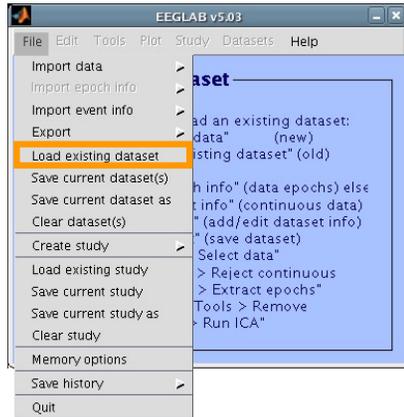


Basic scripting with EEGLAB



(Menus write both dataset and global history)

- Automated processing on groups of subjects (possibly on several processors).
- Richer options for plotting and processing functions (time-frequency decompositions, ...)
- Selecting data/epoch based on event context
- Export data
- Custom processing...



EEGLAB Scripting

Raw data preprocessing

```
datainfo;
pop_editoptions( 'option_storedisk', 1);
outputEEGFolder = 'preprocessed_data';
if ~exist(outputEEGFolder), mkdir(outputEEGFolder); end;

for iSubj = 1:length(sInfo)

    % load dataset
    EEG = pop_biosig(sInfo(iSubj).file);
    EEG.setname = sInfo(iSubj).name;

    % preprocess data
    chanFile= 'plugins/dipfit2.3/standard_BEM/elec/standard_1005.elc';
    EEG = pop_chanedit(EEG, 'lookup', fullfile(fileparts(which('eeglab.m')), chanFile));
    EEG = pop_iirfilt( EEG, 0.5, 0, [], 0, 0); % high pass filtering
    EEG = pop_iirfilt( EEG, 0, 55, [], 0, 0); % low pass filtering
    EEG = pop_select(EEG, 'nochannel', sInfo(iSubj).bad_channels); % remove bad channels
    EEG = pop_reref( EEG, []); % average reference (optional)
    EEG = eeg_eegrej( EEG, sInfo(iSubj).bad_data); % remove bad portions of data

    % run ICA
    EEG = pop_runica(EEG, 'icatype', 'sobi');

    % tag bad components
    EEG = pop_findmatchingrejcomps(EEG, 'matchcomps', sInfo(iSubj).bad_comps, 'corrthresh', 0.92);

    % extract data epochs
    EEG = pop_epoch(EEG, { 2 4 } , [-1 2]);

    % save dataset
    EEG.saved = 'no';
    EEG = pop_saveset( EEG, 'filepath', outputEEGFolder, 'filename', [ sInfo(iSubj).name '.set' ] );
end
```

Group level analysis

```
datainfo;
pop_editoptions( 'option_storedisk', 1);
outputEEGFolder = 'preprocessed_data';
studyCommand = {};

% generate STUDY commands
for iSubject = 1:length(sInfo)
    fileName = fullfile(outputEEGFolder, [ sInfo(iSubject).name '.set' ]);
    studyCommand = [ studyCommand { 'index' iSubject 'load' fileName 'subject' ...
        sInfo(iSubject).name } ];
end;

% create data
[STUDY ALLEEG] = std_editset( [], [], 'name', 'test', 'commands', studyCommand, ...
    'updatedat','off', 'filename', 'test_study', 'resave', 'on');
STUDY = std_makedesign(STUDY, ALLEEG, 1, 'name', 'STUDY_design 1', 'delfiles', 'off', ...
    'defaultdesign', 'off', 'variable1', 'type', 'values1', {'2' '4'});

% update workspace variables and redraw EEGLAB
CURRENTSTUDY = 1; EEG = ALLEEG; CURRENTSET = [1:length(EEG)];
[STUDY, ALLEEG] = std_checkset(STUDY, ALLEEG);
eeglab redraw

% precompute and plot data
allchanlocs = eeg_mergelocs(ALLEEG.chanlocs);
[STUDY ALLEEG] = std_precomp(STUDY, ALLEEG, {}, 'interp', 'on', 'recompute', 'on', 'erp', 'on');
STUDY = pop_statparams(STUDY, 'condstats', 'on', 'singletrials', 'on', 'mode', 'fieldtrip', ...
    'fieldtripmethod', 'montecarlo', 'fieldtripmcorrect', 'cluster');
[STUDY erp] = std_erpplot(STUDY, ALLEEG, 'channels', {allchanlocs.labels}, 'topotime', [300 400]);
print results.eps -depsc
```



Using EEGLAB history for basic scripting

Task 1

Create a script from 'eegh' output

Task 2

Adapt your script with variables

Task 3

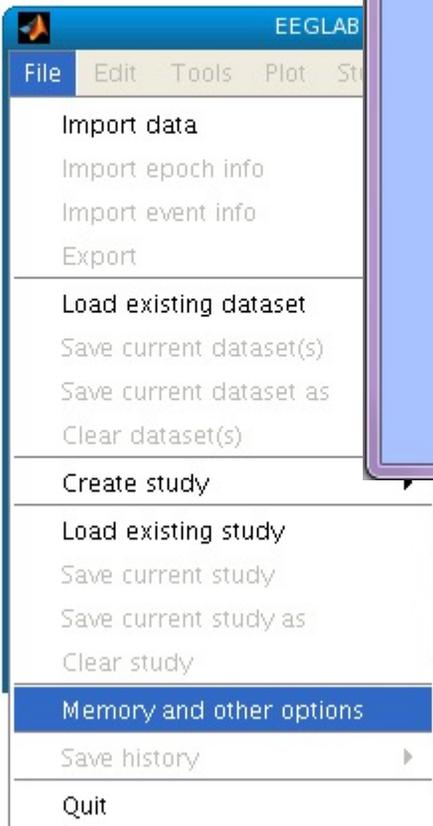
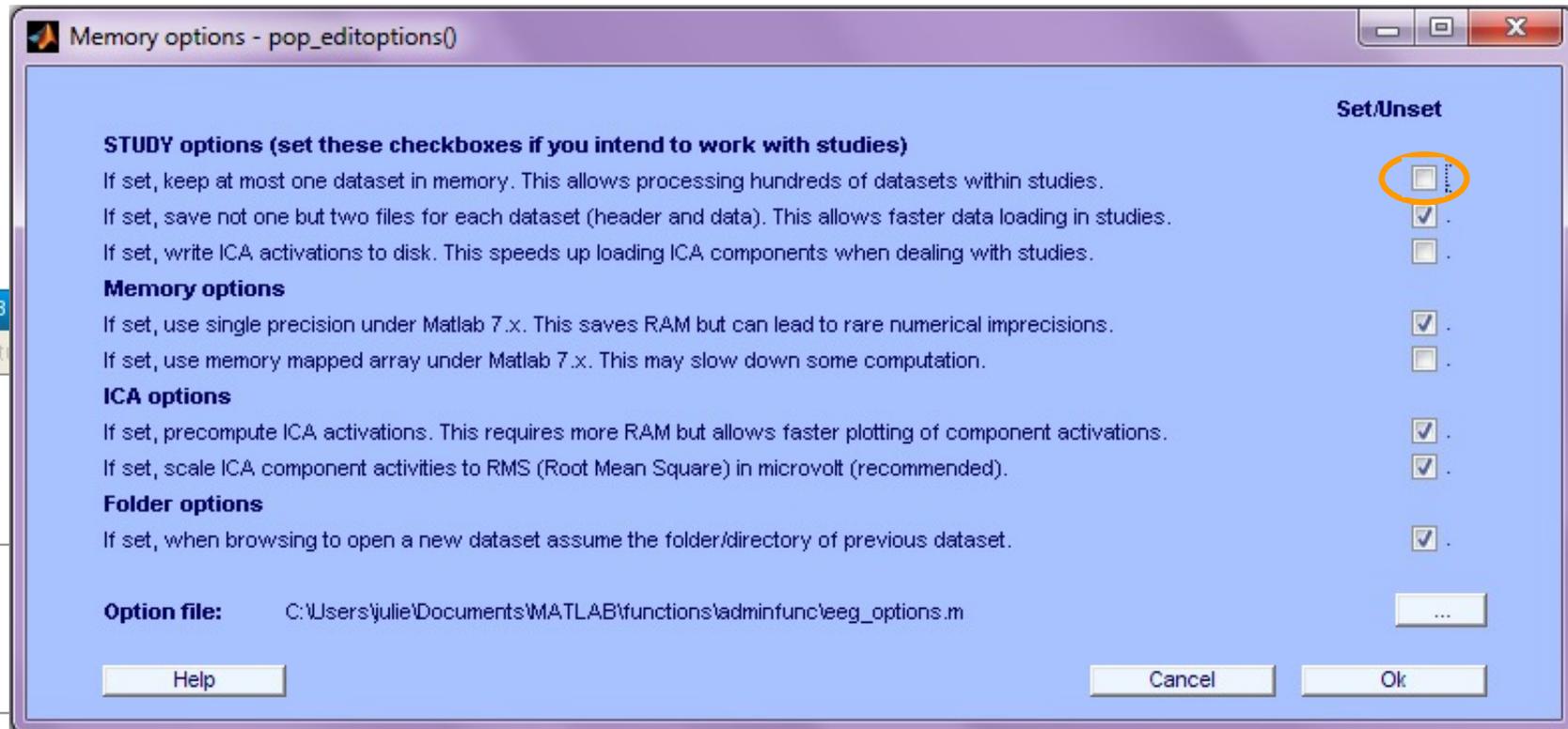
Create a Matlab function

Task 4

Exercise...



