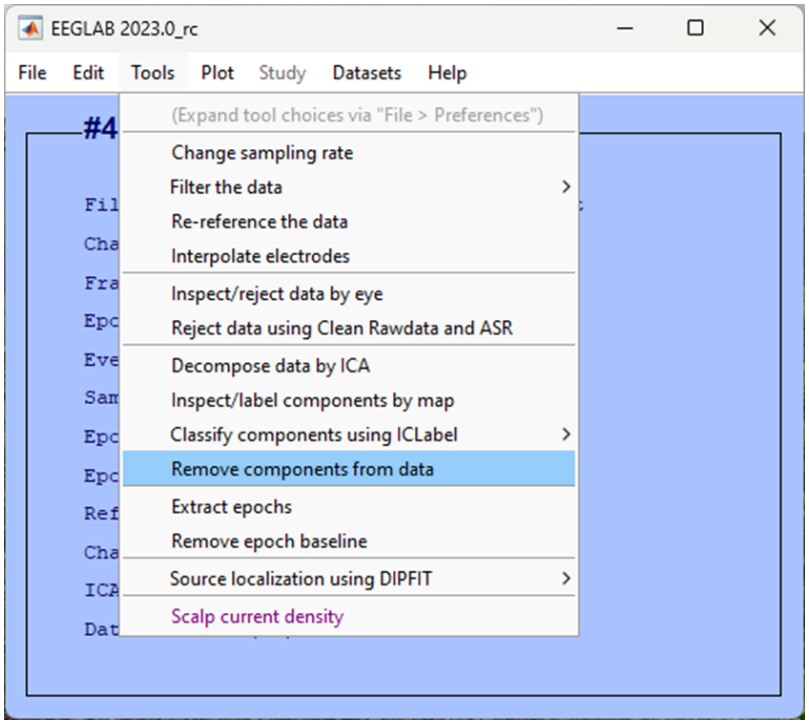


Make a note of where you can see these very clear blinks because we want to go back to this section of the data and compare it to after we've removed the blink component.

Removing the 'blink' component from the data

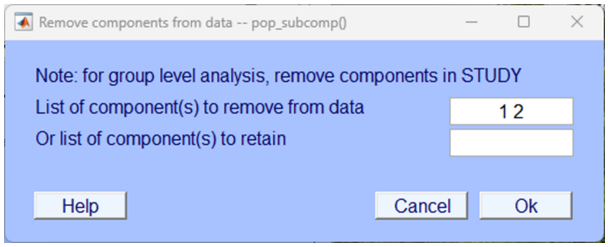
Now we've identified the blink component in the data let's remove it:

```
-> Tools
  -> Remove components from the data
```

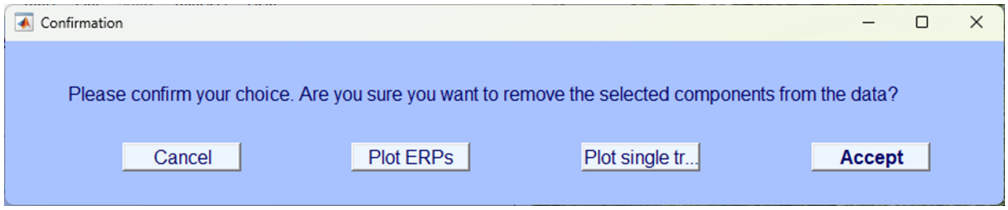


Enter the component(s) that you identified in your data. In our case we found both component 1 and 2 were related to blink activity we add this in the box `List of component(s) to remove from the dta` separated by a space.

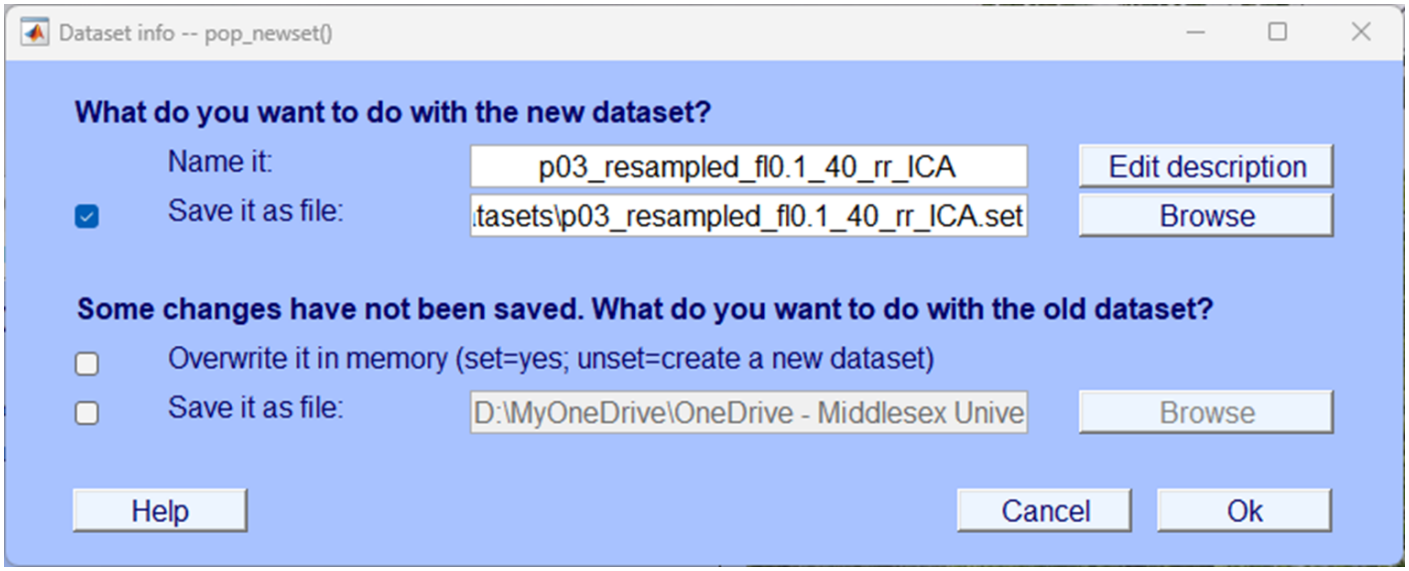
> ⚠ *Your* component(s)
>
> This may be different to the component number we have identified here, if your not sure, seek support in class.



Click `Accept` on the next pop up window.

