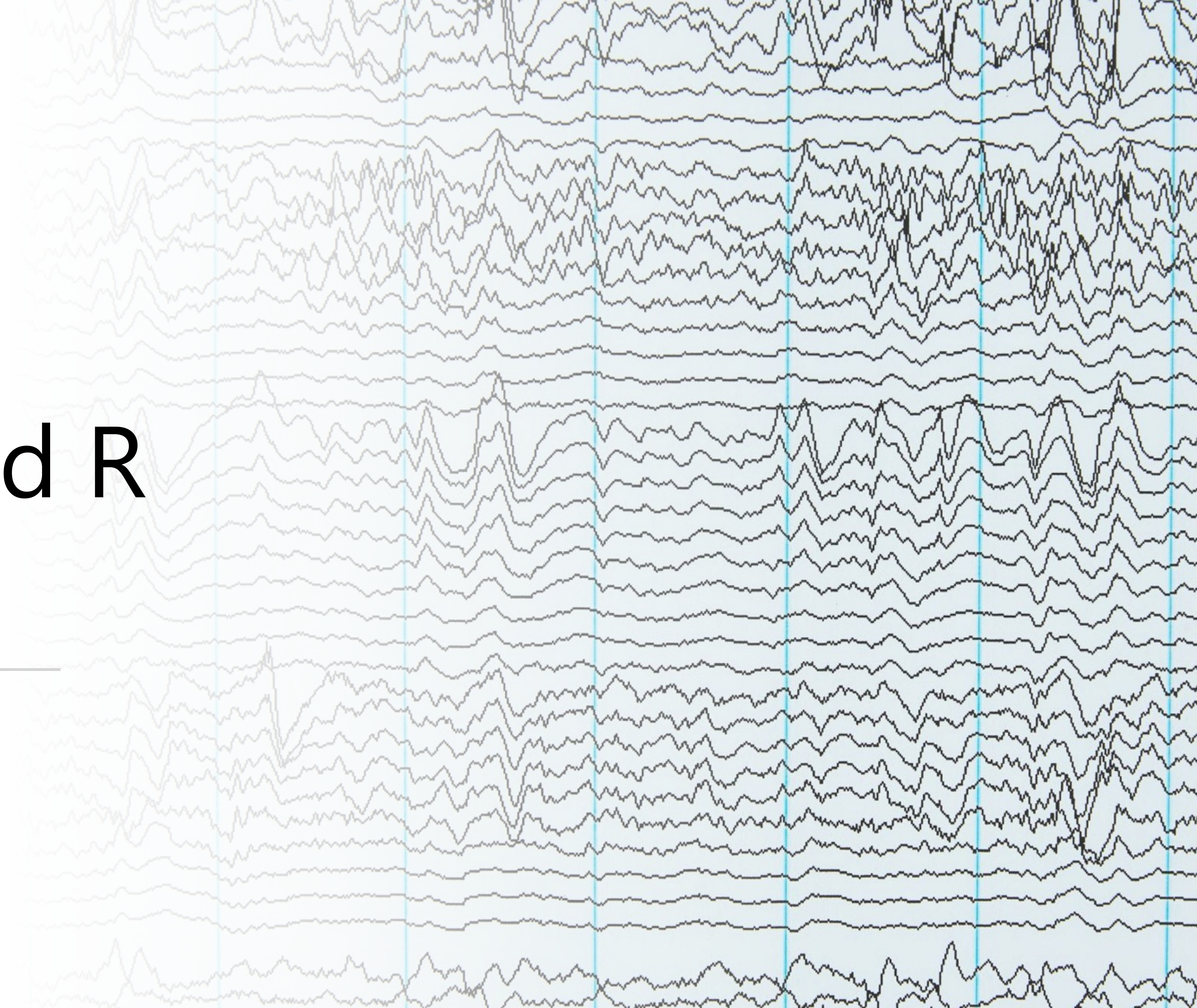
MATLAB and R (for EEG)

Aron Hill

Cognitive Neuroscience Unit,
Deakin University



Why use code?

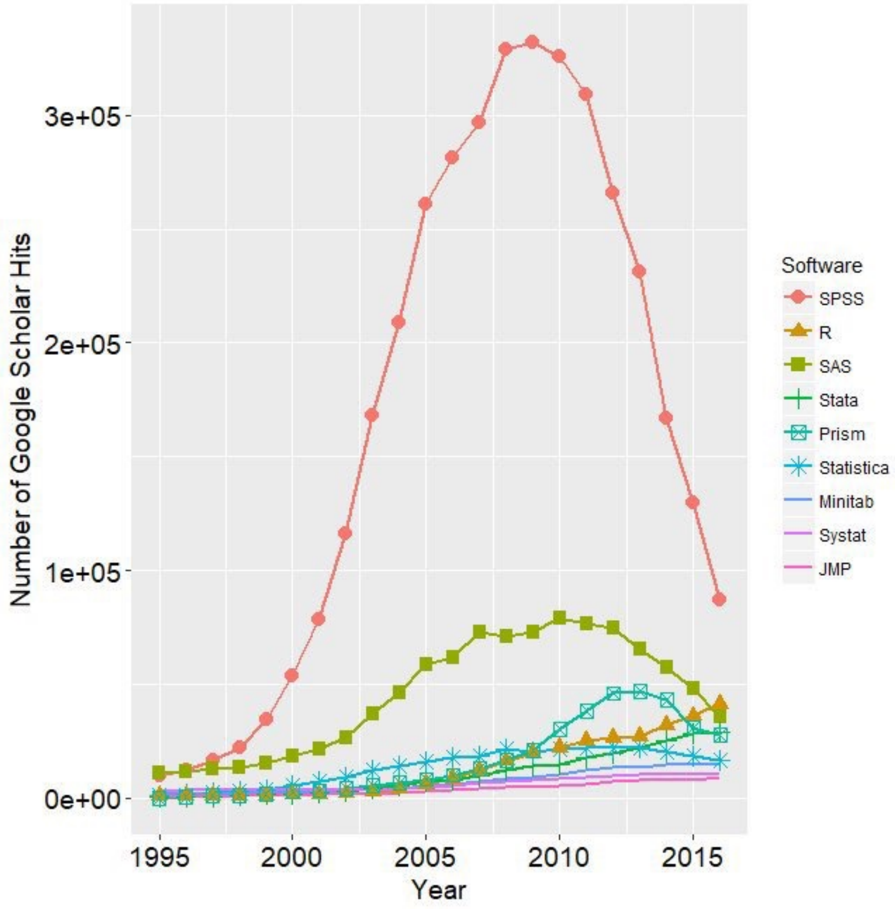
- Increasingly a requirement for working in teams

- Keep track of what you've done

- Do replicable/transparent research
    - Increasing no. of journals ask for code

- Share analysis pipelines
    - E.g., GitHub

- Transferable skillset



Source: @dsquintana

MATLAB® R

- Been around a long time
  - Large user/support base
  - Stood test of time

- Regularly updated

- Cross-platform support (win, mac)

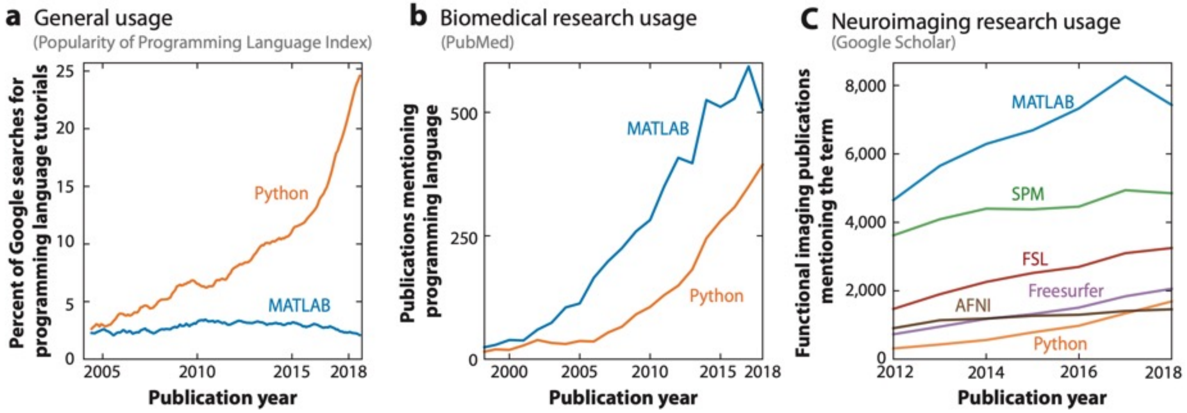
- Many toolboxes/packages available to increase functionality

- Matlab is proprietary software (expensive)

- R is open source (free)

A quick note on Python…

- Increasingly attractive option for EEG analysis
- Rapidly growing userbase
- EEG-focused toolboxes for M/EEG
  - MNE (https://mne.tools/stable/index.htm)
  - FOOOF (https://fooof-tools.github.io/fooof)



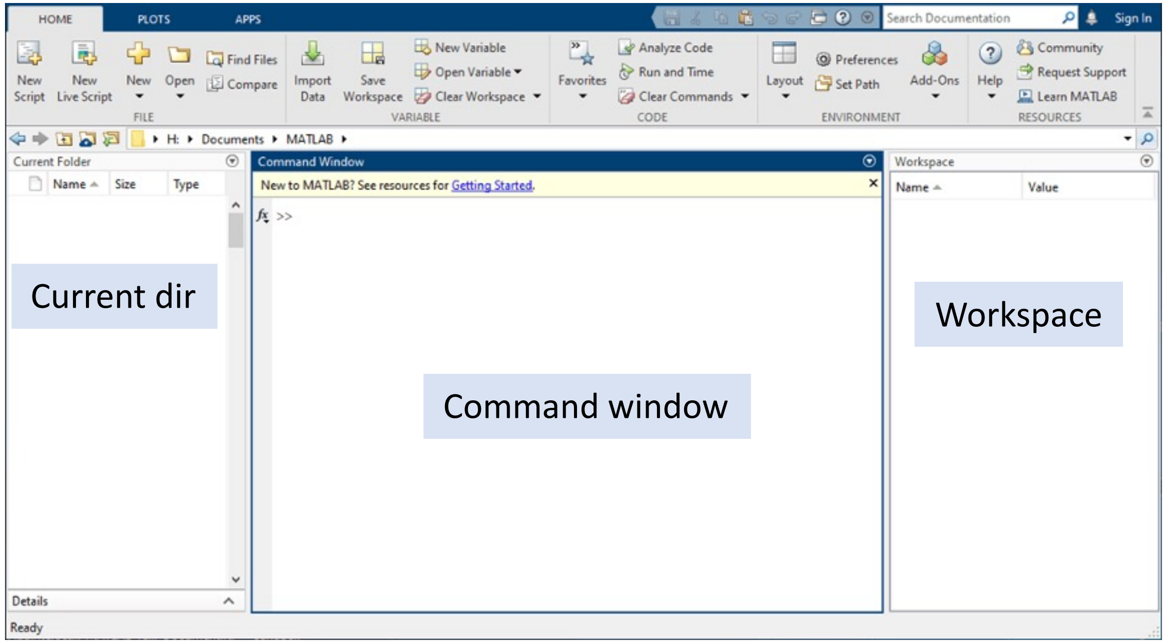
Source: Poldrack et al. (2019). *Annu. Rev. Biomed. Data Sci.*

Useful features

1. Customising your environment
2. Installing toolboxes
3. Building a script and creating sharable code
4. Using and customizing shortcuts
5. Utilising the EEGLAB ‘eegh’ function

1: Customise your environment

MATLAB



New to MATLAB? See resources for Getting Started.

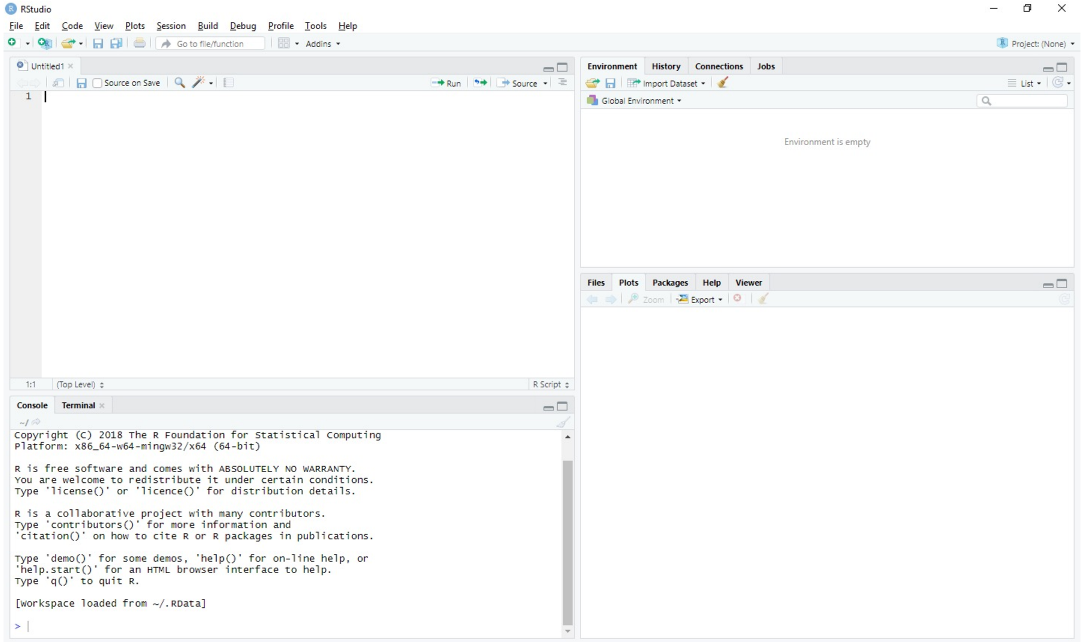
```
fx >>
```

Current dir

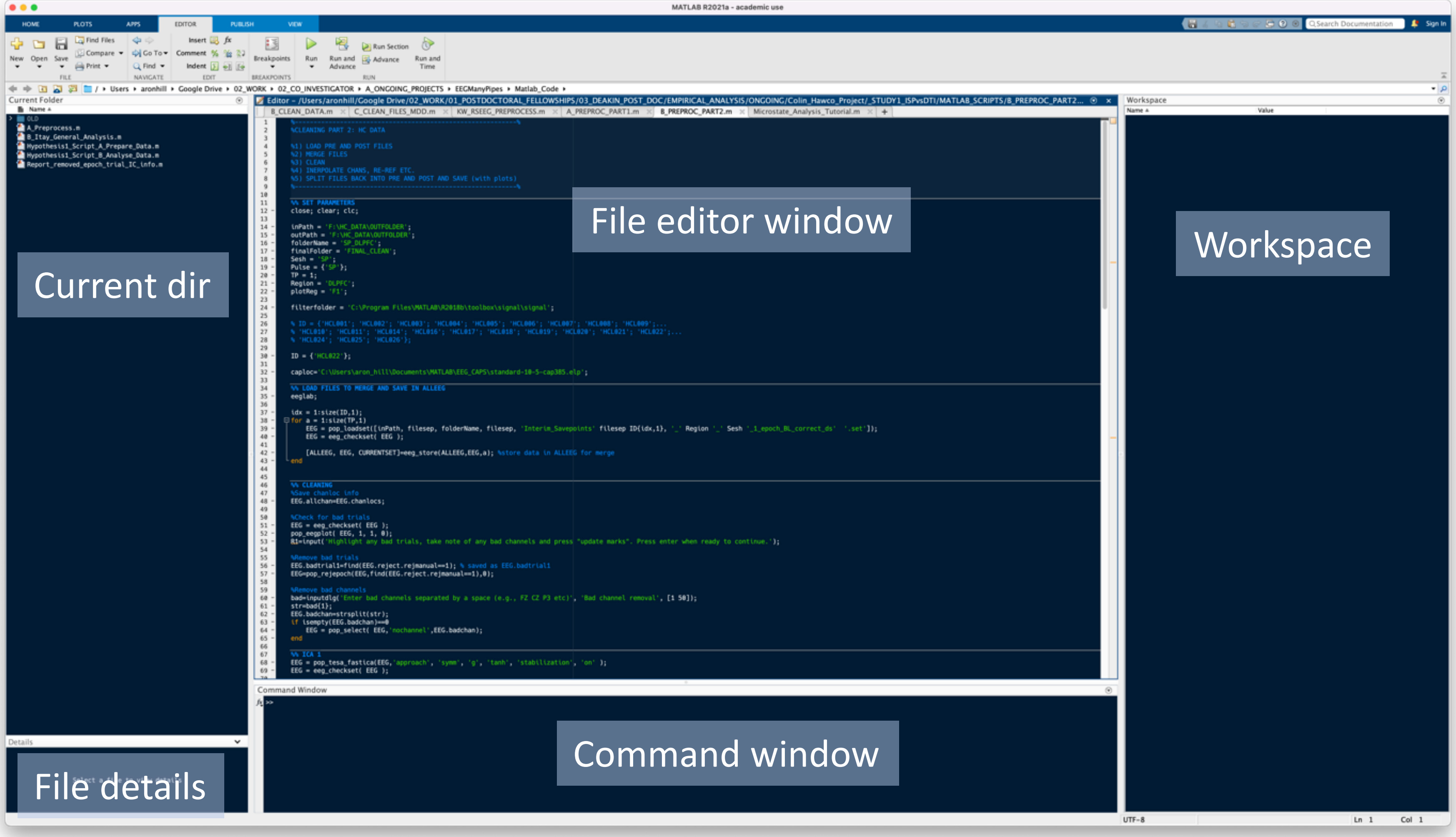
Command window

Workspace

R



```
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[workspace loaded from ~/.RData]

>
```



/ ▸ Users ▸ aronhill ▸ Google Drive ▸ 02_WORK ▸ 02_CO_INVESTIGATOR ▸ A_ONGOING_PROJECTS ▸ EEGManyPipes ▸ Matlab_Code ▸

Current Folder:

- OLD
- A_Preprocess.m
- B_Itay_General_Analysis.m
- Hypothesis1_Script_A_Prepare_Data.m
- Hypothesis1_Script_B_Analyse_Data.m
- Report_removed_epoch_trial_IC_info.m

Editor - /Users/aronhill/Google Drive/02_WORK/01_POSTDOCTORAL_FELLOWSHIPS/03_DEAKIN_POST_DOC/EMPIRICAL_ANALYSIS/ONGOING/Colin_Hawco_Project/_STUDY1_ISPvsDTI/MATLAB_SCRIPTS/B_PREPROC_PART2...

