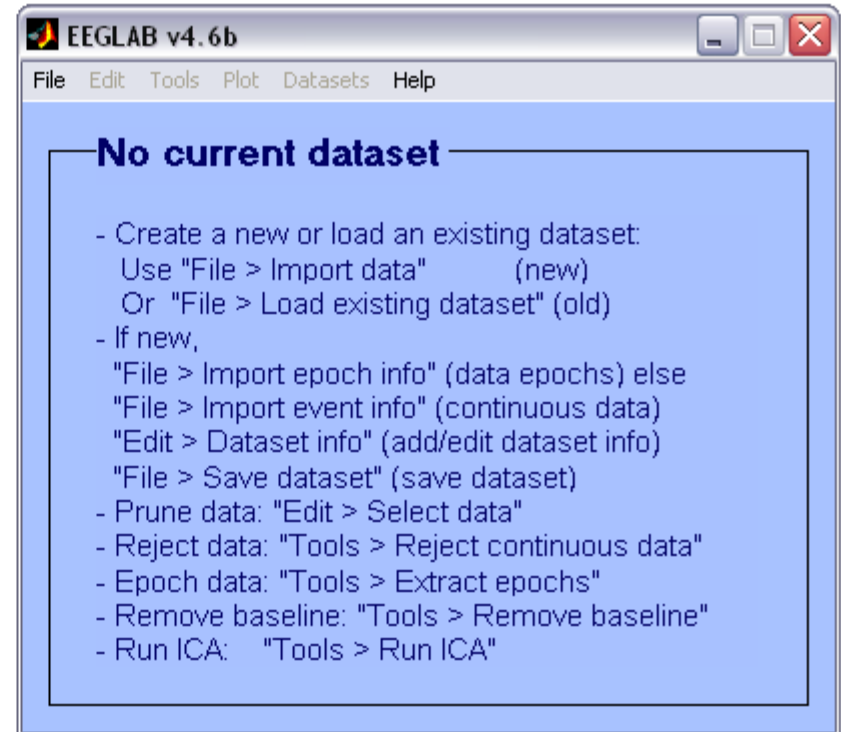


EEGLAB Plugins/Extensions

Starting EEGLAB

```
>> eeglab
eeglab: options file is /Volumes/donnees/data/STUDYst
Adding path to all EEGLAB functions
Adding path to eeglab/external/bioelectromagnetism_lig
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 ee
EEGLAB: adding "bdfimport" plugin (see >> help eegplu
EEGLAB: adding "brainmovie0.1b" plugin (see >> help e
EEGLAB: adding "ctfimport1.03" plugin (see >> help ee
EEGLAB: adding "dipfit2.2" plugin (see >> help eegplugi
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)
>>
```



EEGLAB plugins

eepimport1.02	Data importing for EEprobe data (Oostenveld & ANT company)
bva_io1.30	Brain vision analyzer import/export plugin (Widmann & Delorme)
ctfimport1.01	MEG CTF import plugin (Carver, Weber & Delorme)
dipfit2.0	4-shell and BEM (Oostenveld & Delorme)
fmrib1.2b	Removal of artifact from simultaneously EEG/fMRI recording (Niazi)
icaclust1.00	Clustering ICA components (Serby, Delorme, Makeig)
iirfilt1.0	Non-linear IIR filtering (Pozdin)
loreta1.0	Interface to LORETA-KEY (Delorme)
newtimefreq1.00	Time-freq. decomposition (Delorme)

Better than FIR
Coregistration...

Matlab toolboxes interfaced

BIOSIG	Data importing for rare data binary format (Schloegl)
Fieldtrip	Source localization and time-freq. decompositions (Oostenveld)
ICALAB	20 ICA algorithms (automatically detected by EEGLAB)
SPM2	Spatial normalization of anatomical MRI

Plugin list process - SCCN

Plugin list process - SCCN

92.149.236.22 talk for this ip address log in

page discussion view source history

Plugin list process

Plugin name	Version	Short plugin description	Link	Comments
brainmovie	0.1	Brainmovies (command line only)	Download	User comments
corrmap	1.03	Import BIOPAC data files	Download	User comments
eeg_toolbox	1.0	Interface EEG toolbox functions for ERP peak detection	Download	User comments
ERPLABfilters	1.00	Interface ERPLAB filters (requires seperate ERPLAB instalation)	Download	User comments
fMRIB	1.21	Remove fMRI artifacts from EEG	Download	User comments
MP_clustering	1.00	Measure projection clustering of ICA components	Download	User comments
MutualInfoClustering	1.00	Mutual information clustering	Download	User comments
StudyEnvtopo	0.9	Add envtopo capabilities to STUDY	Download	User comments
VisEd	1.04	Add/Edit dataset events	Download	User comments
ADJUST	1.21	Automatic artifact rejection	Download	User comments
iirfilt	1.02	Non linear filtering	Download	User comments
loreta	1.0	Export and import data to/from LORETA software	Download	User comments
BERGEN	1.1	Removal of fMRI-related gradient artifacts from simultaneous EEG-fMRI data	Download	User comments

Add your plugin to the list

You may add your plugin to the list so users can download it automatically from within EEGLAB. There are 5 tabs:

- Plugin name:** this tab should contain the abbreviated name of your plugin and if necessary a link to the plugin documentation. The plugin documentation may be stored on this wiki.
- Version:** this tab should contain the version of your plugin. The version listed on this page and the one made available in the eegplugin_xxx.m file must be consistent. This allows EEGLAB to automatically check for newer versions of your plugin.
- Short plugin description:** this tab should contain a short plugin description (no more than one line). Additional documentation may be provided as a link in tab 1.

Swartz Center for Computational Neuroscience

home

- SCCN web site
- EEGLAB Wiki
- MoBI Lab Wiki
- SCCN Wiki Home

eeglab wiki pages

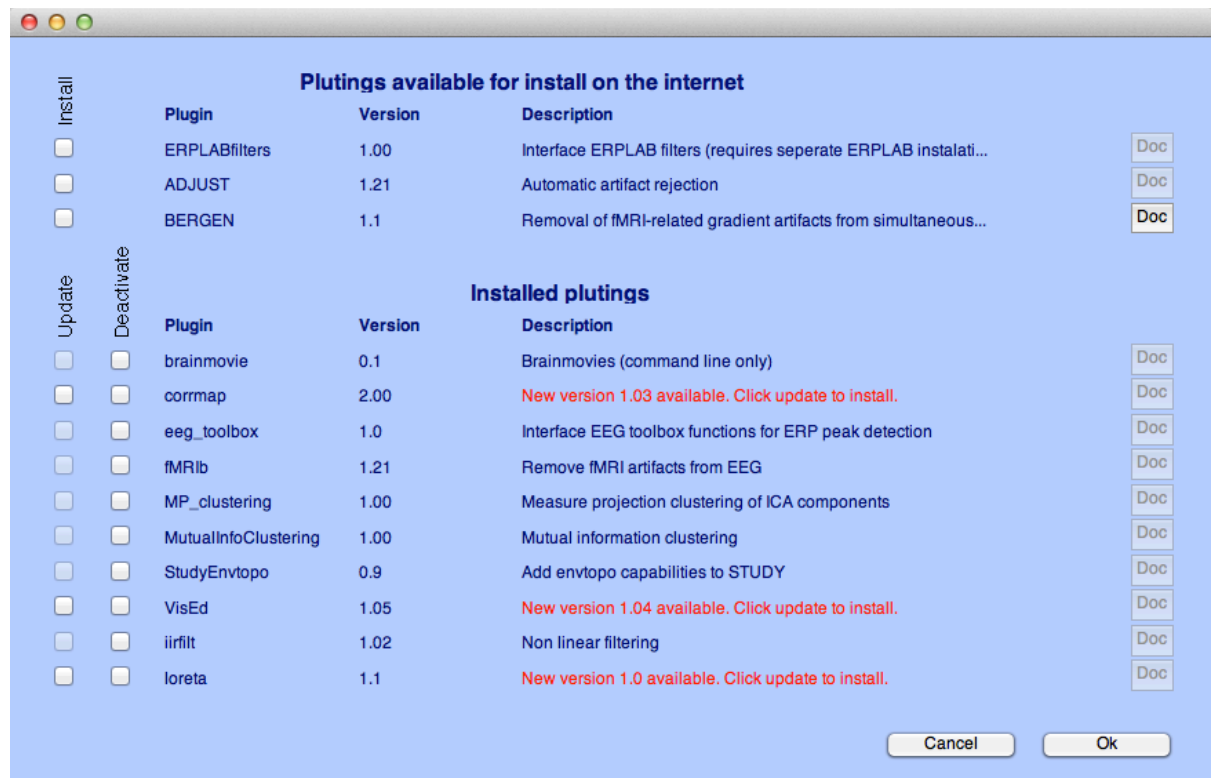
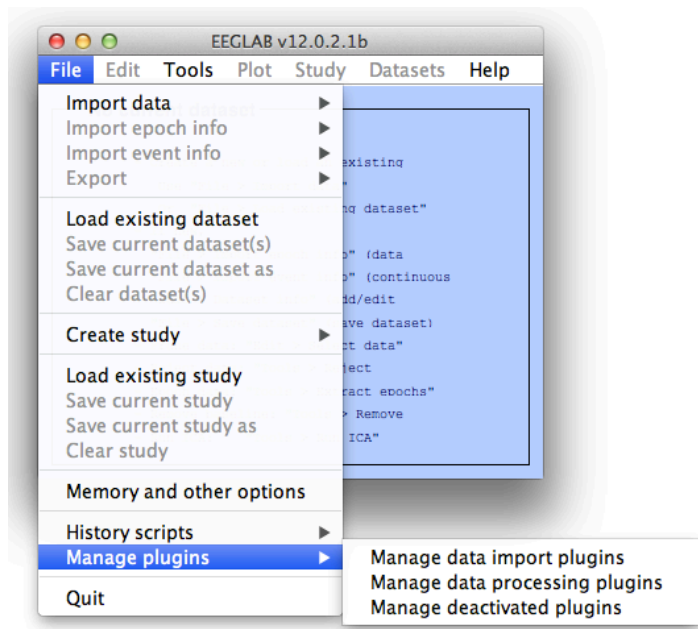
- EEGLAB web page
- EEGLAB Wiki
- EEGLAB Tutorial
- Online EEGLAB Workshop
- Download EEGLAB
- Revision history
- Help EEGLAB