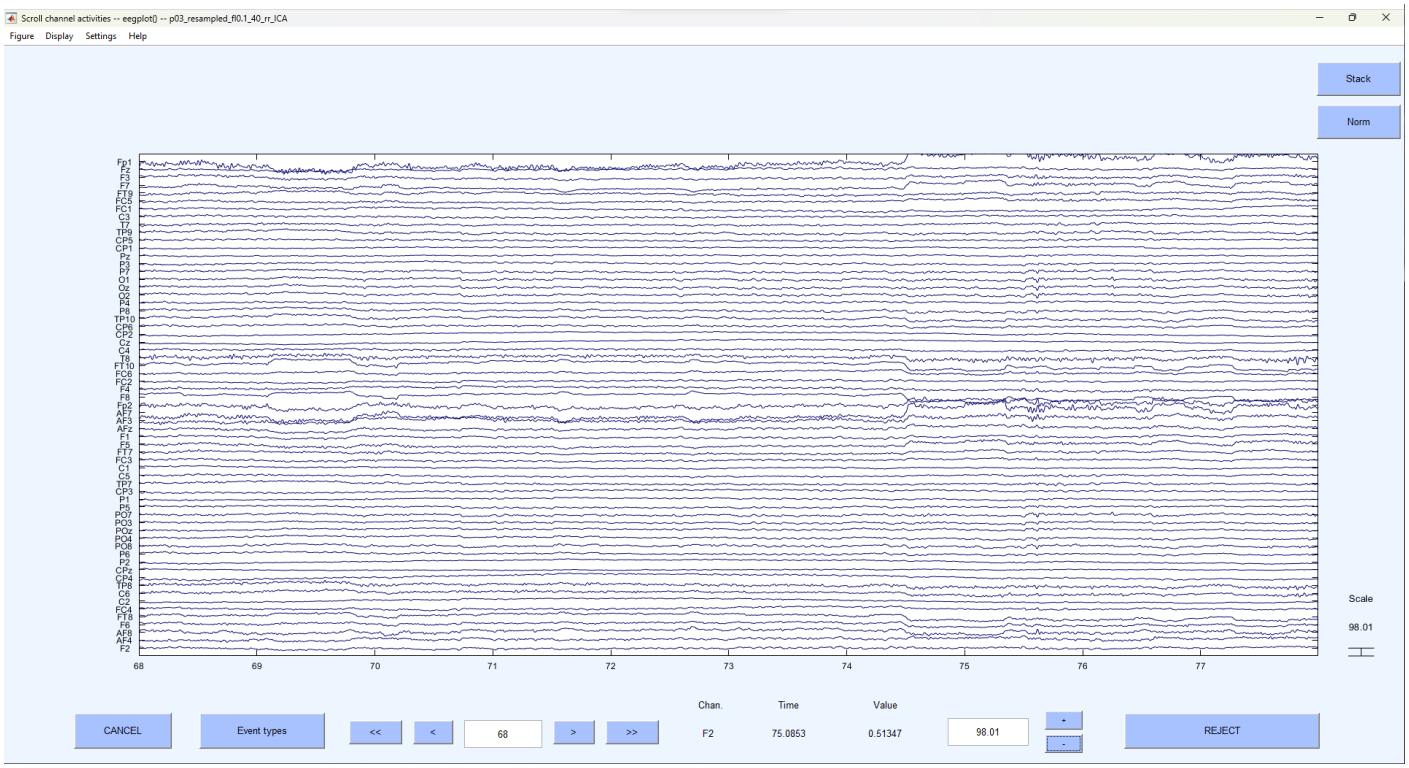


Make sure you click `Save is as file` in the next pop up window and give is an appropriately named suffix, we've used `_ICA`.



Check the blink reduction

Finally, plot your continuous data and go back to the section where you had previously identified blinks. Your dataset should now look very similar but with the blinks removed. If there are still blinks in the data or there appear to be obvious changes to the data that aren't related to the blinks you need to re-consider what components you remove from the data. Below is an image of the same data as that we plotted above but now with the blink components removed:



Test yourself

Question 1 | How is ICA most commonly used in the cleaning of EEG data?

- (A) It enhances high quality signal components
- (B) It separates mixed signals into distinct components
- (C) It amplifies brainwave signals
- (D) It removes high-frequency noise

Question 2 | What characteristic helps to identify blink components during the inspection of ICA components?

- (A) Spatial distribution focused on frontal electrodes

- (B) Consistent distribution of trials
- (C) Spatial distribution focused on occipital electrodes
- (D) Activity power spectrum above 30Hz

Question 3 | What EEG feature is crucial for identifying blink components? _____

Artefact Rejection

We have significantly cleaned the data - we have used filtering to remove frequencies outside of our range of interest, interpolated broken or noisy electrodes and removed components associated with blinks. However, it is still possible that some contamination exists within our data, these could come from a variety of different sources and are often unlikely to be repeated and so identifying them as components is problematic. Rather sections of the data that are contaminated now need to be excluded from any further analysis.

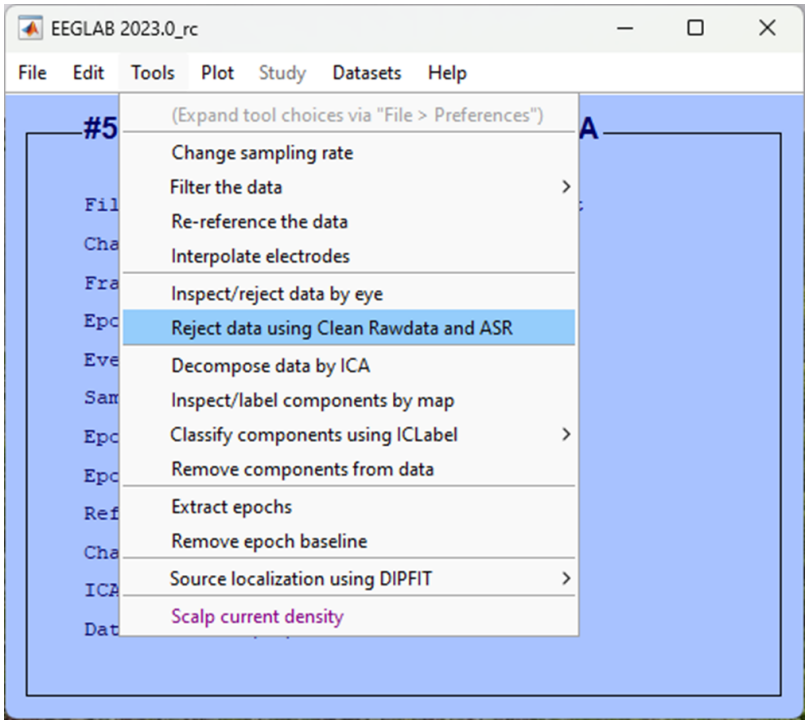
Classically, this done after segmenting the data and marking segments as 'bad' if they contain data that violates certain criteria usually based on relative voltage. We advise sticking to this traditional approach in most instances - however, the standalone EEGLAB compiled version doesn't include the option to do this. Rather, EEGLAB recommends an automated approach to artefact rejection which adopts an automatic approach to identifying artefacts and rejecting data. A full description isn't needed here but the algorithmic approach works by first identifying clean data and then rejects data regions if they exceed 20 times (by default) the standard deviation.

> **Typical artefact rejection**
>
> One of the biggest differences between EEGLABs automatic artefact rejection and more common forms is that EEGLAb rejects section of the continuous data. Typically, you would segment the data first and then conduct artefact rejection.