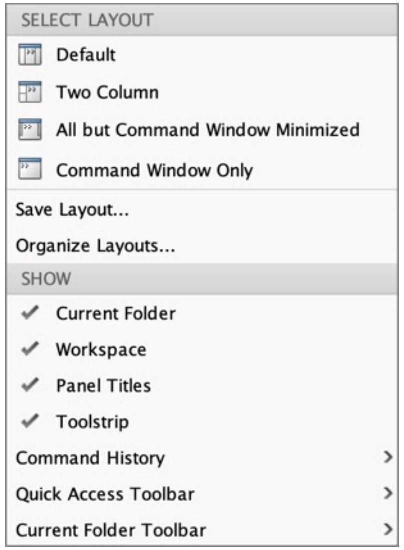
B_CLEAN_DATA.m | C_CLEAN_FILES_MDD.m | KW_RSEEG_PREPROCESS.m | A_PREPROC_PART1.m | B_PREPROC_PART2.m | Microstate_Analysis_Tutorial.m

```
%---------------------------------------------------%
%CLEANING PART 2: HC DATA

%1) LOAD PRE AND POST FILES
%2) MERGE FILES
%3) CLEAN
%4) INERPOLATE CHANS, RE-REF ETC.
%5) SPLIT FILES BACK INTO PRE AND POST AND SAVE (with plots)
%---------------------------------------------------%

%% SET PARAMETERS
close; clear; clc;

inPath = 'F:\HC_DATA\OUTFOLDER';
outPath = 'F:\HC_DATA\OUTFOLDER';
folderName = 'SP_DLPFC';
finalFolder = 'FINAL_CLEAN';
Sesh = 'SP';
Pulse = {'SP'};
TP = 1;
Region = 'DLPFC';
plotReg = 'F1';

filterfolder = 'C:\Program Files\MATLAB\R2018b\toolbox\signal\signal';

% ID = {'HCL001'; 'HCL002'; 'HCL003'; 'HCL004'; 'HCL005'; 'HCL006'; 'HCL007'; 'HCL008'; 'HCL009';...
% 'HCL010'; 'HCL011'; 'HCL014'; 'HCL016'; 'HCL017'; 'HCL018'; 'HCL019'; 'HCL020'; 'HCL021'; 'HCL022';...
% 'HCL024'; 'HCL025'; 'HCL026'};

ID = {'HCL022'};

caploc='C:\Users\aron_hill\Documents\MATLAB\EEG_CAPS\standard-10-5-cap385.elp';

%% LOAD FILES TO MERGE AND SAVE IN ALLEEG
eeglab;

idx = 1:size(ID,1);
for a = 1:size(TP,1)
    EEG = pop_loadset([inPath, filesep, folderName, filesep, 'Interim_Savepoints' filesep ID{idx,1}, '_' Region '_' Sesh '_1_epoch_BL_correct_ds'  '.set']);
    EEG = eeg_checkset( EEG );

    [ALLEEG, EEG, CURRENTSET]=eeg_store(ALLEEG,EEG,a); %store data in ALLEEG for merge
end


%% CLEANING
%Save chanloc info
EEG.allchan=EEG.chanlocs;

%Check for bad trials
EEG = eeg_checkset( EEG );
pop_eegplot( EEG, 1, 1, 0);
R1=input('Highlight any bad trials, take note of any bad channels and press "update marks". Press enter when ready to continue.');

%Remove bad trials
EEG.badtrial1=find(EEG.reject.rejmanual==1); % saved as EEG.badtrial1
EEG=pop_rejepoch(EEG,find(EEG.reject.rejmanual==1),0);

%Remove bad channels
bad=inputdlg('Enter bad channels separated by a space (e.g., FZ CZ P3 etc)', 'Bad channel removal', [1 50]);
str=bad{1};
EEG.badchan=strsplit(str);
if isempty(EEG.badchan)==0
    EEG = pop_select( EEG,'nochannel',EEG.badchan);
end

%% ICA 1
EEG = pop_tesa_fastica(EEG,'approach', 'symm', 'g', 'tanh', 'stabilization', 'on' );
EEG = eeg_checkset( EEG );
```

```
>>
```

MATLAB

Home > Layout



SELECT LAYOUT

- Default
- Two Column
- All but Command Window Minimized
- Command Window Only

Save Layout...

Organize Layouts...

SHOW

- ✓ Current Folder
- ✓ Workspace
- ✓ Panel Titles
- ✓ Toolstrip

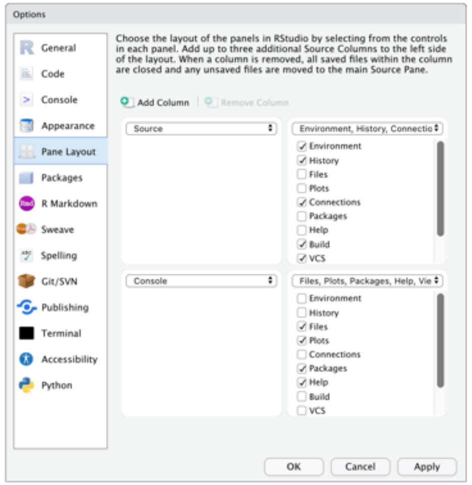
Command History ›

Quick Access Toolbar ›

Current Folder Toolbar ›

R

Tools > Global Options > Pane Layout



Choose the layout of the panels in RStudio by selecting from the controls in each panel. Add up to three additional Source Columns to the left side of the layout. When a column is removed, all saved files within the column are closed and any unsaved files are moved to the main Source Pane.

Source

Environment, History, Connectio

- ☑ Environment
- ☑ History
- ☐ Files
- ☐ Plots
- ☑ Connections
- ☐ Packages
- ☐ Help
- ☑ Build
- ☑ VCS

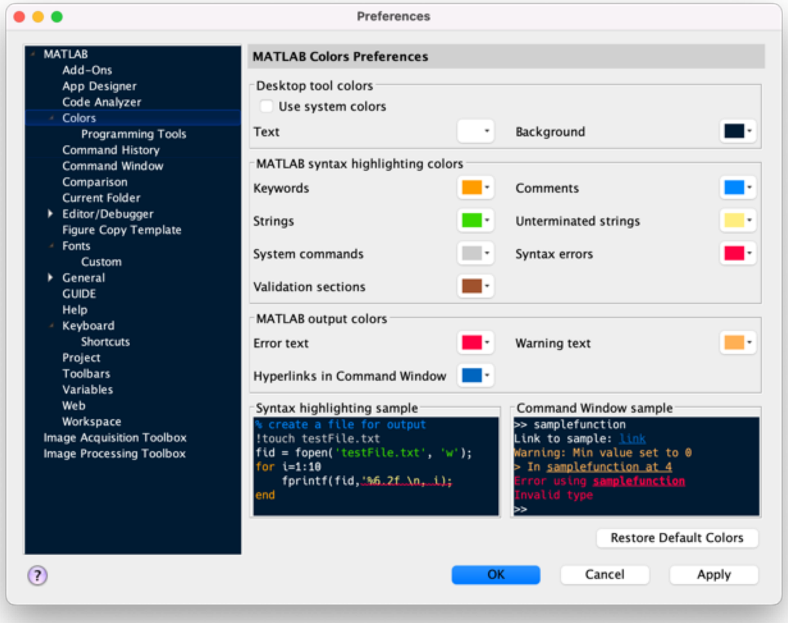
Console

Files, Plots, Packages, Help, Vie

- ☐ Environment
- ☐ History
- ☑ Files
- ☑ Plots
- ☐ Connections
- ☑ Packages
- ☑ Help
- ☐ Build
- ☐ VCS

Change colour scheme…

Option 1: MATLAB Preferences



MATLAB Colors Preferences

Desktop tool colors

□ Use system colors

Text	□	Background	■

MATLAB syntax highlighting colors

Keywords	🟧	Comments	🟦
Strings	🟩	Unterminated strings	🟨
System commands	⬜	Syntax errors	🟥
Validation sections	🟫		

MATLAB output colors

Error text	🟥	Warning text	🟧
Hyperlinks in Command Window	🟦		

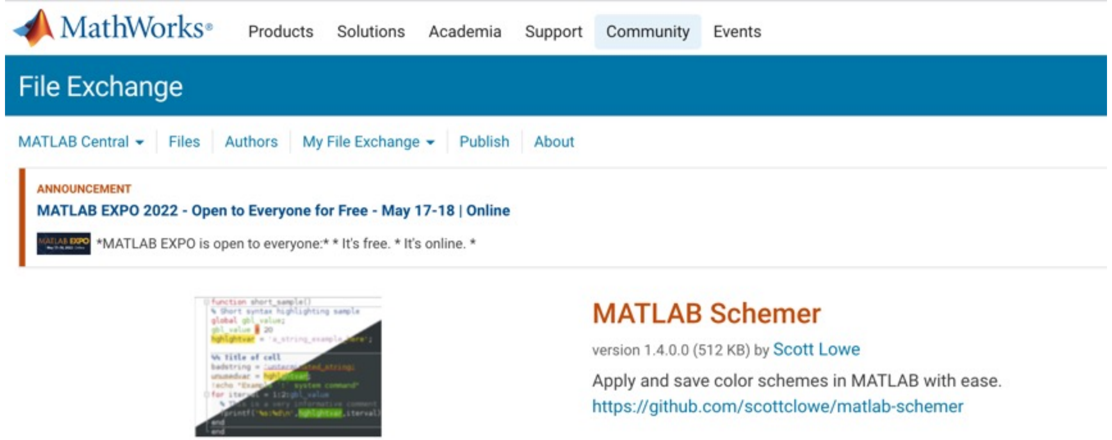
Syntax highlighting sample

```
% create a file for output
!touch testFile.txt
fid = fopen('testFile.txt', 'w');
for i=1:10
    fprintf(fid,'%6.2f \n, i);
end
```

Command Window sample

```
>> samplefunction
Link to sample: link
Warning: Min value set to 0
> In samplefunction at 4
Error using samplefunction
Invalid type
>>
```

Option 2: Custom made toolboxes



ANNOUNCEMENT

MATLAB EXPO 2022 - Open to Everyone for Free - May 17-18 | Online

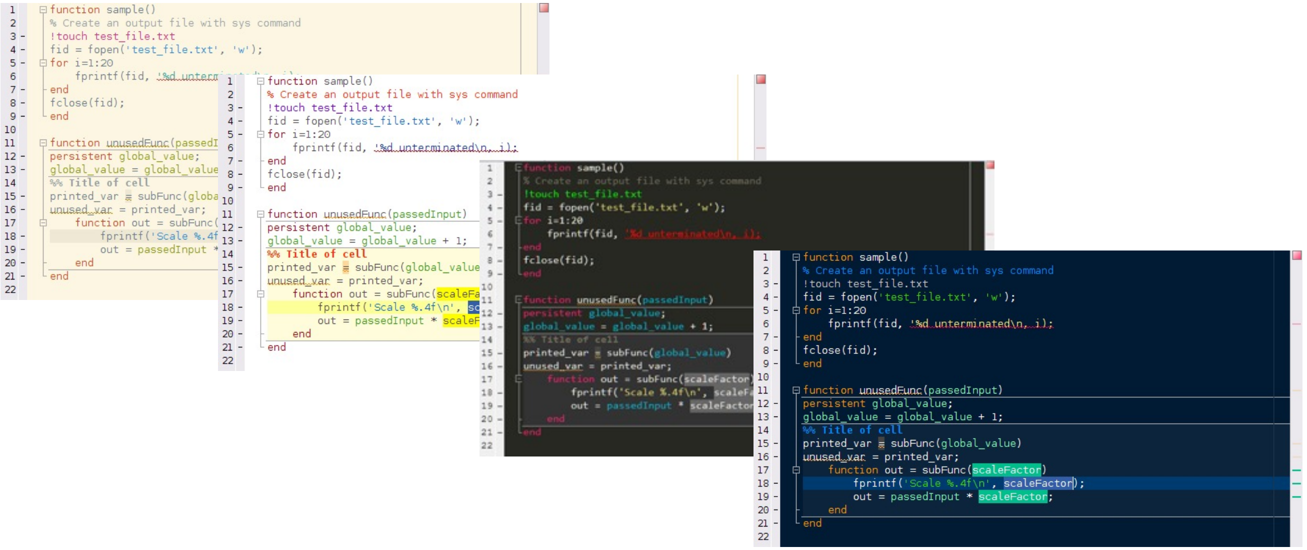
MATLAB EXPO is open to everyone: * It's free. * It's online. *

MATLAB Schemer

version 1.4.0.0 (512 KB) by Scott Lowe

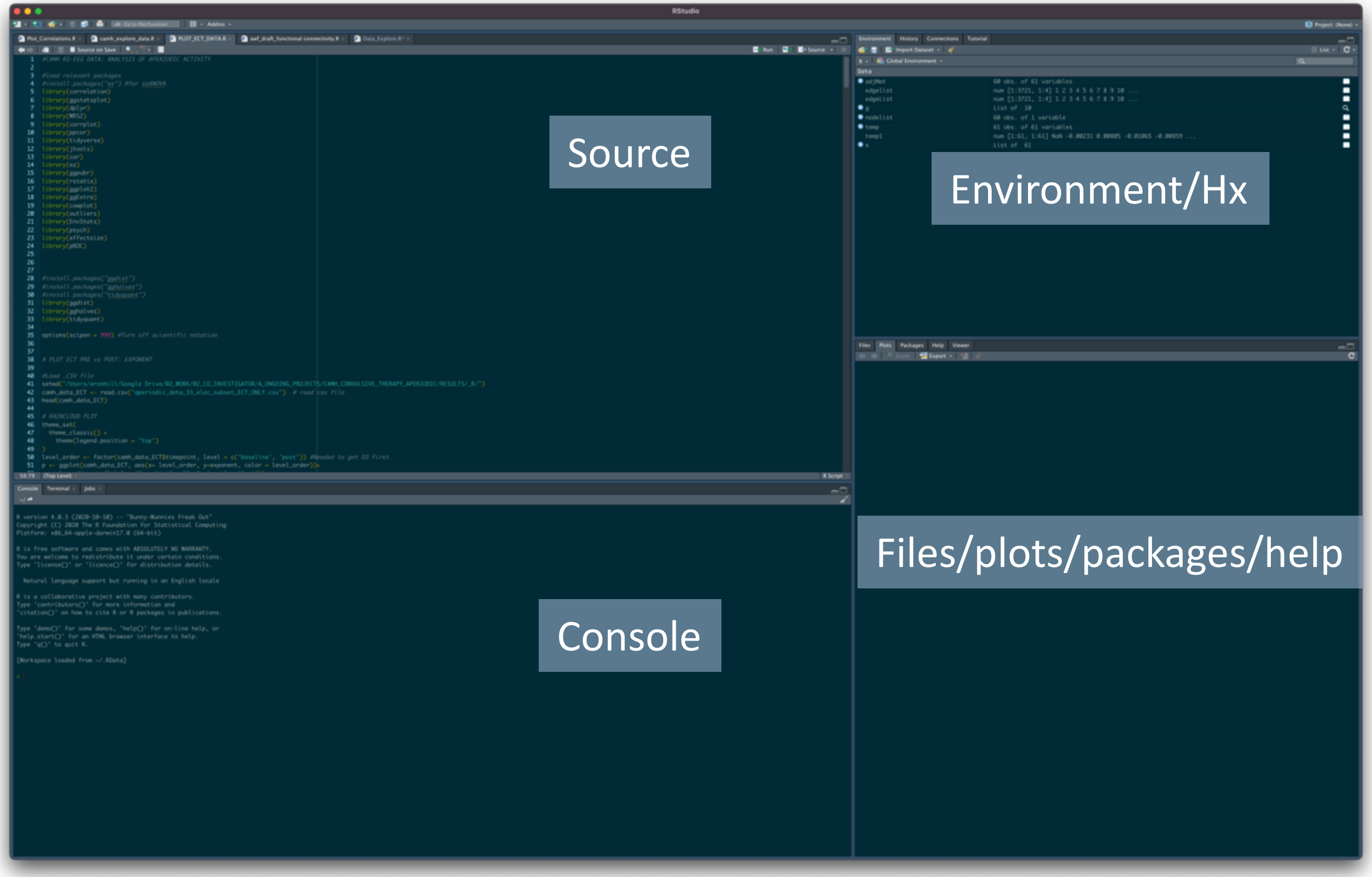
Apply and save color schemes in MATLAB with ease.