sccn toolboxes

- EEGLAB
- NFT
- BCILAB
- SIFT
- MoBILAB
- MPT

wiki tools

- Sandbox
- Basic Wiki Syntax
- Wiki Help
- New Users



Writing EEGLAB plugins

- Assuming that you have a signal processing function called xxxxx → Process any Input data Timef()
 - a pop_xxxxx function will interface your signal processing function → Process EEG structure Pop_timef()
 - a eegplugin_xxxxx function will add the menu to the main interface (and history etc...)
-
- ```
graph TD; A[xxxxx] --> B[Process any Input data]; B --> C[Timef()]; D[pop_xxxxx] --> E[Process EEG structure]; E --> F[Pop_timef()]; E --> B; F --> C;
```

# 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

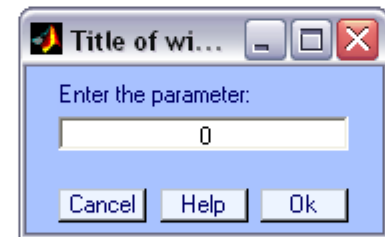
```
function [EEG, com] = pop_sample(EEG, param1);

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
```



# eegplugin functions

- eegplugin\_xxxx function

```
% eegplugin_erp() - plot ERP plugin
```

```
function eegplugin_erp(fig, try_strings, catch_strings);
```

```
% create menu
```

```
plotmenu = findobj(fig, 'tag', 'plot'); % find plot menu
```

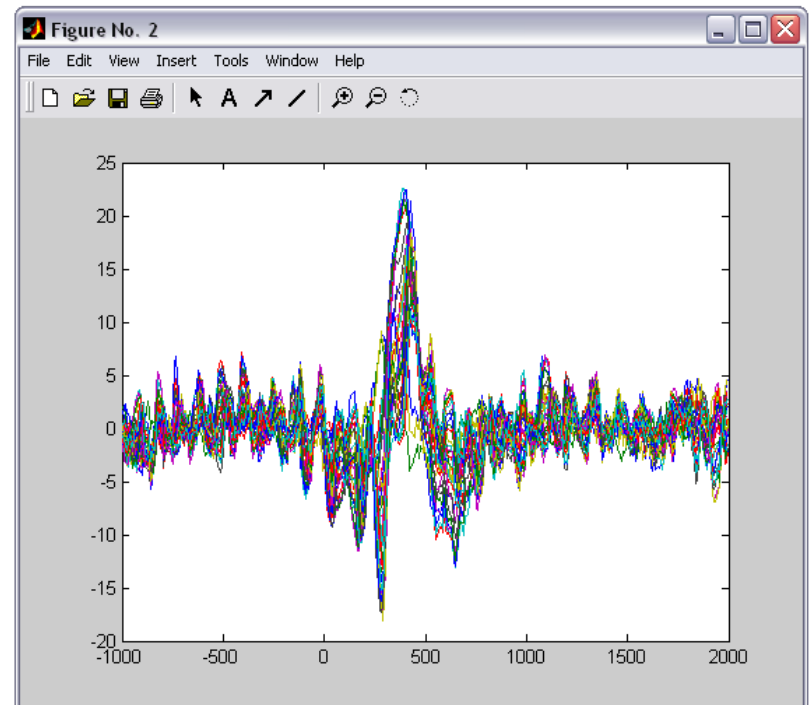
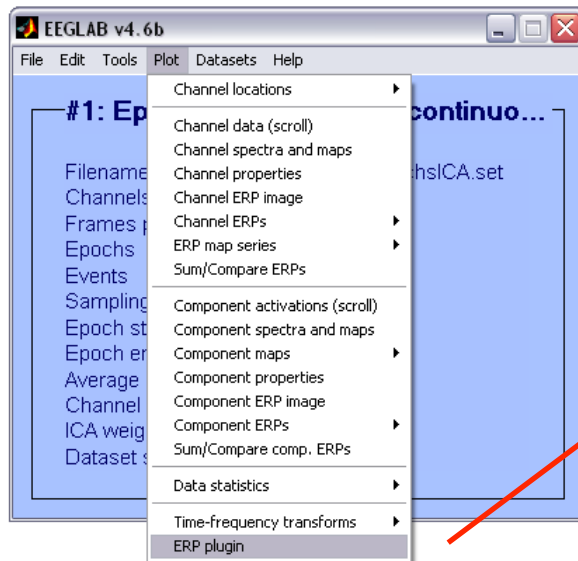
```
% create submenu
```

```
uimenu(plotmenu, 'label', 'ERP plugin', ...
 'callback', 'figure; plot(EEG.times, mean(EEG.data,3));');
```



# 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 "icacust1.00" plugin (see >> help eegplugin_icacust)
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_ne
>>
```



# PCA plugin

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

```
vers = 'pca1.00';
if nargin < 3, error('eegplugin_pca requires 3 arguments'); end;
```

```
% add icaclust folder to path
if ~exist('eegplugin_pca')
 p = which('eegplugin_pca');
 p = p(1:findstr(p,'eegplugin_pca.m')-1);
 addpath(p);
end;
```

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

```
% PCA command
```

```
cmd = ['[tmp1 EEG.icawinv] = runpca(EEG.data(:,:));'];
cmd = [cmd 'EEG.icaweights = pinv(EEG.icawinv);'];
cmd = [cmd 'EEG.icasphere = eye(EEG.nbchan);'];
cmd = [cmd 'clear tmp1;'];
```

```
% 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

# Submit plugin


[http://sccn.ucsd.edu/eeglab/plugin\\_uploader/upload\\_form.php](http://sccn.ucsd.edu/eeglab/plugin_uploader/upload_form.php)