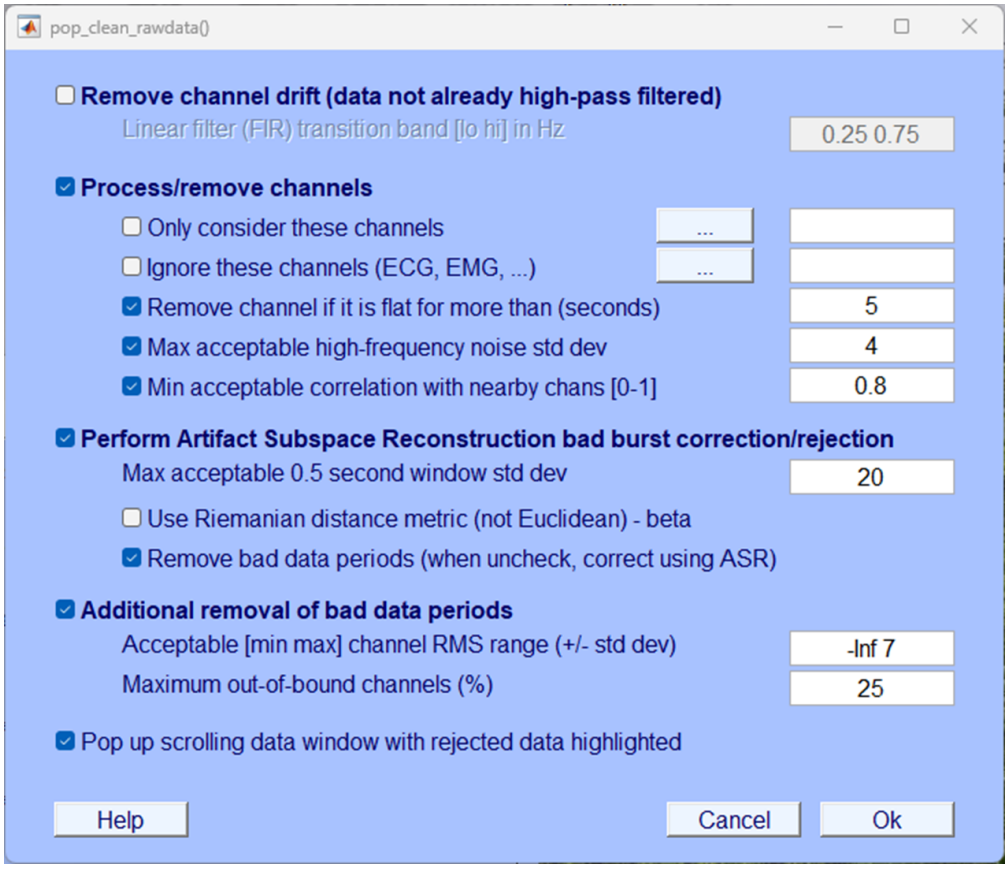
How

For the most part we aren't going to tinker with the recomended settings that EEGLAB provide and accpet their recomendations for rejecting sections of the data:

```
-> Tools
  -> Reject data using Clean Rawdata and ASR
```

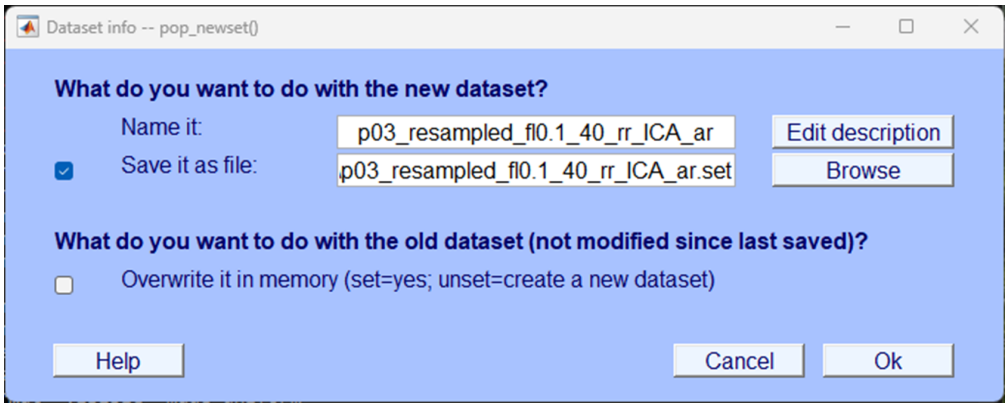


Accept the defaults in the coming pop up window and simply click `Ok`.



Completing the artefact scanning and rejection can take a while so wait for the `Done.` message on the EEGLAB background terminal.

Make sure you select `Save it as file:` and choose a sensible suffic so you can later easily identify the data and what has been done to it, we've used `_ar`.



Test yourself

Question 1 | In the traditional approach of artefact rejection, what are segments marked as 'bad' based on?

- (A) Absolute voltage
- (B) Relative voltage
- (C) Frequency information
- (D) Electrode positions

Question 2 | In EEGLAB's automated artefact rejection, data regions are rejected if they exceed how many times the standard deviation?

- (A) 10 times
- (B) 15 times
- (C) 20 times
- (D) 25 times

Question 3 | What is one significant difference between EEGLAB's automated artefact rejection and traditional artefact rejection approaches?

- (A) EEGLAB rejects segments after segmentation
- (B) EEGLAB uses frequency-based rejection
- (C) EEGLAB requires manual marking of segments
- (D) EEGLAB rejects sections of continuous data

Segmentation

Our data is now sufficiently clean and we now want to divide up the data into sections of interest to us. In this experiment we are looking at brain data for when an individual looked at images to-be-remembered under two conditions - during rhythmic presentation and arrhythmic presentation. Based on the behavioural data (Jones & Ward (2019)) we know that those images that were presented rhythmically to participants were better remembered than those that were presented arrhythmically.Now we want to carve up the brain data to examine what were the differences in the brain when these images were being encoded. Brain data here, and EEG specifically, is particularly useful as during the encoding of images we require no behavioural output from the participant - that means that we can't look at behaviour to make an inference about underlying cognitive processes. However, the EEG data here may provide us with some insight when comparing conditions.

As well as carving up the continuous data around the events of interest, we are also going to '*baseline correct*'. Baseline correction is effectively a simple correction of EEG data to, usually, the pre-stimulus interval. Specifically, EEG data before the event of interest is assumed to have a mean of zero voltage and the EEG activity is simply shifted in the voltage domain to meet this assumption.

We will now carve up the continuous EEG data around 'triggers' that correspond to the presentation of images to-be-remembered under both conditions. Specifically, we will carve up the brain data 100ms before the image appears and 1000ms after it is on the screen. The two triggers that correspond to the presentation of images under different conditions are `S 10` which corresponds to the *rhythmic* condition and `S 12` which corresponds to the *arrhythmic* condition. We'll carve up the EEG seperately for each condition.

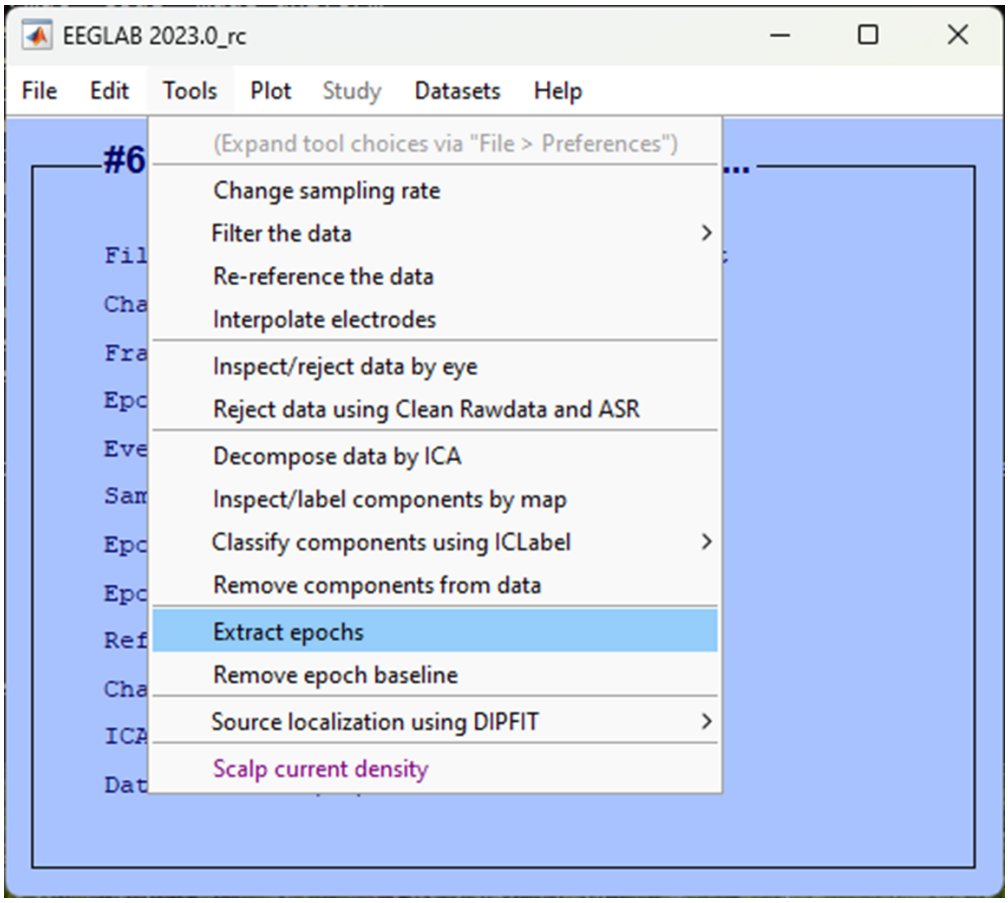
> **i** Terms
>
> Again, EEG terms can be a little confusing two words are often used to mean the same thing:
>
> 1. **Epoch**
> 2. **Segment**
>
> Both usually refer to a small section of the EEG data usually around a specific event - a stimulus or response.

