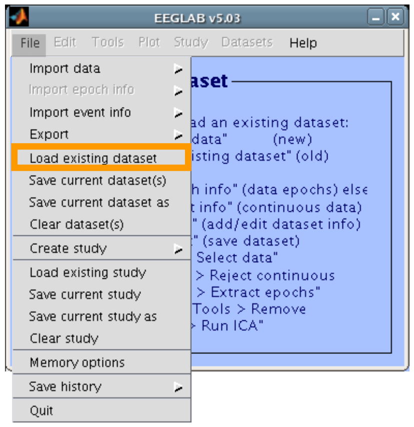
Command line tools



(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…

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...

Using EEGLAB history for basic scripting

Task 1

Create a script from ‘eegh’ output

Task 2

Adapt your script with variables

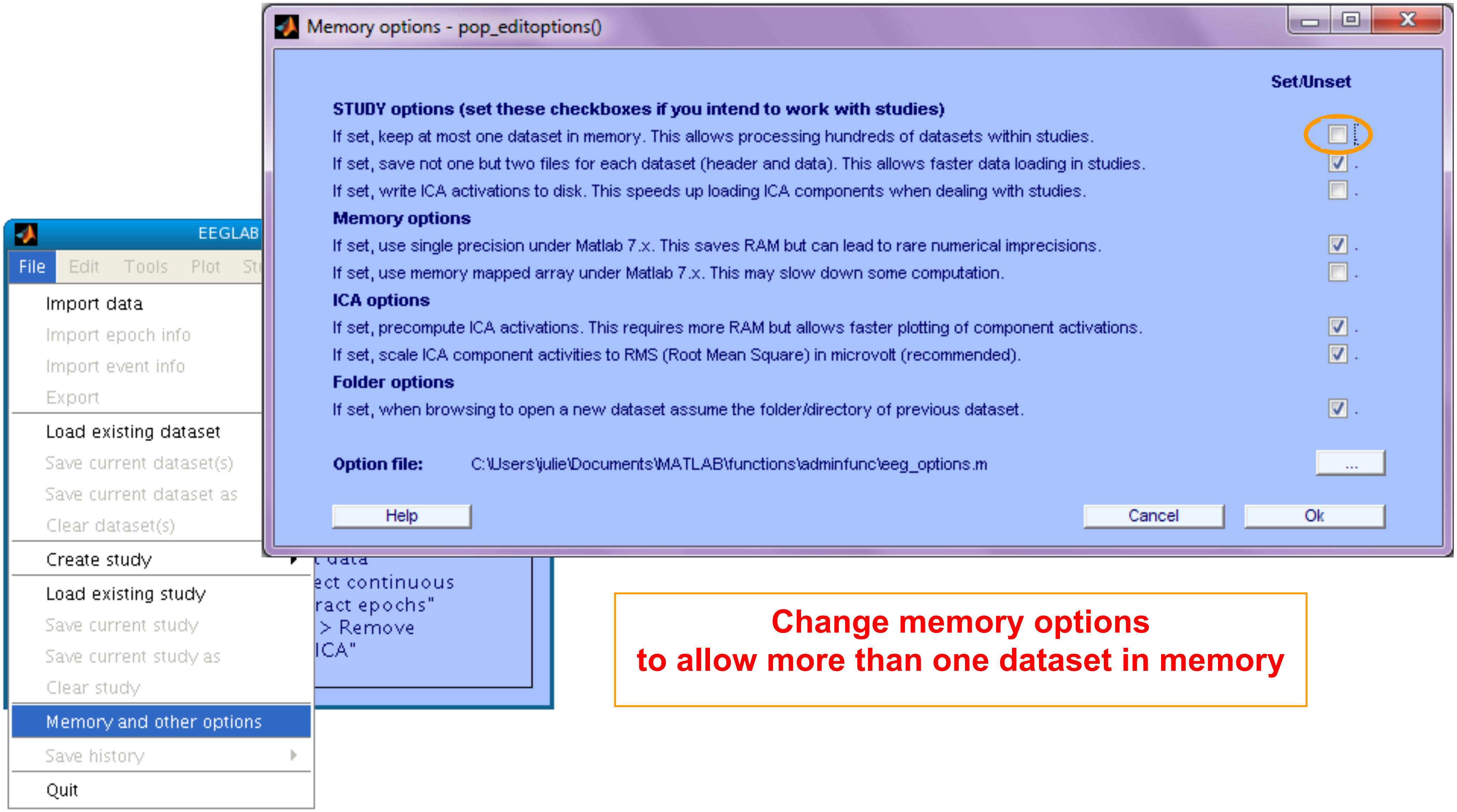
Task 3

Create a Matlab function

Task 4

Exercise…

Memory options



Memory options - pop_editoptions()

Set/Unset

STUDY options (set these checkboxes if you intend to work with studies)

If set, keep at most one dataset in memory. This allows processing hundreds of datasets within studies. ☐

If set, save not one but two files for each dataset (header and data). This allows faster data loading in studies. ✓

If set, write ICA activations to disk. This speeds up loading ICA components when dealing with studies. ☐

Memory options

If set, use single precision under Matlab 7.x. This saves RAM but can lead to rare numerical imprecisions. ✓

If set, use memory mapped array under Matlab 7.x. This may slow down some computation. ☐

ICA options

If set, precompute ICA activations. This requires more RAM but allows faster plotting of component activations. ✓

If set, scale ICA component activities to RMS (Root Mean Square) in microvolt (recommended). ✓

Folder options

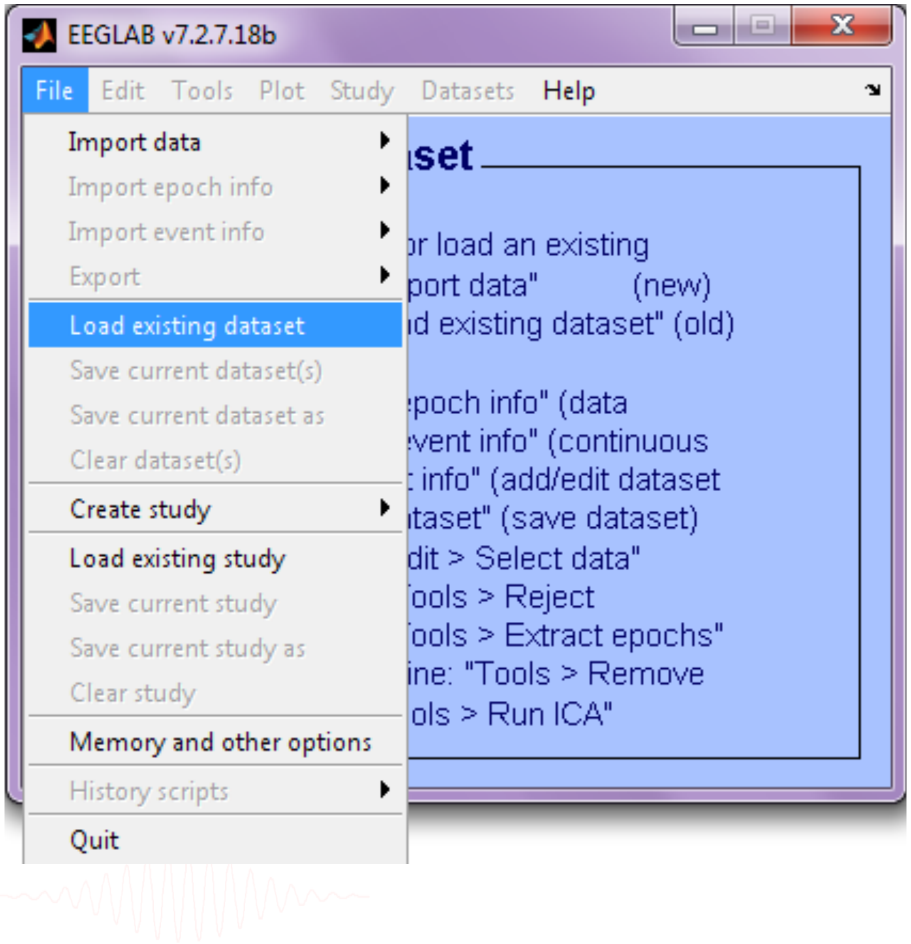
If set, when browsing to open a new dataset assume the folder/directory of previous dataset. ✓