https://github.com/scottclowe/matlab-schemer



```
function sample()
% Create an output file with sys command
!touch test_file.txt
fid = fopen('test_file.txt', 'w');
for i=1:20
    fprintf(fid, '%d unterminated\n, i);
end
fclose(fid);
end

function unusedFunc(passedInput)
persistent global_value;
global_value = global_value + 1;
%% Title of cell
printed_var = subFunc(global_value)
unused_var = printed_var;
    function out = subFunc(scaleFactor)
        fprintf('Scale %.4f\n', scaleFactor);
        out = passedInput * scaleFactor;
    end
end
```

Change colour scheme (R)



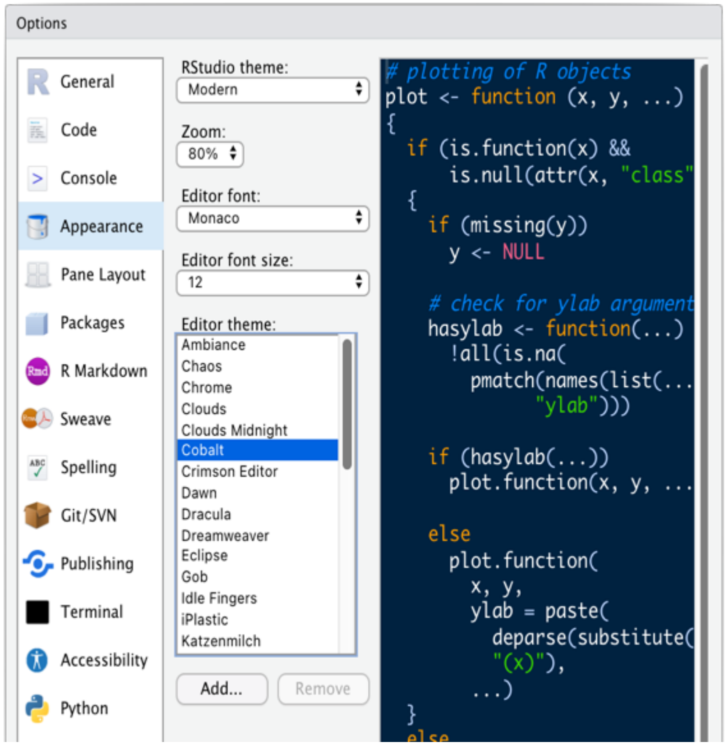
```
#CAMH RS-EEG DATA: ANALYSIS OF APERIODIC ACTIVITY

#load relevant packages
#install.packages("ez") #for ezANOVA
library(correlation)
library(ggstatsplot)
library(dplyr)
library(MASS)
library(corrplot)
library(ppcor)
library(tidyverse)
library(jtools)
library(car)
library(ez)
library(ggpubr)
library(rstatix)
library(ggplot2)
library(ggExtra)
library(cowplot)
library(outliers)
library(EnvStats)
library(psych)
library(effectsize)
library(pROC)



#install.packages("ggdist")
#install.packages("gghalves")
#install.packages("tidyquant")
library(ggdist)
library(gghalves)
library(tidyquant)

options(scipen = 999) #Turn off scientific notation


# PLOT ECT PRE vs POST: EXPONENT

#Load .CSV file
setwd("/Users/aronhill/Google Drive/02_WORK/02_CO_INVESTIGATOR/A_ONGOING_PROJECTS/CAMH_CONVULSIVE_THERAPY_APERIODIC/RESULTS/_R/")
camh_data_ECT <- read.csv("aperiodic_data_33_elec_subset_ECT_ONLY.csv")  # read csv file
head(camh_data_ECT)

# RAINCLOUD PLOT
theme_set(
  theme_classic() +
    theme(legend.position = "top")
)
level_order <- factor(camh_data_ECT$timepoint, level = c("baseline", 'post')) #Needed to get EO first.
p <- ggplot(camh_data_ECT, aes(x= level_order, y=exponent, color = level_order))+
```

```
adjMat          60 obs. of 61 variables
edgelist        num [1:3721, 1:4] 1 2 3 4 5 6 7 8 9 10 ...
edgeList        num [1:3721, 1:4] 1 2 3 4 5 6 7 8 9 10 ...
g               List of 10
nodelist        60 obs. of 1 variable
temp            61 obs. of 61 variables
temp1           num [1:61, 1:61] NaN -0.00231 0.00805 -0.01065 -0.00859 ...
x               List of 61
```

```
R version 4.0.3 (2020-10-10) -- "Bunny-Wunnies Freak Out"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Workspace loaded from ~/.RData]

> 
```

Change Theme

Tools → Global Options...



RStudio theme: Modern

Zoom: 80%

Editor font: Monaco

Editor font size: 12

Editor theme: Cobalt

```
# plotting of R objects
plot <- function (x, y, ...)
{
  if (is.function(x) &&
      is.null(attr(x, "class"
  {
    if (missing(y))
      y <- NULL

    # check for ylab argument
    hasylab <- function(...)
      !all(is.na(
        pmatch(names(list(...
          "ylab")))

    if (hasylab(...))
      plot.function(x, y, ...

    else
      plot.function(
        x, y,
        ylab = paste(
          deparse(substitute(
            "(x)"),
          ...)
  }
  else
```

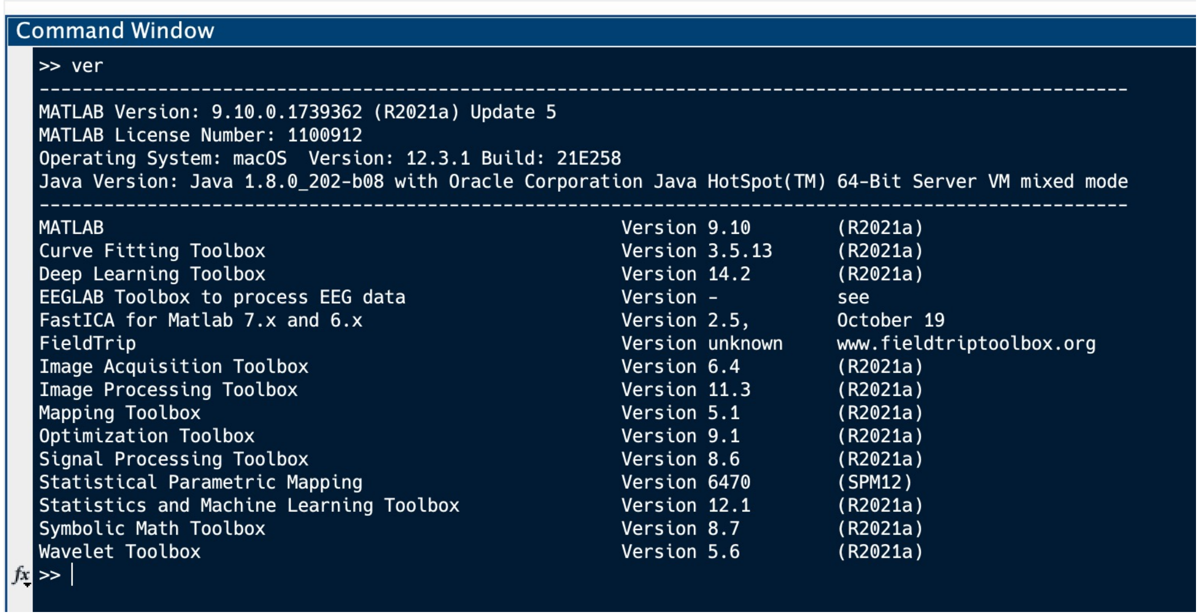
2. Installing Toolboxes

- Some toolboxes come as ‘add-ons’ to MATLAB and need to be installed via the MATLAB installer.
- What is available depends on university license
  - *Signal processing toolbox*
    - Filters, power spectra, wavelets – needed for some functions in EEGLAB/Fieldtrip
  - *Statistics and machine learning toolbox*
    - Large no. of functions needed for some EEGLAB/Fieldtrip functions
  - *Image processing toolbox*
    - Image processing, analysis, and visualization functions
- **Other third party toolboxes are free/open source**
  - EEGLAB: https://sccn.ucsd.edu/eeglab/ressources.php
  - Fieldtrip: https://www.fieldtriptoolbox.org/faq/requirements/

Checking installed toolboxes

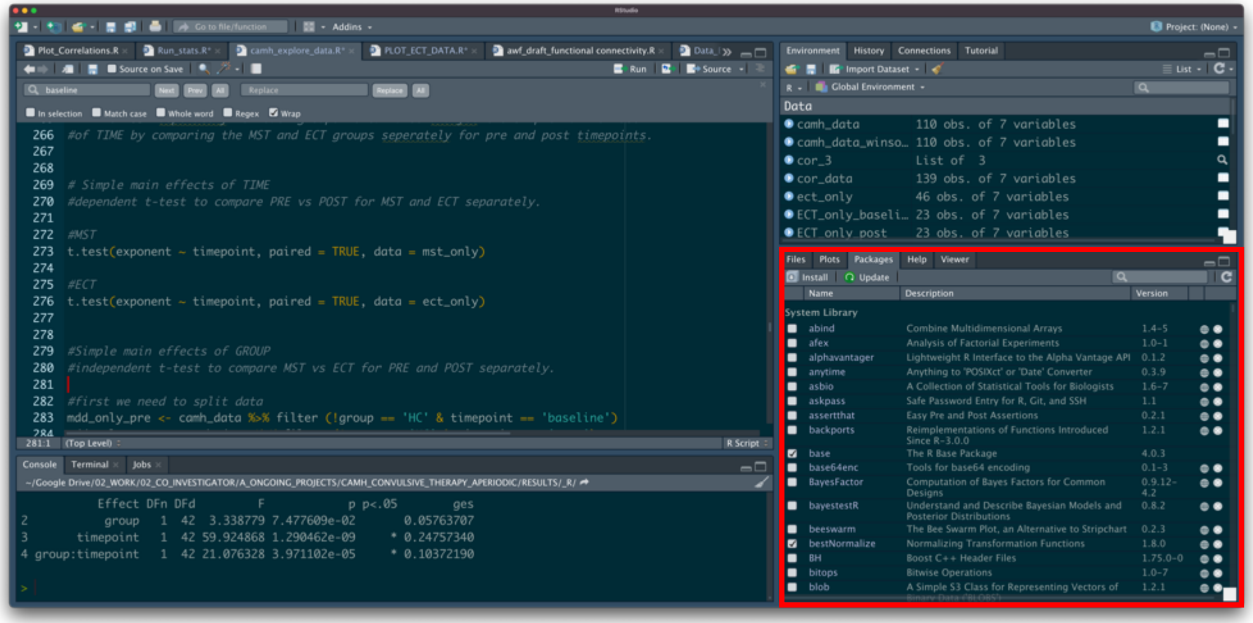
- Use the *'ver'* command to show a list of installed toolboxes
- Also displays MATLAB version information

MATLAB



```
>> ver
-----------------------------------------------------------------------------------------------------
MATLAB Version: 9.10.0.1739362 (R2021a) Update 5
MATLAB License Number: 1100912
Operating System: macOS  Version: 12.3.1 Build: 21E258
Java Version: Java 1.8.0_202-b08 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode
-----------------------------------------------------------------------------------------------------
MATLAB                                              Version 9.10        (R2021a)
Curve Fitting Toolbox                               Version 3.5.13      (R2021a)
Deep Learning Toolbox                               Version 14.2        (R2021a)
EEGLAB Toolbox to process EEG data                  Version -           see
FastICA for Matlab 7.x and 6.x                      Version 2.5,        October 19
FieldTrip                                           Version unknown     www.fieldtriptoolbox.org
Image Acquisition Toolbox                           Version 6.4         (R2021a)
Image Processing Toolbox                            Version 11.3        (R2021a)
Mapping Toolbox                                     Version 5.1         (R2021a)
Optimization Toolbox                                Version 9.1         (R2021a)
Signal Processing Toolbox                           Version 8.6         (R2021a)
Statistical Parametric Mapping                      Version 6470        (SPM12)
Statistics and Machine Learning Toolbox             Version 12.1        (R2021a)
Symbolic Math Toolbox                               Version 8.7         (R2021a)
Wavelet Toolbox                                     Version 5.6         (R2021a)
>> |
```

R



```
#of TIME by comparing the MST and ECT groups seperately for pre and post timepoints.


# Simple main effects of TIME
#dependent t-test to compare PRE vs POST for MST and ECT separately.

#MST
t.test(exponent ~ timepoint, paired = TRUE, data = mst_only)

#ECT
t.test(exponent ~ timepoint, paired = TRUE, data = ect_only)


#Simple main effects of GROUP
#independent t-test to compare MST vs ECT for PRE and POST separately.

#first we need to split data
mdd_only_pre <- camh_data %>% filter (!group == 'HC' & timepoint == 'baseline')
```

```
~/Google Drive/02_WORK/02_CO_INVESTIGATOR/A_ONGOING_PROJECTS/CAMH_CONVULSIVE_THERAPY_APERIODIC/RESULTS/_R/
          Effect DFn DFd         F            p p<.05        ges
2          group   1  42  3.338779 7.477609e-02       0.05763707
3      timepoint   1  42 59.924868 1.290462e-09     * 0.24757340
4 group:timepoint   1  42 21.076328 3.971102e-05     * 0.10372190

>
```

Data

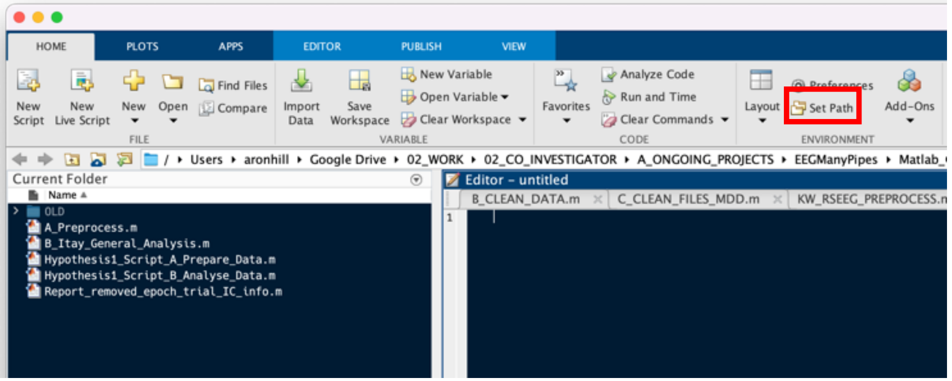
camh_data	110 obs. of 7 variables
camh_data_winso…	110 obs. of 7 variables
cor_3	List of 3
cor_data	139 obs. of 7 variables
ect_only	46 obs. of 7 variables
ECT_only_baseli…	23 obs. of 7 variables
ECT_only_post	23 obs. of 7 variables

System Library

	Name	Description	Version
□	abind	Combine Multidimensional Arrays	1.4-5
□	afex	Analysis of Factorial Experiments	1.0-1
□	alphavantager	Lightweight R Interface to the Alpha Vantage API	0.1.2
□	anytime	Anything to 'POSIXct' or 'Date' Converter	0.3.9
□	asbio	A Collection of Statistical Tools for Biologists	1.6-7
□	askpass	Safe Password Entry for R, Git, and SSH	1.1
□	assertthat	Easy Pre and Post Assertions	0.2.1
□	backports	Reimplementations of Functions Introduced Since R-3.0.0	1.2.1
☑	base	The R Base Package	4.0.3
□	base64enc	Tools for base64 encoding	0.1-3
□	BayesFactor	Computation of Bayes Factors for Common Designs	0.9.12-4.2
□	bayestestR	Understand and Describe Bayesian Models and Posterior Distributions	0.8.2
□	beeswarm	The Bee Swarm Plot, an Alternative to Stripchart	0.2.3
☑	bestNormalize	Normalizing Transformation Functions	1.8.0
□	BH	Boost C++ Header Files	1.75.0-0
□	bitops	Bitwise Operations	1.0-7
□	blob	A Simple S3 Class for Representing Vectors of Binary Data ('BLOBS')	1.2.1