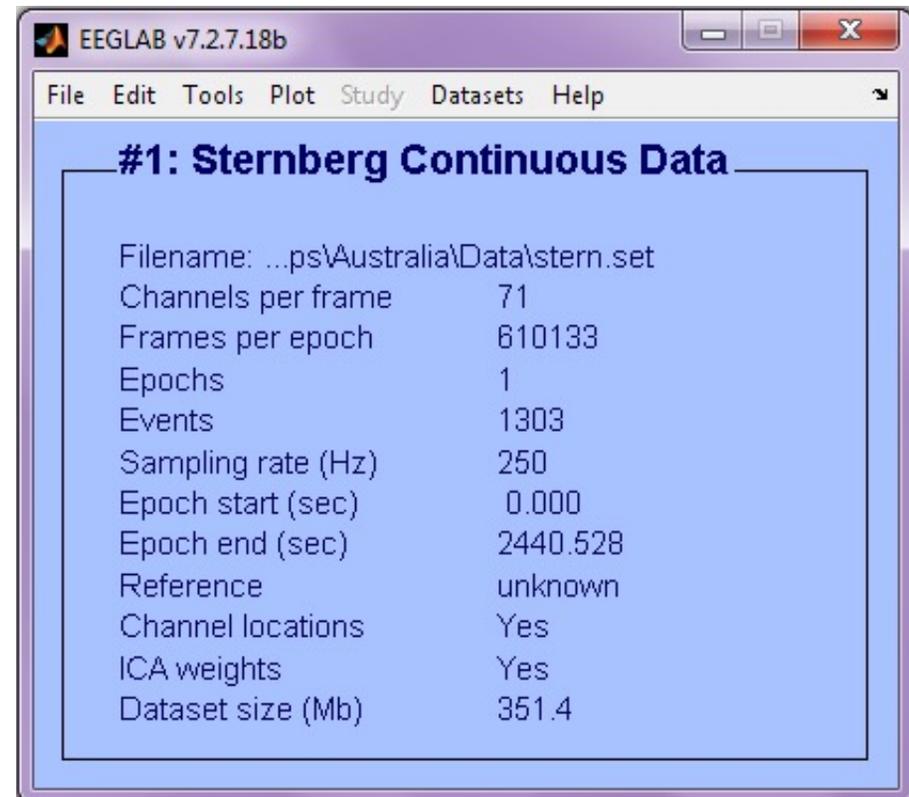
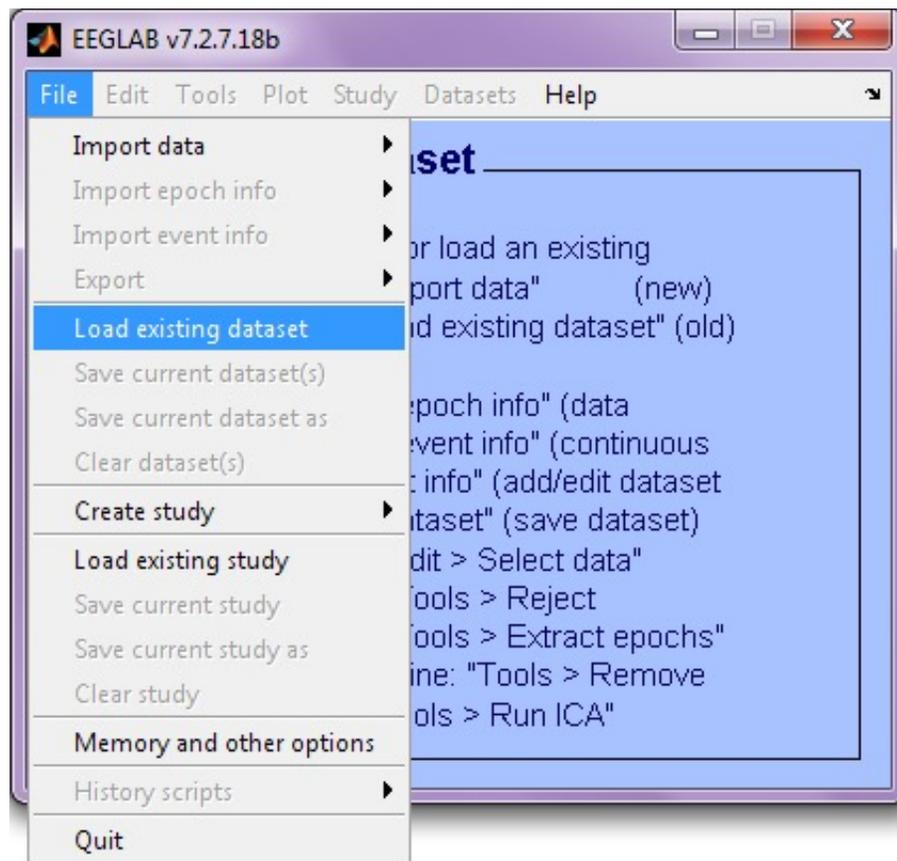
Memory options



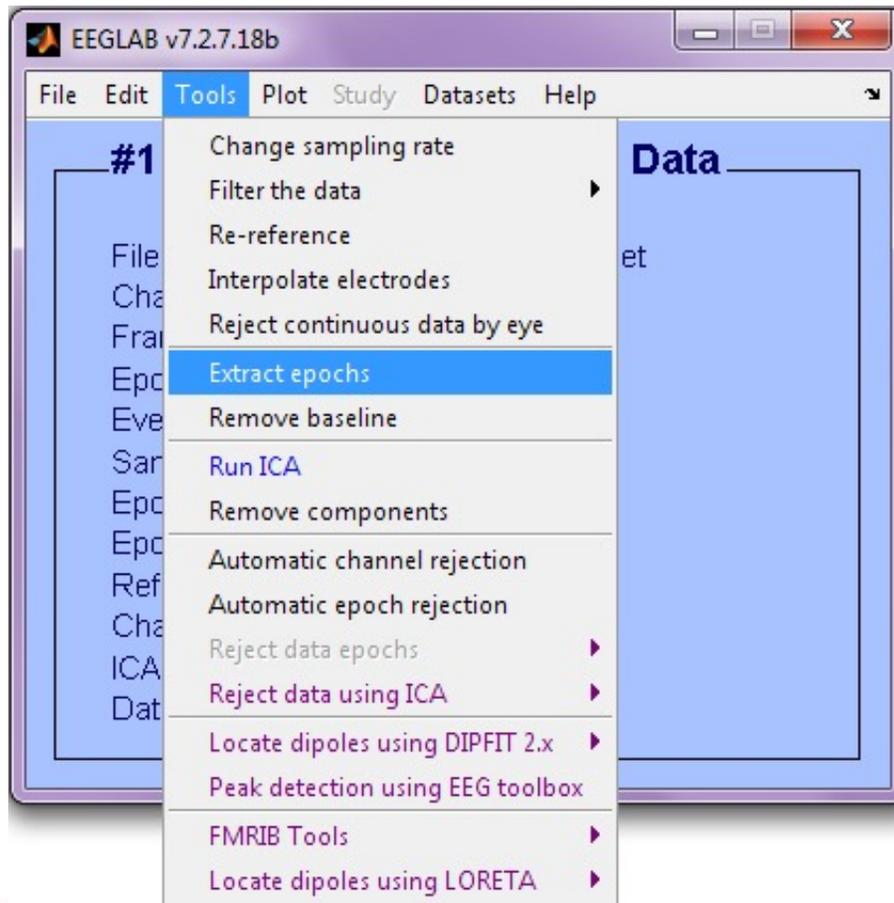
**Change memory options
to allow more than one dataset in memory**