Option file: C:\Users\julie\Documents\MATLAB\functions\adminfunc\eeg_options.m

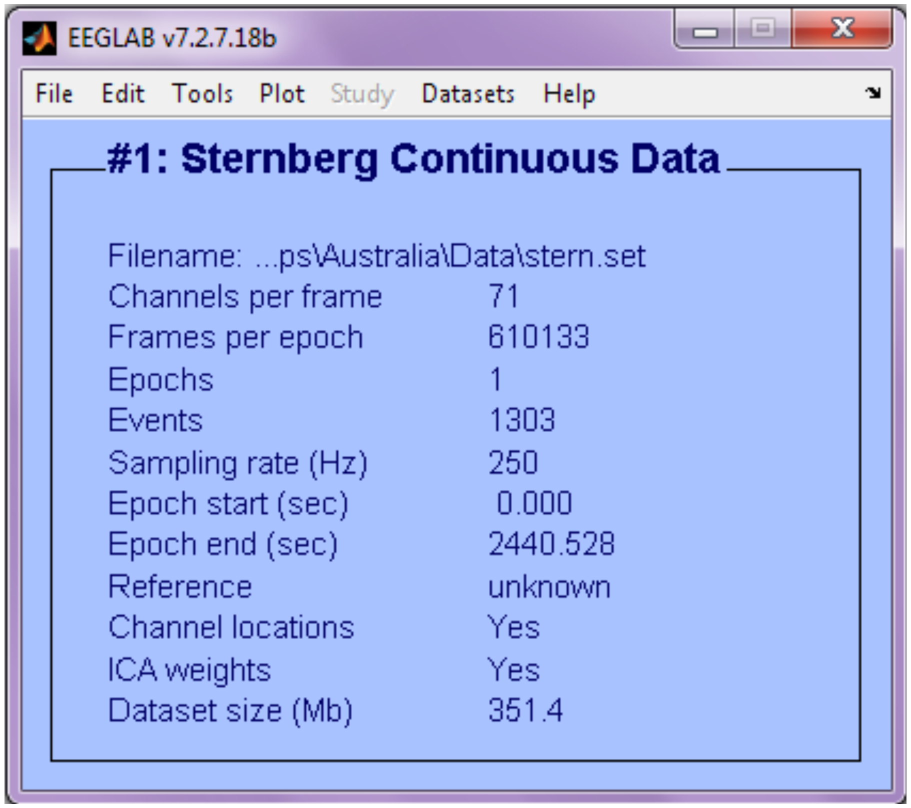
Change memory options to allow more than one dataset in memory

Create a script from ‘eegh’ output

Start by loading a continuous dataset



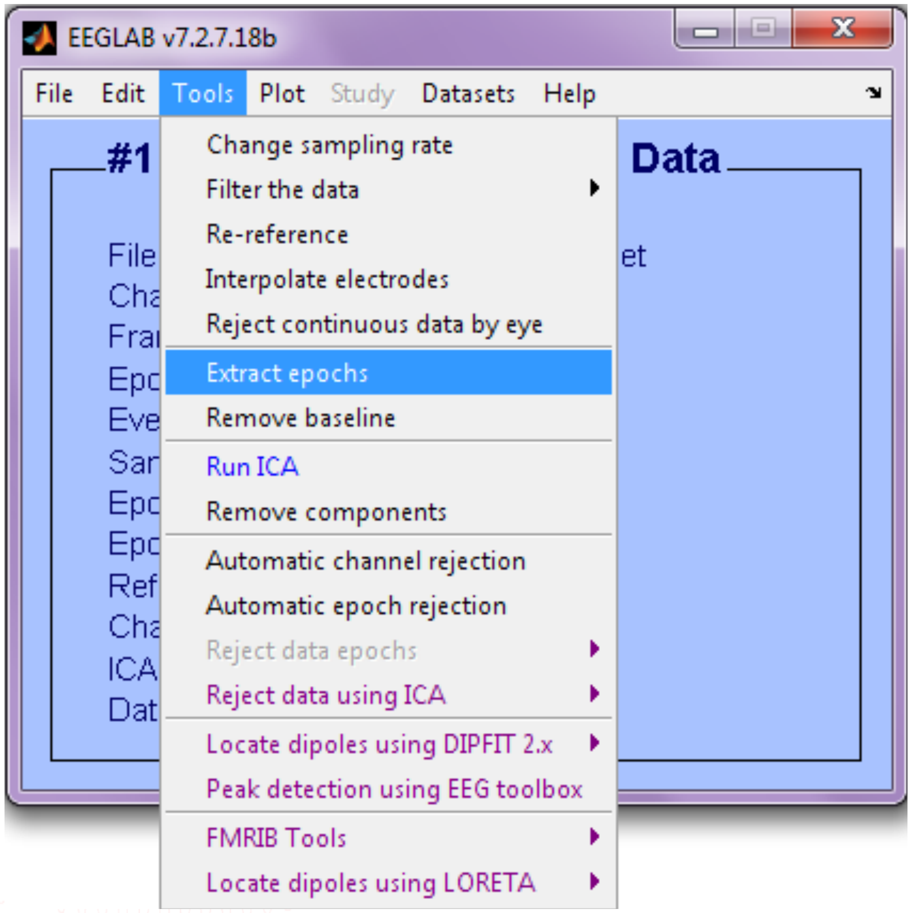
```
set
r load an existing
port data"          (new)
d existing dataset" (old)

poch info" (data
vent info" (continuous
 info" (add/edit dataset
taset" (save dataset)
dit > Select data"
ools > Reject
ools > Extract epochs"
ine: "Tools > Remove
ols > Run ICA"
```



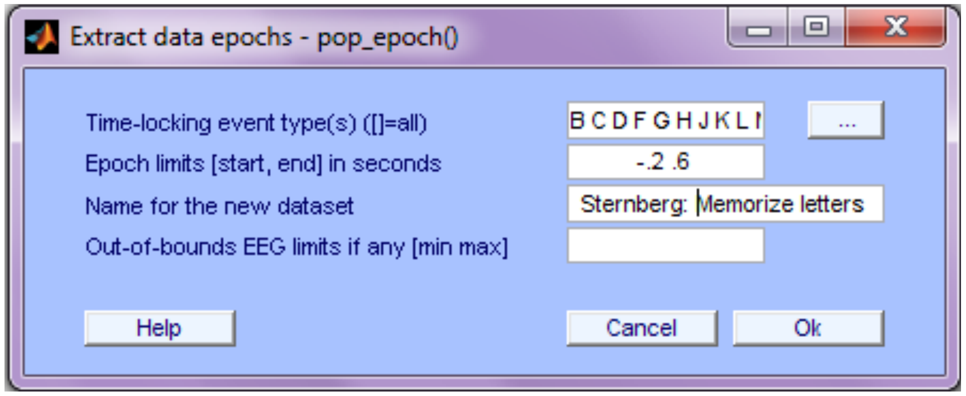
#1: Sternberg Continuous Data

```
Filename: ...ps\Australia\Data\stern.set
Channels per frame        71
Frames per epoch          610133
Epochs                    1
Events                    1303
Sampling rate (Hz)        250
Epoch start (sec)          0.000
Epoch end (sec)           2440.528
Reference                 unknown
Channel locations         Yes
ICA weights               Yes
Dataset size (Mb)         351.4
```

Create a script from ‘eegh’ output



Epoch on Memorize letters


Extract data epochs - pop_epoch()