How

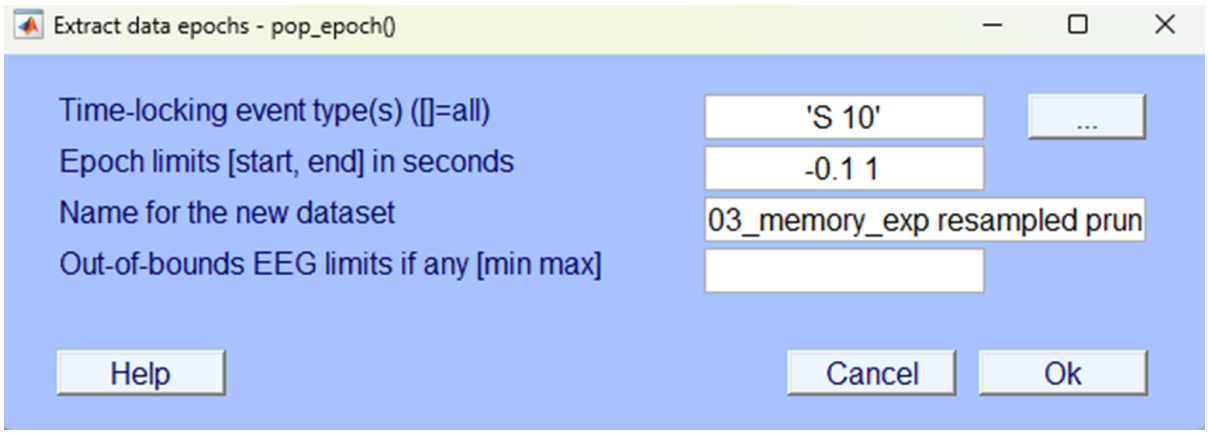
Segment

EEGLAB allows you to segment the data and, at the same time, conduct a baseline correction.

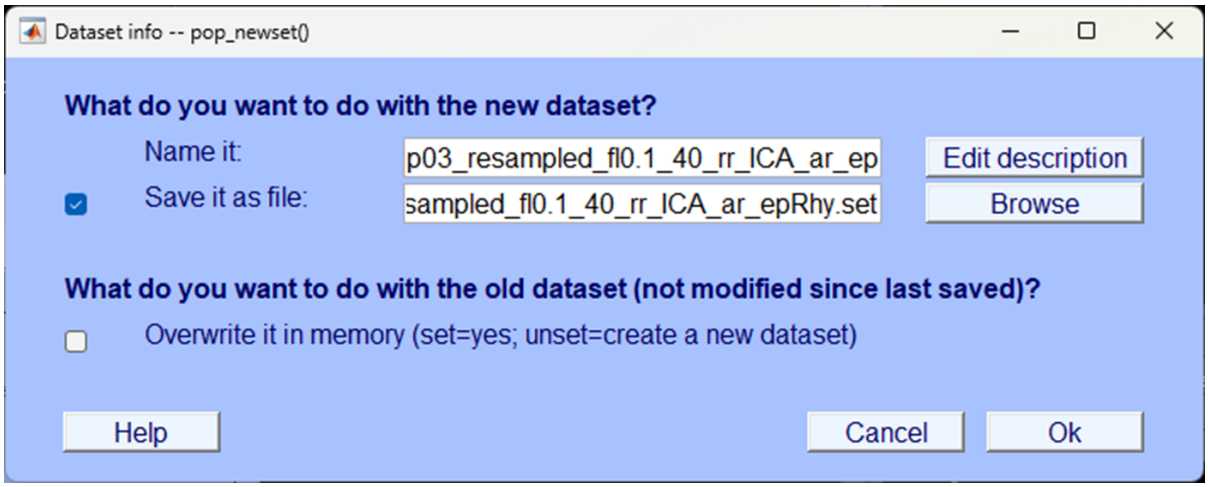
```
-> Tools
  -> Extract epochs
```



In the next pop up window we want to specify the events, we'll first lock to the presentation of images during the rhythmic condition so click on the three dots `...` and select the trigger `S 10`. We want to specify 100 ms before the image and up to one second after as our epoch. To do so, in the `Epoch limits [start, end] in seconds` enter `0.1 1`.

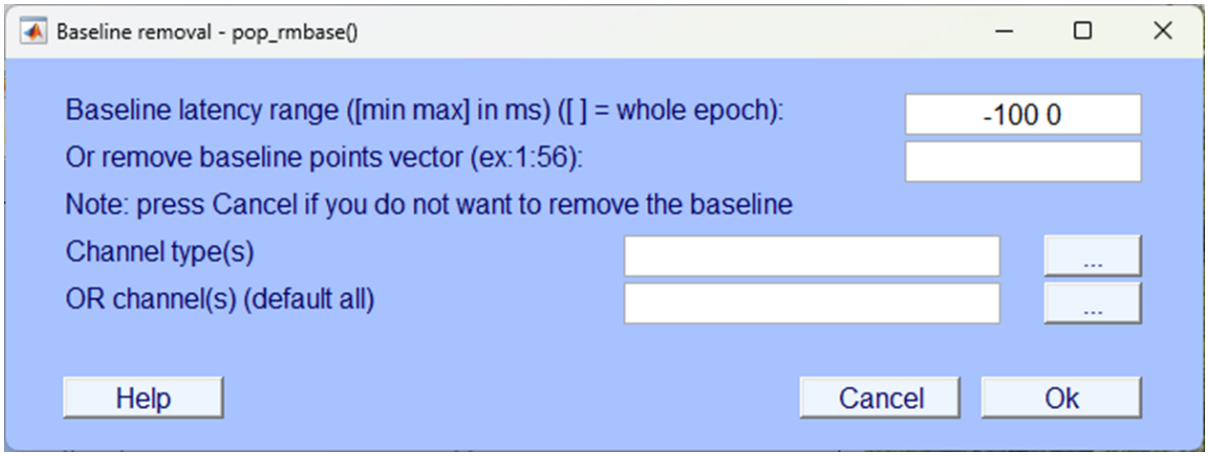


In the next pop up save it so you can identify the dataset as one that is segmented specifically for the rhythmic condition.

