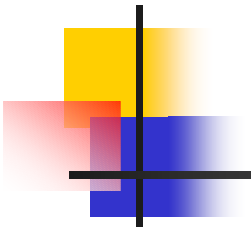
“Open source data handling: An introduction to Scilab

Les Hatton

Professor of Forensic Software Engineering
Kingston University, UK
L.Hatton@kingston.ac.uk, www.leshatton.org

Version 1.3: 01/Oct/2009

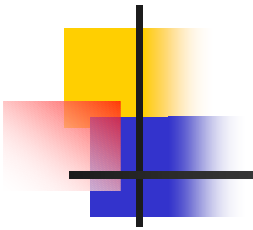
Overview



- What and where is it ?
- Downloading and installation
- In and out
- Objects and expressions
- File handling and scripts
- Data analysis

Copyright Les Hatton 2007-

What and where is it ?



- http://www.scilab.org/
- Open source product of INRIA, France
- Mature, stable and very comprehensive data handling package
- Quite compatible with Matlab
- Interactive and batch with a GUI
- Windows and Linux (and others)
- Current version 5.1.1, (April 2009)

Copyright Les Hatton 2007-

Downloading and installation

- Windows
  - Self installing executable – double-click to install as usual (around 90Mb expanding into about 260Mb).
- Linux
  - Compressed tar ball .tar.gz.

    % zcat xxxxx.tar.gz | tar xvf –

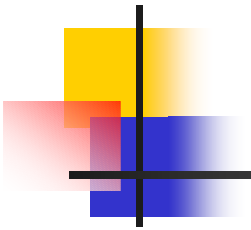
After installation, you can load it to a memory key on either and it will run from the memory key:-

- .../bin/Wscilex.exe

(Wscilex is the GUI version).

Copyright Les Hatton 2007-

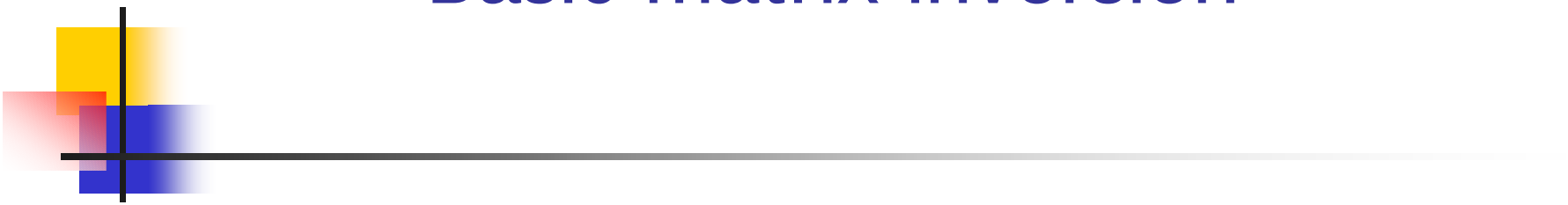
Basics



- The menu bar
  - File -> Quit or quit from the command line
  - Interrupt / Resume
  - Applications
  - Help
  - Demonstrations; TclTk -> Scale
- Variables and expressions
  - a= 2, a, save a, clear a, a, load a, a
  - a+b etc. %e, %pi
  - Vectors v = [1,2], v', w = [3,4], w', v*w'; w'*v
  - X=-10:2:10, y=sin(x*%pi/10), plot(x,y), help plot
- Diary function – diary('session.txt') ⇔diary(0)

Copyright Les Hatton 2007-

Basic matrix inversion



- Solving Ax=B
- A = [1,2,-3
  3,4,5
  7,8,9]
- B = [1;2;3]
- x = inv(A)*B OR
- x = linsolve(A,B)

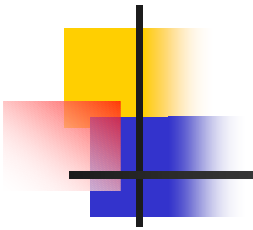
Copyright Les Hatton 2007-

Programming and scripts

- Some constructs
  - for m=1:10, a(m) = m^2, end
  - s = 100; while s>50, disp(s^2), s = s-5, end
  - x=10;y=5; if x>y disp(x), else disp(y), end
  - r=7; select r, case 1, disp(r), case 2, disp(r^2), case 7, disp(r^3), end
- Scripts
  - Where am I ? File -> Change/display current working directory (to Datasets for demos)
  - Enter into text editor as script1.txt
  - exec('script1.txt') or exec script1.txt

Copyright Les Hatton 2007-

Functions



- Inline

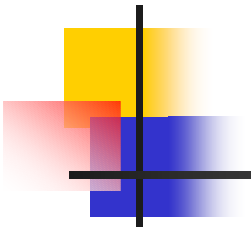
```
deff('[r,theta]=cartpol(x,y)',['r=sqrt(x^2+y^2)';'theta=atan(y,x)'
    ]);
[radius,angle] = cartpol(3.,4.);
```