Time-locking event type(s) ([]=all) B C D F G H J K L N

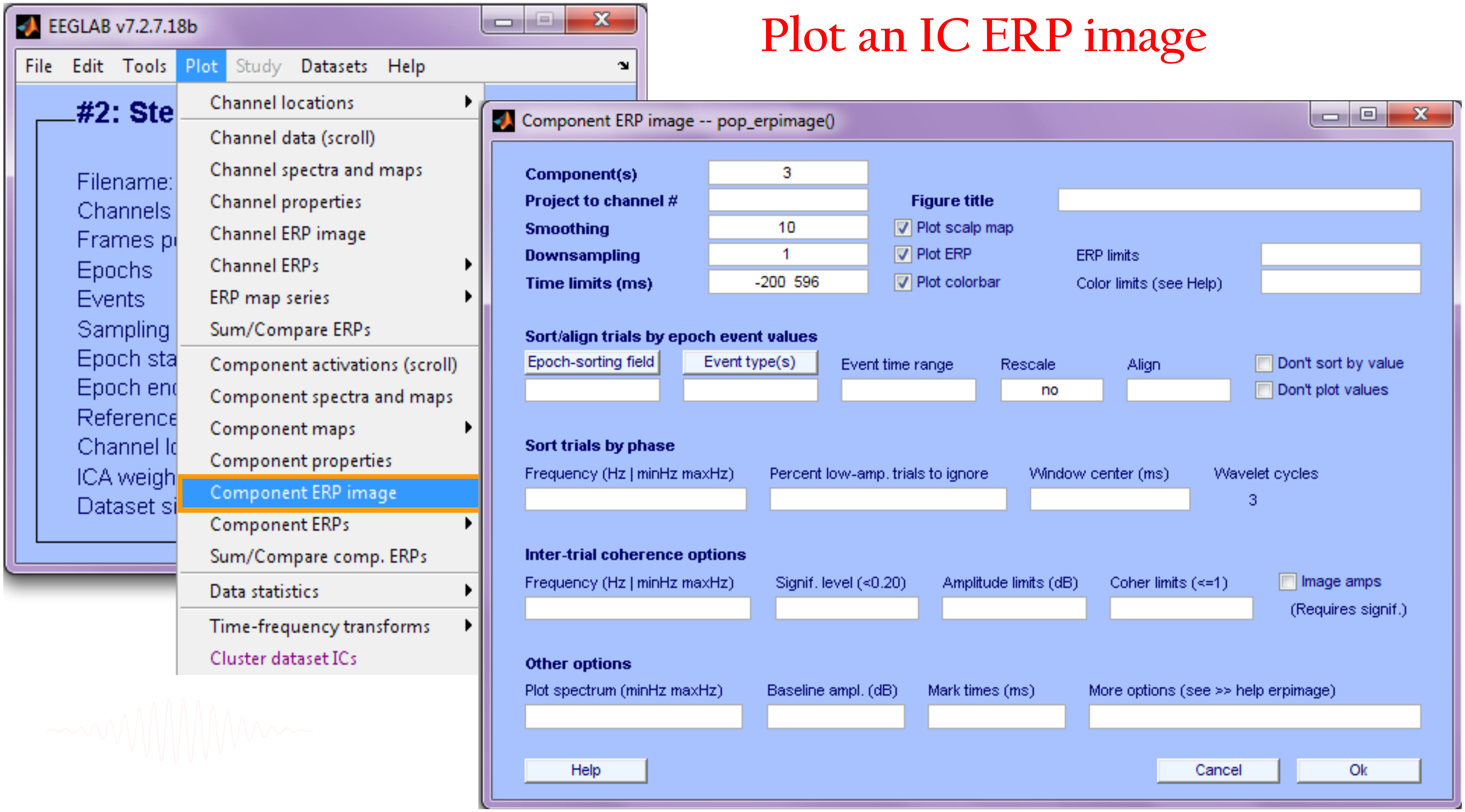
Epoch limits [start, end] in seconds -.2 .6

Name for the new dataset Sternberg: Memorize letters

Out-of-bounds EEG limits if any [min max]

Create a script from 'eegh' output

Plot an IC ERP image



Component(s): 3
Project to channel #:
Figure title:
Smoothing: 10
Downsampling: 1
Time limits (ms): -200 596
[✓] 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

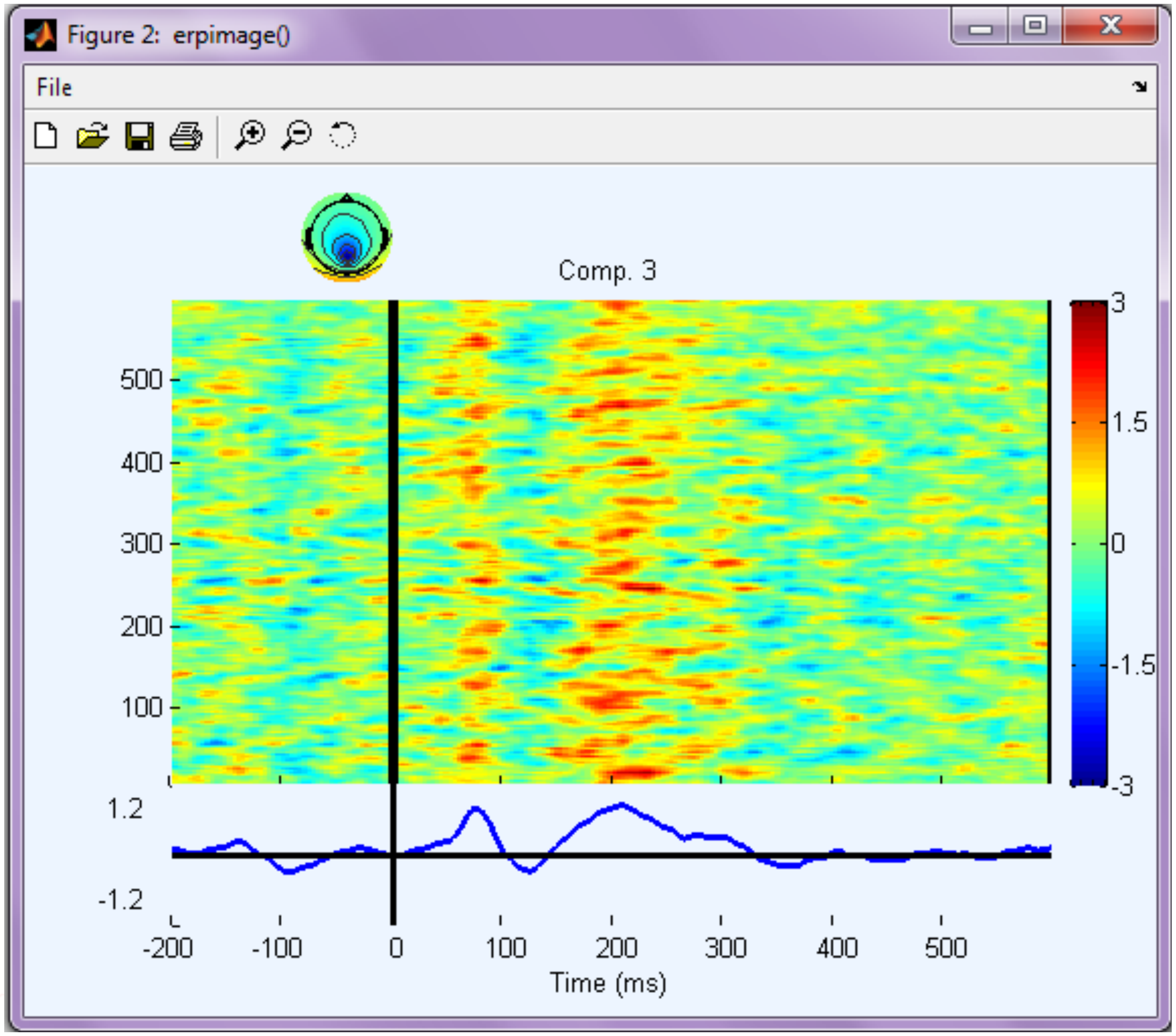
Inter-trial coherence options

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):