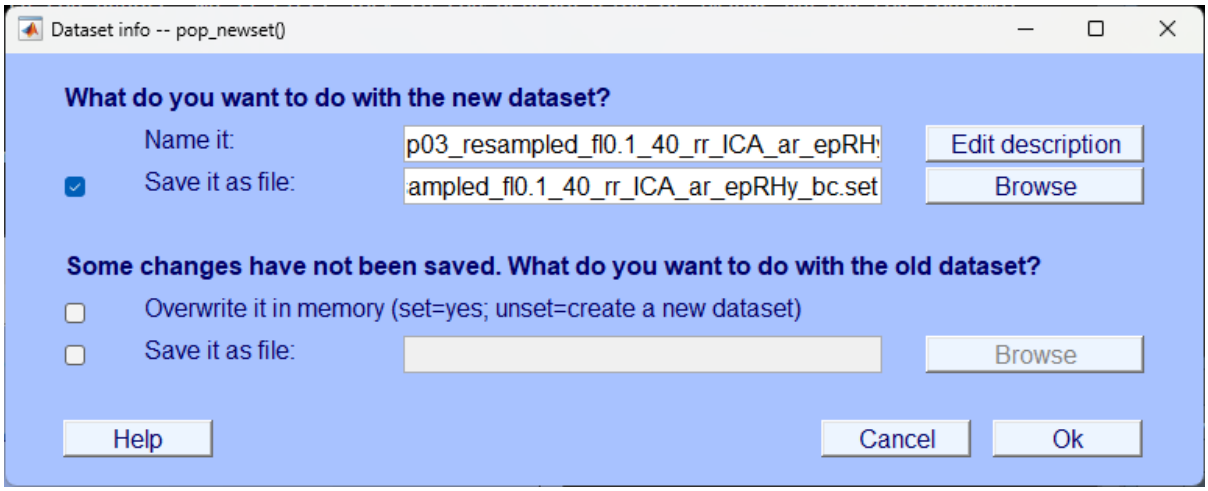


Baseline correct

An automatic pop up will prompt you to baseline correct, the default should use the pre-stimulus period as a baseline and this is what we want so you can simply accept by clicking `Ok`.

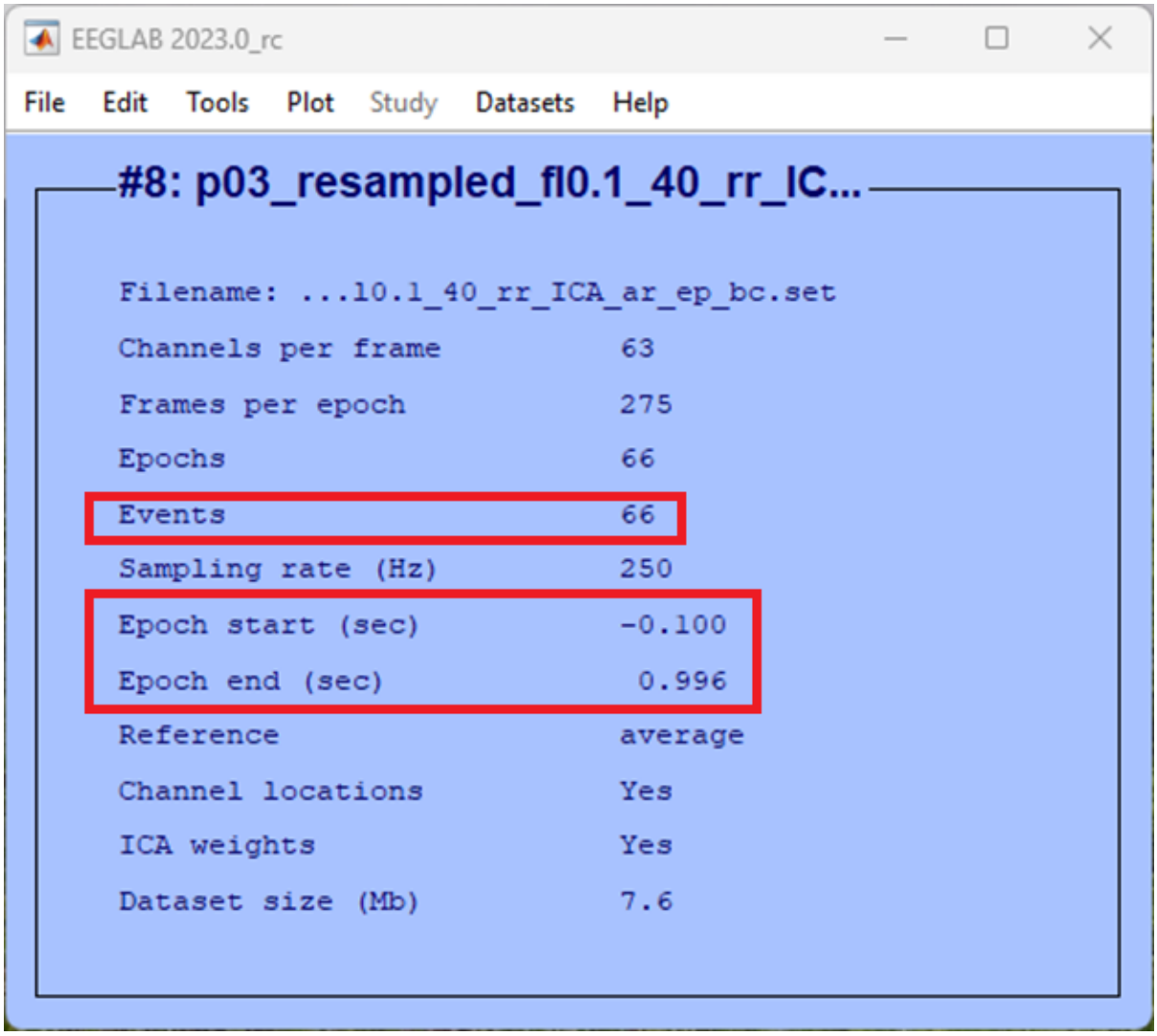


You'll be prompted again to save the data after baseline correcting, you can add the suffix `_bc` for 'baseline correct' and save.