Create a script from 'eegh' output

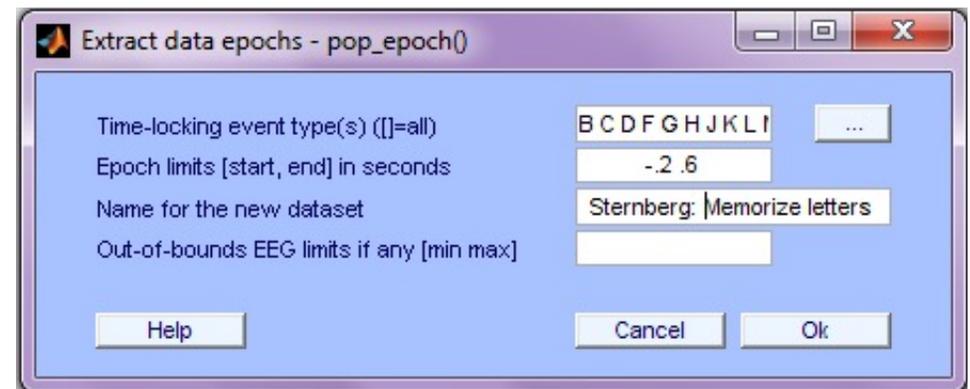
Start by loading a continuous dataset



Create a script from 'eegh' output



Epoch on Memorize letters



Create a script from 'eegh' output

Plot an IC ERP image

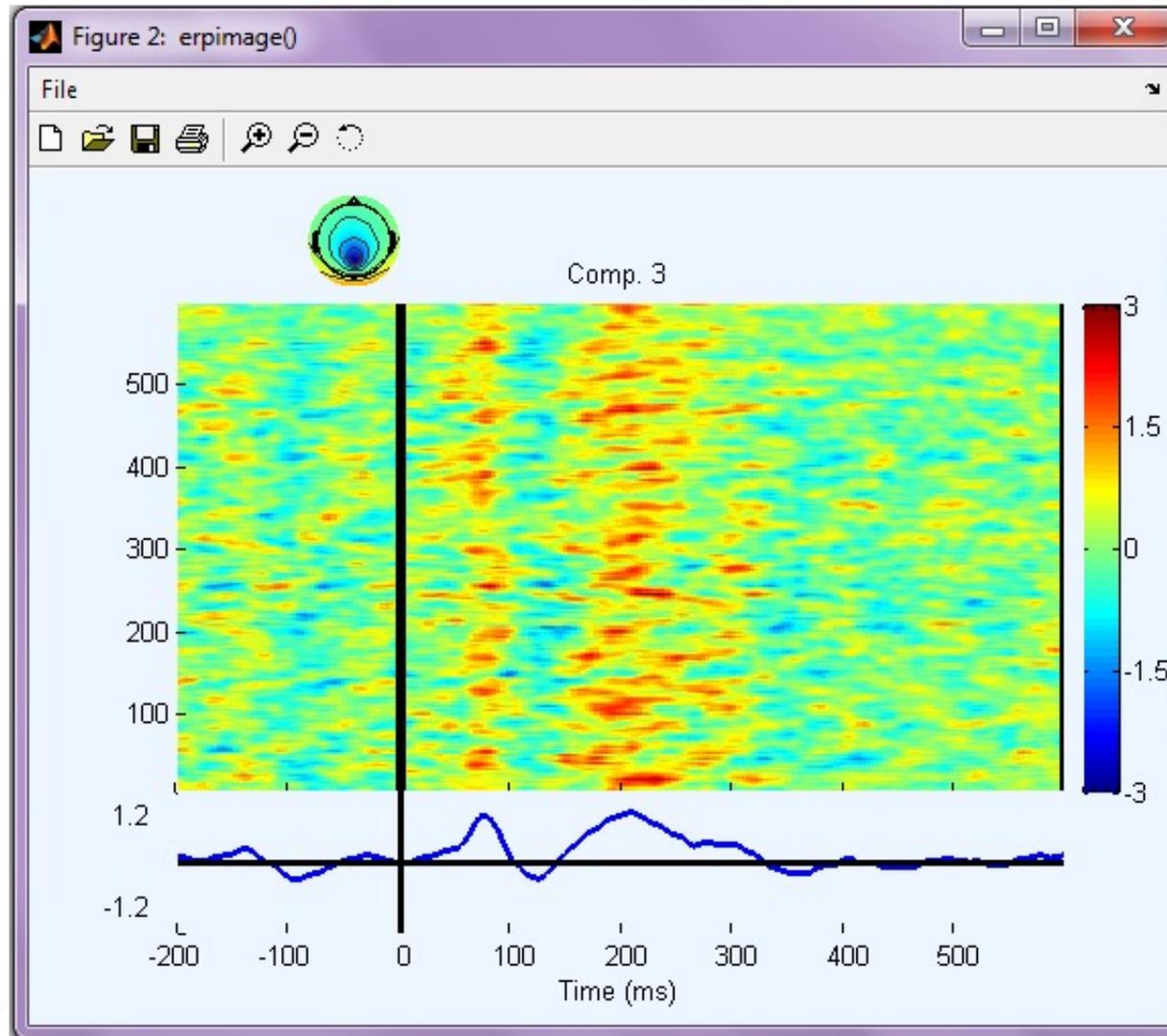
The image shows the EEGLAB v7.2.7.18b interface. The 'Plot' menu is open, with 'Component ERP image' selected. The 'Component ERP image -- pop_erpimage()' dialog box is displayed, showing various settings for plotting an IC ERP image. The settings are as follows:

Parameter	Value
Component(s)	3
Project to channel #	
Smoothing	10
Downsampling	1
Time limits (ms)	-200 596
Figure title	
Plot scalp map	<input checked="" type="checkbox"/>
Plot ERP	<input checked="" type="checkbox"/>
Plot colorbar	<input checked="" type="checkbox"/>
ERP limits	
Color limits (see Help)	
Sort/align trials by epoch event values	
Epoch-sorting field	Event type(s)
Event time range	
Rescale	no
Align	
Don't sort by value	<input type="checkbox"/>
Don't plot values	<input type="checkbox"/>
Sort trials by phase	
Frequency (Hz minHz maxHz)	
Percent low-amp. trials to ignore	
Window center (ms)	
Wavelet cycles	3
Inter-trial coherence options	
Frequency (Hz minHz maxHz)	
Signif. level (<0.20)	
Amplitude limits (dB)	
Coher limits (<=1)	
Image amps (Requires signif.)	<input type="checkbox"/>
Other options	
Plot spectrum (minHz maxHz)	
Baseline ampl. (dB)	
Mark times (ms)	
More options (see >> help erpimage)	