Create a script from ‘eegh’ output



Retrieve commands from eegh

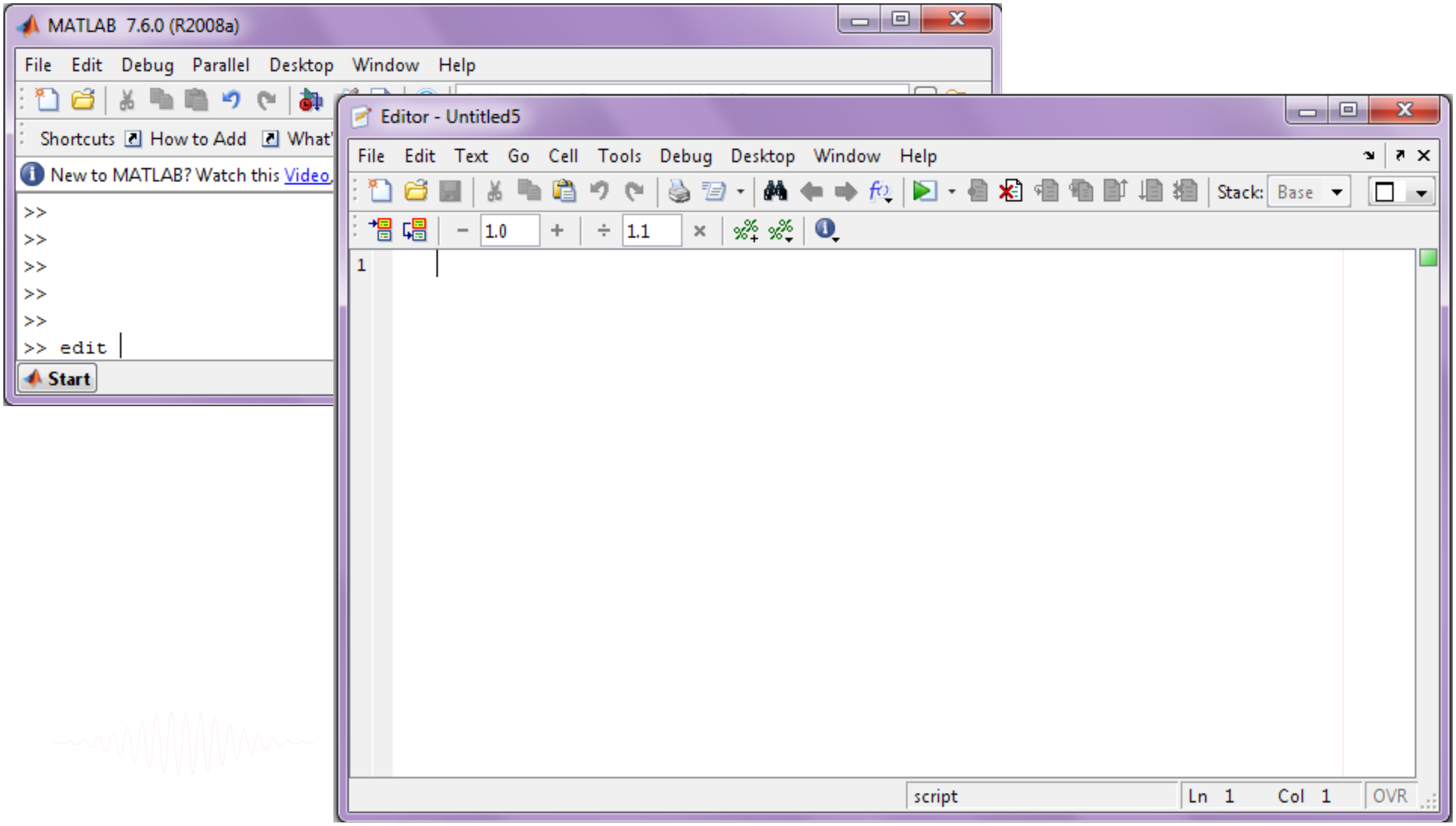
Write a script to do this:

```
>> 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
});
```

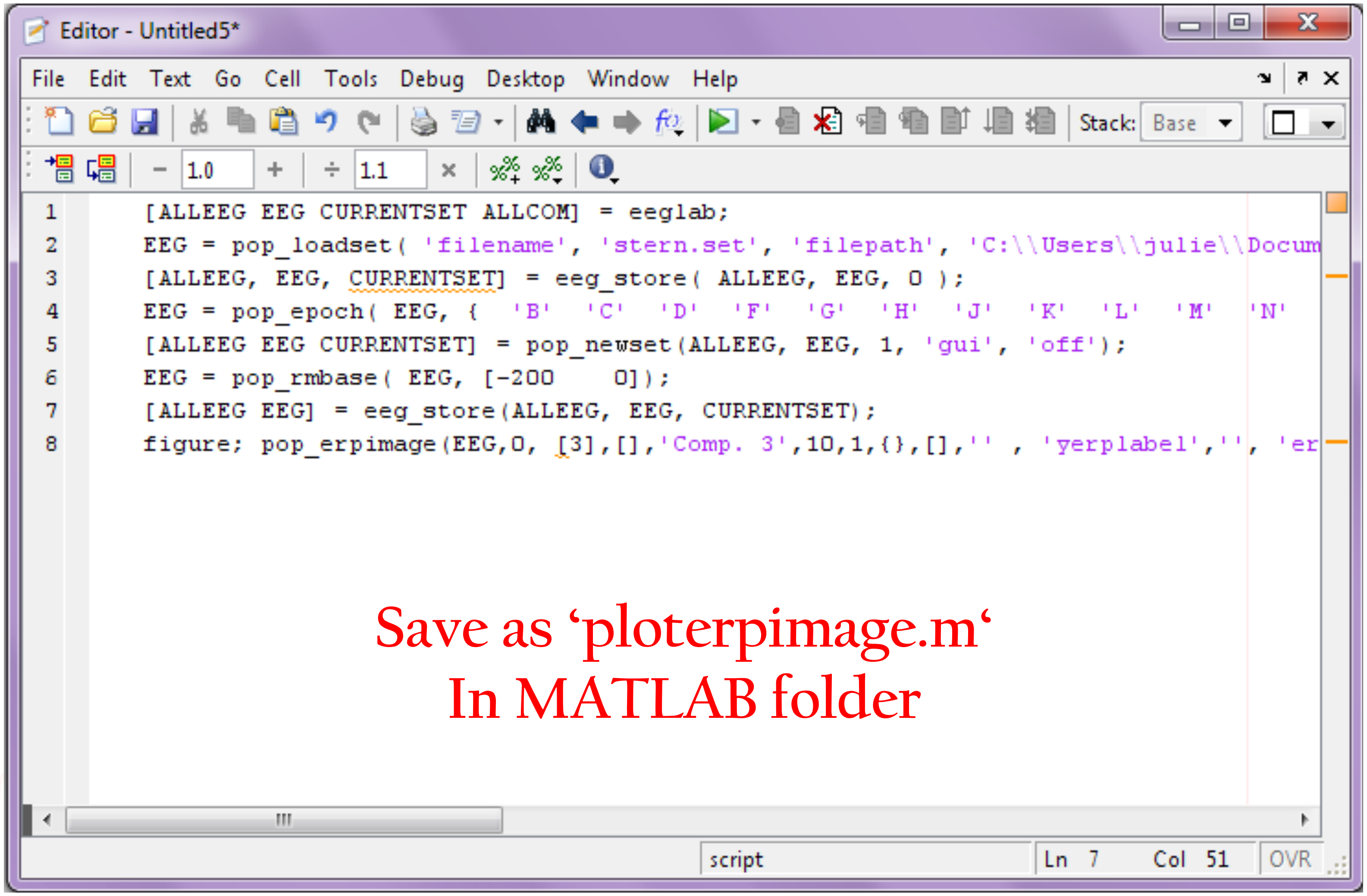
Create a Matlab script


MATLAB 7.6.0 (R2008a)
File Edit Debug Parallel Desktop Window Help
Shortcuts How to Add What'
New to MATLAB? Watch this Video,
Start
Editor - Untitled5
File Edit Text Go Cell Tools Debug Desktop Window Help
Stack: Base
1.0
1.1
script
Ln 1 Col 1
OVR

```
>>
>>
>>
>>
>>
>> edit
```