Plot the data

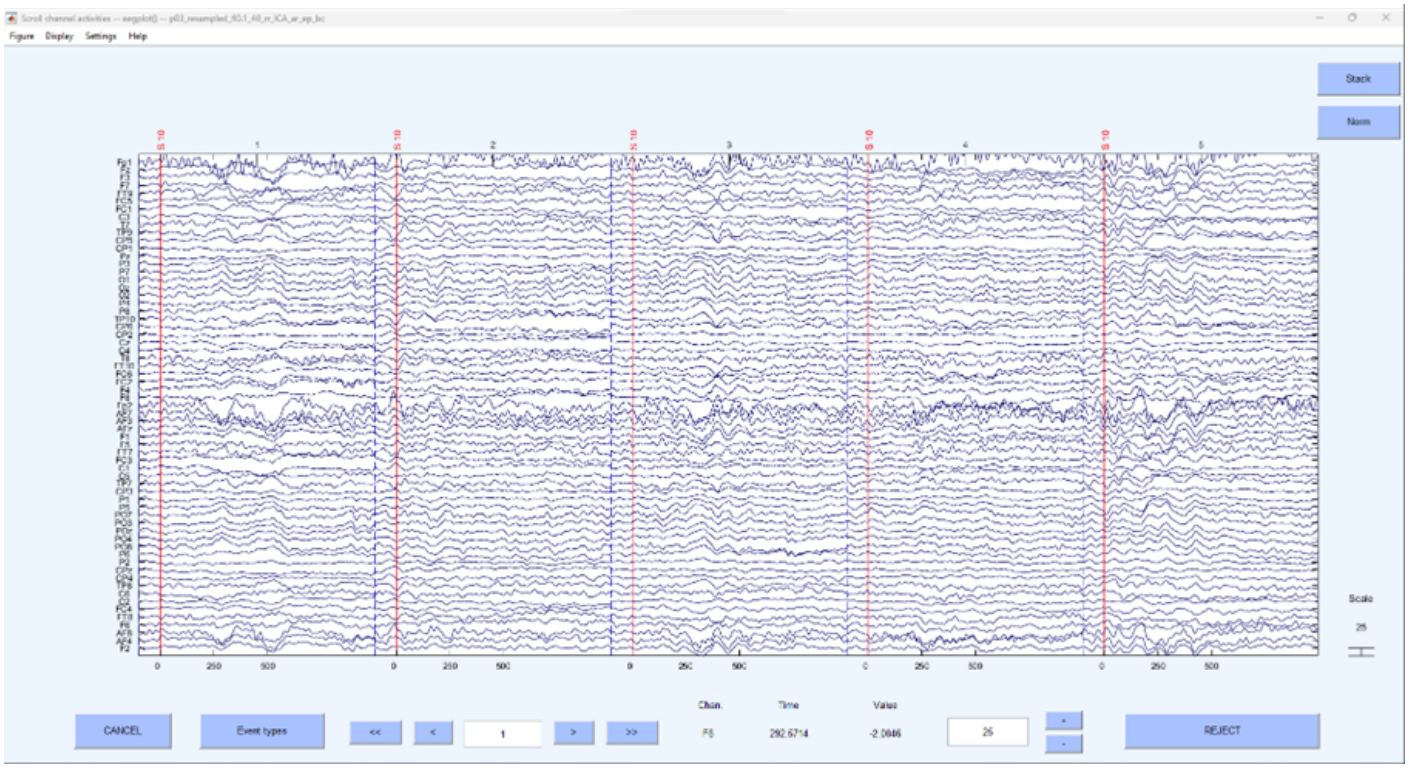
You can now see in the EEGLAB home window that the dataset consisted of segmented data with an epoch that starts at -100 ms from the event and ends just before 1000 ms. Further, there are far fewer events in the data file as we've selected only those relevant to this condition.



If we plot the data we can see it is no longer continuous, click:

```
-> Plot
  -> Channel data (scroll)
```

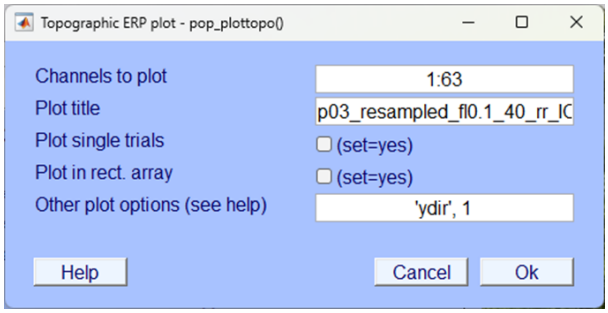
You should see something quite different to what you've seen before using this function, a little like what#s below.



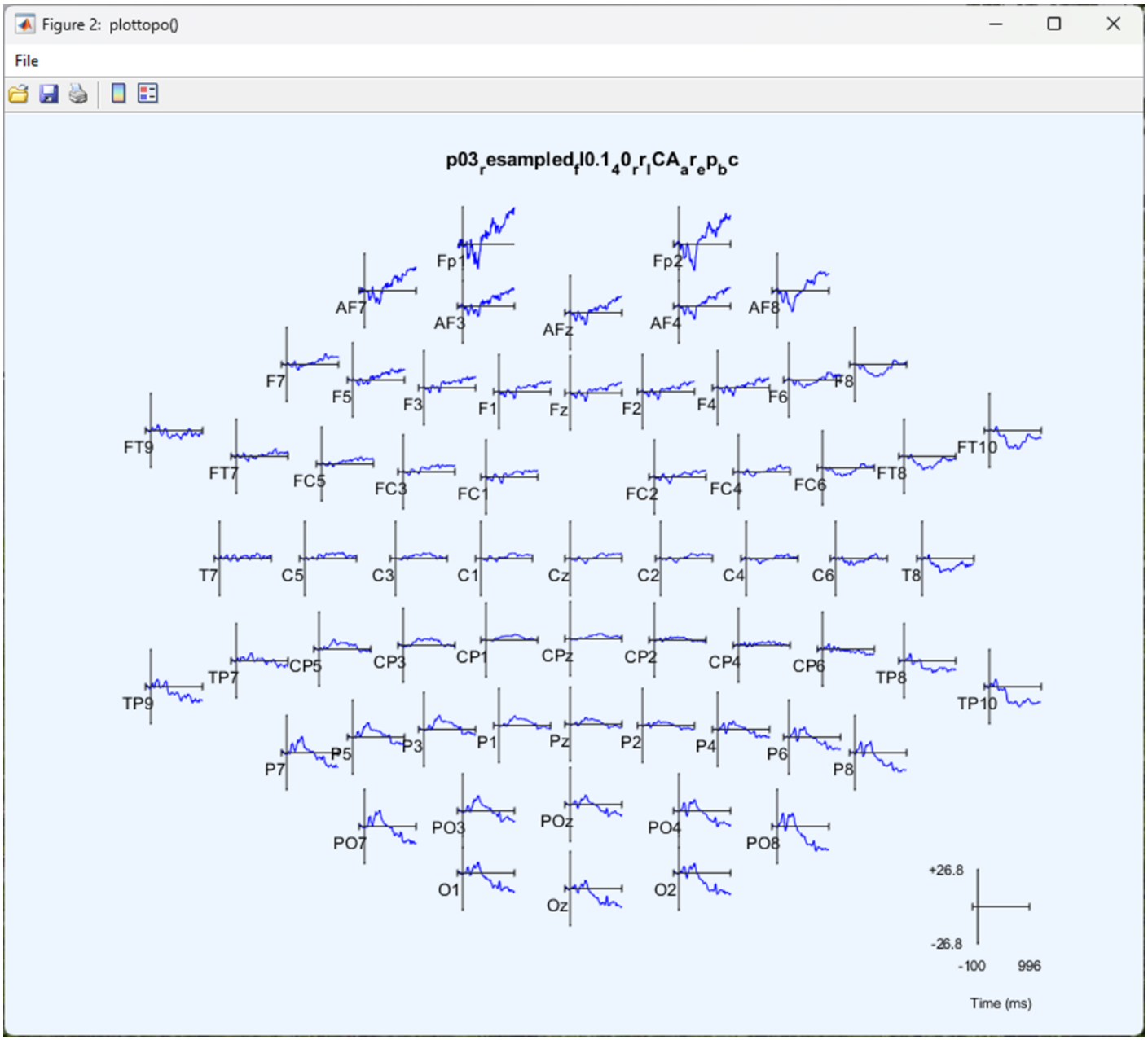
This is no longer continuous data but rather the segments around the specific even of interest that have been stitched together. To generate an ERP we can average across each segment and plot the average, this is simple enough to do in EEGLAB, click:

```
-> Plot
   -> Channel ERPs
     -> In scalp/rect. array
```

