

User defined functions in Matlab

Ken Nyholm

Outline

- Userdefined functions in Matlab
- In-line functions
- Functions in seperate files

Functions in Matlab

- User defined functions were presented in the slide-set called "An Introduction to Matlab"
- Here we go a bit more in detail
- Such functions are mainly used to:
 - Automise tasks that are to be performed several times
 - Implement theoretical models so we can calculate the function value for given and varying inputs and the arguments the maximise/minimise the function
- It is the latter application that we focus on here

Functions in Matlab

- Matlab uses a function "handle" when functions are defined or used
- If a function is defined in a file the following structure has to be use:

```
function [output] = function_name(input)
```

- A function defined in this way can be called from another file/program or from command-line mode by:

```
[my_output] = function_name(my_input)
```

Functions in Matlab

- And it can be used as an input to a built-in Matlab function, for example *fmincon*, using the handles `@` or `'function_name'`, in the following ways:

```
[Est,fval,flag,output,lambda,grad,hess] = ...  
    fmincon(@var_p_lik, pStart,[],[],[],[],LB,UB,[],options_);
```

```
[param, fval, exitflag, outpt, la, g, H] = ...  
    fmincon('regime_likeli',S_param,A,A_u,[],[],lb,ub,[],options);
```

In-line functions

- In-line functions refer to the concept of defining the object of interest in just one line within the program that conducts the optimisation or function value calculation
- This can be achieved by:

```
27 % --- Loglikelihood function ---
28 - lnL = @(B_0) T/2*log(2*pi*B_0(end,1))+...
29           1/(2*B_0(end,1))*sum( (Y-X*B_0(1:nVars)).^2 );
30 % --- setting up the optimisation inputs ---
31 - options_ = optimset('LevenbergMarquardt', 'on', 'LargeScale', 'off', 'Display', 'on', 'TolX', 1e-12, 'Tol
32 - lb       = [-10.*ones(nVars,1); 0.001];
33 - ub       = [10.*ones(nVars,1); 100];
34 - [B_hat,Fval,Exitflag,Output,Lambda,Grad,Hess] = fmincon(lnL, B_0, [], [], [], [], lb, ub, [], options_);
```