## EEGLAB Plug-in Upload

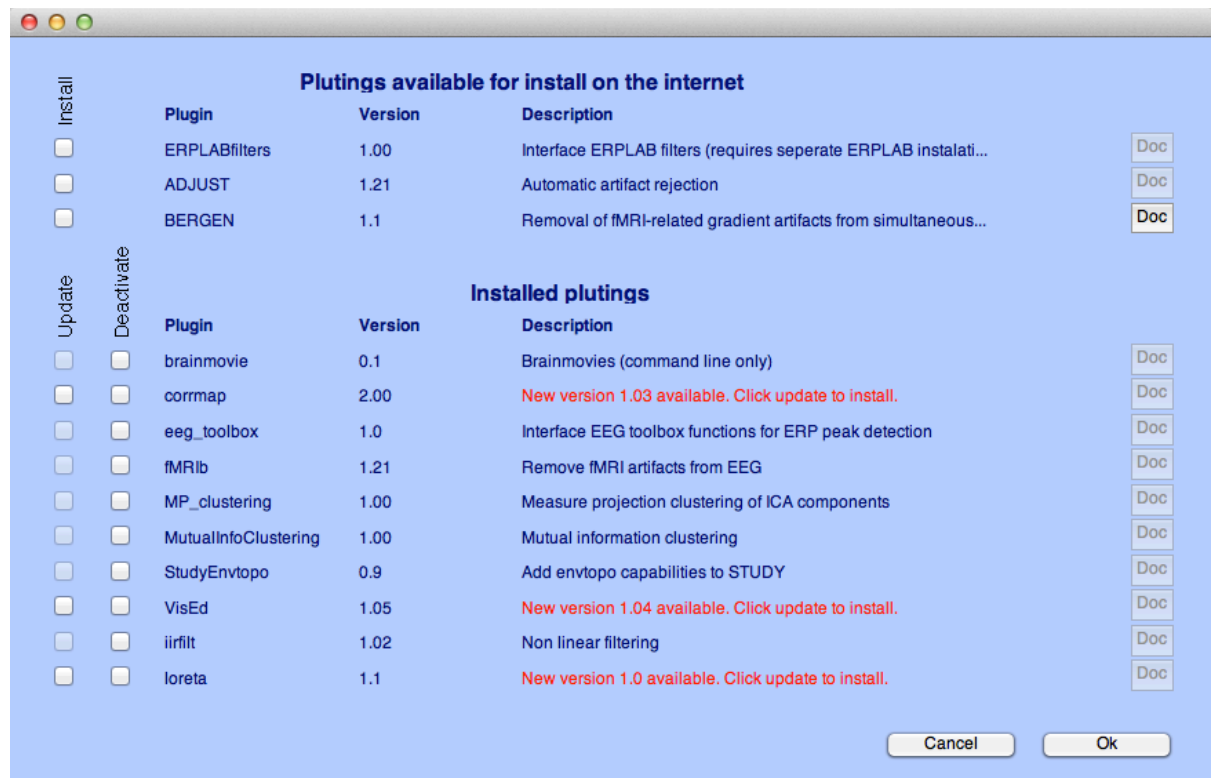
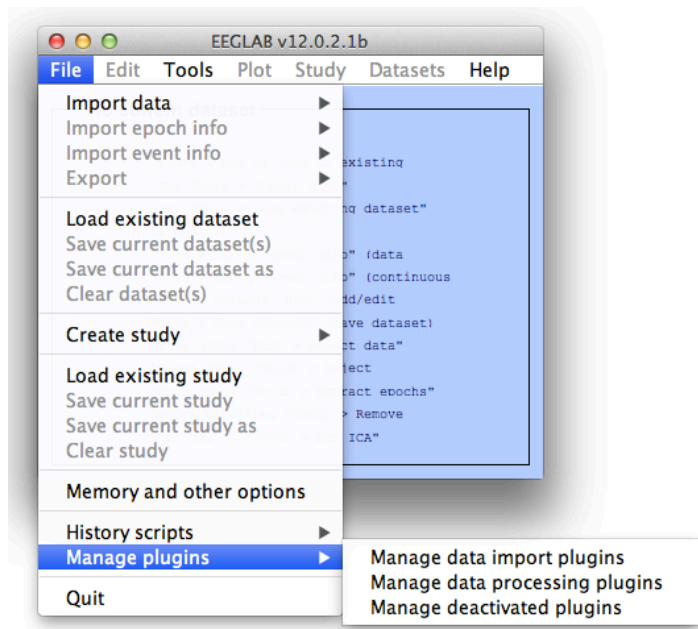
Your name:  
Your email address:  
Re-enter email address:  
Your institution (name, city, country):  
Plug-in description (20 words max):  
Version of the plug-in (e.g. 1.3):  
Documentation URL (web address):  
ZIP file URL (if more than 5Mb):  
or ZIP file to upload (Max 5Mb):

No file selected.

93628366154

  
stop spam.  
read books.

[Privacy & Terms](#)



# EEGLAB documentation

**EEGLAB Home Page**

[sccn.ucsd.edu/eeglab/](http://sccn.ucsd.edu/eeglab/)

**EEGLAB Tutorial Index**

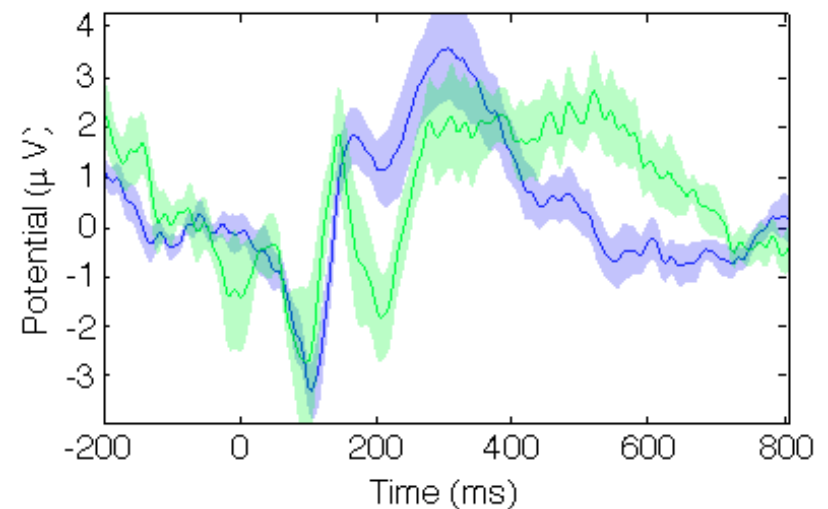
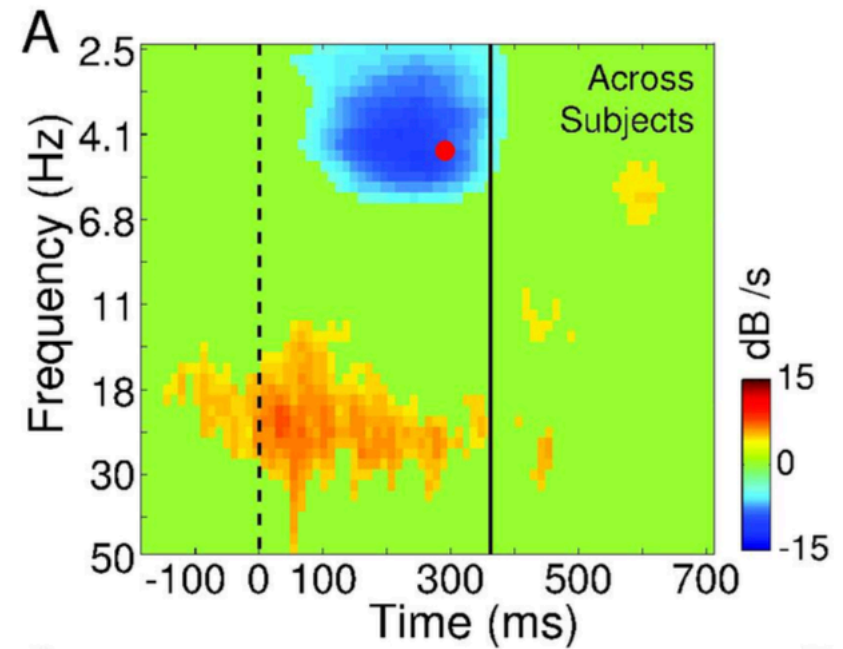
[sccn.ucsd.edu/wiki/EEGLAB](http://sccn.ucsd.edu/wiki/EEGLAB)