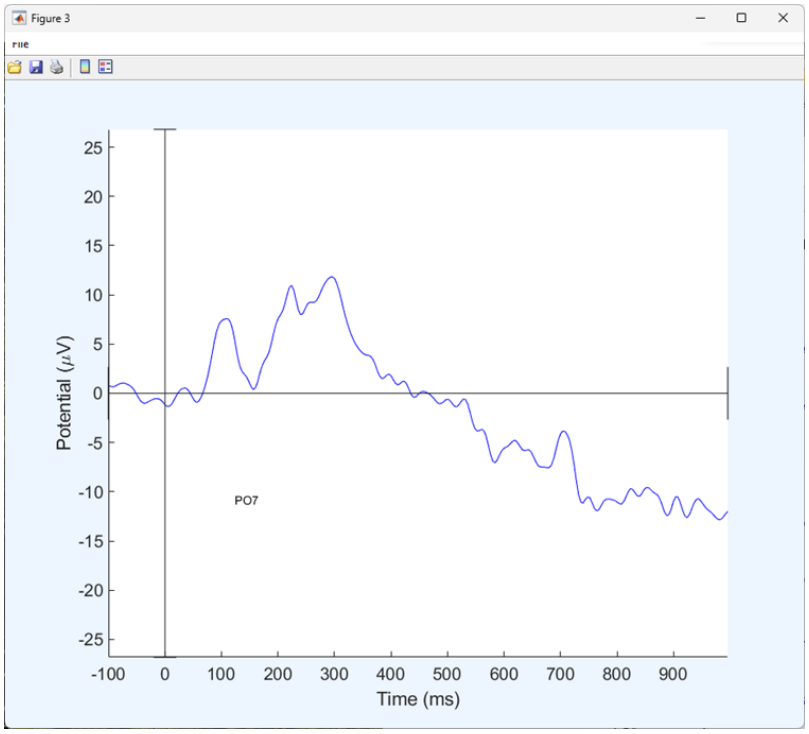
Accept the defaults in the pop up window and click `Ok`.



You should now be looking at an 'Event Related Potential' or 'ERP' - all activity within one condition, averaged over repeated trials and centred around an event. You should have one plot for each electrode and the electrodes are plotted where they appear on a topographical representation of the scalp.



You can click on any electrode for a closer look at the ERP for that electrode.

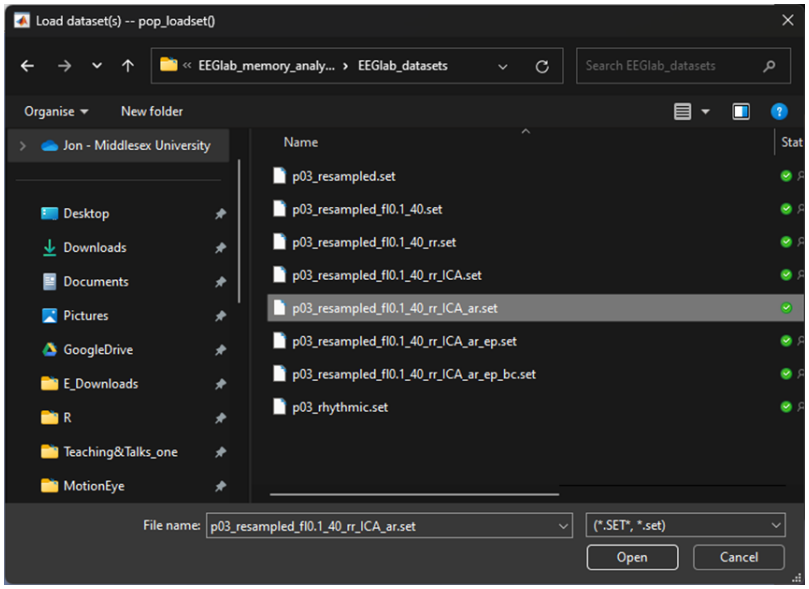


Segment the next condition

We now want to extract the data averaged around the 'arrhythmic' condition. To do this we need to load a dataset before we ran the segmentation procedure, in our case we labled the file `P03_resampled_fl0.1_40_rr_ICA)ar`, click:

```
-> File
  -> Load existing dataset
```

Select your file that has been resampled, filtered, interpolated (if needed), re-referenced, ICA blink reduction and aretfact rejected.



Once you've loaded the file - we want to segment based on the arrhythmic condition. To do so follow through the previous steps in this chapter from Segment. But this time, segment around the trigger `S 13`.

> **!** Trigger values
>
> When you are segmenting the data it's important you remember that you need to do so twice for each participant, remember the trigger codes:
> `S 10` = **Rhythmic condition**
> `S 13` = **Arrhythmic condition**

Question 1 | In EEG analysis, what's another word for 'segment'? _____

Question 2 | What is EEG activity that shifts to a mean of zero during baseline correction?

- (A) Post-stimulus interval
- (B) Pre-stimulus interval
- (C) Alpha band interval
- (D) Theta band interval

All Participants

Great - we have now worked through cleaning the data as needed to extract an Event Related Potential (ERP) for one participant. To complete the rest of the exercise in order to produce an ERP averaged across all participants - a 'Grand Average ERP' you will need to complete all the steps again for each participant in the downloaded data. Importantly, you need to make sure you save both ERP datasets for each condition (Rhythmic and Arrhythmic) for every participant.

Repeating the steps.

You will need to go back through the steps you've completed in the order they appeared.

1. Import the data so it can be read by EEGLAB.

2. Visually examine the data briefly to make sure it's loaded properly.

3. Down sample the data so subsequent analysis isn't so computationally demanding.

4. Filter the data to remove unwanted frequencies.
5. Interpolate bad electrodes only if needed - some data won't need any interpolation.
6. Re-reference the data so that the average of all electrodes is taken as the average.
7. Remove blink artefacts using ICA.
8. Check for any remaining artefacts and reject sections of data that include them.
9. Segment the continuous data and average across both conditions to produce your ERPs.

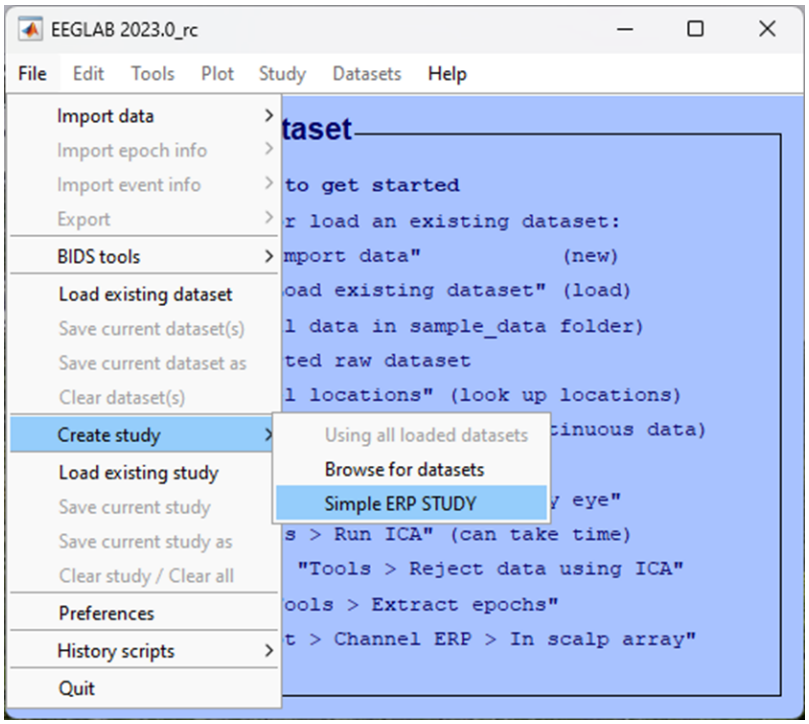
Grand average

A grand average ERP is the average of all ERPs for each individual participant - it isn't used for statistical analysis but provides an excellent visual representation of the ERP data across participants for each condition. It's not the only type of representation of ERPs but certainly the most common. Once you've completed the analysis steps for all of the participant's data in the downloadable data you can construct an average for all the participants and compare conditions.

