Time Series Econometrics/Econometrics II

PC Tutorial: Introduction to Matlab

Richard Schnorrenberger
richard.schn@stat-econ.uni-kiel.de

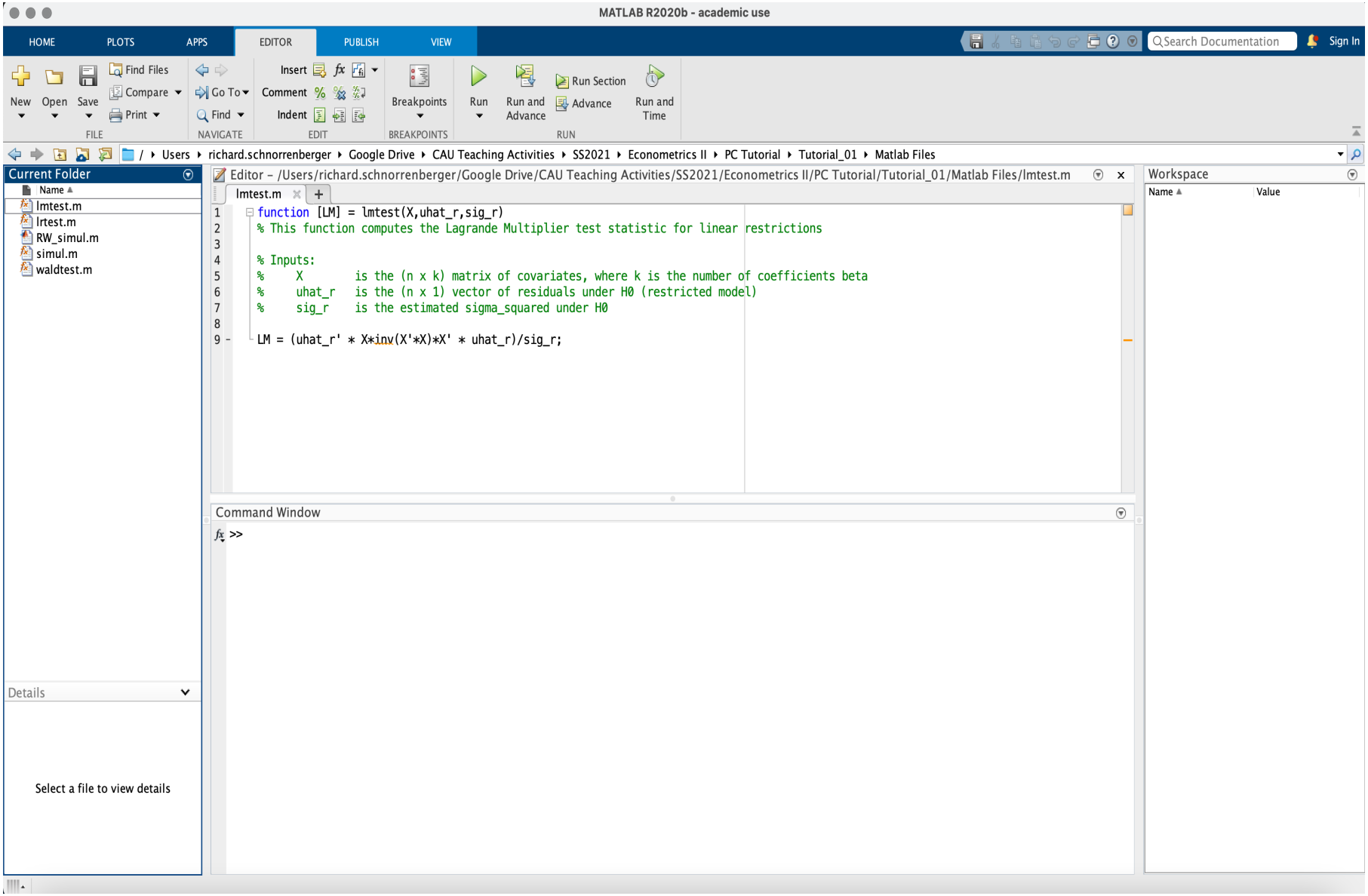
Institute of Statistics and Econometrics
University of Kiel

Summer Semester 2024

Matlab

- Matlab stands for the Matrix Laboratory.
- It has a very simple syntax.
- High performance and runs faster than most statistical software packages.
- Matlab toolboxes and MathWorks documentation. See, for instance, MathWorks' documentation on the Econometrics Toolbox.
- Data visualization.