Create a Matlab script

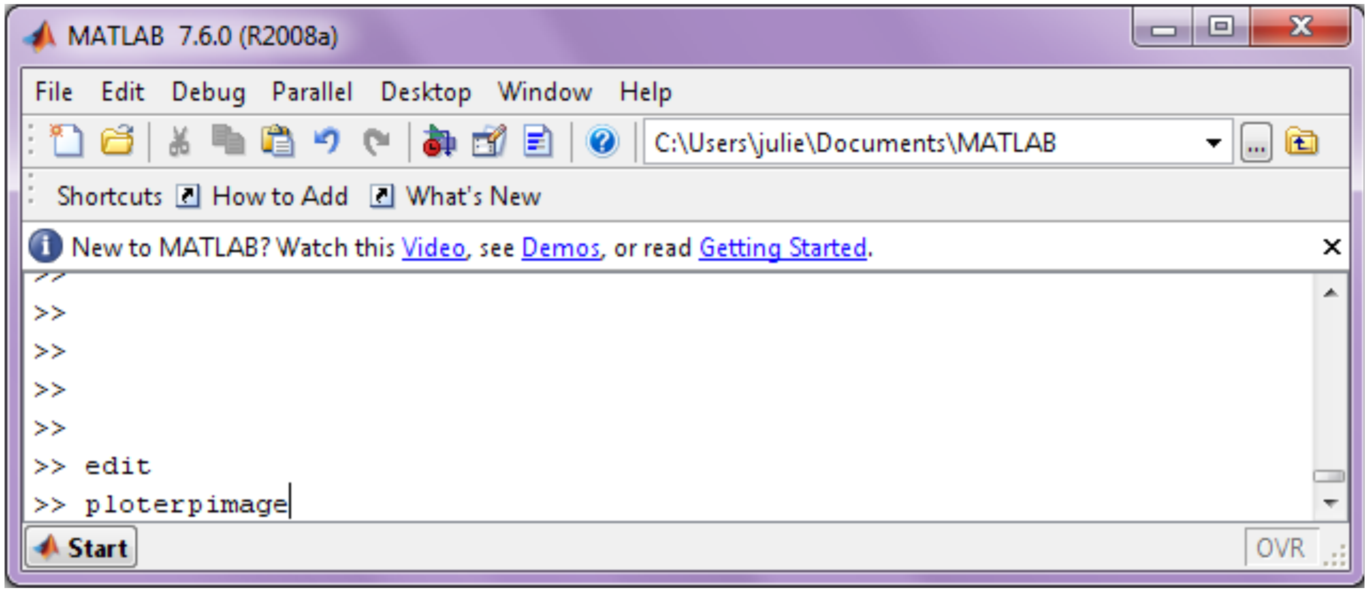
Copy and paste from Matlab window:



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

Save as 'ploterpimage.m'
In MATLAB folder

Run your new script


MATLAB 7.6.0 (R2008a)
File Edit Debug Parallel Desktop Window Help
C:\Users\julie\Documents\MATLAB
Shortcuts How to Add What's New
Start
OVR

New to MATLAB? Watch this Video, see Demos, or read Getting Started.

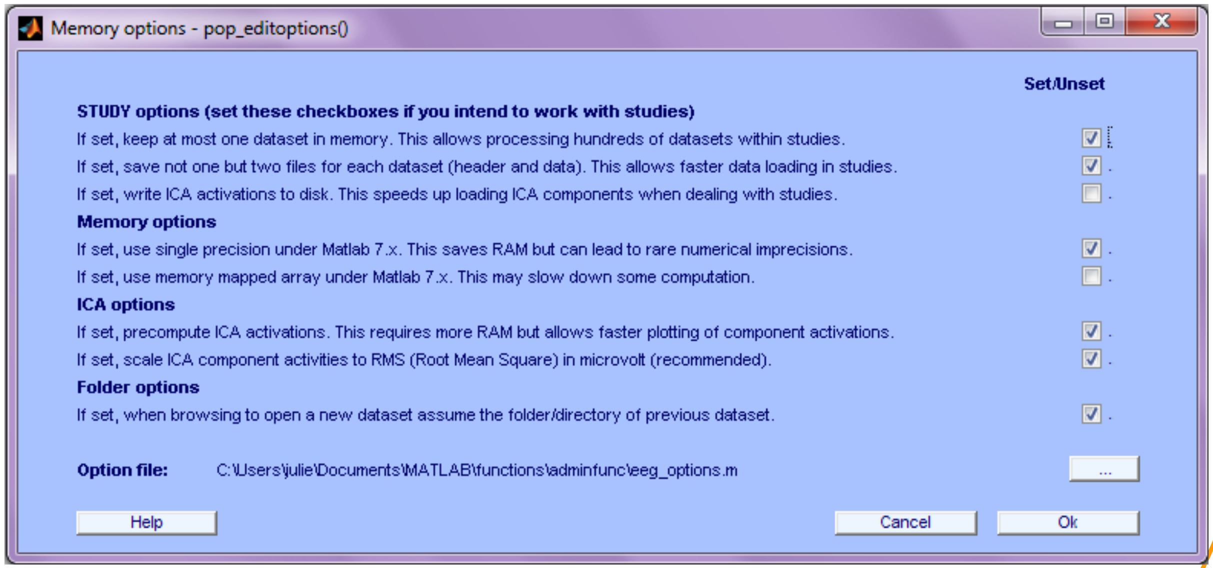
```
>>
>>
>>
>>
>> edit
>> ploterpimage
```

Exercise page 1

```
>> eeglab

% load dataset,
% 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
```

Advanced Scripting in EEGLAB

STUDY scripts



STUDY options (set these checkboxes if you intend to work with studies)

If set, keep at most one dataset in memory. This allows processing hundreds of datasets within studies. ✓

If set, save not one but two files for each dataset (header and data). This allows faster data loading in studies. ✓

If set, write ICA activations to disk. This speeds up loading ICA components when dealing with studies. ☐

Memory options

If set, use single precision under Matlab 7.x. This saves RAM but can lead to rare numerical imprecisions. ✓

If set, use memory mapped array under Matlab 7.x. This may slow down some computation. ☐

ICA options

If set, precompute ICA activations. This requires more RAM but allows faster plotting of component activations. ✓

If set, scale ICA component activities to RMS (Root Mean Square) in microvolt (recommended). ✓

Folder options

If set, when browsing to open a new dataset assume the folder/directory of previous dataset. ✓

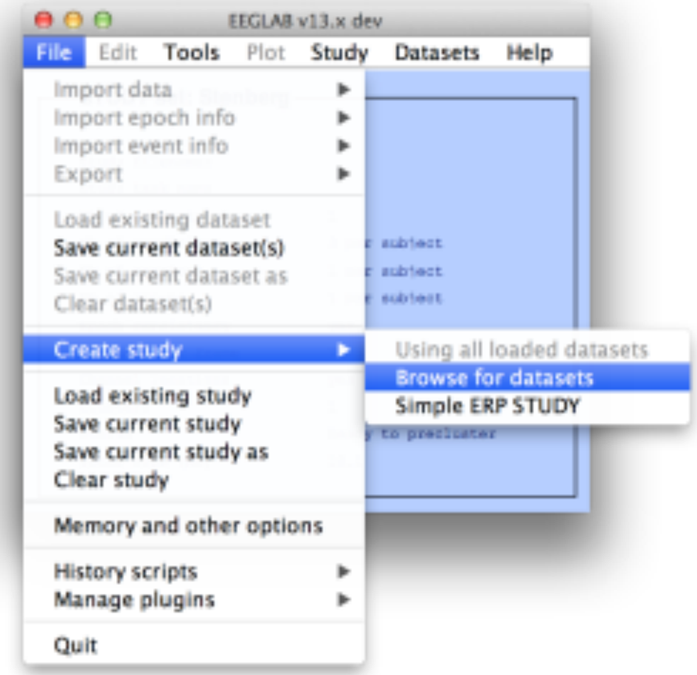
Option file: C:\Users\julie\Documents\MATLAB\functions\adminfunc\eeg_options.m