Buttons: Help, Cancel, Ok



Create a script from 'eegh' output



Retrieve commands from eegh

Write a script to do this:

```
>> eegh
```



Retrieve commands from eegh

```
>> eegh
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;

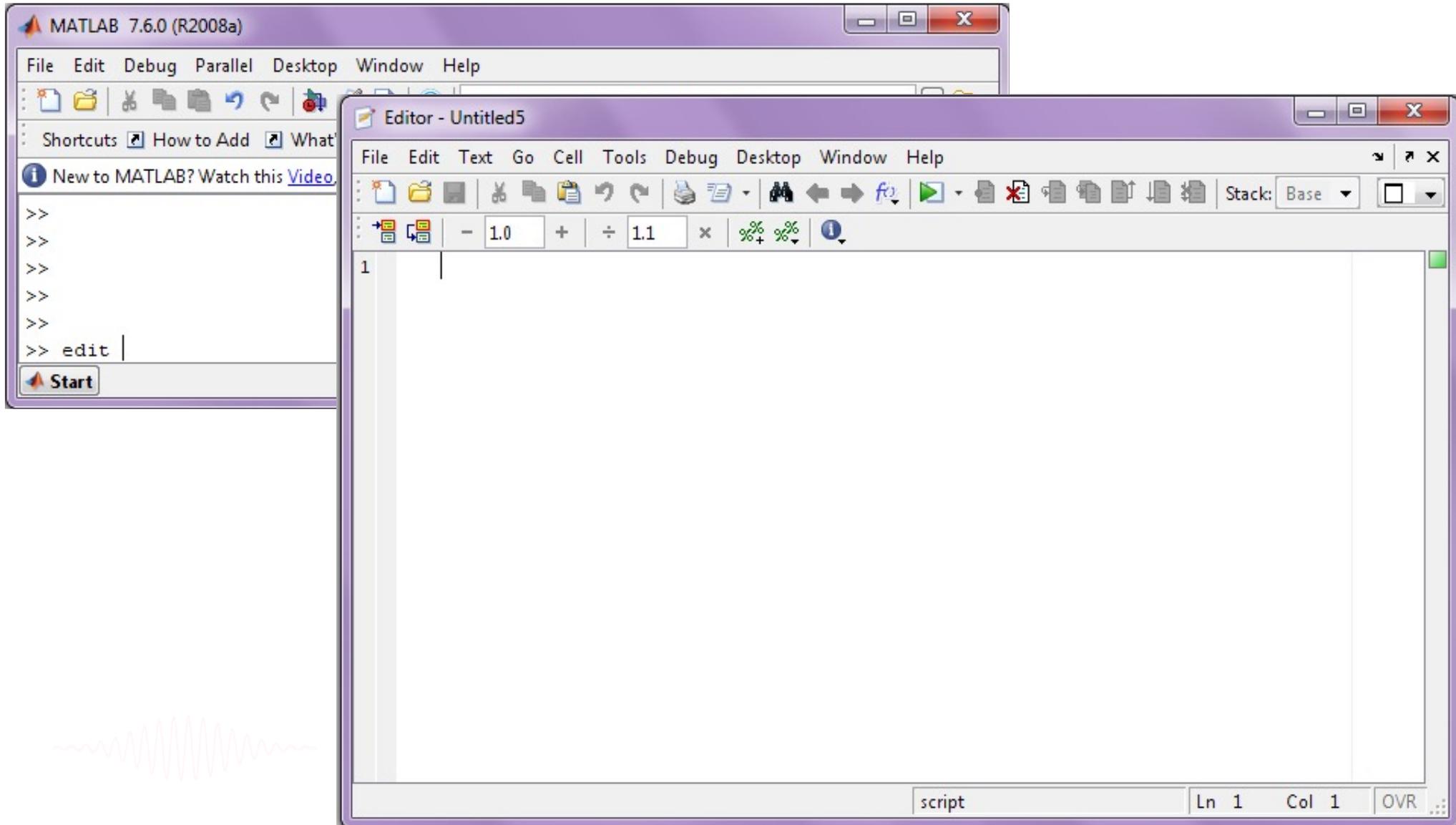
EEG = pop_loadset('filename', 'stern_125Hz.set');
[ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG, 0);

EEG = pop_epoch( EEG, {'B' 'C' 'D' ... }, [-0.2 0.6], 'newname',
'Memorize epochs', 'epochinfo', 'yes');
[ALLEEG EEG CURRENTSET] = eeg_store(ALLEEG, EEG, 1);
EEG = pop_rmbase( EEG, [-200 0]);
[ALLEEG EEG] = eeg_store(ALLEEG, EEG, CURRENTSET);

figure; pop_erpimage(EEG,0, [3],[], 'Comp.
3',10,1,{},[],'', 'yerplabel', '', 'erp', 'on', 'cbar',
'on', 'topo', {mean(EEG.icawinv(:,[3]),2) EEG.chanlocs EEG.chaninfo
});
```

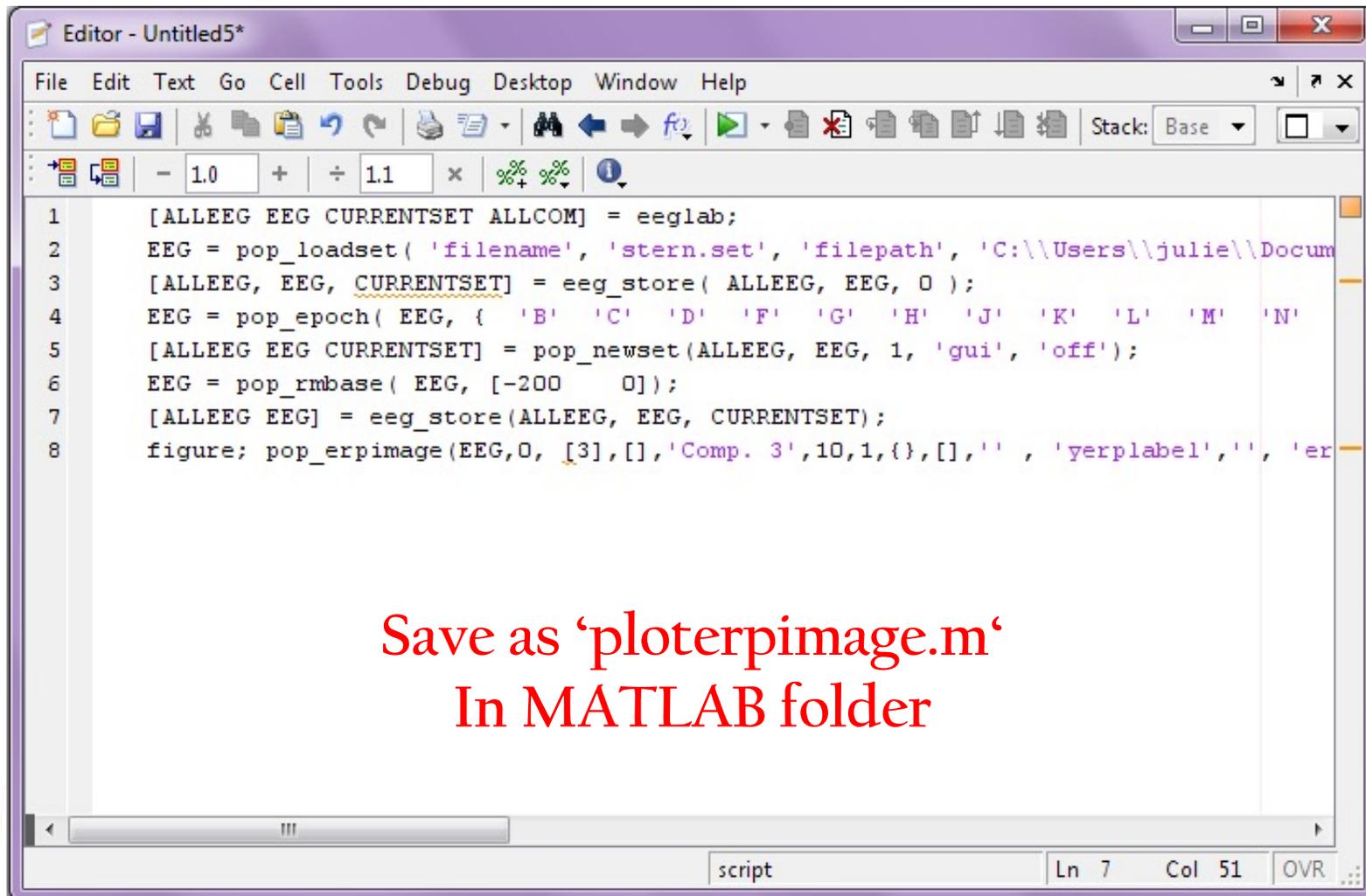


Create a Matlab script



Create a Matlab script

Copy and paste from Matlab window:



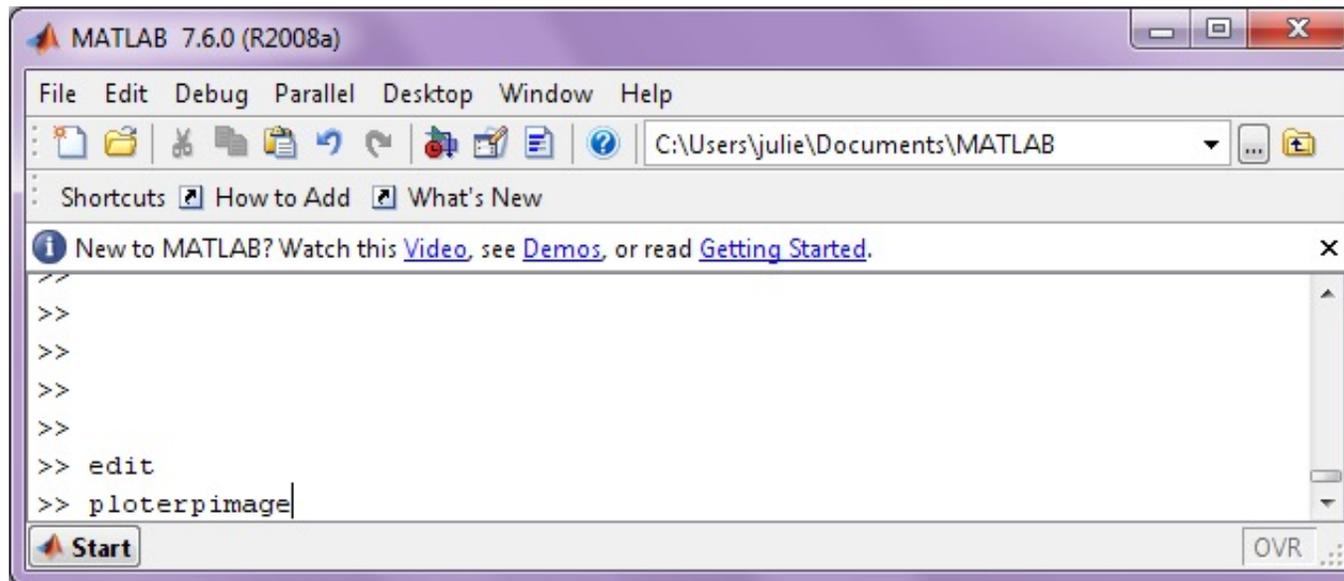
The screenshot shows a Matlab Editor window titled 'Editor - Untitled5*'. The window contains a script with the following code:

```
1 [ALLEEG EEG CURRENTSET ALLCOM] = eeglab;  
2 EEG = pop_loadset( 'filename', 'stern.set', 'filepath', 'C:\\Users\\julie\\Docum  
3 [ALLEEG, EEG, CURRENTSET] = eeg_store( ALLEEG, EEG, 0 );  
4 EEG = pop_epoch( EEG, { 'B' 'C' 'D' 'F' 'G' 'H' 'J' 'K' 'L' 'M' 'N'  
5 [ALLEEG EEG CURRENTSET] = pop_newset( ALLEEG, EEG, 1, 'gui', 'off' );  
6 EEG = pop_rmbase( EEG, [-200 0] );  
7 [ALLEEG EEG] = eeg_store( ALLEEG, EEG, CURRENTSET );  
8 figure; pop_erpimage( EEG, 0, [3], [], 'Comp. 3', 10, 1, {}, [], ' ', 'yerplabel', ' ', 'er
```

Below the code, the text 'Save as 'ploterpimage.m'' and 'In MATLAB folder' is displayed in red. The status bar at the bottom of the window shows 'script', 'Ln 7', 'Col 51', and 'OVR'.

Save as 'ploterpimage.m'
In MATLAB folder

Run your new script



Exercise page 1

```
>> eeglab
```

```
% load dataset stern_125hz.set in data folder  
% epoch on 'memorize letter' B, C, etc...  
% plot erpimage for component 3
```

```
>> eegh
```

```
% open Matlab editor
```

```
>> edit
```

```
% copy & paste eegh results into a new  
% file and save it (ploterpimage.m)
```

```
>> clear
```

```
>> close all
```

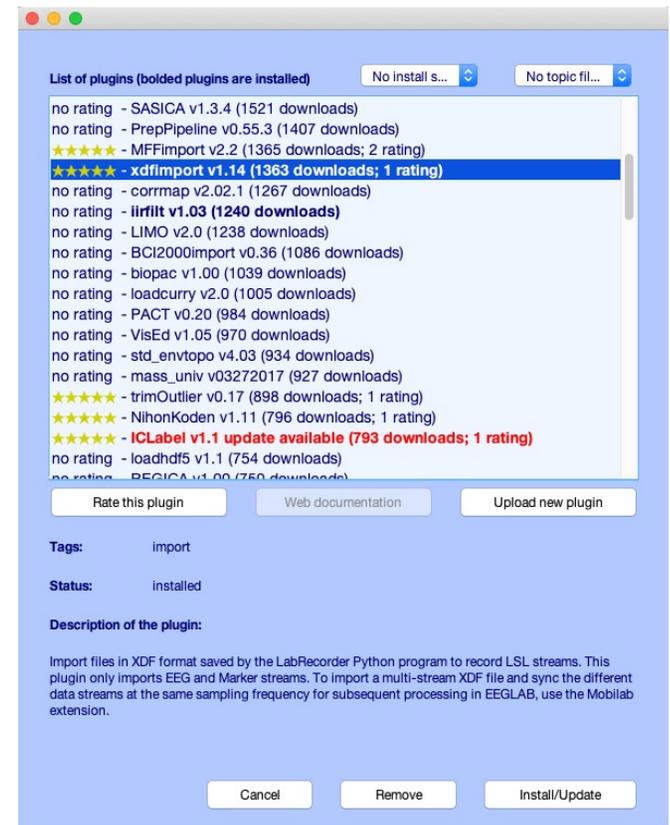
```
>> ploterpimage
```

```
>> eeglab redraw
```



Command line tools

write a simple plugin



List of plugins (bolded plugins are installed)

- no rating - SASICA v1.3.4 (1521 downloads)
- no rating - PrepPipeline v0.55.3 (1407 downloads)
- ★★★★★ - MFFImport v2.2 (1365 downloads; 2 rating)
- ★★★★★ - **xdfimport v1.14 (1363 downloads; 1 rating)**
- no rating - corrmmap v2.02.1 (1267 downloads)
- no rating - **iirfilt v1.03 (1240 downloads)**
- no rating - LIMO v2.0 (1238 downloads)
- no rating - BCI2000Import v0.36 (1086 downloads)
- no rating - biopac v1.00 (1039 downloads)
- no rating - loadcurry v2.0 (1005 downloads)
- no rating - PACT v0.20 (984 downloads)
- no rating - VisEd v1.05 (970 downloads)
- no rating - std_envtopo v4.03 (934 downloads)
- no rating - mass_univ v03272017 (927 downloads)
- ★★★★★ - trimOutlier v0.17 (898 downloads; 1 rating)
- ★★★★★ - NihonKoden v1.11 (796 downloads; 1 rating)
- ★★★★★ - **ICLabel v1.1 update available (793 downloads; 1 rating)**
- no rating - loadhdf5 v1.1 (754 downloads)
- no rating - BEGICA v1.00 (750 downloads)

Tags: import

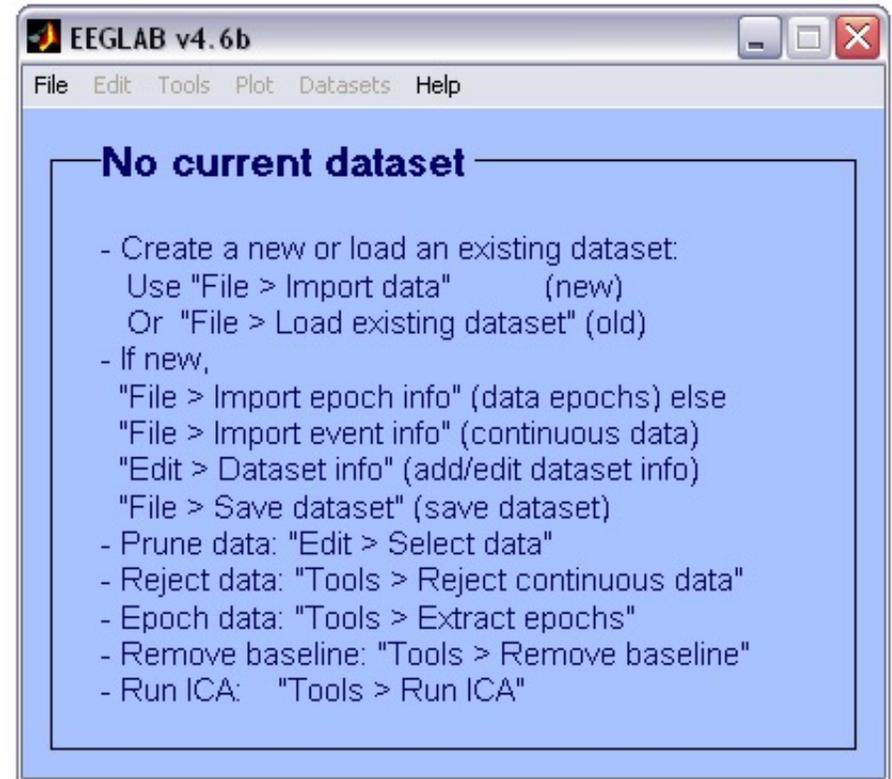
Status: installed

Description of the plugin:

Import files in XDF format saved by the LabRecorder Python program to record LSL streams. This plugin only imports EEG and Marker streams. To import a multi-stream XDF file and sync the different data streams at the same sampling frequency for subsequent processing in EEGLAB, use the Mobilab extension.