2. Installing Toolboxes

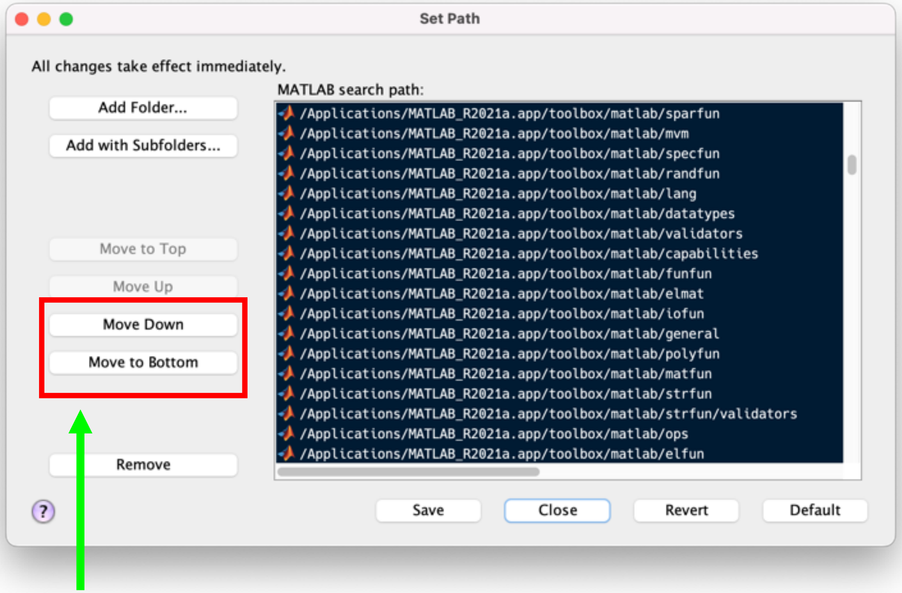
- Mathworks toolboxes can be selected at MATLAB installation
- Third party toolboxes need to be installed manually
    - Important to keep track of installed toolboxes
        - This will making backing up easy prior to MATLAB upgrades etc.
        - Generally best to keep in MATLAB directory
            - Mac: /Users/aronhill/Documents/MATLAB
            - Can create ‘External_Toolboxes’ folder for storage
    - Add files to path in MATLAB
        - Needed in order for MATLAB to search for files/functions



/ ▸ Users ▸ aronhill ▸ Google Drive ▸ 02_WORK ▸ 02_CO_INVESTIGATOR ▸ A_ONGOING_PROJECTS ▸ EEGManyPipes ▸ Matlab_

B_CLEAN_DATA.m | C_CLEAN_FILES_MDD.m | KW_RSEEG_PREPROCESS.m

```
OLD
A_Preprocess.m
B_Itay_General_Analysis.m
Hypothesis1_Script_A_Prepare_Data.m
Hypothesis1_Script_B_Analyse_Data.m
Report_removed_epoch_trial_IC_info.m
```



All changes take effect immediately.

MATLAB search path:

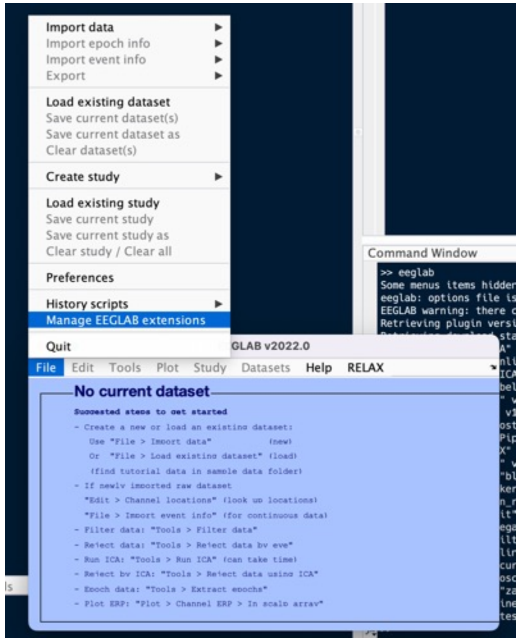
```
/Applications/MATLAB_R2021a.app/toolbox/matlab/sparfun
/Applications/MATLAB_R2021a.app/toolbox/matlab/mvm
/Applications/MATLAB_R2021a.app/toolbox/matlab/specfun
/Applications/MATLAB_R2021a.app/toolbox/matlab/randfun
/Applications/MATLAB_R2021a.app/toolbox/matlab/lang
/Applications/MATLAB_R2021a.app/toolbox/matlab/datatypes
/Applications/MATLAB_R2021a.app/toolbox/matlab/validators
/Applications/MATLAB_R2021a.app/toolbox/matlab/capabilities
/Applications/MATLAB_R2021a.app/toolbox/matlab/funfun
/Applications/MATLAB_R2021a.app/toolbox/matlab/elmat
/Applications/MATLAB_R2021a.app/toolbox/matlab/iofun
/Applications/MATLAB_R2021a.app/toolbox/matlab/general
/Applications/MATLAB_R2021a.app/toolbox/matlab/polyfun
/Applications/MATLAB_R2021a.app/toolbox/matlab/matfun
/Applications/MATLAB_R2021a.app/toolbox/matlab/strfun
/Applications/MATLAB_R2021a.app/toolbox/matlab/strfun/validators
/Applications/MATLAB_R2021a.app/toolbox/matlab/ops
/Applications/MATLAB_R2021a.app/toolbox/matlab/elfun
```

Tip: Files contained at the top of the search path take precedence over those lower down. When adding new toolboxes to the MATLAB path, generally a good idea to move to bottom so MATLAB functions are given priority.

EEGLAB toolboxes

For EEGLAB, additional toolboxes can be added in two ways:

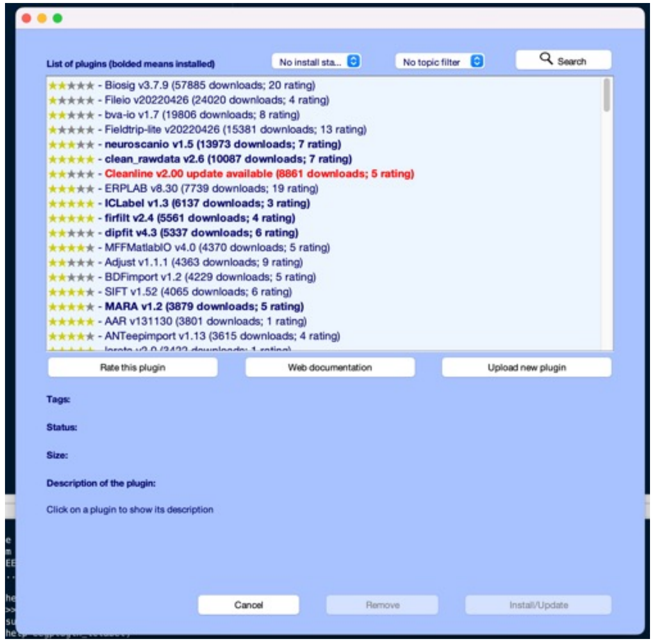
- 1) Directly via the GUI
- 2) download and add to EEGLAB plugins folder
  - EEGLAB will then search this folder



```
>> eeglab
Some menus items hidder
eeglab: options file is
EEGLAB warning: there c
Retrieving plugin versi
```

No current dataset

```
Suggested steps to get started
- Create a new or load an existing dataset:
   Use "File > Import data"           (new)
   Or  "File > Load existing dataset" (load)
   (find tutorial data in sample data folder)
- If newly imported raw dataset
  "Edit > Channel locations" (look up locations)
  "File > Import event info" (for continuous data)
- Filter data: "Tools > Filter data"
- Reject data: "Tools > Reject data by eye"
- Run ICA: "Tools > Run ICA" (can take time)
- Reject by ICA: "Tools > Reject data using ICA"
- Epoch data: "Tools > Extract epochs"
- Plot ERP: "Plot > Channel ERP > In scalp array"
```



List of plugins (bolded means installed)

★★★★★ - Biosig v3.7.9 (57885 downloads; 20 rating)
★★★★★ - Fileio v20220426 (24020 downloads; 4 rating)
★★★★★ - bva-io v1.7 (19806 downloads; 8 rating)
★★★★★ - Fieldtrip-lite v20220426 (15381 downloads; 13 rating)
★★★★★ - **neuroscanio v1.5 (13973 downloads; 7 rating)**
★★★★★ - **clean_rawdata v2.6 (10087 downloads; 7 rating)**
★★★★★ - **Cleanline v2.00 update available (8861 downloads; 5 rating)**
★★★★★ - ERPLAB v8.30 (7739 downloads; 19 rating)
★★★★★ - **ICLabel v1.3 (6137 downloads; 3 rating)**
★★★★★ - **firfilt v2.4 (5561 downloads; 4 rating)**
★★★★★ - **dipfit v4.3 (5337 downloads; 6 rating)**
★★★★★ - MFFMatlabIO v4.0 (4370 downloads; 5 rating)
★★★★★ - Adjust v1.1.1 (4363 downloads; 9 rating)
★★★★★ - BDFimport v1.2 (4229 downloads; 5 rating)
★★★★★ - SIFT v1.52 (4065 downloads; 6 rating)
★★★★★ - **MARA v1.2 (3879 downloads; 5 rating)**
★★★★★ - AAR v131130 (3801 downloads; 1 rating)
★★★★★ - ANTeepimport v1.13 (3615 downloads; 4 rating)

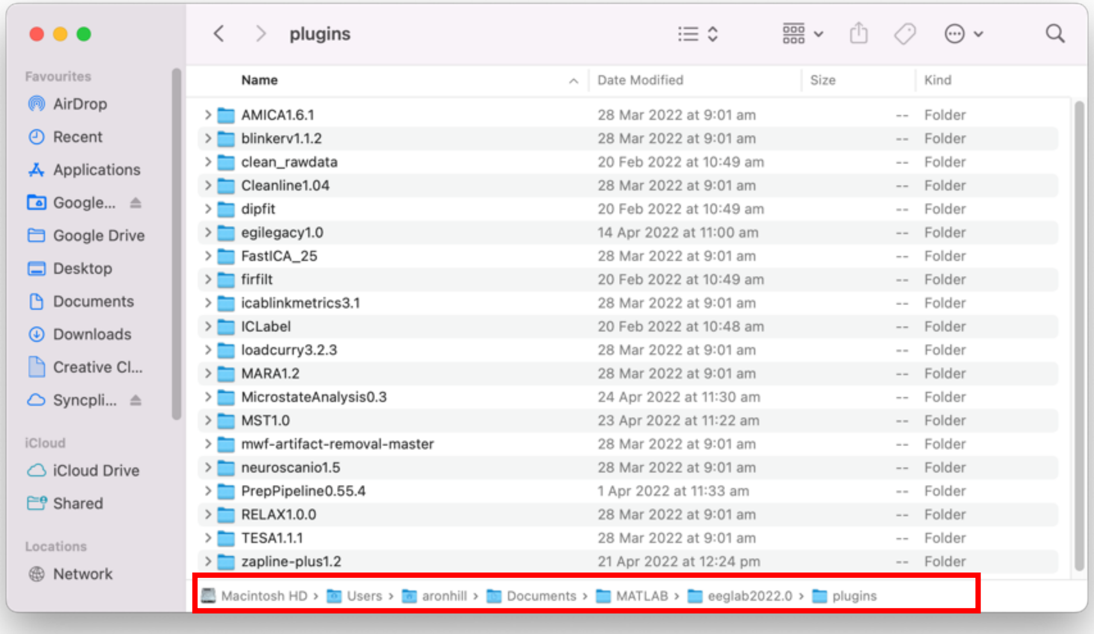
Tags:

Status:

Size:

Description of the plugin:

Click on a plugin to show its description



Name	Date Modified	Size	Kind
AMICA1.6.1	28 Mar 2022 at 9:01 am	--	Folder
blinkerv1.1.2	28 Mar 2022 at 9:01 am	--	Folder
clean_rawdata	20 Feb 2022 at 10:49 am	--	Folder
Cleanline1.04	28 Mar 2022 at 9:01 am	--	Folder
dipfit	20 Feb 2022 at 10:49 am	--	Folder
egilegacy1.0	14 Apr 2022 at 11:00 am	--	Folder
FastICA_25	28 Mar 2022 at 9:01 am	--	Folder
firfilt	20 Feb 2022 at 10:49 am	--	Folder
icablinkmetrics3.1	28 Mar 2022 at 9:01 am	--	Folder
ICLabel	20 Feb 2022 at 10:48 am	--	Folder
loadcurry3.2.3	28 Mar 2022 at 9:01 am	--	Folder
MARA1.2	28 Mar 2022 at 9:01 am	--	Folder
MicrostateAnalysis0.3	24 Apr 2022 at 11:30 am	--	Folder
MST1.0	23 Apr 2022 at 11:22 am	--	Folder
mwf-artifact-removal-master	28 Mar 2022 at 9:01 am	--	Folder
neuroscanio1.5	28 Mar 2022 at 9:01 am	--	Folder
PrepPipeline0.55.4	1 Apr 2022 at 11:33 am	--	Folder
RELAX1.0.0	28 Mar 2022 at 9:01 am	--	Folder
TESA1.1.1	28 Mar 2022 at 9:01 am	--	Folder
zapline-plus1.2	21 Apr 2022 at 12:24 pm	--	Folder

Macintosh HD › Users › aronhill › Documents › MATLAB › eeglab2022.0 › plugins

R Packages

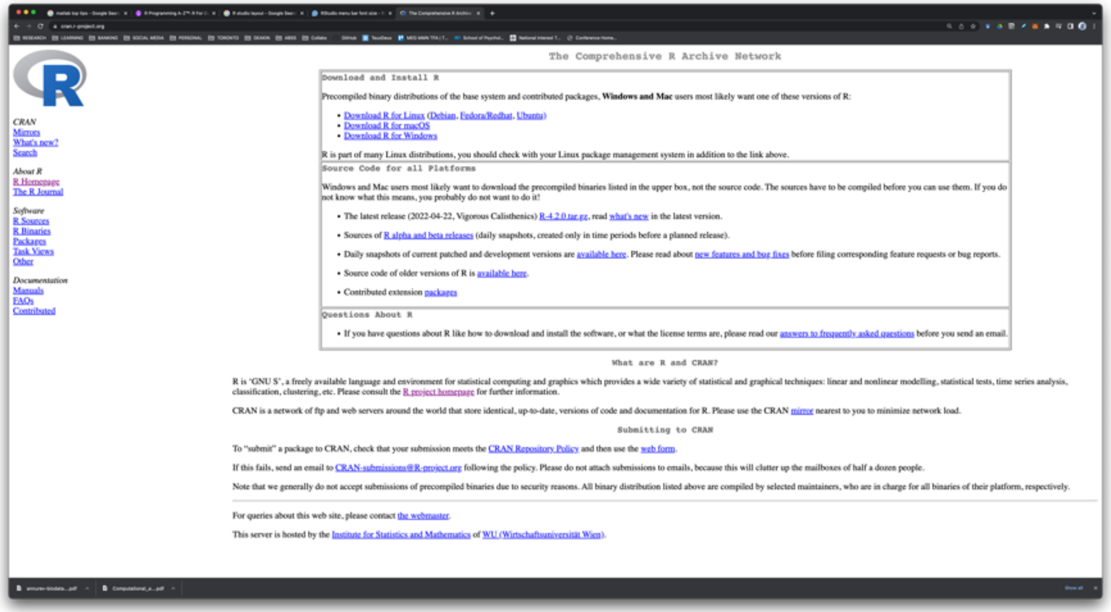
- Packages are the backbone of R – greatly expand its capabilities

- Stored on CRAN repository

- Packages can be easily installed from CRAN directly using code:
- Install.packages(“*package name*”)
  - E.g., install.packages(“ggplot2”)

Then to activate:

- Library(*package name*)
  - E.g., library(ggplot2)



The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2022-04-22, Vigorous Calisthenics) R-4.2.0.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

What are R and CRAN?

R is 'GNU S', a freely available language and environment for statistical computing and graphics which provides a wide variety of statistical and graphical techniques: linear and nonlinear modelling, statistical tests, time series analysis, classification, clustering, etc. Please consult the R project homepage for further information.

CRAN is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R. Please use the CRAN mirror nearest to you to minimize network load.