- Or as a file

```
getf('cartpol.txt');
[radius,angle] = cartpol(3.,4.);
```


Copyright Les Hatton 2007-

Files and plotting



Files

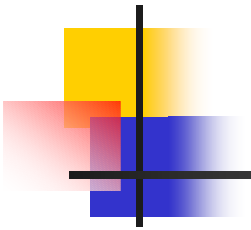
- u=file('open','euro2008.dat','old')
- euro=read(u,-1,1)
- file(’close’,u)
- u=file('open','usd2008.dat','old')
- usd=read(u,-1,1)
- file(’close’,u)

And plotting

- plot(euro,usd)
- plot(euro); x=1:1:length(euro); plot(x,euro,x,usd);

Copyright Les Hatton 2007-

Statistics



Simple stats

- meuro = mean(euro)
- stdeuro = st_deviation(euro)
- coefs=regress(euro,usd)

  (Gives usd = 1.1104439 + 0.6519457 euro)

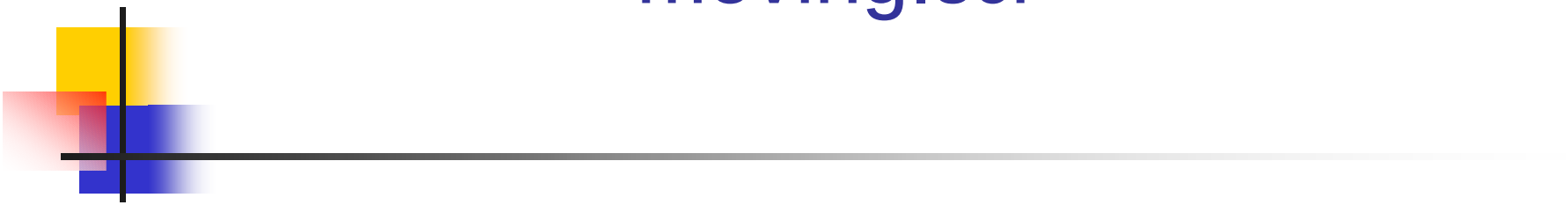
More complex statistics

- getf moving.sci; (view this)
- exec euroavg.txt; (view this)

Source of financial data:- http://fx.sauder.ubc.ca/

Copyright Les Hatton 2007-

moving.sci



```
function [ukp] = moving(u,k)
//
// Calculate moving average using convolution.
//
n = length(u);
uu = ones(1,k);
uu2 = convol(uu,u)/k;
//
// We now have to trim the first part.
//
nuu2 = length(uu2);
ukp = uu2(k:nuu2-k+1);
```


Copyright Les Hatton 2007-

euroavg.txt

```
// Script to read Euro currency movements against the pound
// , calculate a moving average and then plot both. together.
// Date 01-Oct-2009.
//
u=file('open','euro2008.dat','old');
euro=read(u,-1,1);
file('close',u);
//
// Calculate a moving average and trim the initial part from
// the input data.
//
eavg = moving(euro,50);
eurotrim = euro(50:length(euro));
//
// and plot.
//
x=1:1:length(eavg);
plot(x,eurotrim,x,eavg);
```


Copyright Les Hatton 2007-

More exciting stuff

- Suppose we have y = b0 + b1.x +b2.x2 + b3.x3
- And the following data:

x1	x2	x3	y
1.2	3.1	2.0	5.7
2.5	3.1	2.5	8.2
3.5	4.5	2.5	5.0
4.0	4.5	3.0	8.2
6.0	5.0	3.5	9.5

- exec('test_regress.txt') ➔ best-fitting bi.

Copyright Les Hatton 2007-

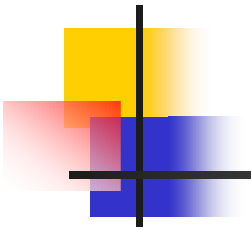
test_regress.txt

```
//  Example to perform multiple linear regression, 03-01-2008
//  Urroz, vol 2, p. 381
clear
//
//  Parameters
//
x1 = [1.2, 2.5, 3.5, 4.0, 6.0]
x2 = [3.1, 3.1, 4.5, 4.5, 5.0]
x3 = [2.0, 2.5, 2.5, 3.0, 3.5]
//
//  Observation
//
y  = [5.7, 8.2, 5.0, 8.2, 9.5]

X  = [ ones(5,1) x1' x2' x3']
//
//  The vector of coefficients is given by ...
//
b  = inv(X'*X) * X' * y'
```


Copyright Les Hatton 2007-

References



- Urroz, G.E. (2001) “Numerical and statistical methods with SCILAB for science and engineering”, vols 1 and 2 www.greatunpublished.com

  (Little out of date but worth looking at)
- www.scilab.org and follow the links to the documentation.
- home.hit.no/~finnh/scilab_scicos
- Google away happily

Copyright Les Hatton 2007-