# Computer Problem Sheet 1: Introduction to Matlab

Most Bayesians econometricians create their own programs using programs such as Matlab or Gauss. When writing my textbook I used Matlab and the website associated with it (http://wiley.co.uk/koopbayesian) contains many Matlab programs. I will use Matlab for this course. Matlab and Gauss have very similar structures so, even if you are a Gauss user, you may find my Matlab programs useful. In this short course there is not enough time to teach you everything about Matlab. Rather I attempt to teach you the basics of relevance for Bayesian econometrics. With a knowledge of these basics, you can then use Matlab online help or manuals to figure out more complicated things. It is also worth noting that, increasingly, Bayesian researchers are making their Matlab or Gauss programs available on their websites (e.g. James LeSage has a good website: http://www.spatial-econometrics.com/). Even if you are going to use someone else's programs, it is useful to have some basic Matlab or Gauss skills to understand and, if necessary, adapt their code.

My way of teaching Matlab will be indirect: I will provide sample programs and then get you to figure out what each step does. Then I will ask you to create similar programs. If you do not like this style of teaching and prefer a direct method where you learn all the commands at a general level, there are many web-based tutorials you should feel free to use. You can find some of these by clicking on demos at the top of the command window. This is the Matlab online tutorial which you may want to experiment with (particularly if you are unfamiliar with programming). A few other useful websites which include Matlab tutorials and demonstrations include

http://www.mathworks.com/academia/student_center/tutorials/launchpad.html,
http://www.math.siu.edu/matlab/tutorials.html,
http://www.cyclismo.org/tutorial/matlab/
and http://amath.colorado.edu/computing/Matlab/tutorials.html.

## Overview

Matlab is a matrix programming language. Unlike other programs you may have used (e.g. Microfit, Eviews, Stata, etc.), Matlab is exclusively a programming language. For instance, in these other languages you would just click on a button to do OLS or type a simple command (e.g. "regress y x"). In Matlab you have to actually program up OLS. The OLS estimator is given as:

$$\widehat{\beta} = \left(X'X\right)^{-1} X'y.$$

In Matlab, the format for this command is:

$$\mathsf{bhat} = \mathsf{inv}\left(\mathsf{x}'{*}\mathsf{x}\right)\mathsf{x}'{*}\mathsf{y};$$

and you would actually have to type this line and then run the program containing it.

A Matlab program is a series of commands such as this. These commands will be listed in a file. Once you have created and saved such a file using the Matlab editor, you can then run the program. Different versions of Matlab have slightly different editors or ways of running programs so I will not give detailed instructions about this (in this regard the layout of Matlab is pretty much the same as any other program so you can figure this out yourself – plus I will demonstrate in the computer labs). Accordingly, below I will focus on program writing.

Some of these questions will require you to use the Help facility in Matlab. It is good practice to become familiar with this facility.

The Matlab Environment:

- The Command Window - To execute commands one at a time.
- Current Directory - Active file directory
- Workspace - Active variables
- Command History - Past commands

- Command files - 'm-files'
- Output files - 'out-files'

## Basic Matlab Matrix Commands

Note the statements beginning with % are ignored by Matlab. Programmers often place comments in their programs in this way to explain key steps in their programs.

*A sample program*

```
% This is a simple program which illustrates Matlab matrix commands
x = [1 2 3; 4 5 6; 7 8 9]
y = [11 12 13; 14 15 16; 17 18 19];
y
z=x+y
w=x-y
u= x*y
a=cat(2,x,y)
b=cat(1,x,y)
c=x(:,2)
d=y(:,1)
e=x(2,3)
% Save the matrix b to an ascii file
save ('data1.txt', 'b', '-ASCII');
```

```
% Retrieve the data in the file data1.txt
xx=load('data1.txt');
xx
```

**Exercise 1**

a) Create this program and run it in Matlab and examine the output Matlab produces. Describe what each line of this program does. Note, what difference does the semi-colon make?

b) Add a line to this program which creates a new matrix, f, which is the transpose of x.

c) Add a line to this program which creates a new matrix, g, which is the identity matrix plus x.

d) Add a line to this program which creates a new matrix, ginv, which is the inverse of g. What happens if you try to take the inverse of x?

e) In the command window, type `who` and press enter, then type `whos` and press enter. What do these commands do?

## Create an Artificial Data Set then do OLS Estimation

Matlab has many subroutines (i.e. built-in little programs) that you can call automatically as part of your program. (Note: You can also create your own subroutines). Here we show you how to use Matlab's subroutines for random number generation from the Normal and Uniform distributions to create an artificial data set from the regression model: $y_i = \alpha + \beta x_i + \varepsilon_i$ for $i = 1, .., 100$. We set $\alpha = 1$, $\beta = 2$ and let the $x_i$ and $\varepsilon_i$ be random draws from the $U(0, 1)$ and $N(0, 1)$ distribution, respectively.

*A sample program*

```
%This is a program which artificially creates a data set and then does OLS estimation using it
%First part of this program artificially simulates data set
n=100;
alpha=1;
beta=2;
e = randn(n,1);
x=rand(n,1);
y=alpha + x*beta + e;
%following line adds intercept to x. explain why
x=cat(2,ones(n,1),x);
%Following part of the program does OLS estimation
%the OLS estimator.
bhat = inv(x'*x)*x'*y;
bhat
%the OLS residuals
resids = y - x*bhat;
%The OLS estimator of the error variance
s2 = resids'*resids/(n-2);
```

'The OLS estimator of the error variance is'
s2

**Exercise 2**

a) Create this program and run it in Matlab and examine the output in Matlab. Describe what each line of this program does.

b) Extend this program to calculate the $R^2$ of this regression and print out the result.

c) Extend this program to calculate the covariance matrix of the OLS estimators (i.e. $var\left(\widehat{\beta}_{OLS}\right) = s^2\left(X'X\right)^{-1}$) and print out the result.

## For Loops and If/Then Statements

When doing Monte Carlo integration we repeatedly take draws from the posterior distribution. Matlab does this kind of repeated action using constructs called "for loops".

*A sample program*

```
%This is a program which illustrates do loops and if statements
%first create a column vector to work with
x= [1; 2; 7; 5; 9; 3; 6; 9; 1; 11; 1]
%the following command sums up the elements of a column vector
xsum=sum(x);
xsum1=0;
for i=1:11
xsum1=xsum1 + x(i,1);
end
['xsum and xsum1']
cat(2,xsum, xsum1)
%now illustrate the if command
xsum2=0;
for i=1:11
if x(i,1)>4
xsum2=xsum2 + x(i,1);
end
end
xsum2
```

**Exercise 3**

a) Create this program and run it in Matlab and examine the output. Describe what each line of this program does. In particular, why are xsum1 and xsum the same as one another? What does the "if" statement do? What is "xsum2"?

b) The sample program *sums* various column vectors. Modify this program to calculate *averages* (i.e. means).

c) Supplementatry question. Use the loop to compute the variance of the elements of the vector x.

## Supplementary Exercise

Do this exercise if you have found the previous exercises simple and are interested in digging into Matlab more deeply. In the previous exercises we have created our own data/matrices and printed out our results to the screen. In practice, you may wish to load in some data from some other source and print out final results to a file which you can keep.

a) The website http://www.wiley.co.uk/wileychi/koop/ has many different data sets. Use the data set HPRICE.XLS. This file is in Excel format (which Matlab can directly import, but I prefer to use txt files). Using manuals or online help facilities figure out how to do OLS estimation involving this data in Matlab. You may like to convert the data to ASCII text format and add a load command to the Exercise 2 program (deleting the code which artificially generates a data set).