Submitting to CRAN

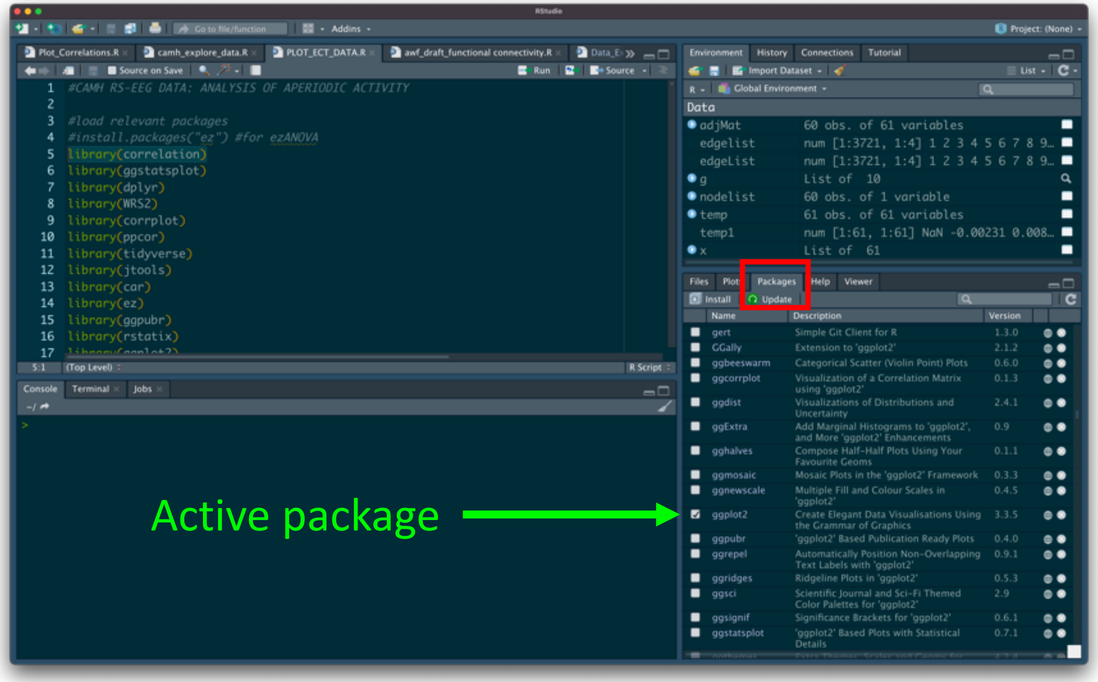
To "submit" a package to CRAN, check that your submission meets the CRAN Repository Policy and then use the web form.

If this fails, send an email to CRAN-submissions@R-project.org following the policy. Please do not attach submissions to emails, because this will clutter up the mailboxes of half a dozen people.

Note that we generally do not accept submissions of precompiled binaries due to security reasons. All binary distribution listed above are compiled by selected maintainers, who are in charge for all binaries of their platform, respectively.

For queries about this web site, please contact the webmaster.

This server is hosted by the Institute for Statistics and Mathematics of WU (Wirtschaftsuniversität Wien).



```r
#CAMH RS-EEG DATA: ANALYSIS OF APERIODIC ACTIVITY

#load relevant packages
#install.packages("ez") #for ezANOVA
library(correlation)
library(ggstatsplot)
library(dplyr)
library(WRS2)
library(corrplot)
library(ppcor)
library(tidyverse)
library(jtools)
library(car)
library(ez)
library(ggpubr)
library(rstatix)
library(ggplot2)
```

```
>
```

Data

adjMat	60 obs. of 61 variables
edgelist	num [1:3721, 1:4] 1 2 3 4 5 6 7 8 9…
edgeList	num [1:3721, 1:4] 1 2 3 4 5 6 7 8 9…
g	List of 10
nodelist	60 obs. of 1 variable
temp	61 obs. of 61 variables
temp1	num [1:61, 1:61] NaN -0.00231 0.008…
x	List of 61

	Name	Description	Version
☐	gert	Simple Git Client for R	1.3.0
☐	GGally	Extension to 'ggplot2'	2.1.2
☐	ggbeeswarm	Categorical Scatter (Violin Point) Plots	0.6.0
☐	ggcorrplot	Visualization of a Correlation Matrix using 'ggplot2'	0.1.3
☐	ggdist	Visualizations of Distributions and Uncertainty	2.4.1
☐	ggExtra	Add Marginal Histograms to 'ggplot2', and More 'ggplot2' Enhancements	0.9
☐	gghalves	Compose Half-Half Plots Using Your Favourite Geoms	0.1.1
☐	ggmosaic	Mosaic Plots in the 'ggplot2' Framework	0.3.3
☐	ggnewscale	Multiple Fill and Colour Scales in 'ggplot2'	0.4.5
☑	ggplot2	Create Elegant Data Visualisations Using the Grammar of Graphics	3.3.5
☐	ggpubr	'ggplot2' Based Publication Ready Plots	0.4.0
☐	ggrepel	Automatically Position Non-Overlapping Text Labels with 'ggplot2'	0.9.1
☐	ggridges	Ridgeline Plots in 'ggplot2'	0.5.3
☐	ggsci	Scientific Journal and Sci-Fi Themed Color Palettes for 'ggplot2'	2.9
☐	ggsignif	Significance Brackets for 'ggplot2'	0.6.1
☐	ggstatsplot	'ggplot2' Based Plots with Statistical Details	0.7.1

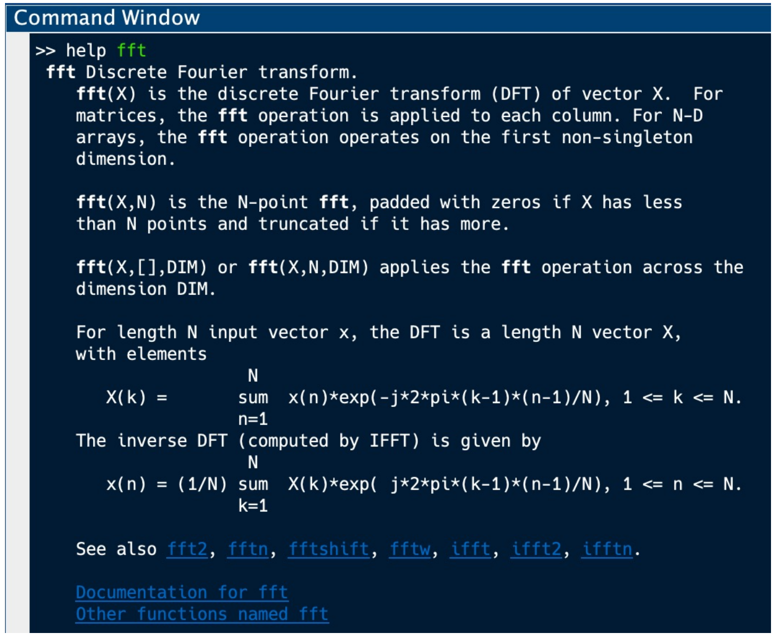
Active package

Use inbuilt help functions

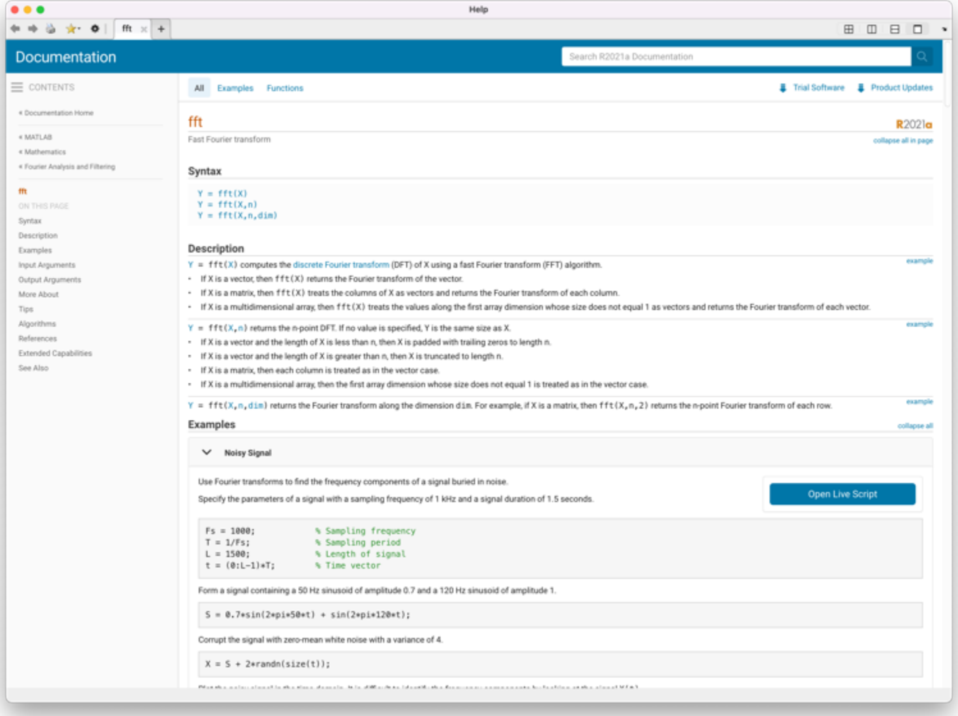
- Help *name of function*
  - Get help file associated with function
- Edit *name of function*
  - Opens file – useful to get more specific details
  - Also helpful as it opens in new window
- Doc *name of function*
  - More detailed info (where available), often including examples and figures

Mathworks discussion forum:

https://au.mathworks.com/matlabcentral/answers/index



```
>> help fft
 fft Discrete Fourier transform.
    fft(X) is the discrete Fourier transform (DFT) of vector X.  For
    matrices, the fft operation is applied to each column. For N-D
    arrays, the fft operation operates on the first non-singleton
    dimension.

    fft(X,N) is the N-point fft, padded with zeros if X has less
    than N points and truncated if it has more.

    fft(X,[],DIM) or fft(X,N,DIM) applies the fft operation across the
    dimension DIM.

    For length N input vector x, the DFT is a length N vector X,
    with elements
                     N
       X(k) =       sum  x(n)*exp(-j*2*pi*(k-1)*(n-1)/N), 1 <= k <= N.
                    n=1
    The inverse DFT (computed by IFFT) is given by
                     N
       x(n) = (1/N) sum  X(k)*exp( j*2*pi*(k-1)*(n-1)/N), 1 <= n <= N.
                    k=1

    See also fft2, fftn, fftshift, fftw, ifft, ifft2, ifftn.

    Documentation for fft
    Other functions named fft
```



fft

Fast Fourier transform

Syntax

```
Y = fft(X)
Y = fft(X,n)
Y = fft(X,n,dim)
```

Description

`Y = fft(X)` computes the discrete Fourier transform (DFT) of X using a fast Fourier transform (FFT) algorithm.

- If X is a vector, then `fft(X)` returns the Fourier transform of the vector.
- If X is a matrix, then `fft(X)` treats the columns of X as vectors and returns the Fourier transform of each column.
- If X is a multidimensional array, then `fft(X)` treats the values along the first array dimension whose size does not equal 1 as vectors and returns the Fourier transform of each vector.

`Y = fft(X,n)` returns the n-point DFT. If no value is specified, Y is the same size as X.

- If X is a vector and the length of X is less than n, then X is padded with trailing zeros to length n.
- If X is a vector and the length of X is greater than n, then X is truncated to length n.
- If X is a matrix, then each column is treated as in the vector case.
- If X is a multidimensional array, then the first array dimension whose size does not equal 1 is treated as in the vector case.

`Y = fft(X,n,dim)` returns the Fourier transform along the dimension dim. For example, if X is a matrix, then `fft(X,n,2)` returns the n-point Fourier transform of each row.

Examples

Noisy Signal

Use Fourier transforms to find the frequency components of a signal buried in noise.

Specify the parameters of a signal with a sampling frequency of 1 kHz and a signal duration of 1.5 seconds.

```
Fs = 1000;            % Sampling frequency
T = 1/Fs;             % Sampling period
L = 1500;             % Length of signal
t = (0:L-1)*T;        % Time vector
```

Form a signal containing a 50 Hz sinusoid of amplitude 0.7 and a 120 Hz sinusoid of amplitude 1.

```
S = 0.7*sin(2*pi*50*t) + sin(2*pi*120*t);
```

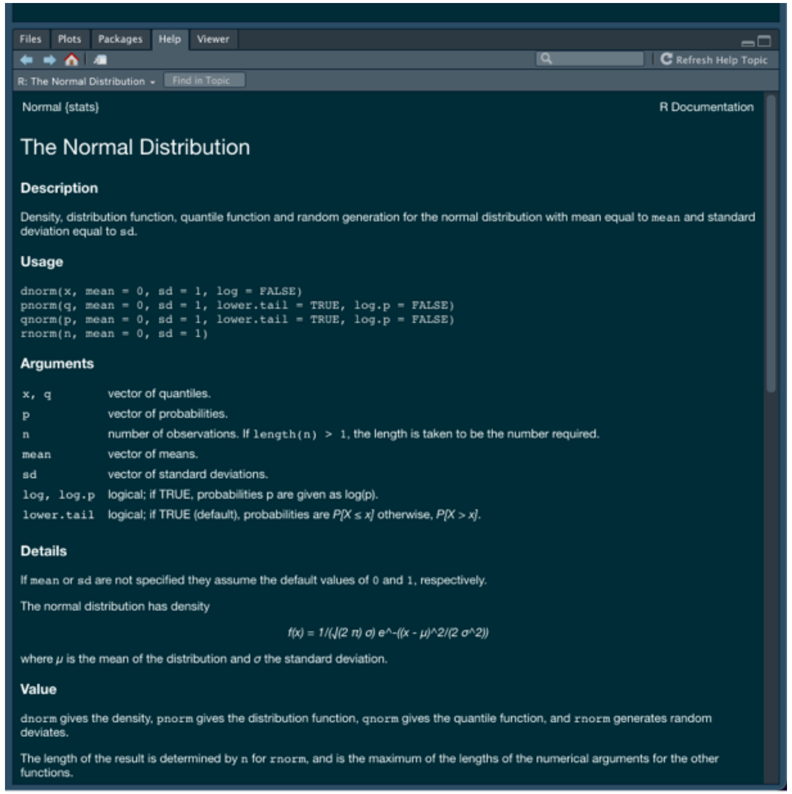
Corrupt the signal with zero-mean white noise with a variance of 4.

```
X = S + 2*randn(size(t));
```

Getting help (R)

- As with Matlab, R has excellent help on functions
  - ?*name of function*
- Consider using tidyverse packages in place of base R packages – often faster and more intuitive

?rnorm()



Normal {stats}   R Documentation

The Normal Distribution

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.

Details

If `mean` or `sd` are not specified they assume the default values of `0` and `1`, respectively.

The normal distribution has density

f(x) = 1/(√(2 π) σ) e^-((x - μ)^2/(2 σ^2))

where μ is the mean of the distribution and σ the standard deviation.

Value

`dnorm` gives the density, `pnorm` gives the distribution function, `qnorm` gives the quantile function, and `rnorm` generates random deviates.

The length of the result is determined by `n` for `rnorm`, and is the maximum of the lengths of the numerical arguments for the other functions.

