% MATLAB WORKSHOP

```
{ 'Gabe Nespoli';
  'July 2016';
  'Ryerson University' }
```

% The plan

% **1.** Intro to MATLAB / PHZLAB

% **2.** Writing scripts (PHZLAB and/or EEGLAB)

% Why programming?

% **flexibility:** do whatever you want

% **power:** do it all with one click

% **understanding:** know what you are doing; make your own assumptions

% 1. Intro to MATLAB / PHZLAB
% ----------------------------

% getting oriented

% numeric arrays

% plotting

% structure variables

% character arrays (strings)

% cell arrays

% parameter-value pairs

% getting oriented

```
% current folder
% command window
% workspace
% editor

% installing PHZLAB by adding it to the path
```

% MATLAB is case-sensitive

```
matlab ~= MATLAB

data ~= Data

PHZ ~= phz
```

% brackets & spacers

```
() % brackets
[] % square brackets
{} % curly brackets
```

```
␣ % space
, % comma
; % semi colon
: % colon
```

```
% % comments
>> clear all % clear all vars from workspace
>> clc % clear stuff from command window
<arrow up> % scroll through command history
```

% numeric arrays: vectors

```
% [Square brackets for numeric arrays]
% Space or comma to separate columns
>> num = 1 % number / vector of size 1-by-1
>> num = [1] % same thing
>> num = [5 6 7 8 9] % vector of size 1-by-5
num =
     5     6     7     8     9

% Indexing vectors
>> num(1) % the first element
>> num(1:3) % the first three elements
>> num(1:2:5) % the elements with odd indices
```

% numeric arrays: matrices

```
% Semi-colon to separate rows
>> num = [4 5 6; 7 8 9] % matrix (2-by-3)
num =
      1      2      3
      4      5      6

% Indexing matrices
>> num(1,2) % row 1, column 2
>> num(1,:) % row 1, all columns
```

% e.g.: Tic Tac Toe

```
>> T = [0 0 0; 0 0 0; 0 0 0]
>> T = zeros(3,3)
T =
     0     0     0
     0     0     0
     0     0     0
```

% plotting

```
>> y = [1 2 3 4 5]
>> plot(y)

>> y = y * 5
>> x = 0:4
>> plot(x,y)
>> scatter(x,y)
>> bar(x,y)

>> stdError = ones(1,5)
>> errorbar(x,y,stdError)
```

% structure variables

```
% a struct is multiple variables in one

% load 'phz_sample_data/scr_all.phz'
>> PHZ = phz_load

% it operates like a dictionary
% access its fields with periods
>> PHZ.srate % numeric specifying sampling rate
>> PHZ.region % struct of 1-by-2 vectors
>> PHZ.region.baseline
```

% digital sampling

```
% you have a sensor (e.g., electrode) that measures
  something
% the computer can write down the value at the sensor
>> data = 0.5; times = 0;
% it does this many times per second
>> data = [data 0.6]; times = [times 1];
>> data = [data 0.4]; times = [times 2];

% PHZ.data is a matrix of size trials-by-samples/time
>> size(PHZ.data)
ans =
        2282        8001
```

% e.g.: plotting real data

```
% 1. Plot the first trial
   (hint: first row, all columns)
% 2. Plot the first trial with time on the
   x-axis (hint: use PHZ.times)
% 3. Plot the mean of all trials (hint: type
   help mean for info on the mean function)
```

% strings

```
% character array (a.k.a., string) in single quotes
>> str = 'string'

% concatenate a string with square brackets
>> str = ['This' 'is' 'a' 'string' 'too']
str =
Thisisastringtoo

% index strings just like numeric vectors
>> str(8:13)

% PHZ.study and PHZ.datatype are strings
```

% cell arrays

```
% What if you want to index strings, not
  characters?

% cell array of strings

>> C = {'cell' 'array' 'of' 'strings'}


% Indexing cell arrays

>> C{1} = 'cell'

>> C{1}(2) = 'e'


% PHZ.history is a cell array of strings
```

% parameter-value pairs

```
%  function(input,'Param1',Value1,...)
%  function(input,'Param1',Value1,'Param2',Value2,etc.)

%  the parameter is always string
%  the value can be any variable type, but different
   parameters expect different variable types

>> phz_plot(PHZ)
>> phz_plot(PHZ,'summary','group')
>> phz_plot(PHZ,'summary','group','feature','mean')
```

% plotting in PHZLAB

% **1.** Plot the data summarized trials. (hint: use a parameter/value pair)

% **2.** Plot the data summarized by *both* group *and* trials. (hint: use a cell array)

% **3.** Plot the max value of the target region. (hints: use the region param/value pair; look at the help for phz_feature for available features and their names)

% **4.** Subtract the mean of the baseline region, then plot the mean of the target region (hint: use the blsub function or parameter/value pair)

% PHZLAB Toolbox

```
% A toolbox is a folder of MATLAB functions

>> phzlab % a list of available functions

% file I/O (incl. loading data from Biopac)
% processing data
% plotting data
% exporting data (for R, SPSS, etc.)
```

% loading data from Biopac

```
>> PHZ = phz_create('blank')
>> PHZ = phz_create % choose a raw data file

% auto-fill grouping fields
...,'namestr','participant-group-session',...

% specify the channel
...,'channel',4,...
```

% PHZ files

```
% grouping variables:
>> PHZ.participant
>> PHZ.group
>> PHZ.condition
>> PHZ.session
>> PHZ.trials

>> PHZ.resp % behavioural responses
>> PHZ.proc % keeps track of all processing
>> PHZ.history % keeps track of warnings
```

% processing data

```
>> phz_transform % Transform data (e.g., square root)
>> phz_filter % Butterworth filtering.
>> phz_rectify % Full- or half-wave rectification.
>> phz_smooth % Sliding window averaging (incl. RMS)
>> phz_epoch % Split a single channel of data into
              trials.
>> phz_trials % Add names to each trial of epoched data.
>> phz_rej % Remove trials with values exceeding a
            threshold or SD.
>> phz_blsub % Subtract mean of baseline region from each
              trial.
>> phz_norm % Normalize across specified grouping
             variables.
```

% analyzing data

```
>> phz_subset % Keep data only from specified
              grouping variables.

>> phz_region % Keep only data from a certain
              time region.

>> phz_feature % Convert data to the
               specified feature.

>> phz_summary % Average across grouping
               variables.
```

% plotting data

```
>> phz_plot(PHZ)
>> phz_plot(PHZ,'feature','mean')
>> phz_plot(PHZ,'region','target')
>> phz_plot(PHZ,'subset',PHZ.resp.q1_rt < 7)
```

% exporting data

```
>> phz_writetable

% input is the same as phz_plot
% but here are some useful modifications:
  • ...,'summary',{'participant',...},...
  • ...,'save','filename.csv',...
```

```
% 2. Writing scripts
% ------------------
```

```
% scripts vs. functions
% loops
% getting files
```

% functions vs. scripts

% functions

- you have to specify the input and where to put the output

```
>> output = myFunction(input)
```

% scripts

- no input or output
- usually loads and saves files from/to disk

```
>> myScript
```

% loops

```
for i = [1 2 3 4 5]
    disp(i)
end

files = {'a' 'b' 'c'};
for i = 1:length(files)
    disp(files{i})
end
```

% getting files (uigetfile)

```
% in the command window
>> uigetfile
>> [files,folder] = uigetfile

% in a script
[files,folder] = uigetfile;
files = cellstr(files);
```

% Best practices

% MATLAB is a language like English

% make code readable

- indenting
- descriptive variable names

% good commenting

% comments that aren't redundant with code

% A few notes

% Don't forget your friends

- help functions
- the internet
- forum posts are often the best source of information (e.g., MATLAB forums, stackoverflow)

% Learn by doing

- programming is hard to learn by studying
- find a project to work on (i.e., find something to program)