Matlab Interface



Vectors and Matrices

- % Create a row vector: a = [1 2 3 4 5]
- % Create a column vector: a = [1 2 3 4 5]' or a = [1; 2; 3; 4; 5]
- % Create a vector with evenly spaced elements: t = 0:1:100
- % Create matrices: A = [1 2; 3 4]; B = A'; C = [1 2 3; 5 6 7; 9 10 11]
- % Create an $N \times N$ empty matrix: nan(N)
- % Create an $N \times N$ matrix of zeros: zeros(N)
- % Create an $N \times N$ matrix of ones: ones(N)
- % Create an $N \times N$ identity matrix: eye(N)
- % Create diagonal matrix or get the main diagonal elements of matrix:
  d = diag([1 2 3]); diag(d)
- % Get the size of the row or column dimension of a matrix A: size(A,1), size(A,2)
- % Combine two vectors/matrices A and B: C = [A B], C = [A; B]
- % Select the entry (i, j) in matrix A: A(i,j)
- % Select all entries in row m or column n: A(m,:), A(:,n)
- % Get a selection of elements in row m or column n: A(m,j1:j2), A(i1:i2,n)
- % Select last entry or elements in the last row/column: A(end), A(end,:), A(:,end)
- % Specify elements of a vector/matrix by indexing:
  a(end) = 2, X(:,1) = ones(size(X,1),1), A(end,1:2) = zeros(1,2)

For more commands on vectors (arrays) and matrices see MathWorks documentation.

Matrix Operations

Let A and B be matrices with A $\in$ M(m x n) and B $\in$ M(r x s) and c be a scalar.

Matrix Operation	Matlab Command	Condition
add scalar	A+c	
add matrix	A+B	m = r, n = s
subtract scalar	A-c	
subtract matrix	A-B	m = r, n = s
multiply scalar	c*A	
matrix multiplication	A*B	n = r
element-wise multiplication	A.*B	m=r, m = s
to the power	A^c	m = n
element-wise power	A.^c	
transpose	A′	
invert	inv(A)	A invertible
invert faster and efficiently (a)	eye(size(A,1))/A	A invertible
invert faster and efficiently (b)	A\eye(size(A,1))	A invertible
matrix "division"	A*inv(B) = A/B = B\A	B invertible

Control Structures:

Control structures such as for, while, and if-else are implemented through *loops* and *conditionals* as follows:

```
for i = 1:100 (loop counter)
    Statements
end
```

```
if x > y (condition)
    Statements
end
```

```
while x == 0 (condition)
    Statements
end
```

```
if x > y (condition)
    Statements
    else if x == y (condition)
        Statements
    else
        Statements
end
```

Conditional and Logical Operators:

Conditional Operators

Operator	Relation	Application
$==$	Equality	$x == y$
$>$	Larger than	$x > y$
$<$	Smaller than	$x < y$
$>=$	Larger or equal	$x >= y$
$<=$	Smaller or equal	$x <= y$
$\sim=$	Different	$x \sim= y$

Logical Operators

Operator	Condition	Application
&	And	$x > 0$ & $y > 0$
—	Or	$x > 0$ — $y > 0$
$\sim$	Not	$\sim (x > 0)$
all	All elements...	all$(x > 0)$
any	Any element...	any$(x > 0)$