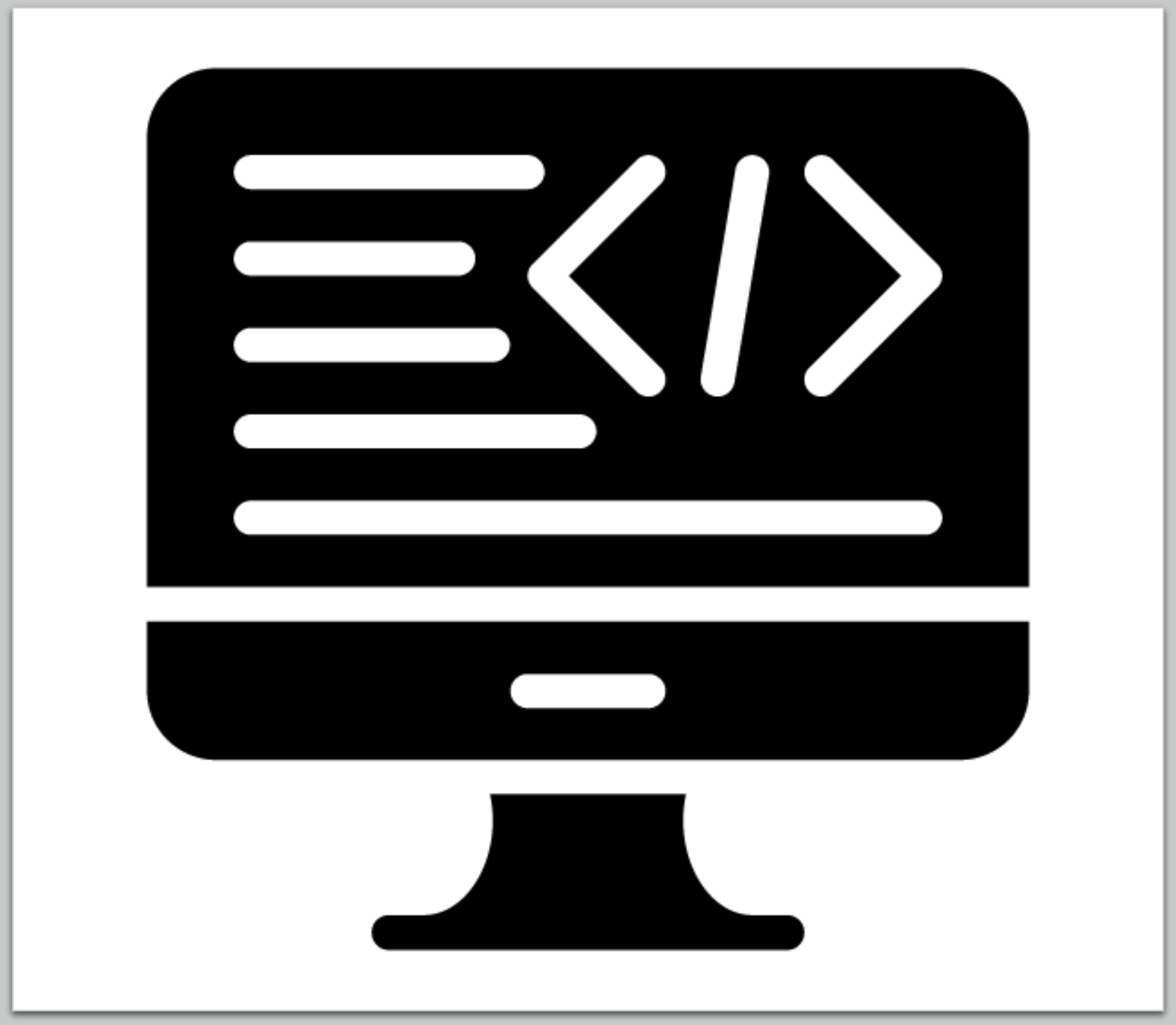
Building a script

- Creating a script from scratch can be daunting

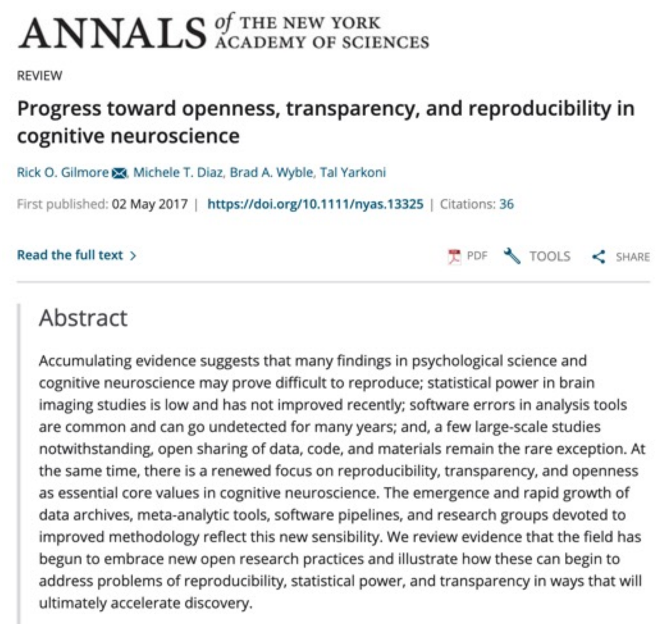
- Can often be helpful to first outline sections of a script using comments and then fill in the gaps

- This can help structure the script around a specific objective

- Also helps to break down complex analyses into achievable steps



Write sharable code



ANNALS *of* THE NEW YORK ACADEMY OF SCIENCES

REVIEW

Progress toward openness, transparency, and reproducibility in cognitive neuroscience

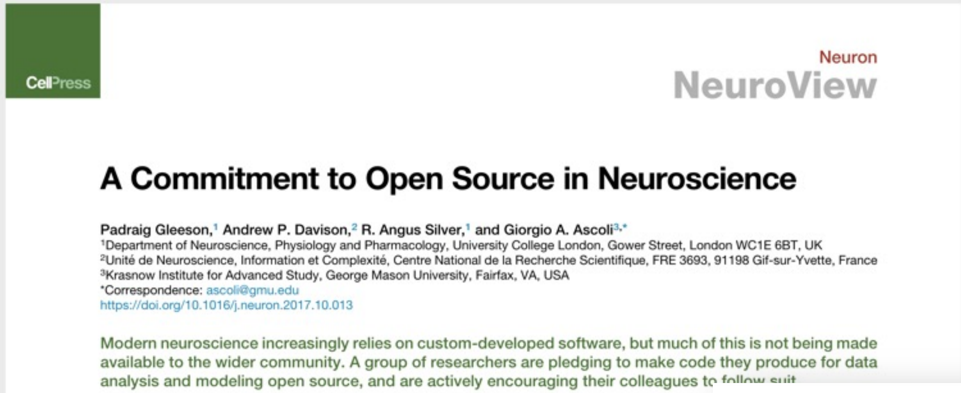
Rick O. Gilmore, Michele T. Diaz, Brad A. Wyble, Tal Yarkoni

First published: 02 May 2017 | https://doi.org/10.1111/nyas.13325 | Citations: 36

Read the full text › PDF TOOLS SHARE

Abstract

Accumulating evidence suggests that many findings in psychological science and cognitive neuroscience may prove difficult to reproduce; statistical power in brain imaging studies is low and has not improved recently; software errors in analysis tools are common and can go undetected for many years; and, a few large-scale studies notwithstanding, open sharing of data, code, and materials remain the rare exception. At the same time, there is a renewed focus on reproducibility, transparency, and openness as essential core values in cognitive neuroscience. The emergence and rapid growth of data archives, meta-analytic tools, software pipelines, and research groups devoted to improved methodology reflect this new sensibility. We review evidence that the field has begun to embrace new open research practices and illustrate how these can begin to address problems of reproducibility, statistical power, and transparency in ways that will ultimately accelerate discovery.



CellPress

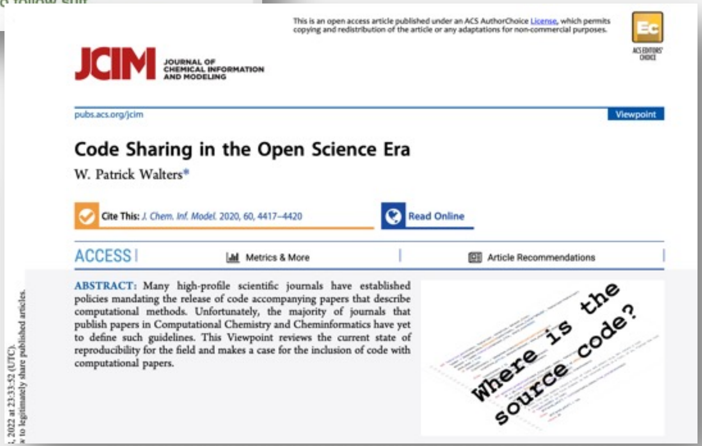
Neuron

NeuroView

A Commitment to Open Source in Neuroscience

Padraig Gleeson,[1] Andrew P. Davison,[2] R. Angus Silver,[1] and Giorgio A. Ascoli[3,*]
[1]Department of Neuroscience, Physiology and Pharmacology, University College London, Gower Street, London WC1E 6BT, UK
[2]Unité de Neuroscience, Information et Complexité, Centre National de la Recherche Scientifique, FRE 3693, 91198 Gif-sur-Yvette, France
[3]Krasnow Institute for Advanced Study, George Mason University, Fairfax, VA, USA
*Correspondence: ascoli@gmu.edu
https://doi.org/10.1016/j.neuron.2017.10.013

Modern neuroscience increasingly relies on custom-developed software, but much of this is not being made available to the wider community. A group of researchers are pledging to make code they produce for data analysis and modeling open source, and are actively encouraging their colleagues to follow suit



This is an open access article published under an ACS AuthorChoice License, which permits copying and redistribution of the article or any adaptations for non-commercial purposes.

Ec ACS EDITORS' CHOICE

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

pubs.acs.org/jcim Viewpoint

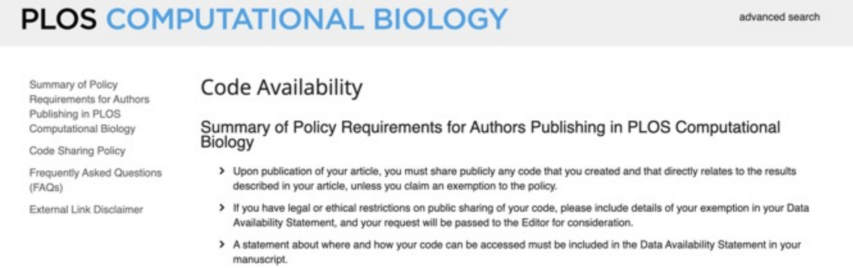
Code Sharing in the Open Science Era

W. Patrick Walters*

Cite This: *J. Chem. Inf. Model.* 2020, 60, 4417–4420 Read Online

ACCESS | Metrics & More | Article Recommendations |

ABSTRACT: Many high-profile scientific journals have established policies mandating the release of code accompanying papers that describe computational methods. Unfortunately, the majority of journals that publish papers in Computational Chemistry and Cheminformatics have yet to define such guidelines. This Viewpoint reviews the current state of reproducibility for the field and makes a case for the inclusion of code with computational papers.



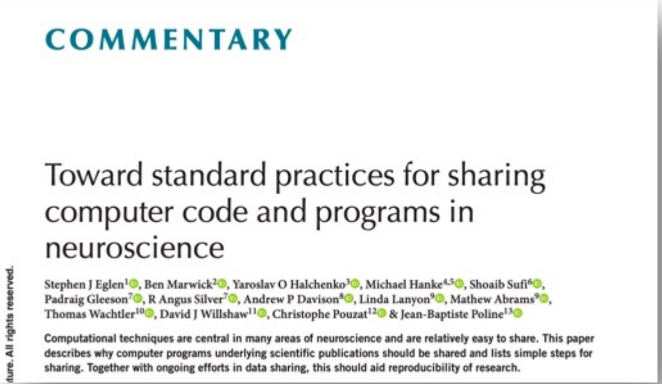
PLOS COMPUTATIONAL BIOLOGY advanced search

Summary of Policy Requirements for Authors Publishing in PLOS Computational Biology
Code Sharing Policy
Frequently Asked Questions (FAQs)
External Link Disclaimer

Code Availability

Summary of Policy Requirements for Authors Publishing in PLOS Computational Biology

- Upon publication of your article, you must share publicly any code that you created and that directly relates to the results described in your article, unless you claim an exemption to the policy.
- If you have legal or ethical restrictions on public sharing of your code, please include details of your exemption in your Data Availability Statement, and your request will be passed to the Editor for consideration.
- A statement about where and how your code can be accessed must be included in the Data Availability Statement in your manuscript.



COMMENTARY

Toward standard practices for sharing computer code and programs in neuroscience

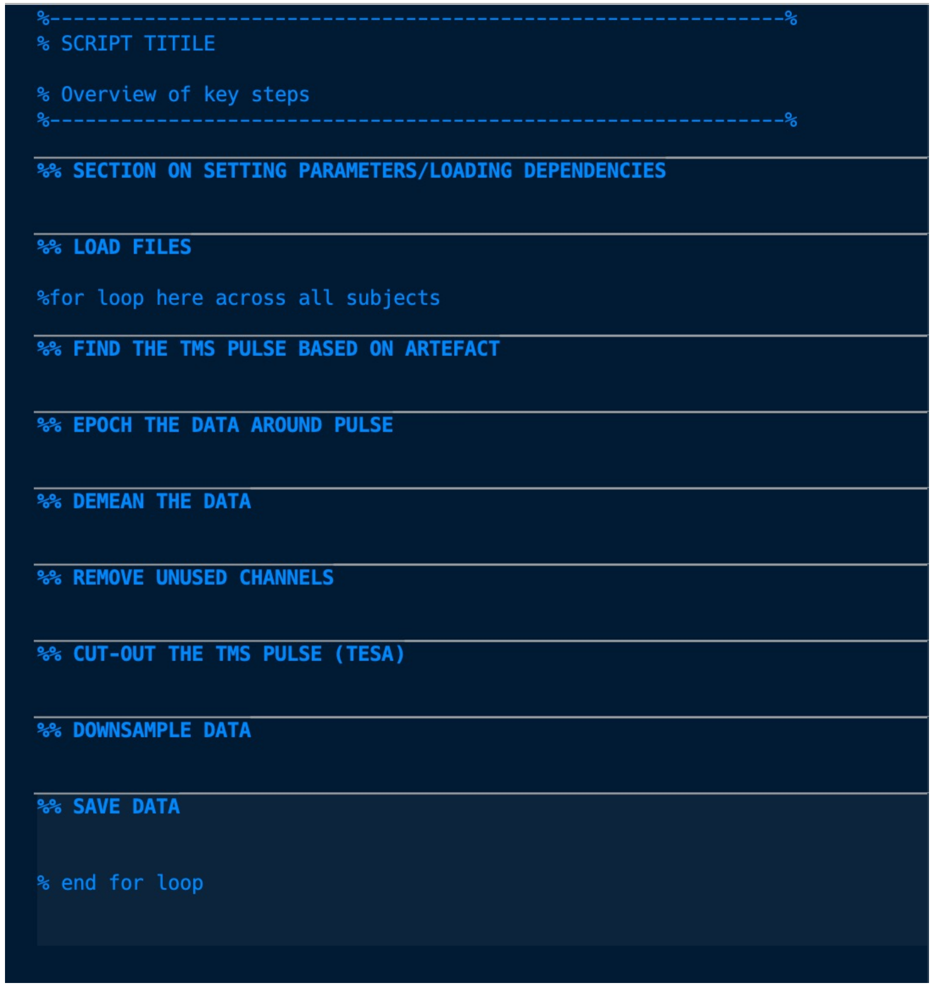
Stephen J Eglen[1], Ben Marwick[2], Yaroslav O Halchenko[3], Michael Hanke[4,5], Shoaib Sufi[6], Padraig Gleeson[7], R Angus Silver[7], Andrew P Davison[8], Linda Lanyon[9], Mathew Abrams[9], Thomas Wachtler[10], David J Willshaw[11], Christophe Pouzat[12] & Jean-Baptiste Poline[13]

Computational techniques are central in many areas of neuroscience and are relatively easy to share. This paper describes why computer programs underlying scientific publications should be shared and lists simple steps for sharing. Together with ongoing efforts in data sharing, this should aid reproducibility of research.

GitHub

Example MATLAB Script

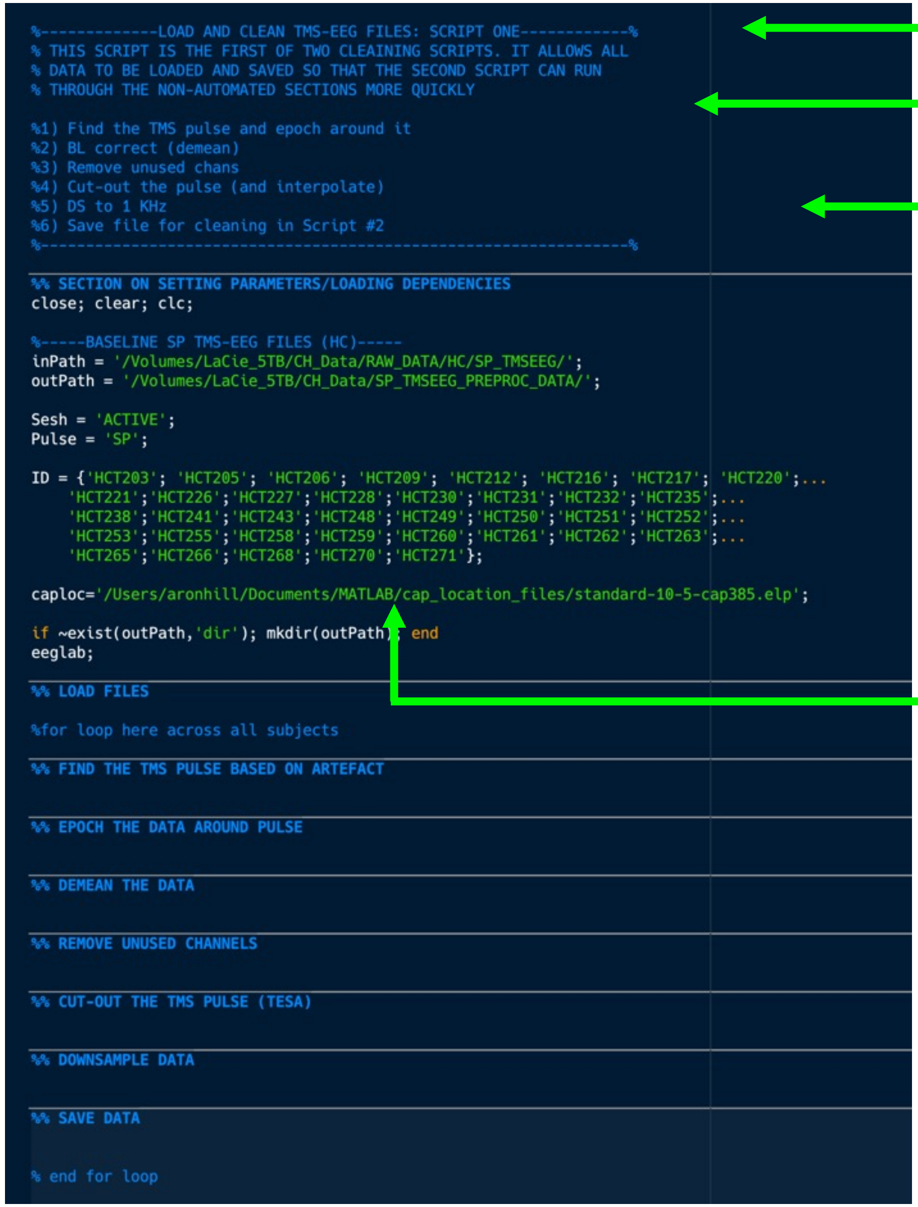
Load EEG files and preprocess



```
%---------------------------------------------------------------%
% SCRIPT TITILE

% Overview of key steps
%---------------------------------------------------------------%

%% SECTION ON SETTING PARAMETERS/LOADING DEPENDENCIES


%% LOAD FILES

%for loop here across all subjects

%% FIND THE TMS PULSE BASED ON ARTEFACT


%% EPOCH THE DATA AROUND PULSE


%% DEMEAN THE DATA


%% REMOVE UNUSED CHANNELS


%% CUT-OUT THE TMS PULSE (TESA)


%% DOWNSAMPLE DATA


%% SAVE DATA


% end for loop
```