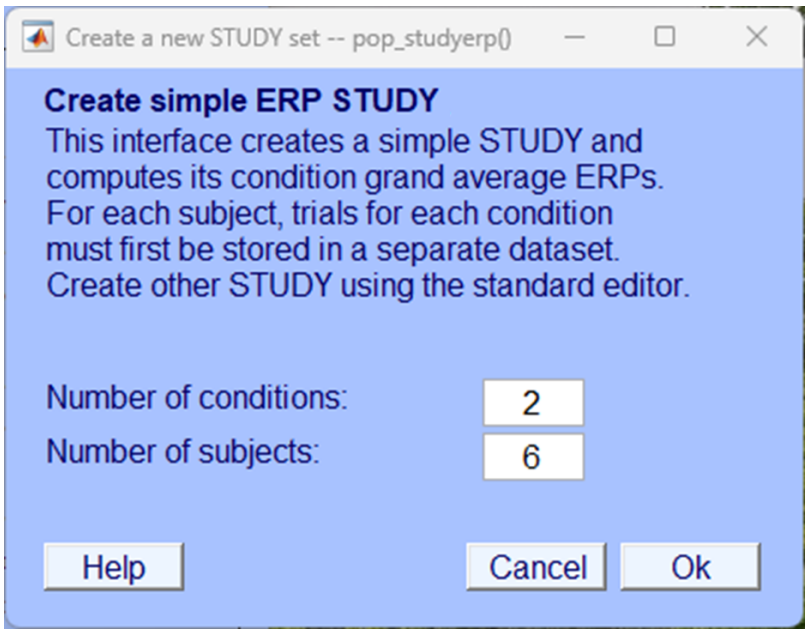
How

To compute a grand average ERP follow these steps:

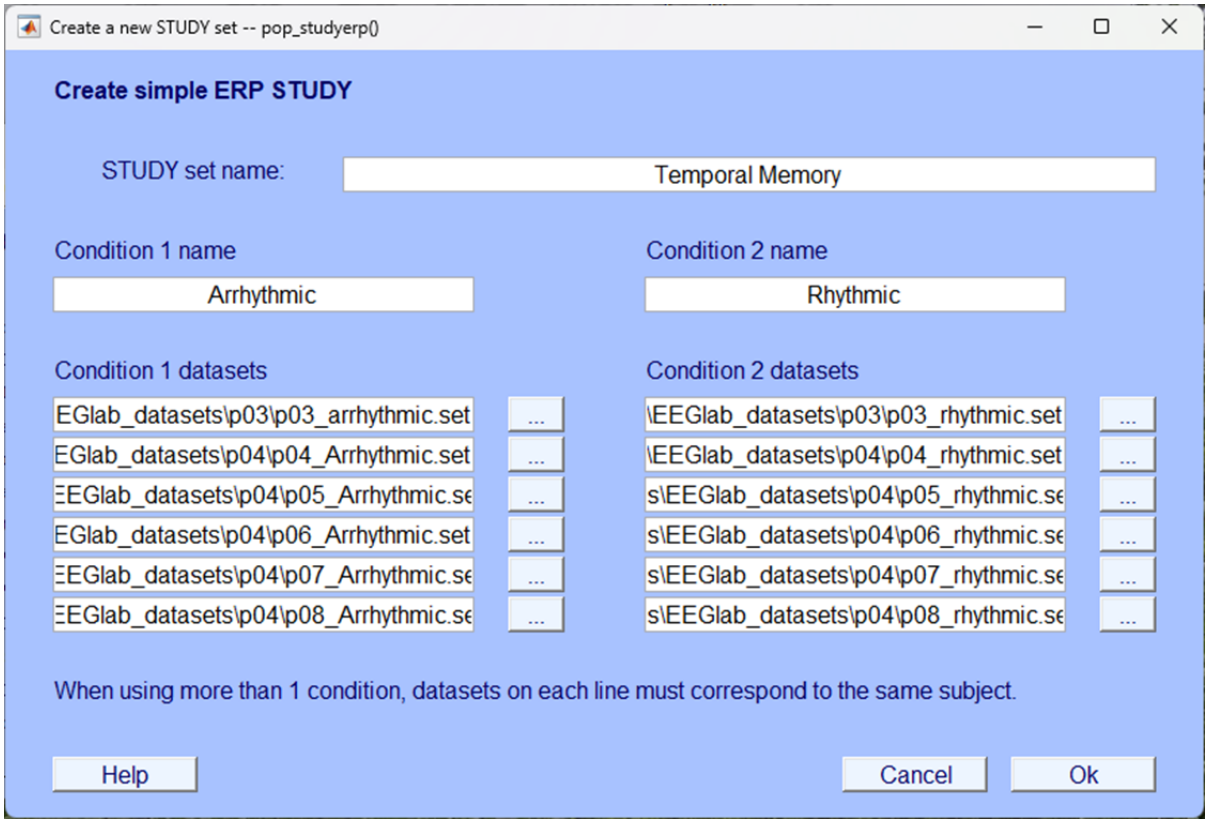
```
-> File
  -> Create study
    -> Simple ERP study
```



If you have analysed all the data from all the participants in the data provided, you should have data from 6 participants, each with 2 conditions. You will need to identify this in the pop up window that appears next:



In the following pop up window you need to include the files that you created in the Segmentation chapter, you should have two files for each participant, one for each condition. Give your study and conditions a name and load up the appropriate files for each participant and each condition. Make sure you load the correct file in to the correct condition as shown below.



After clicking `Ok` the resultant plot that will appear is a grand average ERP for each electrode plotted on a topographic map. This image and the image of the grand average ERPs at electrodes PO7 and PO8 are what is needed for your portfolio. We haven't provided that final output here as you will need to generate that yourself.

End

Congratulations on completing the analysis! There's a lot to learna and a lot to take in so don't worry if it took you longer than expected. If you're unsure about certain sections or have got stuck at any point, remember support is available in class but you can also contact the module leaders to ask any questions via email or to book a one-to-one meeting and go through your questions.

How to Submit

> ! Portfolio Deadline
>
> **You should submit your portfolio and your experiment no later than April 15th 2024, 10am**

EEG write up

Download and complete the EEG template

This template will require you to include an image of the grand average ERP that is the result of the analysis you will have completed if you have gone through the analysis steps correctly.

Written component

Your final portfolio accounts for 80% of your grade for this module. Your EEG analysis and answer to the template questions should form part of your portfolio that should be submitted as one document on MyLearning here (log in to MyLearning required)

Resources

There are multiple sources of information that you should use to help you along your journey to better understanding EEG analysis.

EEGLAB Specific

- EEGLAB tutorial

Matlab support

- Getting started with Matlab

References

Jones, A., & Ward, E. V. (2019). Rhythmic temporal structure at encoding enhances recognition memory. *Journal of Cognitive Neuroscience*, *31*(10), 1549–1562.