Starting EEGLAB

```
>> eeglab
eeglab: options file is /Volumes/donnees/data/STUDYstc
Adding path to all EEGLAB functions
Adding path to eeglab/external/bioelectromagnetism_ligt
Adding path to eeglab/external/biosig-partial
Adding path to eeglab/external/fieldtrip-partial
Adding path to eeglab/external/fieldtrip-partial subfolders
EEGLAB: adding plugin function "eegplugin_VisEd"
EEGLAB: adding "eepimport1.02" plugin (see >> help eepimport)
EEGLAB: adding "bdfimport" plugin (see >> help eegplugin_bdfimport)
EEGLAB: adding "brainmovie0.1b" plugin (see >> help eegplugin_brainmovie)
EEGLAB: adding "ctfimport1.03" plugin (see >> help eegplugin_ctfimport)
EEGLAB: adding "dipfit2.2" plugin (see >> help eegplugin_dipfit)
EEGLAB: adding "EEG toolbox ERP plotting" plugin (see >> help eegplugin_eeg_toolbox)
EEGLAB: adding "erpssimport1.00" plugin (see >> help eegplugin_erpssimport)
EEGLAB: adding "fmrib1.21" plugin (see >> help eegplugin_fmrib)
EEGLAB: adding "iirfilt1.01" plugin (see >> help eegplugin_iirfilt)
EEGLAB: adding "eepimport1.02" plugin (see >> help eegplugin_ascinstep)
EEGLAB: adding "loreta1.0" plugin (see >> help eegplugin_loreta)
EEGLAB: adding "Butter1.0" plugin (see >> help eegplugin_ERPLAB_filters)
EEGLAB: adding "Measure_Product1.0" plugin (see >> help eegplugin_mp_clustering)
EEGLAB: adding plugin function "eegplugin_miclust"
EEGLAB: adding "4dneuroimaging1.00" plugin (see >> help eegplugin_4dneuroimaging)
>>
```



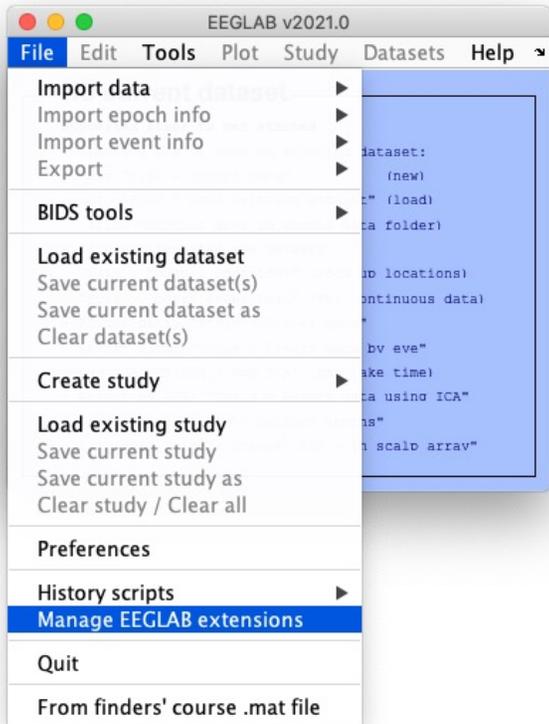
Proper EEGLAB plugins

eepimport1.02	Data importing for EEprobe data (Oostenved & ANT company)
bva_io1.30	Brain vision analyzer import/export plugin (Widmann & Delorme)
ctfimport1.01	MEG CTF import plugin (Carver, Weber & Delorme)
dipfit4.0	4-shell and BEM (Oostenveld & Delorme)
fmrib1.2b	Removal of artifact from simultaneously EEG/fMRI recording (Niazi)
iirfilt1.0	Non-linear IIR filtering (Pozdin)

Matlab toolboxes interfaced as plugins

BIOSIG	Data importing for rare data binary format (Schloegl)
File-IO	Data importing (Oostenveld)
Fieldtrip	Source localization and time-freq. decompositions (Oostenveld)
LIMO-EEG	General linear model and EEG
SIFT	Source information flow toolbox

EEGLAB plugin manager



EEGLAB online plugin page

Plugin list all

sccn.ucsd.edu/eeglab/plugin_uploader/plugin_list_all.php

List of plug-ins available for download in EEGLAB 2019.1 and later versions

You may download the .zip file associated with the plug-ins below and unzip it in the "plugins" subfolder of your EEGLAB distribution. However, this list is primarily provided for informational purposes. To install plug-ins of interest, we recommend that you use the EEGLAB extension manager (in the EEGLAB GUI window, select menu item "File > Manage plug-ins"). Note: it is not possible to install new plug-ins in the EEGLAB compiled version. However, you may use a plug-in when running the compiled version by including the plug-in code base in the EEGLAB script you run.

Search:

Plug-in name	Version	Plug-in description	Link	Tags	Contact	Ratings	Number of downloads
AAR	131130	ICA-based Automatic Artifact Removal. Note that this extension overloads the "pca.m" Matlab function and can lead to unexpected behaviors of EEGLAB when running ICA.	Download	artifact,ica	G. Gomez-Herrero	★★★★★	3287
Adjust	1.1.1	Adjust plug-in to remove artifacts automatically	Download	ica,artifact	James Charles	★★★☆☆	3540
AMICA	1.5.2	Powerful Amica ICA decomposition plug-in (requires available compiled binary). Related: postAMICAUtility	Download	ica		★★★★★	2273
ANTeepimport	1.13	Import ANT .cnt data and trigger files	Download	import		★★★★☆	3059
ARfitStudio	0.41	Cleans event-related transient artifacts using ARfit (beta)	Download	artifact,preprocessing	Miyakoshi and Mullen	☆☆☆☆☆	530

List of plugins (bolded plugins are installed)

No install s... No topic fil...

- no rating - SASICA v1.3.4 (1521 downloads)
- no rating - PrepPipeline v0.55.3 (1407 downloads)
- ★★★★★ - MFFimport v2.2 (1365 downloads; 2 rating)
- ★★★★★ - **xdimport v1.14 (1363 downloads; 1 rating)**
- no rating - corrmmap v2.02.1 (1267 downloads)
- no rating - **lirfilt v1.03 (1240 downloads)**
- no rating - LIMO v2.0 (1238 downloads)
- no rating - BCI2000import v0.36 (1086 downloads)
- no rating - biopac v1.00 (1039 downloads)
- no rating - loadcurry v2.0 (1005 downloads)
- no rating - PACT v0.20 (984 downloads)
- no rating - VisEd v1.05 (970 downloads)
- no rating - std_envtopo v4.03 (934 downloads)
- no rating - mass_univ v03272017 (927 downloads)
- ★★★★★ - trimOutlier v0.17 (898 downloads; 1 rating)
- ★★★★★ - NihonKoden v1.11 (796 downloads; 1 rating)
- ★★★★★ - **ICLabel v1.1 update available (793 downloads; 1 rating)**
- no rating - loadhd5 v1.1 (754 downloads)
- no rating - BEGICA v1.00 (750 downloads)

Rate this plugin Web documentation Upload new plugin

Tags: import

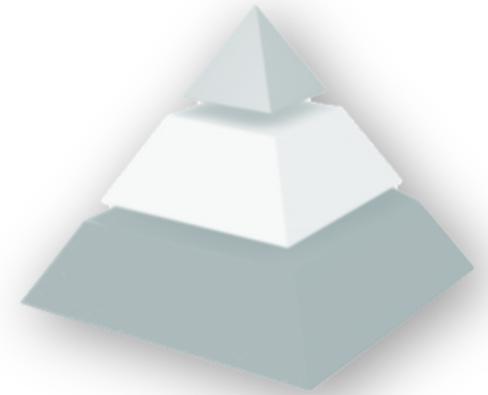
Status: installed

Description of the plugin:

Import files in XDF format saved by the LabRecorder Python program to record LSL streams. This plugin only imports EEG and Marker streams. To import a multi-stream XDF file and sync the different data streams at the same sampling frequency for subsequent processing in EEGLAB, use the Mobilab extension.

Cancel Remove Install/Update

3 levels of functions



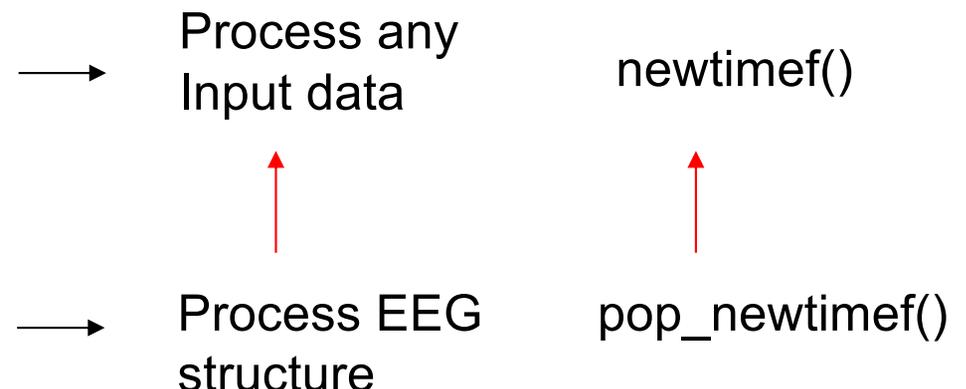
Top: Administrative functions: handle EEG/ALLEEG structures
eeglab(), eeg_checkset(), pop_delset(), ...