Step 1: Create an outline

Step 2: Fill in the blanks…



```
%-------------LOAD AND CLEAN TMS-EEG FILES: SCRIPT ONE------------%
% THIS SCRIPT IS THE FIRST OF TWO CLEAINING SCRIPTS. IT ALLOWS ALL
% DATA TO BE LOADED AND SAVED SO THAT THE SECOND SCRIPT CAN RUN
% THROUGH THE NON-AUTOMATED SECTIONS MORE QUICKLY

%1) Find the TMS pulse and epoch around it
%2) BL correct (demean)
%3) Remove unused chans
%4) Cut-out the pulse (and interpolate)
%5) DS to 1 KHz
%6) Save file for cleaning in Script #2
%-----------------------------------------------------------------%

%% SECTION ON SETTING PARAMETERS/LOADING DEPENDENCIES
close; clear; clc;

%-----BASELINE SP TMS-EEG FILES (HC)-----
inPath = '/Volumes/LaCie_5TB/CH_Data/RAW_DATA/HC/SP_TMSEEG/';
outPath = '/Volumes/LaCie_5TB/CH_Data/SP_TMSEEG_PREPROC_DATA/';

Sesh = 'ACTIVE';
Pulse = 'SP';

ID = {'HCT203'; 'HCT205'; 'HCT206'; 'HCT209'; 'HCT212'; 'HCT216'; 'HCT217'; 'HCT220';...
    'HCT221';'HCT226';'HCT227';'HCT228';'HCT230';'HCT231';'HCT232';'HCT235';...
    'HCT238';'HCT241';'HCT243';'HCT248';'HCT249';'HCT250';'HCT251';'HCT252';...
    'HCT253';'HCT255';'HCT258';'HCT259';'HCT260';'HCT261';'HCT262';'HCT263';...
    'HCT265';'HCT266';'HCT268';'HCT270';'HCT271'};

caploc='/Users/aronhill/Documents/MATLAB/cap_location_files/standard-10-5-cap385.elp';

if ~exist(outPath,'dir'); mkdir(outPath); end
eeglab;

%% LOAD FILES

%for loop here across all subjects

%% FIND THE TMS PULSE BASED ON ARTEFACT


%% EPOCH THE DATA AROUND PULSE


%% DEMEAN THE DATA


%% REMOVE UNUSED CHANNELS


%% CUT-OUT THE TMS PULSE (TESA)


%% DOWNSAMPLE DATA


%% SAVE DATA


% end for loop
```

Heading

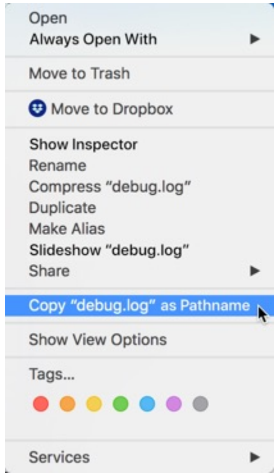
Broad description

More thorough description of key steps

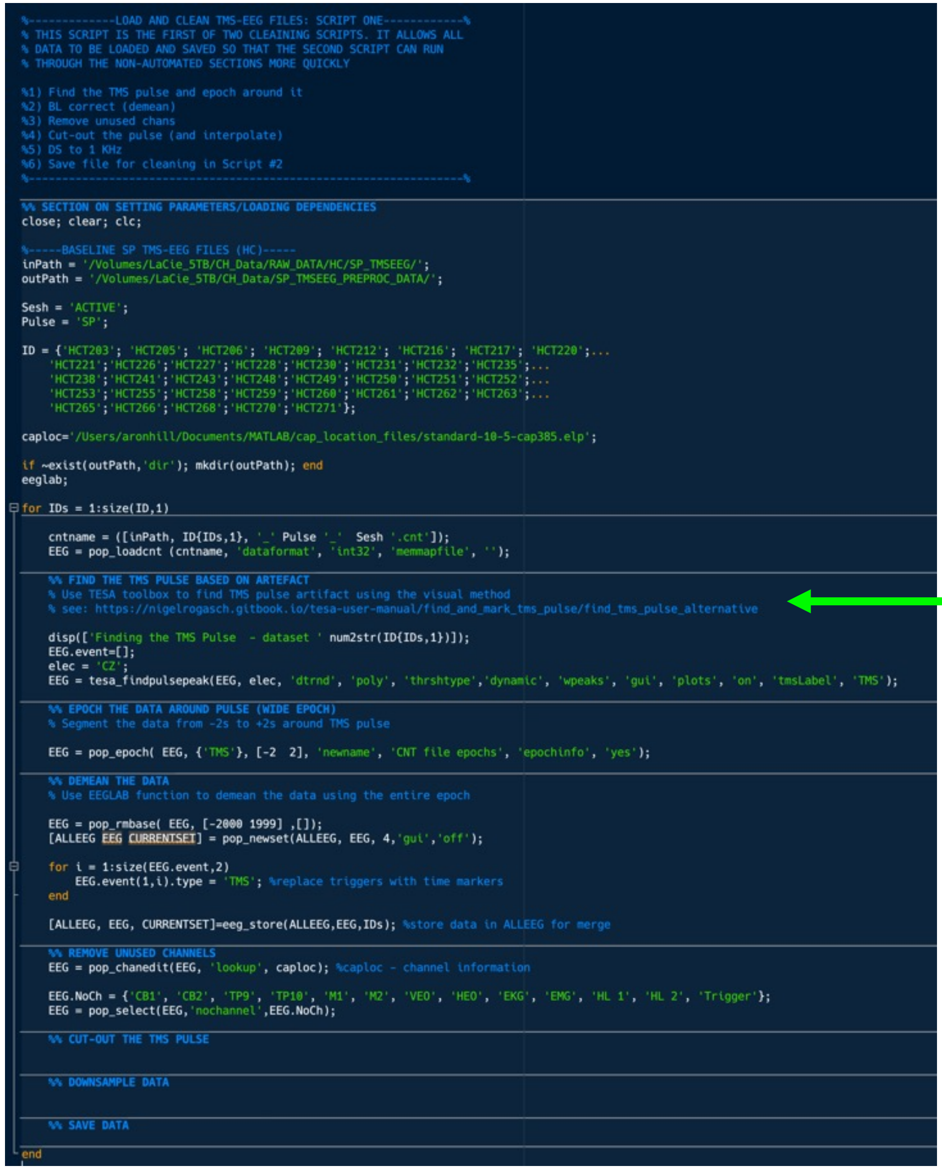
First section typically sets in and outpaths, defines key conditions (used in later loops) and loads subject IDs

Tip: Consider use of 'filesep' function to separate dir names for compatibility between mac/windows

Tip (MAC): Option + R click 'copy as pathname'



Step 3: Continue to build script

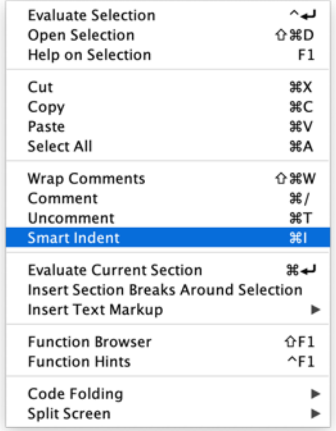


```
%------------LOAD AND CLEAN TMS-EEG FILES: SCRIPT ONE------------%
% THIS SCRIPT IS THE FIRST OF TWO CLEAINING SCRIPTS. IT ALLOWS ALL
% DATA TO BE LOADED AND SAVED SO THAT THE SECOND SCRIPT CAN RUN
% THROUGH THE NON-AUTOMATED SECTIONS MORE QUICKLY

%1) Find the TMS pulse and epoch around it
%2) BL correct (demean)
%3) Remove unused chans
%4) Cut-out the pulse (and interpolate)
%5) DS to 1 KHz
%6) Save file for cleaning in Script #2
%-----------------------------------------------------------------%

%% SECTION ON SETTING PARAMETERS/LOADING DEPENDENCIES
close; clear; clc;

%-----BASELINE SP TMS-EEG FILES (HC)-----
inPath = '/Volumes/LaCie_5TB/CH_Data/RAW_DATA/HC/SP_TMSEEG/';
outPath = '/Volumes/LaCie_5TB/CH_Data/SP_TMSEEG_PREPROC_DATA/';

Sesh = 'ACTIVE';
Pulse = 'SP';

ID = {'HCT203'; 'HCT205'; 'HCT206'; 'HCT209'; 'HCT212'; 'HCT216'; 'HCT217'; 'HCT220';...
    'HCT221';'HCT226';'HCT227';'HCT228';'HCT230';'HCT231';'HCT232';'HCT235';...
    'HCT238';'HCT241';'HCT243';'HCT248';'HCT249';'HCT250';'HCT251';'HCT252';...
    'HCT253';'HCT255';'HCT258';'HCT259';'HCT260';'HCT261';'HCT262';'HCT263';...
    'HCT265';'HCT266';'HCT268';'HCT270';'HCT271'};

caploc='/Users/aronhill/Documents/MATLAB/cap_location_files/standard-10-5-cap385.elp';

if ~exist(outPath,'dir'); mkdir(outPath); end
eeglab;

for IDs = 1:size(ID,1)

    cntname = ([inPath, ID{IDs,1}, '_' Pulse '_'  Sesh '.cnt']);
    EEG = pop_loadcnt (cntname, 'dataformat', 'int32', 'memmapfile', '');

    %% FIND THE TMS PULSE BASED ON ARTEFACT
    % Use TESA toolbox to find TMS pulse artifact using the visual method
    % see: https://nigelrogasch.gitbook.io/tesa-user-manual/find_and_mark_tms_pulse/find_tms_pulse_alternative

    disp(['Finding the TMS Pulse  - dataset ' num2str(ID{IDs,1})]);
    EEG.event=[];
    elec = 'CZ';
    EEG = tesa_findpulsepeak(EEG, elec, 'dtrnd', 'poly', 'thrshtype','dynamic', 'wpeaks', 'gui', 'plots', 'on', 'tmsLabel', 'TMS');

    %% EPOCH THE DATA AROUND PULSE (WIDE EPOCH)
    % Segment the data from -2s to +2s around TMS pulse

    EEG = pop_epoch( EEG, {'TMS'}, [-2  2], 'newname', 'CNT file epochs', 'epochinfo', 'yes');

    %% DEMEAN THE DATA
    % Use EEGLAB function to demean the data using the entire epoch

    EEG = pop_rmbase( EEG, [-2000 1999] ,[]);
    [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 4,'gui','off');

    for i = 1:size(EEG.event,2)
        EEG.event(1,i).type = 'TMS'; %replace triggers with time markers
    end

    [ALLEEG, EEG, CURRENTSET]=eeg_store(ALLEEG,EEG,IDs); %store data in ALLEEG for merge

    %% REMOVE UNUSED CHANNELS
    EEG = pop_chanedit(EEG, 'lookup', caploc); %caploc - channel information

    EEG.NoCh = {'CB1', 'CB2', 'TP9', 'TP10', 'M1', 'M2', 'VEO', 'HEO', 'EKG', 'EMG', 'HL 1', 'HL 2', 'Trigger'};
    EEG = pop_select(EEG,'nochannel',EEG.NoCh);

    %% CUT-OUT THE TMS PULSE


    %% DOWNSAMPLE DATA


    %% SAVE DATA

end
```

Tip: use *cmd + /* to comment/uncomment chunks of highlighted code (*Cntl + Shift + C* in R)

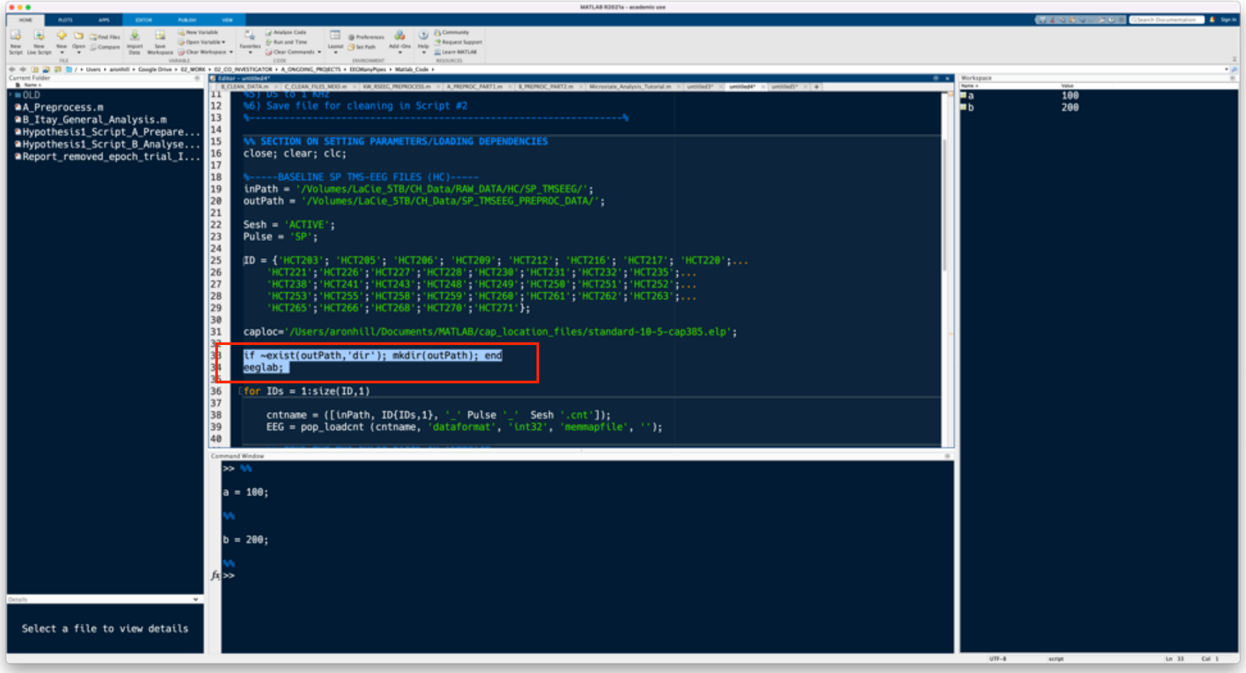
Add further info relating to code chunk

Tip: use 'smart indent' function to keep code neat



Use shortcuts

- Run highlighted section of code
  - MATLAB: Shift + F7 (Mac)
  - R: Cntrl + Enter