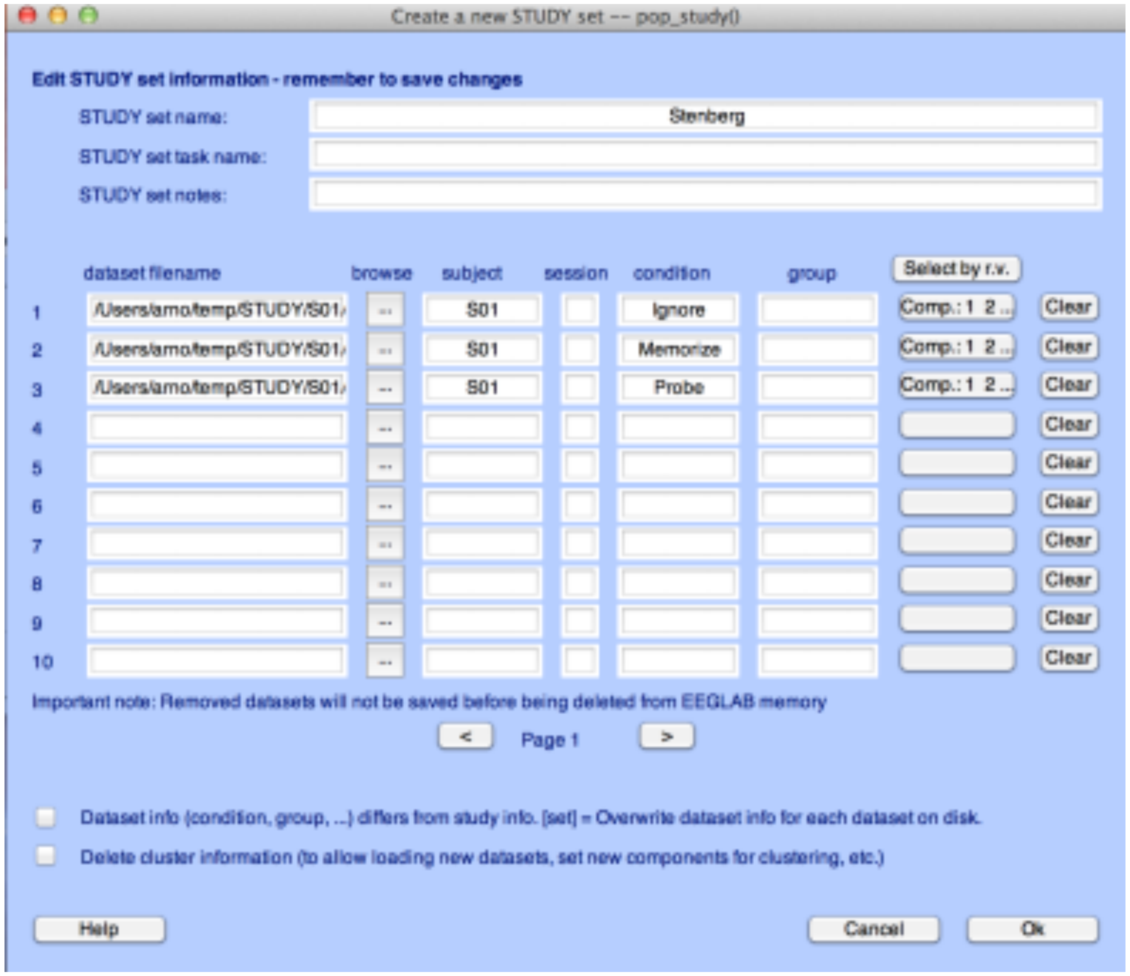
Most important option:

- Allows only one dataset to be loaded at once.
- Most STUDYs are too big to have all data loaded at once.

```
% Set memory options:
pop_editoptions( 'option_storedisk', 1);
```

Edit dataset info




Create a new STUDY set -- pop_study()

Edit STUDY set information - remember to save changes

STUDY set name: Stenberg

STUDY set task name:

STUDY set notes:

	dataset filename	browse	subject	session	condition	group	Select by r.v.	
1	/Users/arno/temp/STUDY/S01/	...	S01		Ignore		Comp.: 1 2 ...	Clear
2	/Users/arno/temp/STUDY/S01/	...	S01		Memorize		Comp.: 1 2 ...	Clear
3	/Users/arno/temp/STUDY/S01/	...	S01		Probe		Comp.: 1 2 ...	Clear
4		...						Clear
5		...						Clear
6		...						Clear
7		...						Clear
8		...						Clear
9		...						Clear
10		...						Clear

Important note: Removed datasets will not be saved before being deleted from EEGLAB memory

< Page 1 >

☐ Dataset info (condition, group, ...) differs from study info. [set] = Overwrite dataset info for each dataset on disk.

☐ Delete cluster information (to allow loading new datasets, set new components for clustering, etc.)

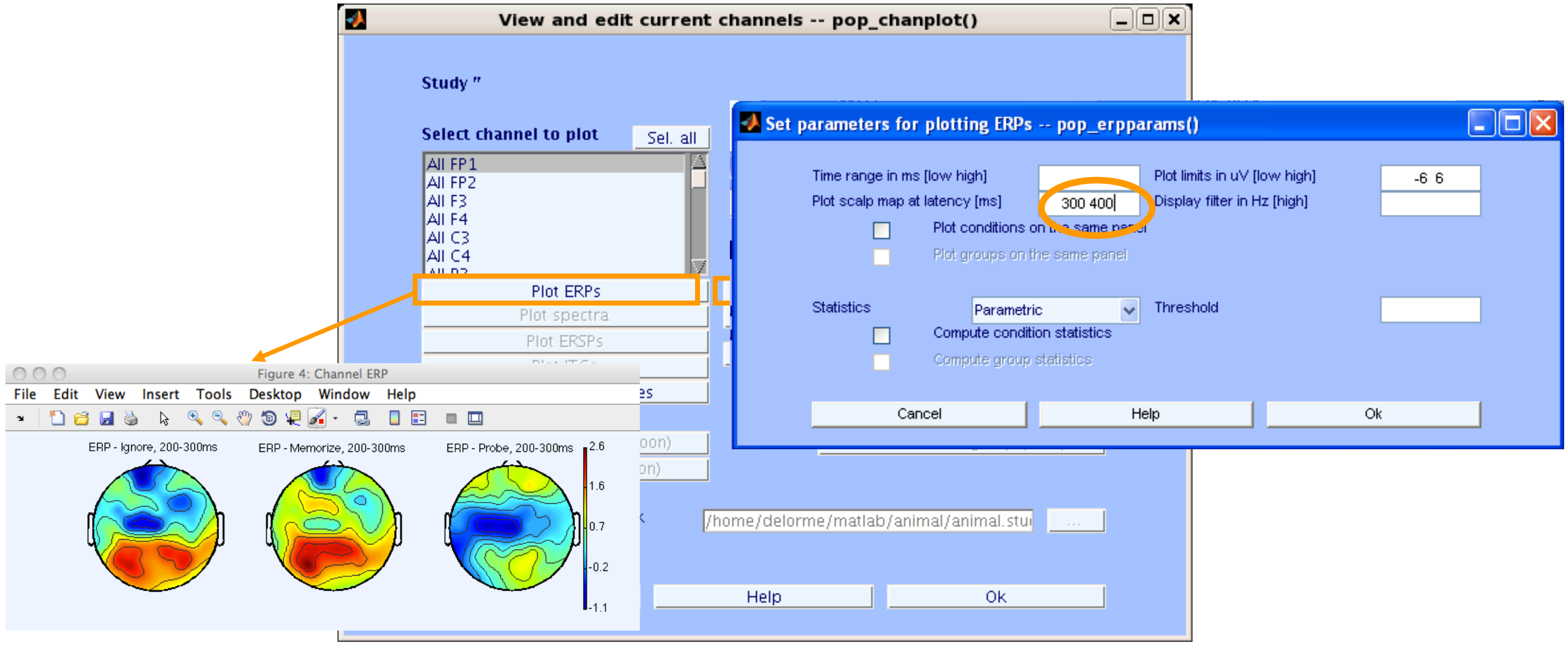
Help Cancel Ok

```
[STUDY ALLEEG] = std_editset( STUDY, ALLEEG, 'name','Stenberg','commands',{{'index' 1
'load' '/data/STUDY/S01/Ignore.set'} {'index' 2 'load' '/data/S01/Memorize.set'} {'index'
3 'load' '/data/S01/Probe.set'} {'index' 1 'subject' 'S01'} {'index' 2 'subject' 'S01'}
{'index' 3 'subject' 'S01'} {'index' 1 'condition' 'Ignore'} {'index' 2 'condition'
'Memorize'} {'index' 3 'condition' 'Probe'}},'updatedat','off' );
```

Looking a the function that create STUDY

```
[STUDY ALLEEG] = std_editset( STUDY, ALLEEG, 'name','Stenberg','commands',
     {{'index' 1 'load' '/data/STUDY/S01/Ignore.set'} …
     {'index' 2 'load' '/data/S01/Memorize.set'} …
     {'index' 3 'load' '/data/S01/Probe.set'} …
     {'index' 1 'subject' 'S01'} …
     {'index’ 2 'subject' 'S01'} …
     {'index’ 3 'subject' 'S01'} …
     {'index' 1 'condition' 'Ignore'} …
     {'index' 2 'condition' 'Memorize'} …
     {'index' 3 'condition' 'Probe'}},'updatedat','off' );
```

```
[STUDY ALLEEG] = std_editset( STUDY, ALLEEG, 'name','Stenberg','commands',
    {{'index' 1 'load' '/data/STUDY/S01/Ignore.set’ 'subject' 'S01' 'condition' 'Ignore'} …
    {'index' 2 'load' '/data/S01/Memorize.set’ 'subject' 'S01' 'condition' 'Memorize'} …
    {'index' 3 'load' '/data/S01/Probe.set’ 'subject' 'S01' 'condition' 'Probe'} …
    },'updatedat','off' );
```