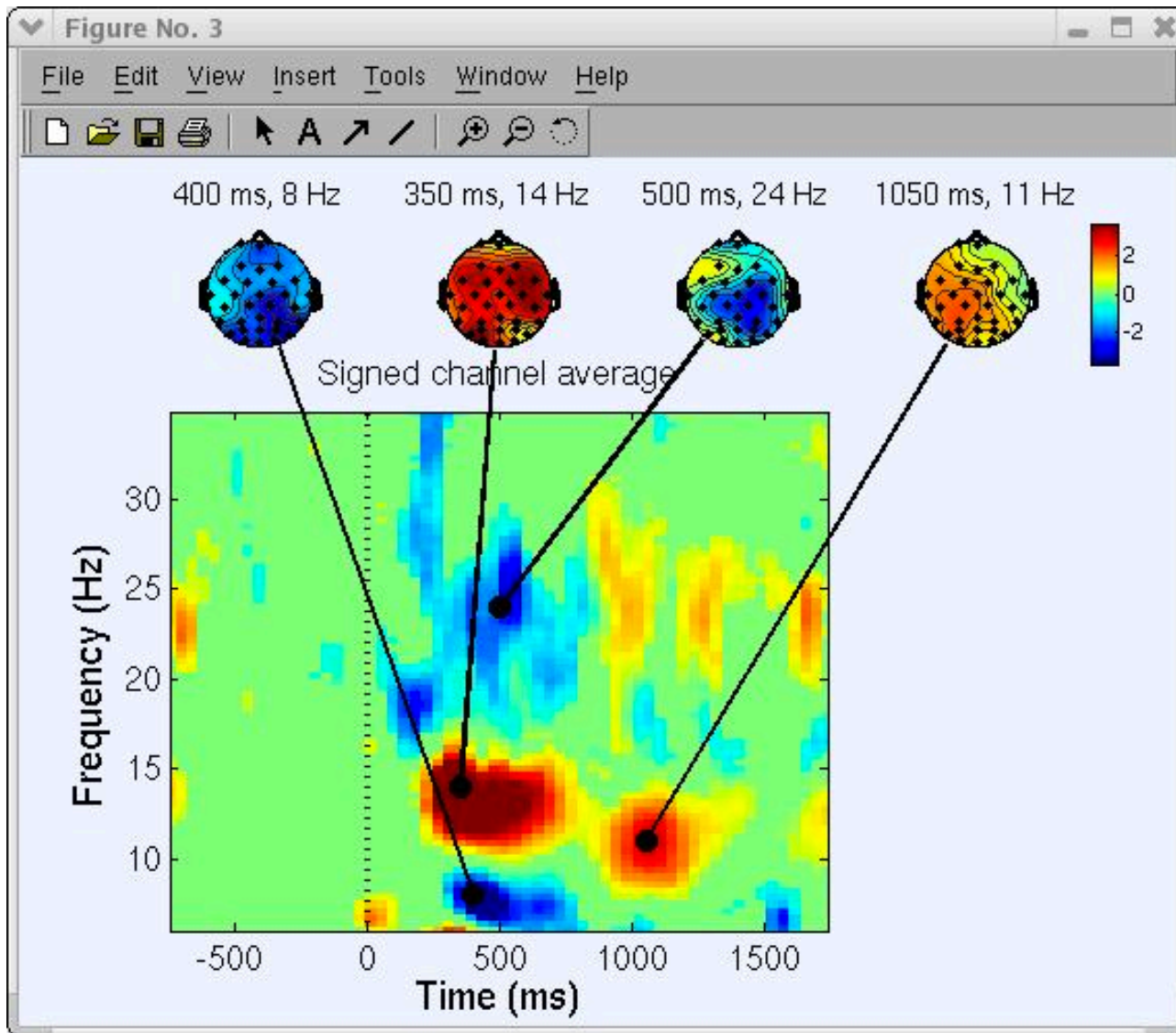
- 200 pages of tutorial (including “how to” for plugins) WEB or PDF
- Function documentation (next slide)
- Send questions to the mailing list [eeglablist@sccn.ucsd.edu](mailto:eeglablist@sccn.ucsd.edu)  
(or search mailing list archive using google)
- Bug submission <http://sccn.ucsd.edu/eeglab/bugzilla>
- Email us (suggestions) [eeglab@sccn.ucsd.edu](mailto:eeglab@sccn.ucsd.edu)
- Workshop with practicum every year

# Basic Scripting in EEGLAB

# Why scripting?

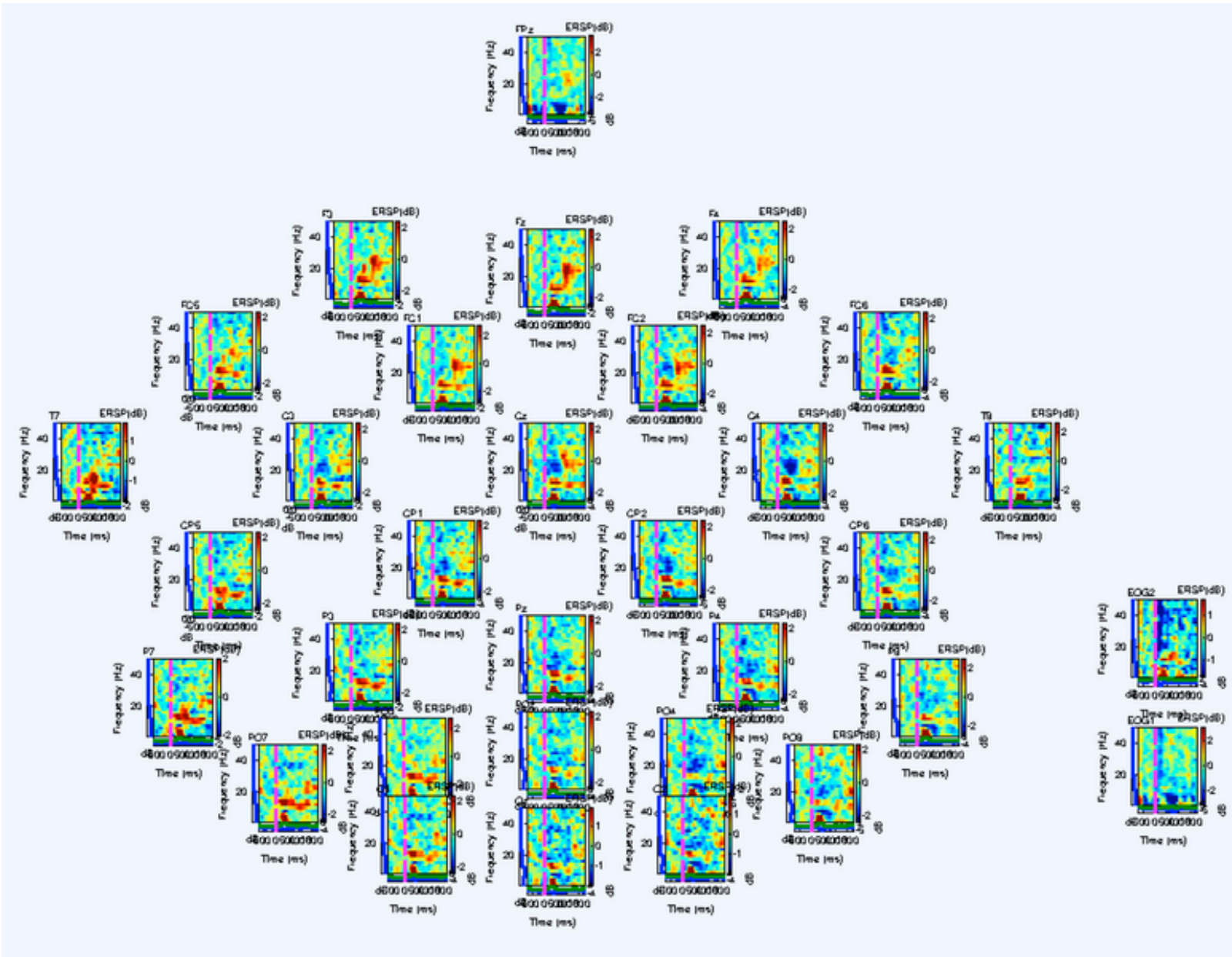
- `corrimage.m` →
- `eeg_regepochs.m`
- `eegmovie.m`
- `fastregress.m`
- `fieldtrip2eeglab.m`
- `fillcurves.m` →
- `compile_eeglab.m`





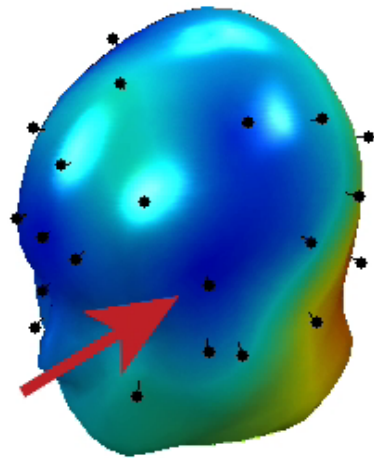
Tftopo function



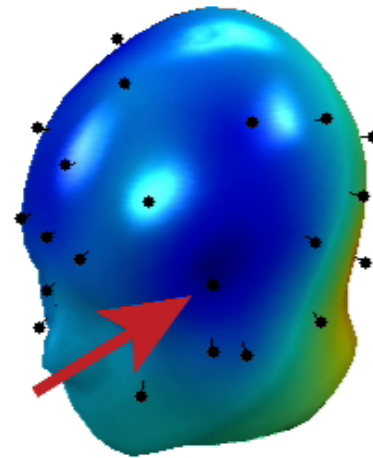


Metaplotopo function

High Acc.