Middle: Pop functions: interactive functions using EEG structure
pop_erpimage(), pop_topoplot(), pop_envtopo(), ...

Bottom: Signal processing functions: perform signal processing
erpimage(), topoplot(), envtopo(), ...



Writing EEGLAB plugins

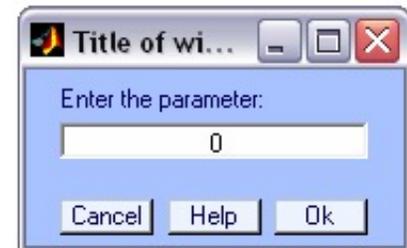
- Assuming that you have a signal processing function called xxxxx → Process any Input data
 - a pop_xxxxx function will interface your signal processing function → Process EEG structure
 - a eegplugin_xxxxx function will add the menu to the main interface (and history etc...)
- newtimef()
- pop_newtimef()
- 



Pop functions

- Called with the EEG structure only `pop_xxxxx(EEG)`, they pop-up a GUI asking for more arguments
- Called with enough arguments, they simply call the signal processing function

```
function [EEG, com] = pop_sample( EEG, param1 );  
  
com = "; % empty history  
if nargin < 2  
    % pop up window if less than 2 arguments  
    result = inputdlg( { 'Enter the parameter:' }, 'Title of window', 1, { '0' } )  
    if length( result ) == 0 return; end;  
  
    param1 = eval( [ '[' result{1} ']' ] ); % the brackets allow to process matlab arrays  
end;  
  
sample( EEG.data, param1); % run sample function  
  
com = sprintf('pop_sample(EEG, %d );', param1); % return history
```



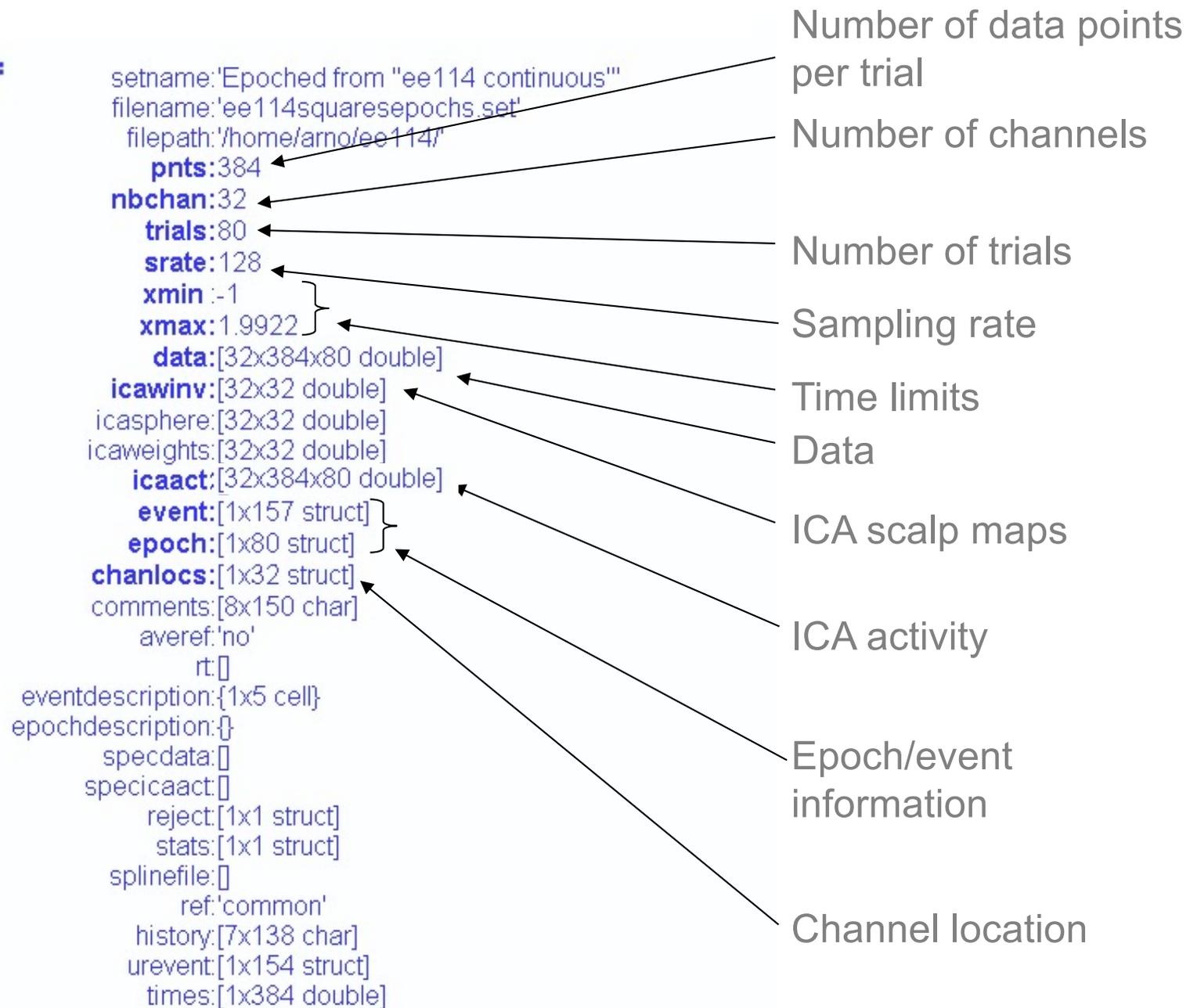
EEGLAB Data Structures

1. EEG
 - root 'dataset' structure
 - .data - the dataset data (2-D, 3-D matrix)
 - .chanlocs - channel locations substructure
 - .event - data events substructure
 - .epoch - data epochs substructure
2. ALLEEG
 - vector of loaded EEG datasets
3. CURRENTSET
 - index in ALLEEG of current EEG dataset
4. STUDY
 - root 'studysset' structure
 - .cluster - component clustering substructure



EEG structure

EEG =



EEG structure

EEG =

```
setname:'Epoched from "ee114 continuous"  
filename:'ee114squaresepochs.set'  
filepath:'/home/arno/ee114/'  
pnts:384  
nbchan:32  
trials:80  
srate:128  
xmin:-1  
xmax:1.9922  
data:[32x384x80 double]  
icawinv:[32x32 double]  
icasphere:[32x32 double]  
icaweights:[32x32 double]  
icaact:[32x384x80 double]  
event:[1x157 struct]  
epoch:[1x80 struct]  
chanlocs:[1x32 struct]  
comments:[8x150 char]  
averef:'no'  
rt:[]  
eventdescription:{1x5 cell}  
epochdescription:[]  
specdata:[]  
specicaact:[]  
reject:[1x1 struct]  
stats:[1x1 struct]  
splinefile:[]  
ref:'common'  
history:[7x138 char]  
urevent:[1x154 struct]  
times:[1x384 double]
```

Number of data points per trial

Number of channels

Number of trials

Sampling rate

Time limits

Data

ICA scalp maps

ICA activity

Epoch/event information

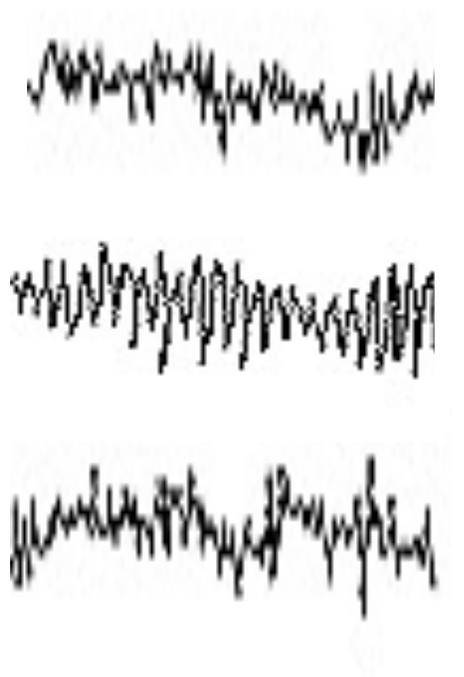
Channel location

The EEG structure can be extended to include new fields

store information for future access

Continuous data

$$\mathbf{EEG.data} = \begin{bmatrix} 2.1 & 3.8 & 4.9 & 5.1 & 4.8 & 3.9 & \dots \\ -1.3 & -2.4 & -0.5 & -0.3 & 1.4 & 2.5 & \dots \\ 5.2 & 4.7 & 3.3 & 1.2 & 0.7 & 1.3 & \dots \end{bmatrix}$$