View and edit current channels -- pop_chanplot()

Study ''

Select channel to plot

/home/delorme/matlab/animal/animal.stu

Set parameters for plotting ERPs -- pop_erpparams()

Time range in ms [low high]		Plot limits in uV [low high]	-6 6
Plot scalp map at latency [ms]	300 400	Display filter in Hz [high]	

☐ Plot conditions on the same panel

☐ Plot groups on the same panel

Statistics: Parametric — Threshold:

☐ Compute condition statistics

☐ Compute group statistics

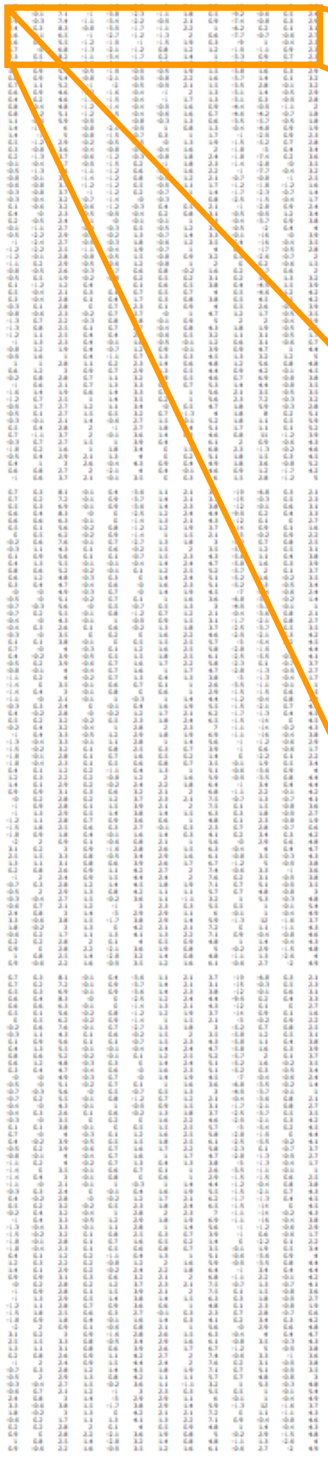
Figure 4: Channel ERP

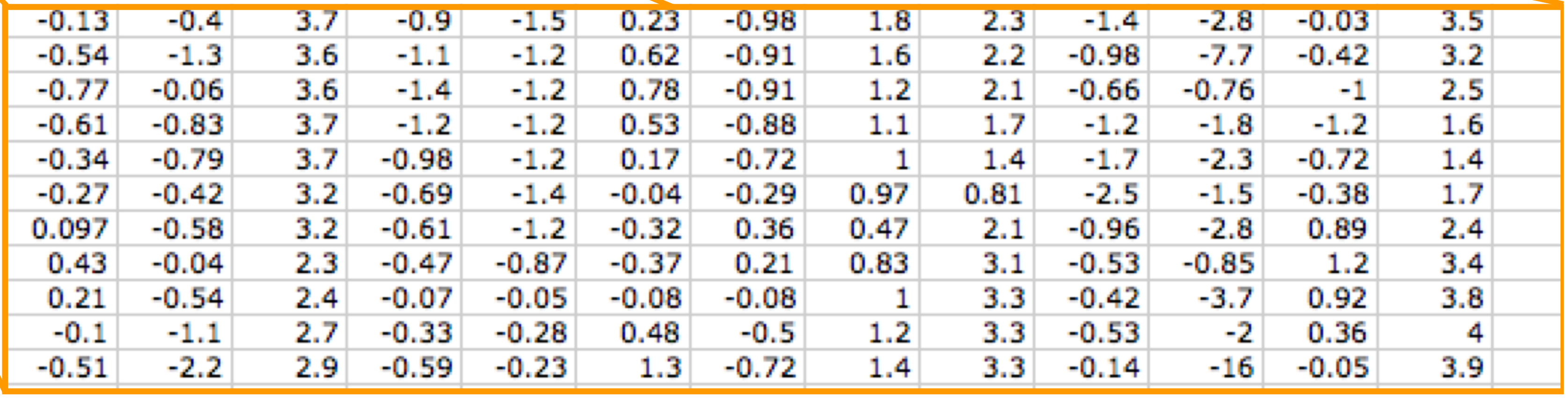
```
STUDY = std_erpplot(STUDY,ALLEEG, 'topotime',[200 300] , 'channels',{'OZ' 'O2' 'FP1' 'FPZ' 'FP2'});
[STUDY erpdata ] = std_erpplot(STUDY,ALLEEG, , ‘topotime’,[200 300] , ‘channels’,{’OZ’ ’O2’});
```

Exporting to excell file

```
[1x67x13 double]        xlswrite('myxlsfile',squeeze(erpdata{1}),1);
[1x67x13 double]        xlswrite('myxlsfile',squeeze(erpdata{2}),2);
[1x67x13 double]        xlswrite('myxlsfile',squeeze(erpdata{3}),3);
```

Exporting text file





-0.13	-0.4	3.7	-0.9	-1.5	0.23	-0.98	1.8	2.3	-1.4	-2.8	-0.03	3.5
-0.54	-1.3	3.6	-1.1	-1.2	0.62	-0.91	1.6	2.2	-0.98	-7.7	-0.42	3.2
-0.77	-0.06	3.6	-1.4	-1.2	0.78	-0.91	1.2	2.1	-0.66	-0.76	-1	2.5
-0.61	-0.83	3.7	-1.2	-1.2	0.53	-0.88	1.1	1.7	-1.2	-1.8	-1.2	1.6
-0.34	-0.79	3.7	-0.98	-1.2	0.17	-0.72	1	1.4	-1.7	-2.3	-0.72	1.4
-0.27	-0.42	3.2	-0.69	-1.4	-0.04	-0.29	0.97	0.81	-2.5	-1.5	-0.38	1.7
0.097	-0.58	3.2	-0.61	-1.2	-0.32	0.36	0.47	2.1	-0.96	-2.8	0.89	2.4
0.43	-0.04	2.3	-0.47	-0.87	-0.37	0.21	0.83	3.1	-0.53	-0.85	1.2	3.4
0.21	-0.54	2.4	-0.07	-0.05	-0.08	-0.08	1	3.3	-0.42	-3.7	0.92	3.8
-0.1	-1.1	2.7	-0.33	-0.28	0.48	-0.5	1.2	3.3	-0.53	-2	0.36	4
-0.51	-2.2	2.9	-0.59	-0.23	1.3	-0.72	1.4	3.3	-0.14	-16	-0.05	3.9

```
dlmwrite('erpfile.txt',squeeze(erpdata{1}),'delimiter', '\t', 'precision', 2);

dlmwrite('erpfile.txt',squeeze(erpdata{2}),'-append', 'roffset', 1,
          'delimiter', '\t', 'precision', 2);

dlmwrite('erpfile.txt',squeeze(erpdata{2}),'-append', 'roffset', 1,
          'delimiter', '\t', 'precision', 2);
```