Functions

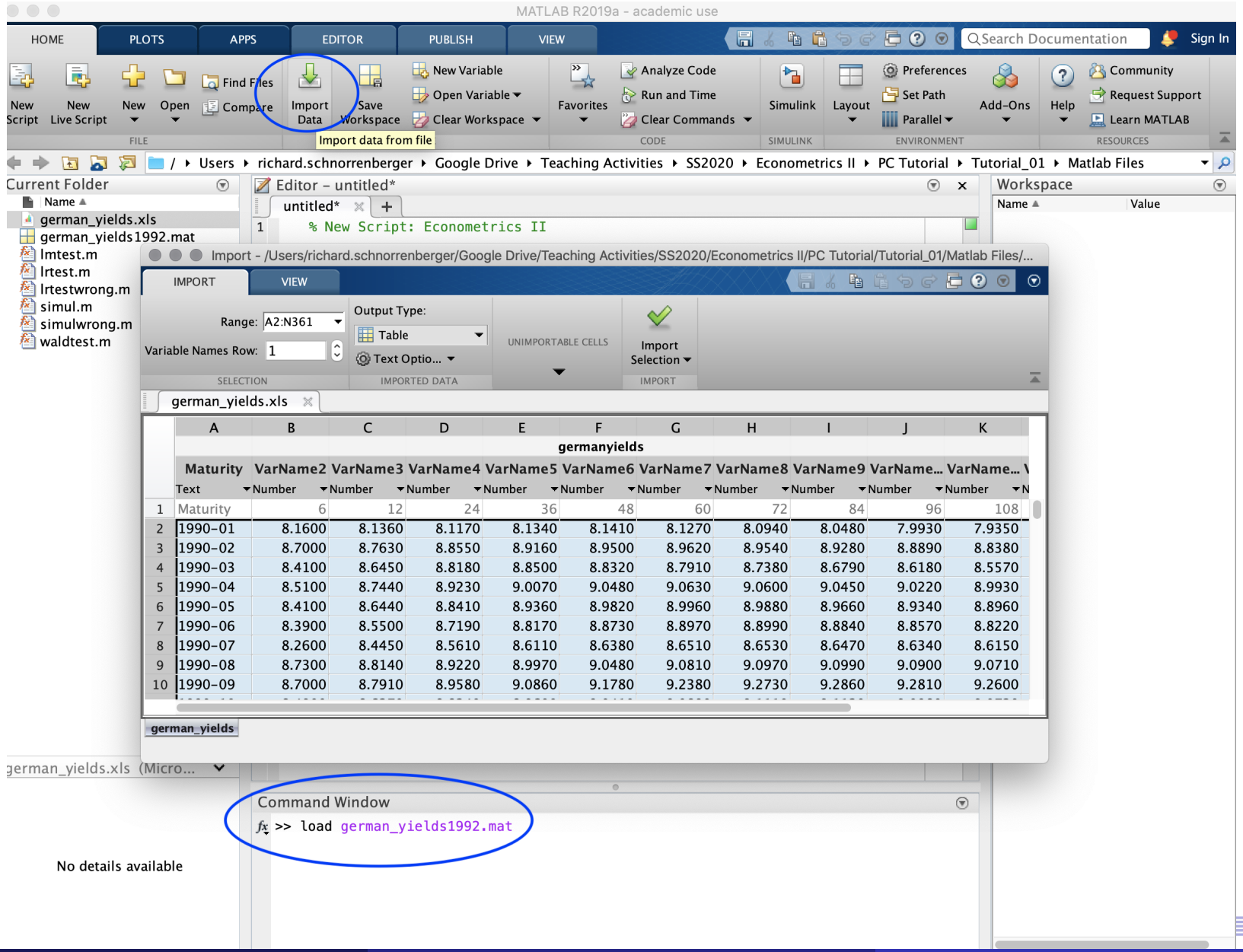
- You can write your own functions in Matlab.
- They are constructed as follows:

```
function [output1, output2, ...] = functionname(input1, input2, ...)
    Statements...
```

- For instance, create the Matlab function to compute the Wald test statistic for linear restrictions:

```
function [W] = waldtest(X,R,r,theta)
% X is the n × k matrix of covariates
% R is the q × k matrix of restrictions
% r is the q × 1 vector of restricted/hypothesized values
% theta is the (k + 1) × 1 vector of parameter estimates
n = size(x,1); % sample size
k = size(x,2); % number of covariates (including intercept)
sig = theta(end); % estimate for sigma
V = sig*inv(x'*x); % covariance matrix based on conditional expectations
W = (R*theta-r)'*inv(R*(V./n)*R')*(R*theta-r); % compute Wald test statistic
```

Loading Data into Matlab



Plots

- Most useful plot functions: plot(x,y), scatter(x,y), plotmatrix(x), histogram(x,nbins).
- For basic plotting functions see the YouTube video. For details on input arguments of the plot function (line style, color, title,...), see the function documentation.
- For detailed information on all types of Matlab plots, see MathWorks online documentation.
- To create sequences of dates and time, see MathWorks online documentation.
- Example:

Random_Walk.m

```
% Simulate a random walk process:
seed = 1; rng(seed);                          % Fix seed of pseudo-random generator
n = 240;                                      % Define number of time periods
t = datetime(2000,1,1) + calmonths(1:n);      % Create dates
u = normrnd(0,1,[n 1]);                       % Generate random numbers
y = zeros(n,1);                               % Initialize y vector
for i = 2:n
    y(i) = y(i-1) + u(i);                     % Compute random walk process
end
figure                                        % Generate plot
plot(t',y,'Color','#0072BD','LineWidth',2)
title('Simulated Random Walk Process')
xlabel('Time')
```

Matlab Plot (Time Series Example)