Data epochs

$$\mathbf{EEG.data} = \begin{bmatrix} 2.1 & 3.8 & 4.9 & 5.1 & 4.8 & 3.9 & \dots \\ -1.3 & -2.4 & -0.5 & -0.3 & 1.4 & 2.5 & \dots \\ 5.2 & 4.7 & 3.3 & 1.2 & 0.7 & 1.3 & \dots \end{bmatrix}$$

Trials 1: EEG.data(:, :, 1)

$$\begin{bmatrix} 2.1 & 3.8 & 4.9 & 5.1 & 4.8 & 3.9 & \dots \\ -1.3 & -2.4 & -0.5 & -0.3 & 1.4 & 2.5 & \dots \\ 5.2 & 4.7 & 3.3 & 1.2 & 0.7 & 1.3 & \dots \end{bmatrix}$$

Trials 2: EEG.data(:, :, 2)

$$\begin{bmatrix} 2.1 & 3.8 & 4.9 & 5.1 & 4.8 & 3.9 & \dots \\ -1.3 & -2.4 & -0.5 & -0.3 & 1.4 & 2.5 & \dots \\ 5.2 & 4.7 & 3.3 & 1.2 & 0.7 & 1.3 & \dots \end{bmatrix}$$

Trials 3: EEG.data(:, :, 3)

Plot ERP for your data

```
>> figure; plot(mean(EEG.data,3)');
```

```
>> figure; plot(EEG.times, mean(EEG.data,3)');
```



eegplugin functions

- eegplugin_erp.m function in “plugins” folder of EEGLAB

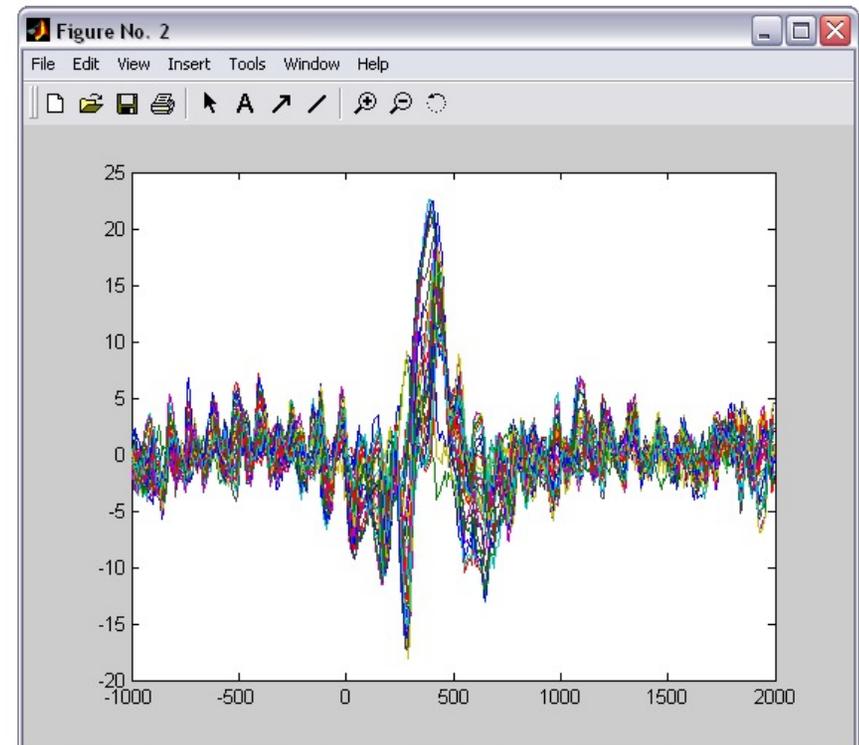
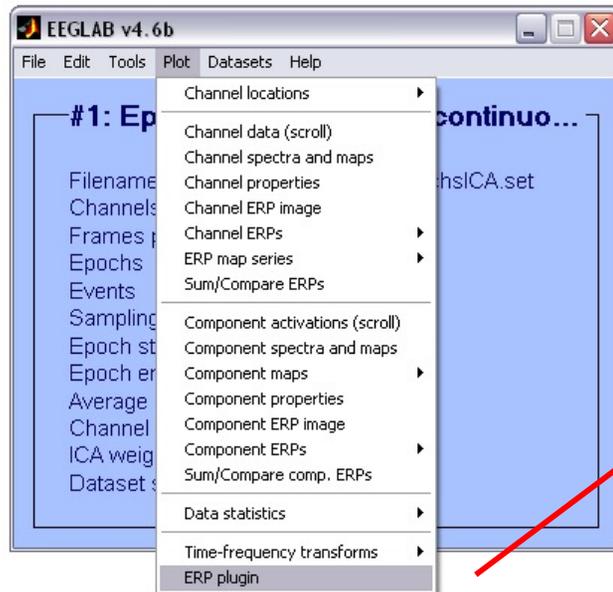
```
% eegplugin_erp() - plot ERP plugin  
  
function eegplugin_erp( fig, try_strings, catch_strings)  
  
% create menu  
plotmenu = findobj(fig, 'tag', 'plot'); % find plot menu  
  
% create submenu  
uimenu( plotmenu, 'label', 'ERP plugin', ...  
'callback', 'figure; plot(EEG.times, mean(EEG.data,3));');
```



To test, load file in sample_data folder eeglab_data_epochs_ica.set

eegplugin functions

```
>> eeglab
eeglab: adding "BIOSIGv0.86" plugin
eeglab: adding "eepimport1.02" plugin (see >> help eegplugin_eepimport)
eeglab: adding "bva_io1.30" plugin (see >> help eegplugin_bva_io)
eeglab: adding "ctfimport1.01" plugin (see >> help eegplugin_ctfimport)
eeglab: adding "dipfit2.0" plugin (see >> help eegplugin_dipfit2_0)
eeglab: adding plugin function "eegplugin_erp"
eeglab: adding "fmrib1.2b" plugin (see >> help eegplugin_fmrib)
eeglab: adding "icaclust1.00" plugin (see >> help eegplugin_icaclust)
eeglab: adding "iirfilt1.0" plugin (see >> help eegplugin_iirfilt)
eeglab: adding "loreta1.0" plugin (see >> help eegplugin_loreta)
eeglab: adding "newtimefreq1.00" plugin (see >> help eegplugin_newtimefreq)
>>
```



PCA plugin

```
function vers = eegplugin_pca(fig, trystrs, catchstrs)

    vers = 'pca1.00';

    % find tools menu
    menu = findobj(fig, 'tag', 'tools');

    % PCA command
    cmd = [ '~ EEG.icawinv = runpca(EEG.data(:,:));' ];
    cmd = [ cmd 'EEG.icaweights = pinv(EEG.icawinv);' ];
    cmd = [ cmd 'EEG.icasphere = eye(EEG.nbchan);' ];

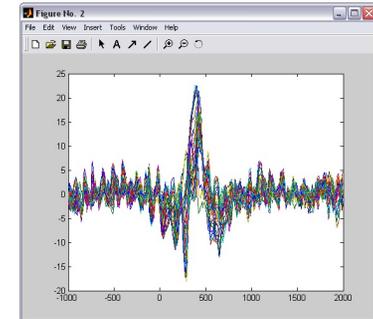
    % create menu
    uimenu( menu, 'Label', 'Run PCA', 'CallBack', cmd, 'separator', 'on');
```

'import data' -> *File > import data menu*
'import epoch' -> *File > import epoch menu*
'import event' -> *File > import event menu*
'export' -> *File > export*
'tools' -> *tools menu*
'plot' -> *plot menu*



Exercise

Write a plugin to plot ERPs



1. Copy and paste code from PDF
2. Save `eegplugin_erp.m` in the *plugins* folder of EEGLAB
3. Restart EEGLAB
4. Load epoched EEGLAB dataset (*eeglab_data_epochs_ica.set* from *sample_data* subfolder of EEGLAB)
5. Use plugin menu