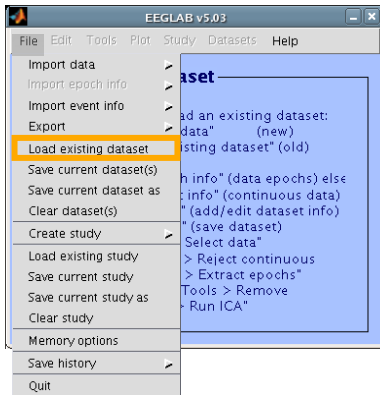


Low Acc.



eegmovie

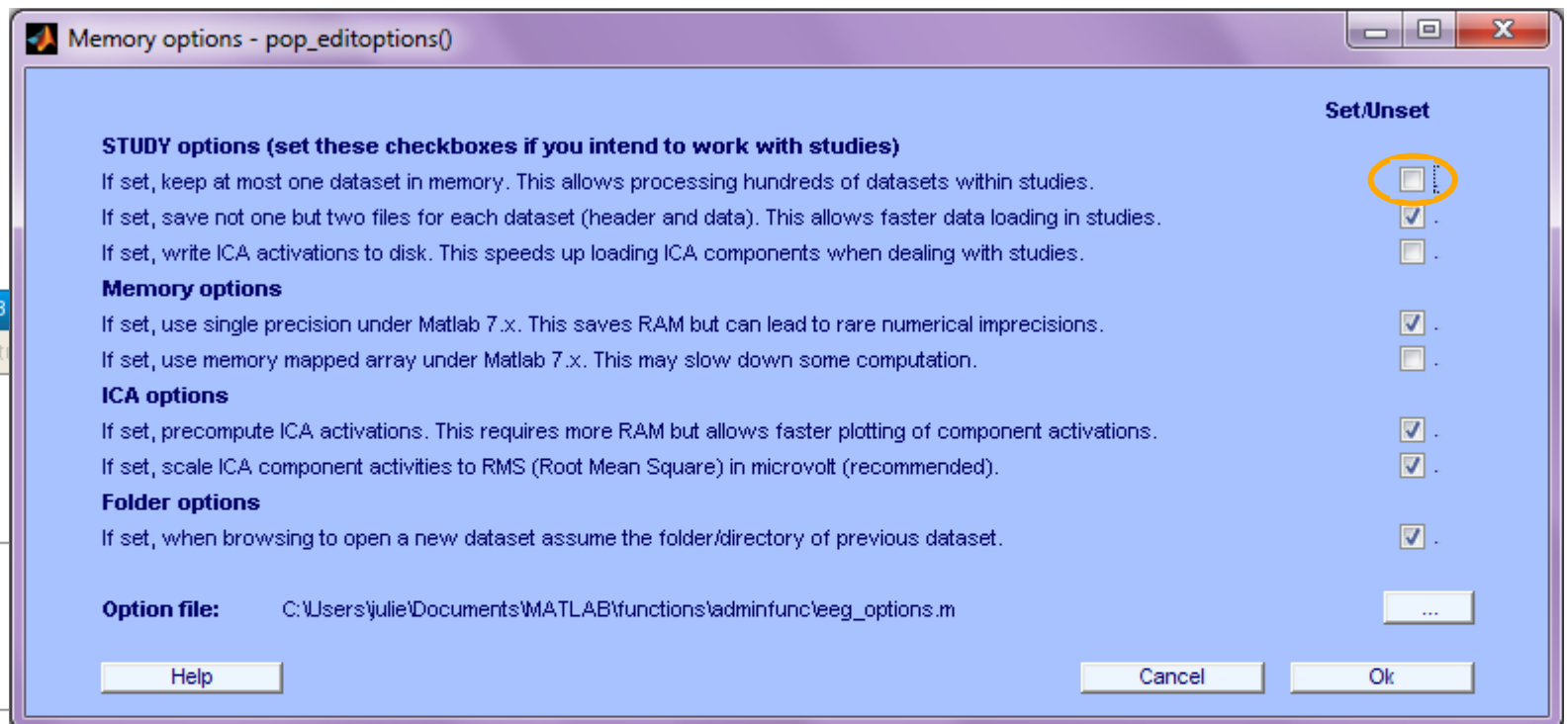
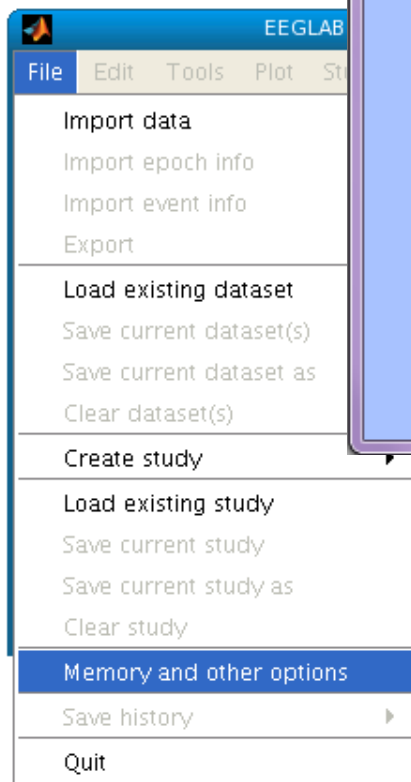
# Command line tools



(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
- Custom processing...

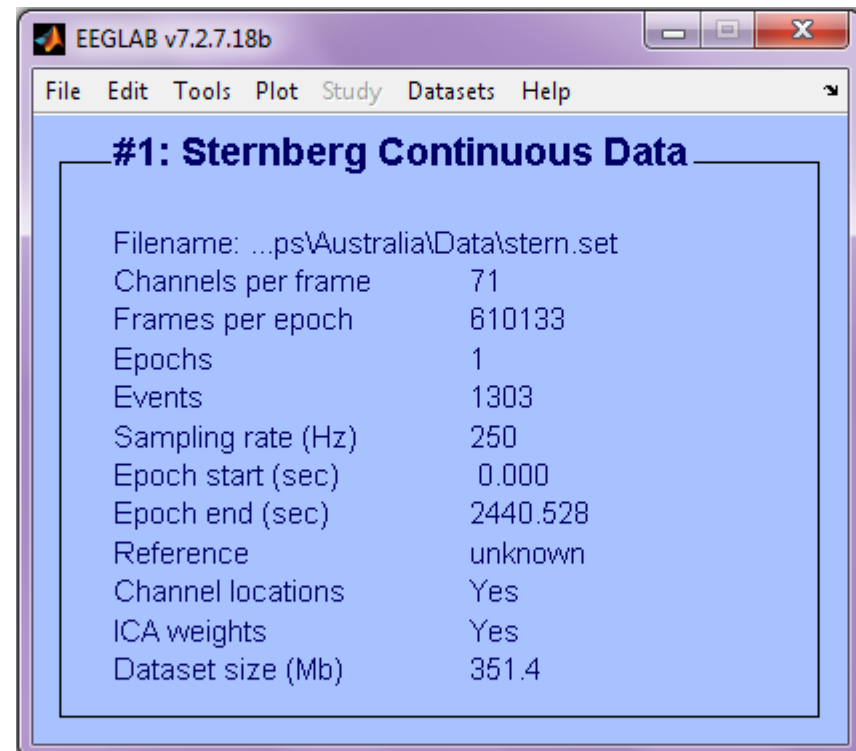
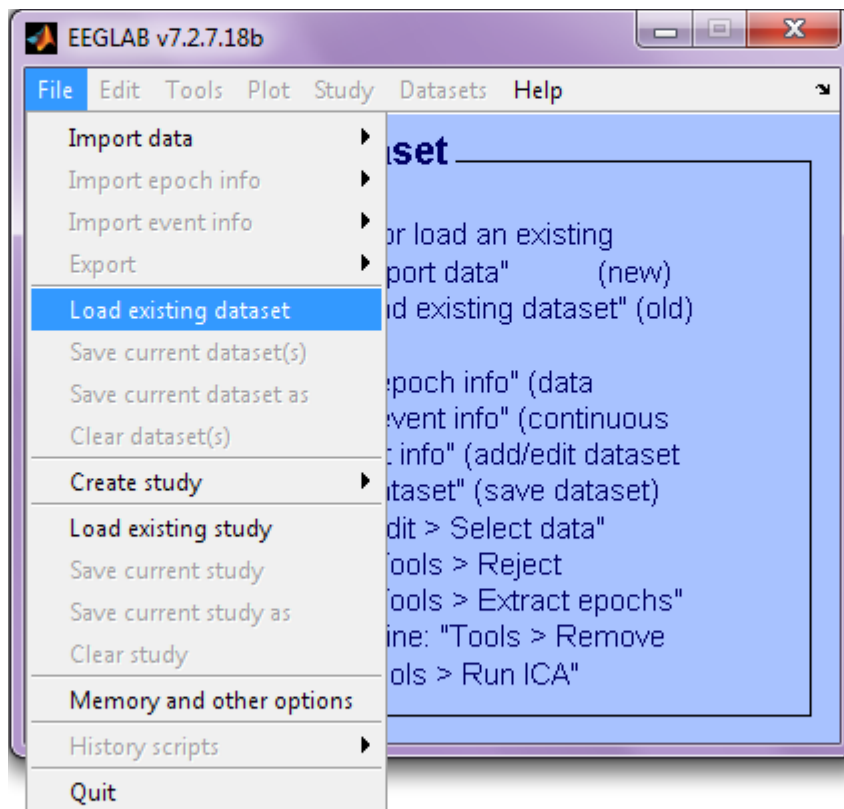
# Memory options