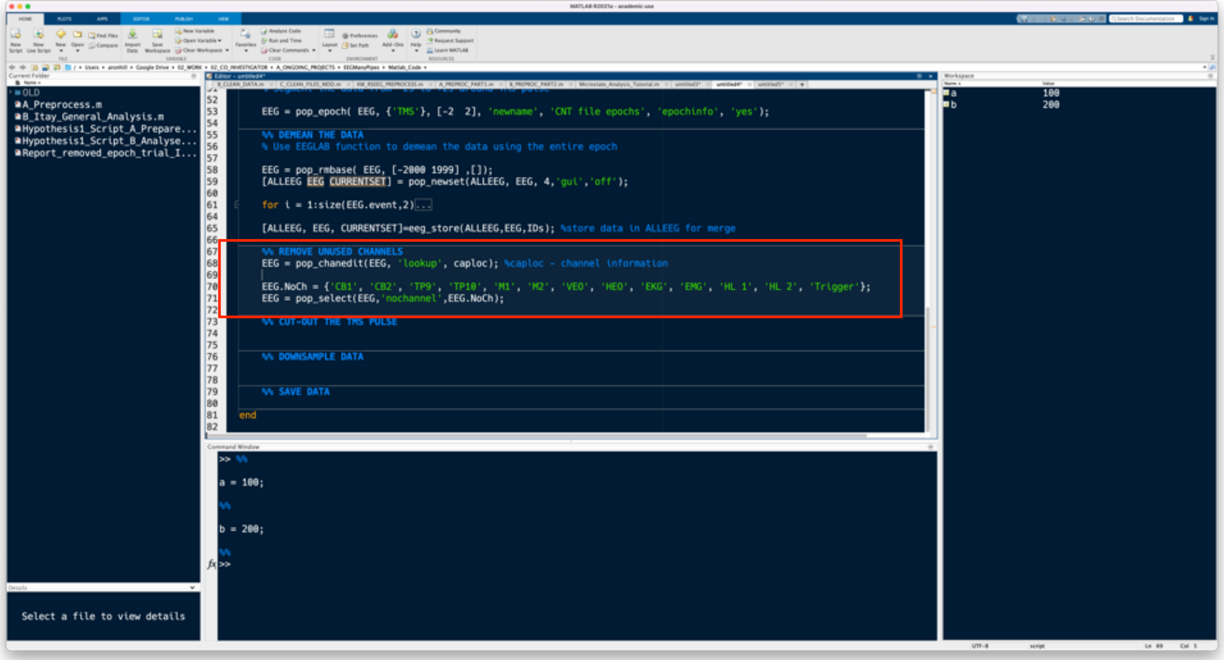


OLD
A_Preprocess.m
B_Itay_General_Analysis.m
Hypothesis1_Script_A_Prepare...
Hypothesis1_Script_B_Analyse...
Report_removed_epoch_trial_I...
Select a file to view details
a 100
b 200

```
%5) DS to 1 KHz
%6) Save file for cleaning in Script #2
%-------------------------------------------------------------%

%% SECTION ON SETTING PARAMETERS/LOADING DEPENDENCIES
close; clear; clc;

%-----BASELINE SP TMS-EEG FILES (HC)-----
inPath = '/Volumes/LaCie_5TB/CH_Data/RAW_DATA/HC/SP_TMSEEG/';
outPath = '/Volumes/LaCie_5TB/CH_Data/SP_TMSEEG_PREPROC_DATA/';

Sesh = 'ACTIVE';
Pulse = 'SP';

ID = {'HCT203'; 'HCT205'; 'HCT206'; 'HCT209'; 'HCT212'; 'HCT216'; 'HCT217'; 'HCT220';...
    'HCT221';'HCT226';'HCT227';'HCT228';'HCT230';'HCT231';'HCT232';'HCT235';...
    'HCT238';'HCT241';'HCT243';'HCT248';'HCT249';'HCT250';'HCT251';'HCT252';...
    'HCT253';'HCT255';'HCT258';'HCT259';'HCT260';'HCT261';'HCT262';'HCT263';...
    'HCT265';'HCT266';'HCT268';'HCT270';'HCT271'};

caploc='/Users/aronhill/Documents/MATLAB/cap_location_files/standard-10-5-cap385.elp';

if ~exist(outPath,'dir'); mkdir(outPath); end
eeglab;

for IDs = 1:size(ID,1)

    cntname = ([inPath, ID{IDs,1}, '_' Pulse '_'  Sesh '.cnt']);
    EEG = pop_loadcnt (cntname, 'dataformat', 'int32', 'memmapfile', '');
```

```
>> %%

a = 100;

%%

b = 200;

%%

>>
```

- Run entire section of code
  - MATLAB: Option + Enter



OLD
A_Preprocess.m
B_Itay_General_Analysis.m
Hypothesis1_Script_A_Prepare...
Hypothesis1_Script_B_Analyse...
Report_removed_epoch_trial_I...
Select a file to view details
a 100
b 200

```
    EEG = pop_epoch( EEG, {'TMS'}, [-2  2], 'newname', 'CNT file epochs', 'epochinfo', 'yes');

    %% DEMEAN THE DATA
    % Use EEGLAB function to demean the data using the entire epoch

    EEG = pop_rmbase( EEG, [-2000 1999] ,[]);
    [ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 4,'gui','off');

    for i = 1:size(EEG.event,2)...

    [ALLEEG, EEG, CURRENTSET]=eeg_store(ALLEEG,EEG,IDs); %store data in ALLEEG for merge

    %% REMOVE UNUSED CHANNELS
    EEG = pop_chanedit(EEG, 'lookup', caploc); %caploc - channel information

    EEG.NoCh = {'CB1', 'CB2', 'TP9', 'TP10', 'M1', 'M2', 'VEO', 'HEO', 'EKG', 'EMG', 'HL 1', 'HL 2', 'Trigger'};
    EEG = pop_select(EEG,'nochannel',EEG.NoCh);

    %% CUT-OUT THE TMS PULSE


    %% DOWNSAMPLE DATA


    %% SAVE DATA

end
```

```
>> %%

a = 100;

%%

b = 200;

%%

>>
```

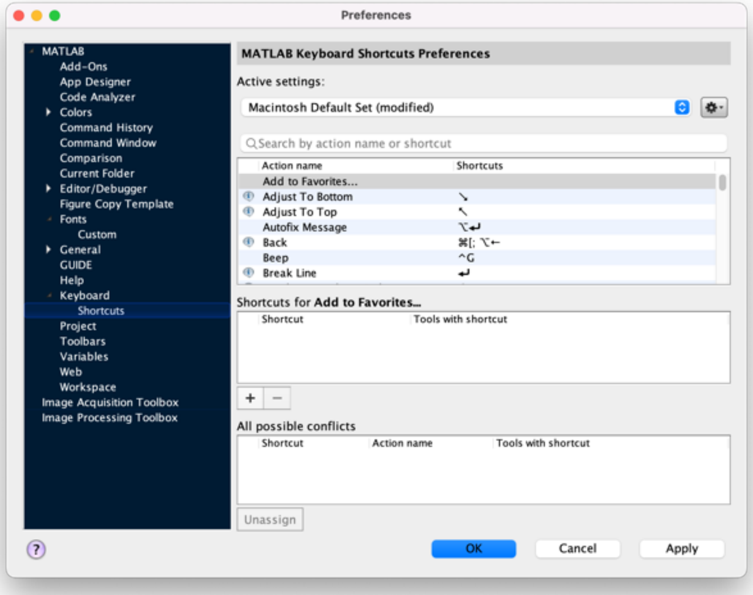
4. Customise shortcuts

- A key frustration when swapping between MATLAB and R can be having different shortcuts for performing the same task

- Tip: Consider customizing shortcuts to make them the same across the two platforms

MATLAB

- Preferences → Keyboard → Shortcuts



MATLAB Keyboard Shortcuts Preferences

Active settings: Macintosh Default Set (modified)

Action name	Shortcuts
Add to Favorites...	
Adjust To Bottom	↘
Adjust To Top	↖
Autofix Message	⌥↵
Back	⌘[; ⌥←
Beep	^G
Break Line	↵

Shortcuts for **Add to Favorites...**

Shortcut	Tools with shortcut

All possible conflicts

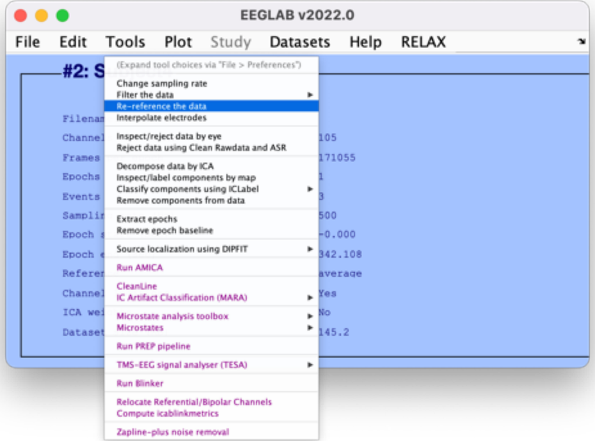
Shortcut	Action name	Tools with shortcut

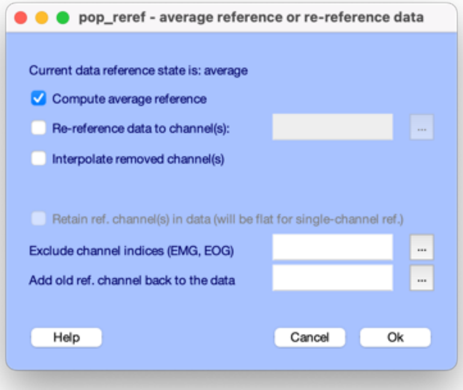
5. Use the ‘eegh’ function

TIP: use the ‘eegh’ function to build scripts

‘Write’ sections of code using the GUI

This can be a lifesaver when starting out coding in EEGLAB





Current data reference state is: average

☑ Compute average reference

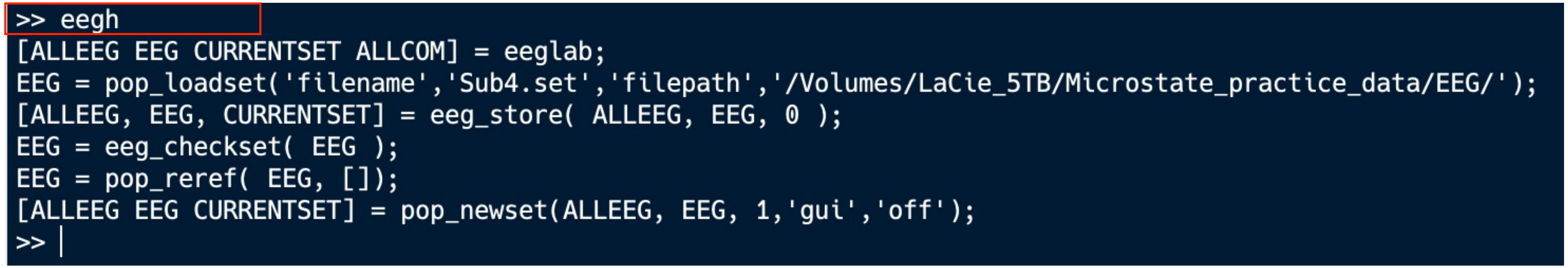
□ Re-reference data to channel(s):

□ Interpolate removed channel(s)

□ Retain ref. channel(s) in data (will be flat for single-channel ref.)

Exclude channel indices (EMG, EOG)

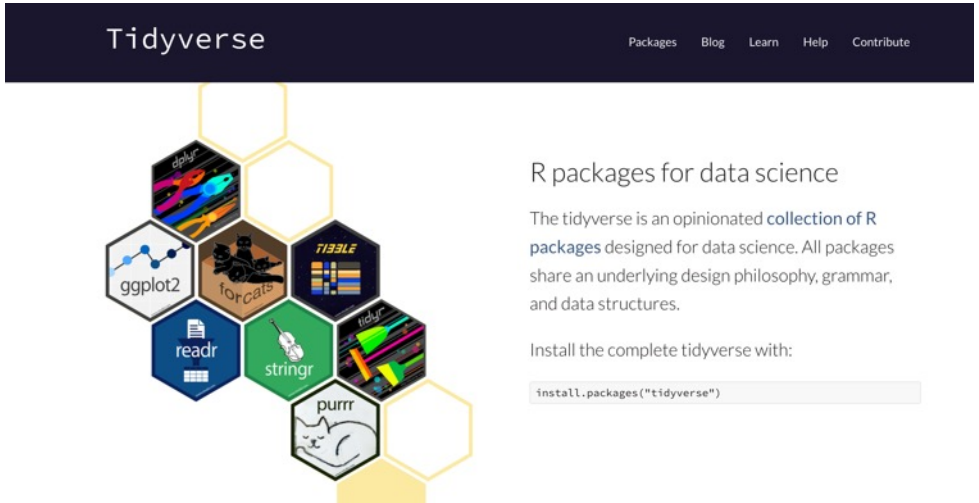
Add old ref. channel back to the data



```
>> eegh
[ALLEEG EEG CURRENTSET ALLCOM] = eeglab;
EEG = pop_loadset('filename','Sub4.set','filepath','/Volumes/LaCie_5TB/Microstate_practice_data/EEG/');
[ALLEEG, EEG, CURRENTSET] = eeg_store( ALLEEG, EEG, 0 );
EEG = eeg_checkset( EEG );
EEG = pop_reref( EEG, []);
[ALLEEG EEG CURRENTSET] = pop_newset(ALLEEG, EEG, 1,'gui','off');
>> 
```

Utilise R’s array of packages

- Base functions in R can be quite limited/unintuitive
- A major feature of R is its vast array of packages
- https://www.tidyverse.org/

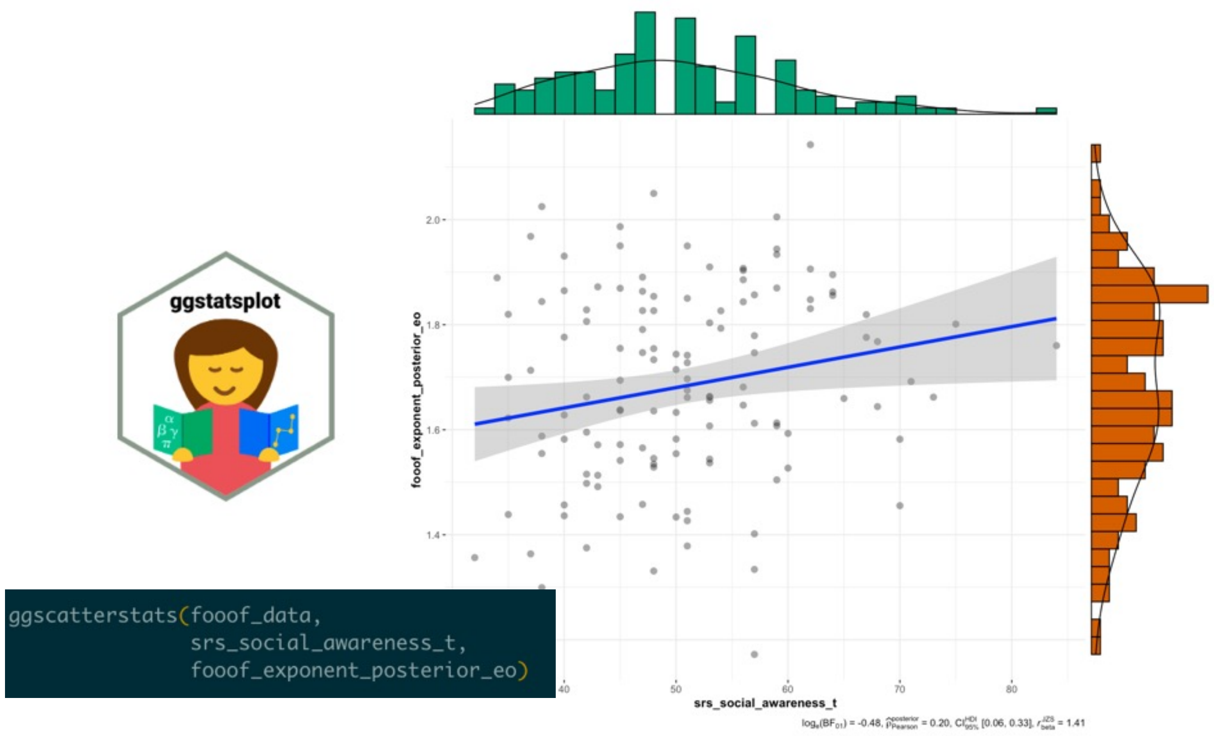


R packages for data science

The tidyverse is an opinionated collection of R packages designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```



```
ggscatterstats(fooof_data,
               srs_social_awareness_t,
               fooof_exponent_posterior_eo)
```

Final Tips

- Find others in your lab/wider university who are at a similar stage to you and meet regularly to share and discuss code, tips, and tricks

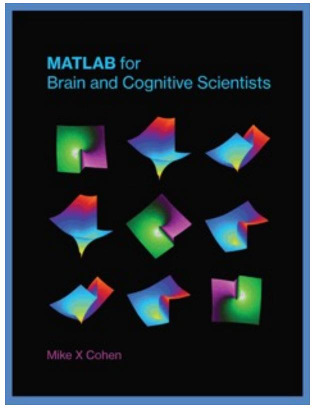
- Practice adapting other people's code to work with your own data

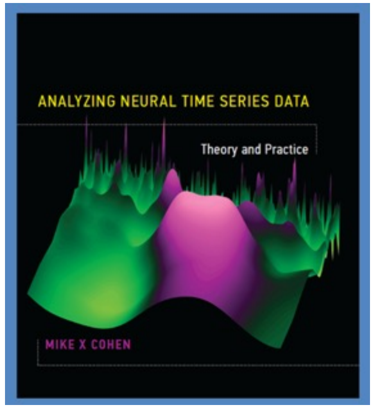
- Take opportunities to work with collaborators experienced in coding, or find a mentor who is willing to help you code

- Dedicate time each week to learning how to code (your future self will thank you!)

Additional Resources

- Mike X Cohen books/courses
  - https://sincxpress.com/
- RELAX pipeline for automated EEG cleaning
  - https://github.com/NeilwBailey/RELAX/releases
- Data wrangling in R (Mike Chapple)
  - Available through Linked in learning
- Nordman et al. (2022) Data Visualization Using R for Researchers Who Do Not Use R (doi: https://doi.org/10.1177/25152459221074654)





Thanks for listening!

Doctors: Googling stuff online does not make you a doctor.

Programmers: