## Functions

## Learning Objectives

* Explain a Matlab function file.
* Define a function that takes parameters.
* Test a function.
* Know why we should divide programs into small, single-purpose functions.

**Part 1 – Inbuilt Functions**

Function syntax

When called in script or command line

[Out1, Out2, ...] = function\_name(In1, In2, ...)

In first line of function definition

function [Out1, Out2, ...] = function\_name(In1, In2, ...)

Functions we’ve used so far

Mean M = mean(A,dim)

Max M = min(A,[],dim)

Num2str s = num2str(A)

Etc.

Input and Output arguments in documentation

Draw concept map of a function

CHALLENGE

%% Challenge #2

% Turn this code into a function that takes in1 and in2 as inputs

% and outputs out 1 and out2

% You should also give your function a name that helps people work out what

% it does!

in1 = 5;

in2 = 4;

out1 = in1 - in2;

out2 = in1 + in2;

%% Extension

% Inputs can be anything (numbers, vectors, strings)

% Try your function from above with vectors? matrices?

**Part 2 – Writing Functions**

Variable scope – variables defined in functions are only visible there.

* For something to be useable in a function you have to either pass it to it, or define it within.

Use challenge answer in a script to explain variable scope (split the screen between script and function)

CHALLENGE

* Write a function that takes two numbers and returns the sum of the numbers and the product of the numbers (two inputs and two outputs)
* Include in the function an “if statement” that tests whether the sum or product is bigger, and returns a third output that is a string stating whether the sum or product is bigger (ie. ‘sum is bigger’ or ‘product is bigger’)

function [vsum, vprod, checksize] = sumprod(v1, v2)

vsum=v1+v2;

vprod=v1\*v2;

if vsum>vprod

checksize='sum is bigger than product';

elseif vprod>vsum

checksize='product is bigger than sum';

else

checksize='sum and product are same size';

end

end

**Part 3 – Commenting Functions**

We need to write documentation about functions to help other people (and our future selves)

Go over answer to sumprod code with comments

Put into sumprod code:

% [vsum, vprod] = sumprod(v)

% Calculate the sum (vsum)

% and the product (vprod)

% of the inputs v1 and v2.

% Also return a string (checksize) stating the relative size of vsum

% and vprod

Things we do to every data set can be put in functions

Want to shift our data so that the mean = whatever we want it to be. (explain why shifting the mean is useful e.g. – want to just compare changes in data and ignore baseline offset)

Use patient data & plot

Subtract mean and plot again (it is now zero mean)

Shift data and plot again (it now has a new baseline so the whole thing is shifted)

out = (data - mean(data))

out = (data - mean(data)) + desired

Challenge

* Make a function called “centre\_shift” that will shift your data to a desired mean, and use it in your analyse\_data.m script
* Add documentation to your function so that if you type “help centre\_shift” in the command line, useful information about how to use the function will show up

Answer:

function out = center\_shift(data, desired)

% Center data around a desired value.

%

% center(DATA, DESIRED)

%

% Returns a new array containing the values in

% DATA centered around the value.

out = (data - mean(data)) + desired;

end