Exercice: run STUDY Script

```
% Create Stern STUDY
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;
pop_editoptions( 'option_storedisk', 1);
subjects = {'S01' 'S02' 'S03' 'S04' 'S05' 'S06' 'S07' 'S08' 'S09' 'S10' 'S11' 'S12'};
filepath = '/Users/arno/temp/STUDY'; % XXXXX Change path here XXXXX
if ~exist(filepath), error('You need to change the path to the STUDY'); end;
commands = {}; % initialize STUDY dataset list

% Loop through all of the subjects in the study to create the dataset
for loopnum = 1:length(subjects) %for each subject
    IgnoreFile   = fullfile(filepath, subjects{loopnum}, 'Ignore.set');
    MemorizeFile = fullfile(filepath, subjects{loopnum}, 'Memorize.set');
    ProbeFile    = fullfile(filepath, subjects{loopnum}, 'Probe.set');
    commands = {commands{:} ...
        {'index' 3*loopnum-2 'load' IgnoreFile   'subject' subjects{loopnum} 'condition' 'Ignore'} ...
        {'index' 3*loopnum-1 'load' MemorizeFile 'subject' subjects{loopnum} 'condition' 'Memorize'} ...
        {'index' 3*loopnum   'load' ProbeFile    'subject' subjects{loopnum} 'condition' 'Probe'}};
end;
% Uncomment the line below to select ICA components with less than 15% residual variance
% commands = {commands{:} {'dipselect', 0.15}};
[STUDY, ALLEEG] = std_editset(STUDY, ALLEEG, 'name','Sternberg','commands',commands,'updatedat','on');

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

[STUDY ALLEEG] = std_precomp(STUDY, ALLEEG, {},'rmicacomps','on','interp','on','recompute','on','erp','on');
STUDY = pop_erpparams(STUDY, 'topotime',[200 300] );
[STUDY erpdata] = std_erpplot(STUDY,ALLEEG,'channels',{'LEYE' 'REYE' 'OZ' 'O2' 'FP1' 'FPZ' 'FP2' 'AF7' ...
            'AF3' 'AFZ' 'AF4' 'AF8' 'F9' 'F7' 'F5' 'F3' 'F1' 'FZ' 'F2' 'F4' 'F6' 'F8' 'F10' 'FT9' ...
            'FT7' 'FC5' 'FC3' 'FC1' 'FCZ' 'FC2' 'FC4' 'FC6' 'FT8' 'FT10' 'T7' 'C5' 'C3' 'C1' 'CZ' ...
            'C2' 'C4' 'C6' 'T8' 'TP9' 'TP7' 'CP5' 'CP3' 'CP1' 'CPZ' 'CP2' 'CP4' 'CP6' 'TP8' 'TP10' ...
            'P7' 'P5' 'P3' 'P1' 'PZ' 'P2' 'P4' 'P6' 'P8' 'PO9' 'PO7' 'PO3' 'POZ' 'PO4' 'PO8' 'PO10' 'O1'});

dlmwrite('erpfile.txt',squeeze(erpdata{1}),'delimiter', '\t', 'precision', 2);
dlmwrite('erpfile.txt',squeeze(erpdata{2}),'-append', 'roffset', 1, 'delimiter', '\t', 'precision', 2);
dlmwrite('erpfile.txt',squeeze(erpdata{2}),'-append', 'roffset', 1, 'delimiter', '\t', 'precision', 2);
```

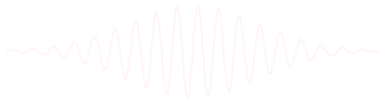
Advanced scripting: EEG pipeline

```
sInfo = [];
sInfo(end+1).file = 'rawdata/S01.raw';
sInfo(end).name = 'S01';
sInfo(end).bad_channels = { 'E1' };
```

Raw data file

Subject name

Subject name



Advanced scripting: EEG pipeline

```
sInfo = [];
sInfo(end+1).file = 'rawdata/S01.raw';
sInfo(end).name = 'S01';
sInfo(end).bad_channels = { 'E1' };
sInfo(end).bad_data  = [726 1495;6098 6831;13245 14057;15715 16399;22756 24457
```

Raw data file

Subject name

Subject name

Copy the output from the eeg_eegrej function in the history

Advanced scripting: EEG pipeline

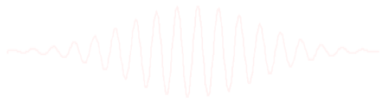
```
sInfo = [];
sInfo(end+1).file = 'rawdata/S01.raw';
sInfo(end).name = 'S01';
sInfo(end).bad_channels = { 'E1' };
sInfo(end).bad_data  = [726 1495;6098 6831;13245 14057;15715 16399;22756];
sInfo(end).bad_comps = [1.6681  1.9870 0.3979 0.4444 -0.2274 -0.1433 -0.2626 -
                        1.1917 -1.4838 0.7469 -1.1599 0.4773 -0.3257  0.3074 -
```

Raw data file

Subject name

Subject name

Copy transposed columns of the inverse weight matrix
EEG.icawinv for your selected artifact components



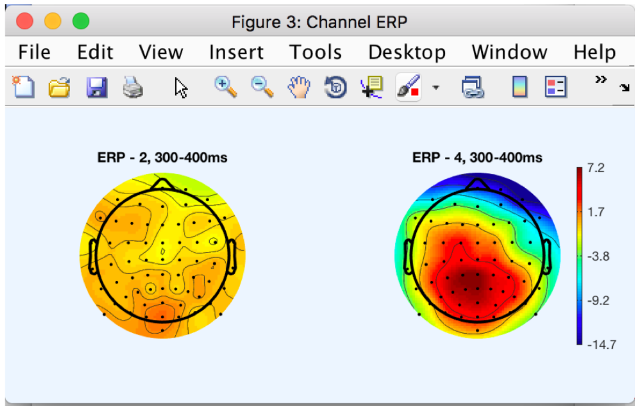
Advanced scripting: EEG pipeline

datainfo.m file

```
sInfo = [];
sInfo(end+1).file = 'S01.raw';
sInfo(end).name = 'S01';
sInfo(end).bad_channels = { 'E1' };
sInfo(end).bad_data  = [726 1495;6098 6831;13245 14057;15715 16399;22756 24457;3074
sInfo(end).bad_comps = [1.6681  1.9870 0.3979 0.4444 -0.2274 -0.1433 -0.2626 -0.108
                        1.1917 -1.4838 0.7469 -1.1599 0.4773 -0.3257  0.3074 -0.163

sInfo(end+1).file = 'S02.raw';
sInfo(end).name = 'S02';
sInfo(end).bad_channels = { };
sInfo(end).bad_data  = [41661 43713;24000 24833;44878 46501;48706 49210;51190 52353
sInfo(end).bad_comps = [0.6960 -0.8637  0.9087 -0.8028  0.4873 -0.2142  0.2737 -0.2
                       -0.0875 -0.4056 -0.0287 -0.3870  0.0600 -0.3716  0.3425 -0.4
                        2.1928  1.5712  0.8622  0.3215 -0.0357 -0.3125 -0.2268 -0.3

sInfo(end+1).file = 'S03.raw';
sInfo(end).name = 'S03';
sInfo(end).bad_channels = { 'E10' 'E19' 'E20' 'E29' };
sInfo(end).bad_data  = [1 10449;19808 21815;25678 27254;29257 30010;34023 36016;367
sInfo(end).bad_comps = [ 1.8583  2.0468 -0.0516  0.3159 -0.4256 -0.2770 -0.3643 -0.
                         1.2189 -0.7385  1.2464 -0.8913  0.5475 -0.3971  0.2987 -0.
                        -0.1248 -0.1358 -0.1954 -0.2533 -0.1555 -0.2313 -0.0351 -0.
```

```matlab
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;

    % proprocess 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
```

Create STUDY



```
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
```

Exercice: build your own pipeline

Suggestion for exercise

1. Load oddball_file.bdf dataset (in Data folder or on the wiki)
2. High pass filter at 0.5Hz (menu Tools > Filter)
3. Reject bad channels by hand (plot spectrum)
4. Re-reference to average ref. (optional) (menu Tools > Re-reference)
5. Reject bad portion of data by hand (menu Tools > Reject continuous…)
6. Build your *datainfo.m* file using EEGLAB history (see scripting lecture)
7. Run ICA; select bad ICA components
8. Epoch data on Oddball (type 4) and Standard (type 2) – save dataset
9. Add components to you *datainfo.m* file
10. Create a STUDY with this single file
11. Compare the ERP for Oddball (type 4) and Standard (type 2) and use single-trial statistics with cluster correction for multiple comparisons
12. Build a script that creates the STUDY and perform the same analysis
13. Save the figure at the end of the script in eps or jpg format ("print –depsc file" command or "print –djpg file" command).
14. Run the full pipeline (dataset processing and STUDY processing)
15. Change the filtering in the pipeline (step 2) and observe effects

Exporting figures

Transparency and complex figures

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. Note that these cannot handle transparency and 3-D graphics.

Transparency: Use the “plot2svg” matlab toolbox to export figure for transparency.