Mathematics for Economists
An Introduction to Matlab[1]

Marcel Bluhm

Wang Yanan Institute for Studies in Economics
Xiamen University

September 10, 2013

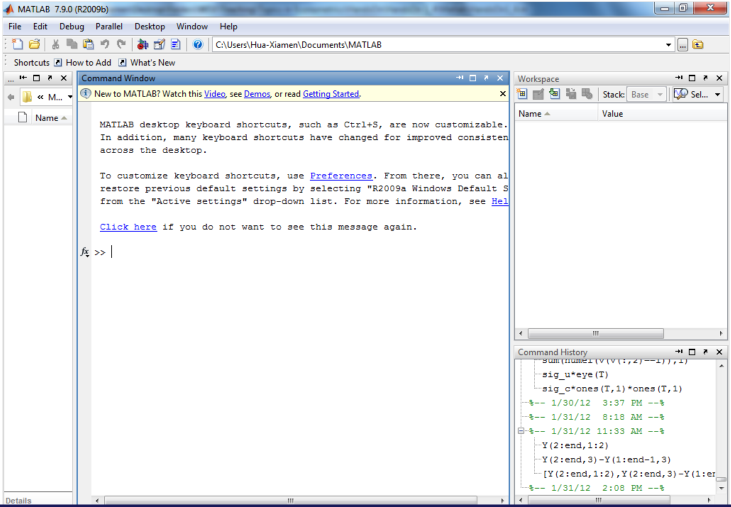
[1] Based on 'Matlab Primer' by Kermit Sigmon

Preliminaries

- Matlab is an interactive, matrix-based programming language
- Using this primer in combination with the user's and reference guide for Matlab is sufficient for this course
- This outline only focuses on the parts necessary for the topical course in econometrics

Preliminaries

- Upon opening the software, the program window consists of the **command window**, **workspace**, and **command history**:



Preliminaries

- Matlab is an expression language:
  - *variable* $=$ *expression* or
  - *expression*
- Help on any function can be obtained via pressing 'F1' in the program window and then entering the search term

Entering Matrices and Vectors

- 1×1 matrices are interpreted as scalars
- Matrices with only one column/row are interpreted as vectors
- Matrices can be set up via
  - entering an explicit list of statements, for example $A = [1\ 1; 1\ 1]$;
  - using built in functions, for example $B = ones(2, 2)$

Entering Matrices and Vectors

Exercise:

Generate variable $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$;

Matrix Operations

Among others, the following matrix operations are available in Matlab:

- $+$ addition
- $-$ subtraction
- $*$ multiplication
- ^ power
- $'$ transpose
- $/$ division

Matrix Building Functions

- $eye(\cdot, \cdot)$ identity matrix
- $zeros(\cdot, \cdot)$ matrix of zeros
- $diag(\cdot, \cdot)$ create and extract diagonals
- $rand(\cdot, \cdot)$ create a uniformly distributed matrix
- $randn(\cdot, \cdot)$ create a normally distributed matrix

Matrices can be built from blocks

- $A = ones(2, 2)$
- $B = [A; zeros(1, 2); eye(2)];$

Matrix Building Functions

Exercise:

1. Generate a standard normally distributed variable, x, with 1000 observations;
2. Generate a new variable $y = 5 + x$;
3. Compute mean and standard deviation of the variable y.

Flow Control Statements

Flow control statements operate like those in most computer languages:

$X = []$
for i $= 1 : 3$
$\quad X = [X, i\hat{\ }2];$
end

Flow Control Statements

Exercise:

1. Write a loop which generates the vector a, containing numbers from 1 to 50;
2. Generate a vector $b = a'$.

Relations

- $<$ less
- $>$ greater then
- $<=$ less then or equal
- $>=$ greater then or equal
- $==$ equal
- $\tilde{} =$ not equal

For example:
$A = [(1:3)', [1;0;1]];$
$B = A(A(:,2) == 1, :);$

Relations

Relations may be connected or quantified by logical operators:

- & and
- || or
- ˜ not

For example:
$A = [(1:3)', (3:1)', [1;0;1]];$
$B = A(A(:,3) == 1 \& A(:,1) == 1, :);$

Scalar Functions

- exp
- log
- sqrt

Matrix/Vector Functions

- max
- min
- sum
- mean
- size

Colon Notation

The colon notation can be used to access submatrices, for example:
$A = [1\ 2\ 3; 4\ 5\ 6; 7\ 8\ 9];$
$A(1, :)$ is a row vector consisting of all entries in the first row of matrix A

M-Files

Matlab can execute a sequence of statements stored on hard disk.
Exercise:

1. Open an empty M-File (via clicking the paper icon in the main program window below 'File')
2. Write the program
   $A = 1;$
   $B = 2;$
   $C = A + B;$
3. Save the program as 'test.m' in a folder of your choice
4. Execute the program via clicking on the 'Run icon in the editor'

Function-Files

Function files provide extensibility to Matlab.
Exercise:

1. Save the following as M-File 'square_op.m':
   function sq= square_op(in)
   $sq = in.\hat{}2;$
2. Entering the statement 'square_op(2)' in the command window or an M-File results in execution of the M-File 'square_op', that is, the result is '4'

Note that functions can have multiple input and output arguments.

Example: Hands-On

- Consider the following model:
  $y_t = c + \beta x_t + \epsilon_t$
  where
  - y is the dependent variable
  - c is a constant
  - β is a coefficient
  - x_t is an independent variable distributed as *N(0,1)*
  - ϵ_t is a white noise error term distributed as *N(0,4)*

1. Generate 200 observations for y_t and x_t following the data generating process (DGP) outlined above
2. Use the ordinary least squares (OLS) estimator to estimate c and β and compute their t-statistics