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

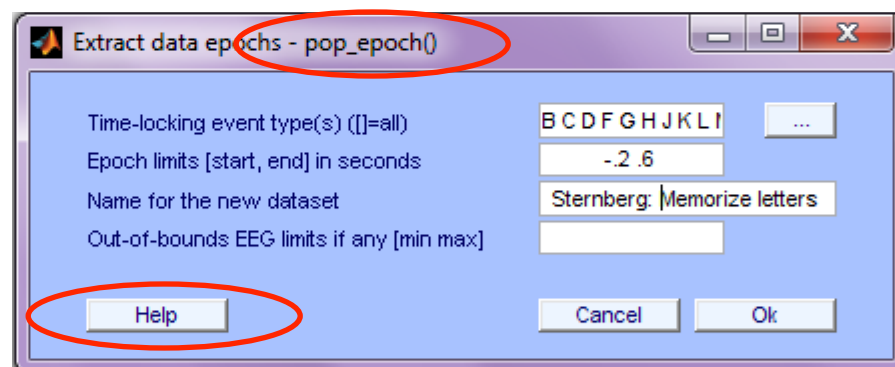
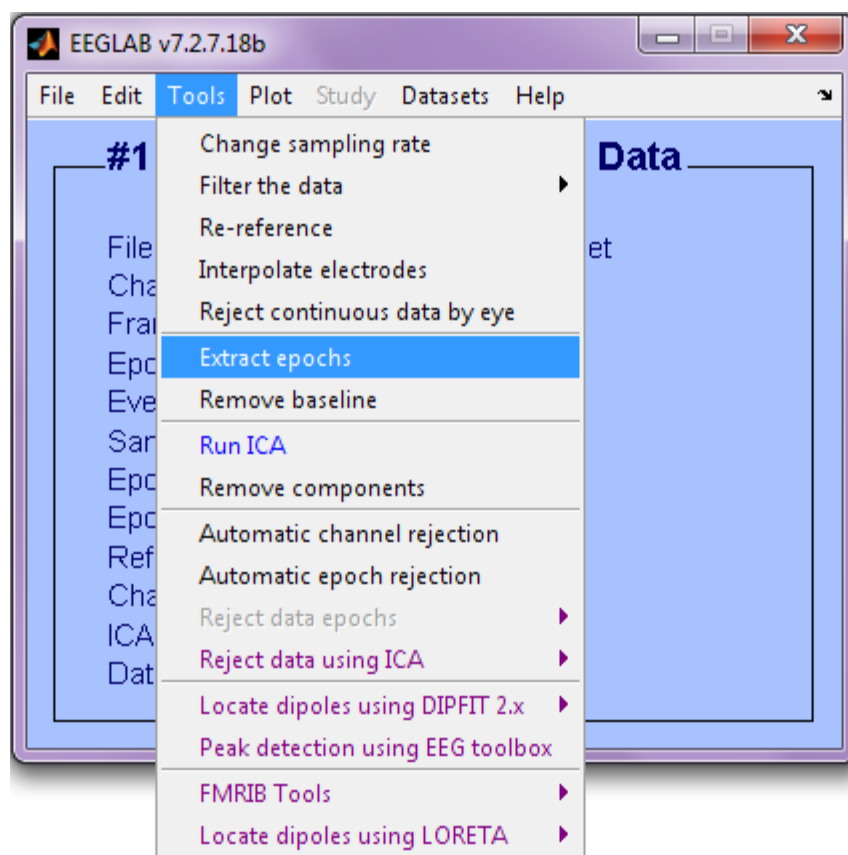
# Create a script from 'eegh' output

**Start by loading a continuous dataset**



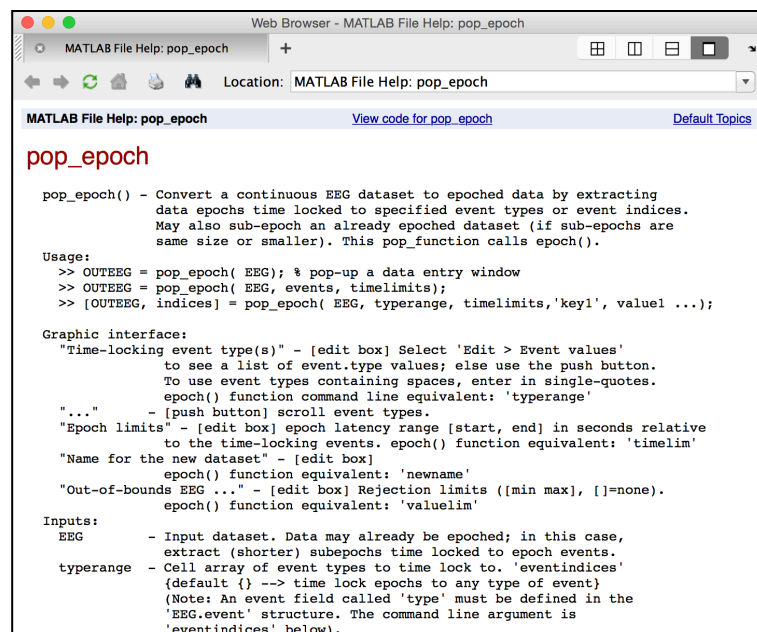
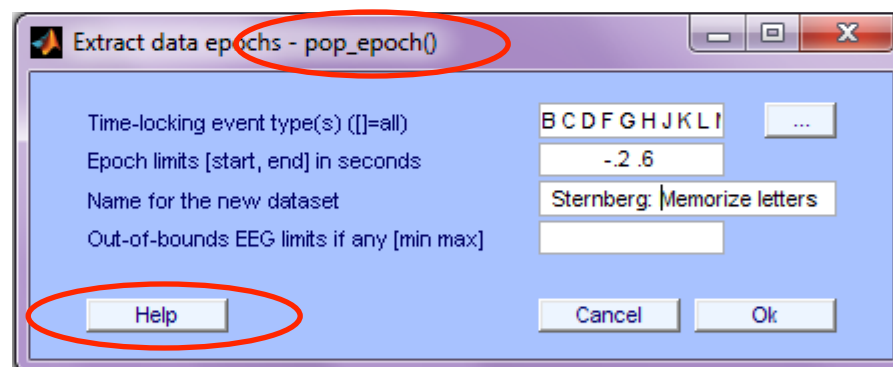
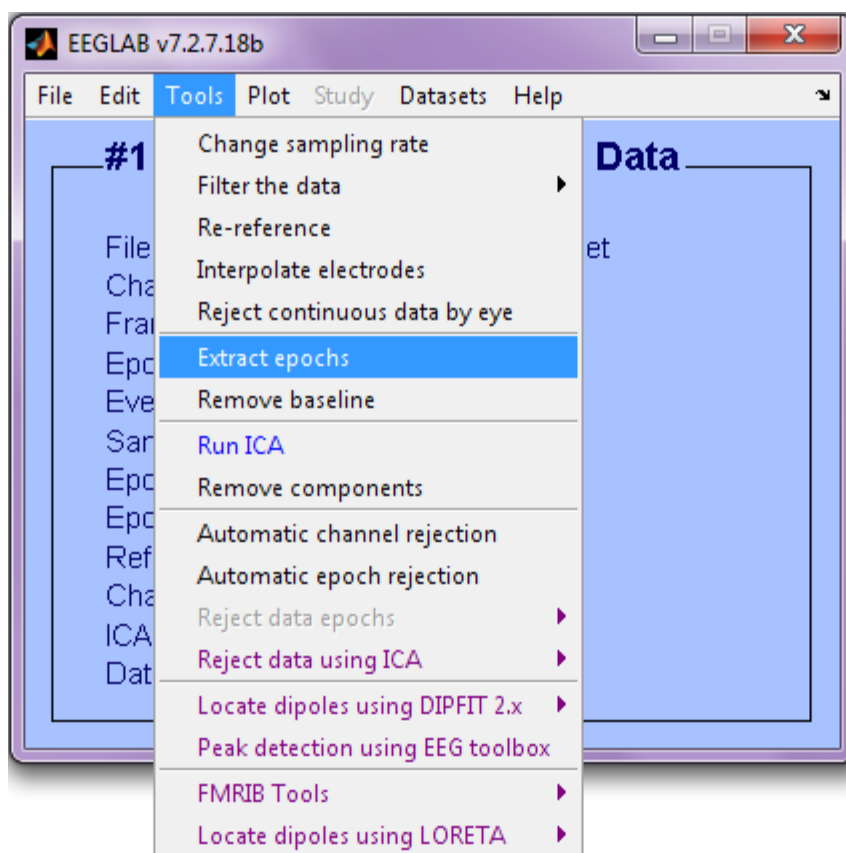
# Create a script from 'eegh' output

## Epoch on Memorize letters



# Create a script from 'eegh' output

## 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 dialog box for 'Component ERP image -- pop\_erpimage()' is displayed, showing various settings for plotting an IC ERP image.

**Component ERP image -- pop\_erpimage()**

**Component(s)**: 3  
**Project to channel #**:  
**Smoothing**: 10  
**Downsampling**: 1  
**Time limits (ms)**: -200 596

**Figure title**:  
 Plot scalp map  
 Plot ERP  
 Plot colorbar

**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  
 Don't plot values

**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.)

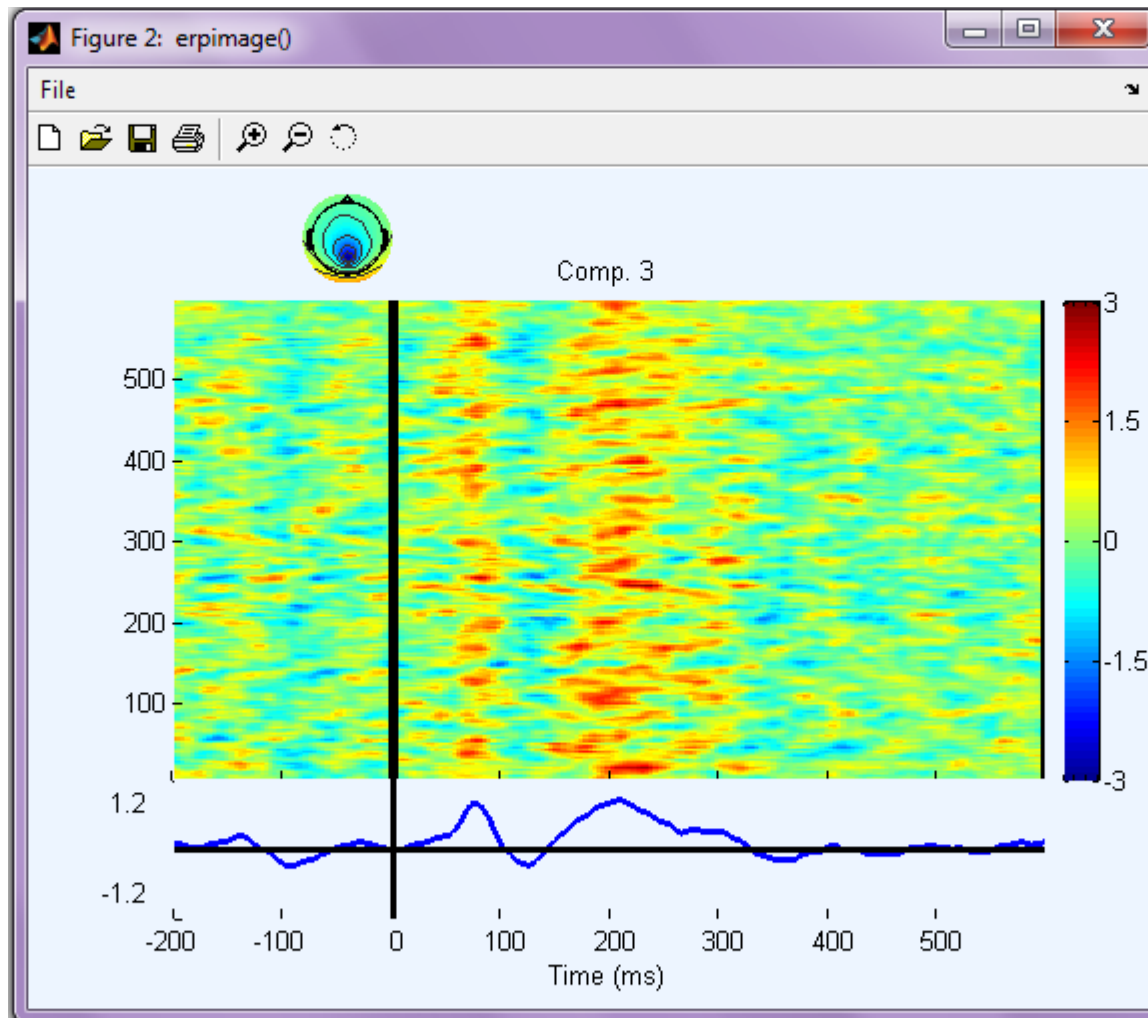
**Other options**

**Plot spectrum (minHz maxHz)**:  
**Baseline ampl. (dB)**:  
**Mark times (ms)**:  
**More options (see >> help erpimage)**:

**Help** **Cancel** **Ok**



# Create a script from 'eegh' output



Retrieve commands from eegh

**Write a script to do this:**

```
>> eegh
```

# Retrieve commands from eegh

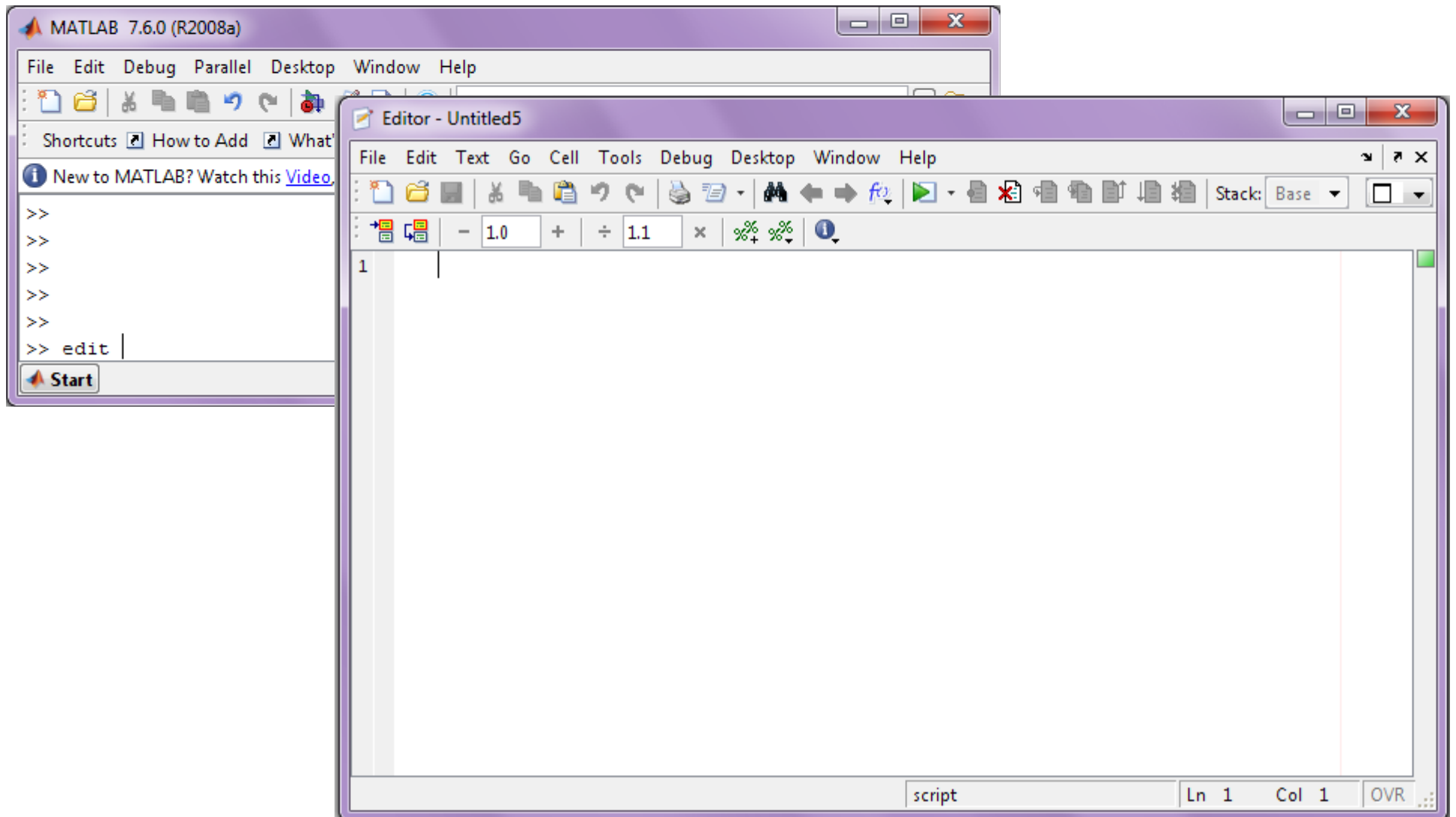
```
>> eegh
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;

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);

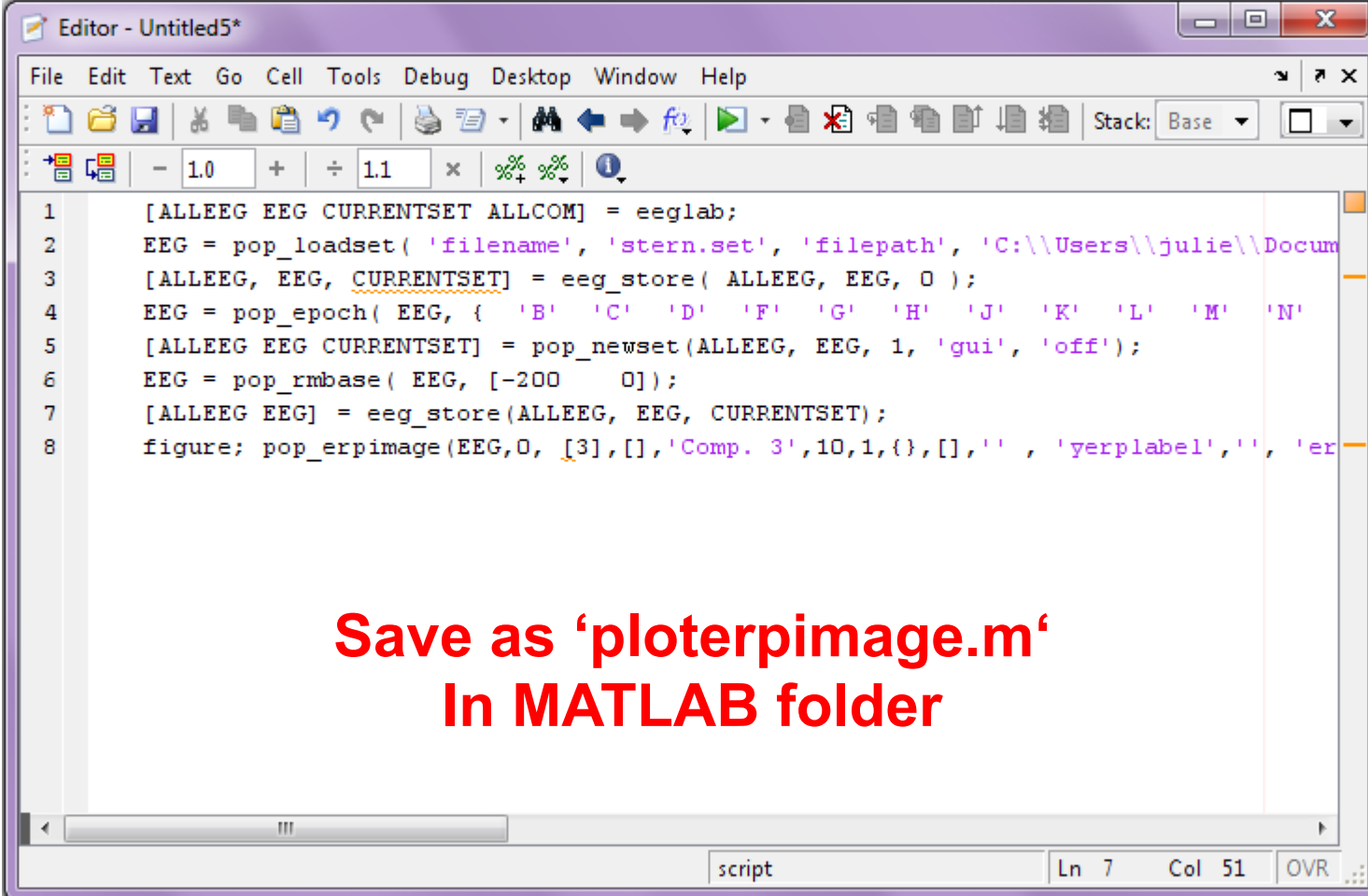
figure; pop_erpimage(EEG,0, [3],[], 'Comp. 3', 10, 1, {},
[], '', 'yerplabel', '', 'erp', 'on', 'cbar', 'on', 'topo',
{mean(EEG.icawinv(:, [3]), 2) EEG.chanlocs EEG.chaninfo });
```

# Create a Matlab script



# Create a Matlab script

**Copy and paste from Matlab window:**



The screenshot shows a Matlab script editor window titled "Editor - Untitled5\*". The window contains 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, there is a red text overlay that reads:

**Save as 'ploterpimage.m'  
In MATLAB folder**

The status bar at the bottom of the window shows "script", "Ln 7", "Col 51", and "OVR".

# Run your new script

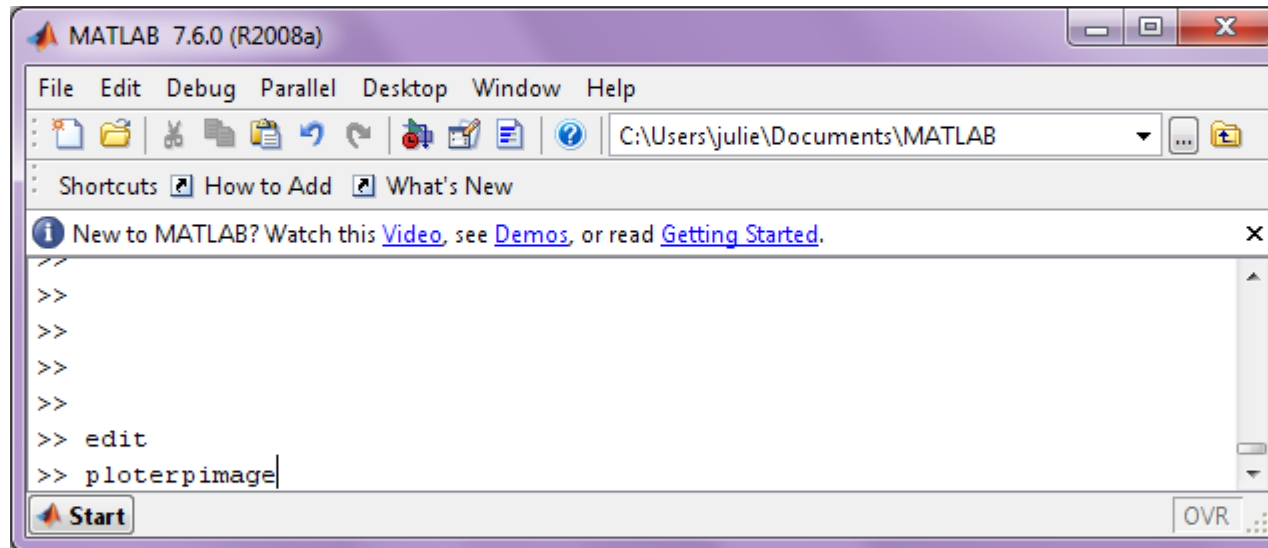


Figure problem



solution



For publication, export as eps (postscript) and edit under postscript editor

For crashes, freezes, etc... use a non Open-GL renderer

At startup type in `set(0,'defaultfigurerenderer','zbuffer' )` or `set(0,'defaultfigurerenderer','painter' )`. Note that these cannot handle transparency and 3-D graphics or type “opengl software”

To export figures for publication, use .eps format (postscript) and edit for instance with adobe illustrator. Use “`set(gcf, 'renderer', 'painter')`” before exporting complex figures. Use the “plot2svg” matlab toolbox to export figure for transparency (Google “plot2svg” – it is the first hit).

# Exercise page 1

```
>> eeglab
```

```
% load dataset stern_125.set
% 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
```