MATLAB Programming

More Flow Control

MATLAB is a programming language

- Flow control:
  - if ï else if ï else
  - for
  - while
- Functions

MATLAB is a programming language

- Flow control:
  - if ï else if ï else
  - for
  - while
- Functions
  - encapsulation
  - logical independence

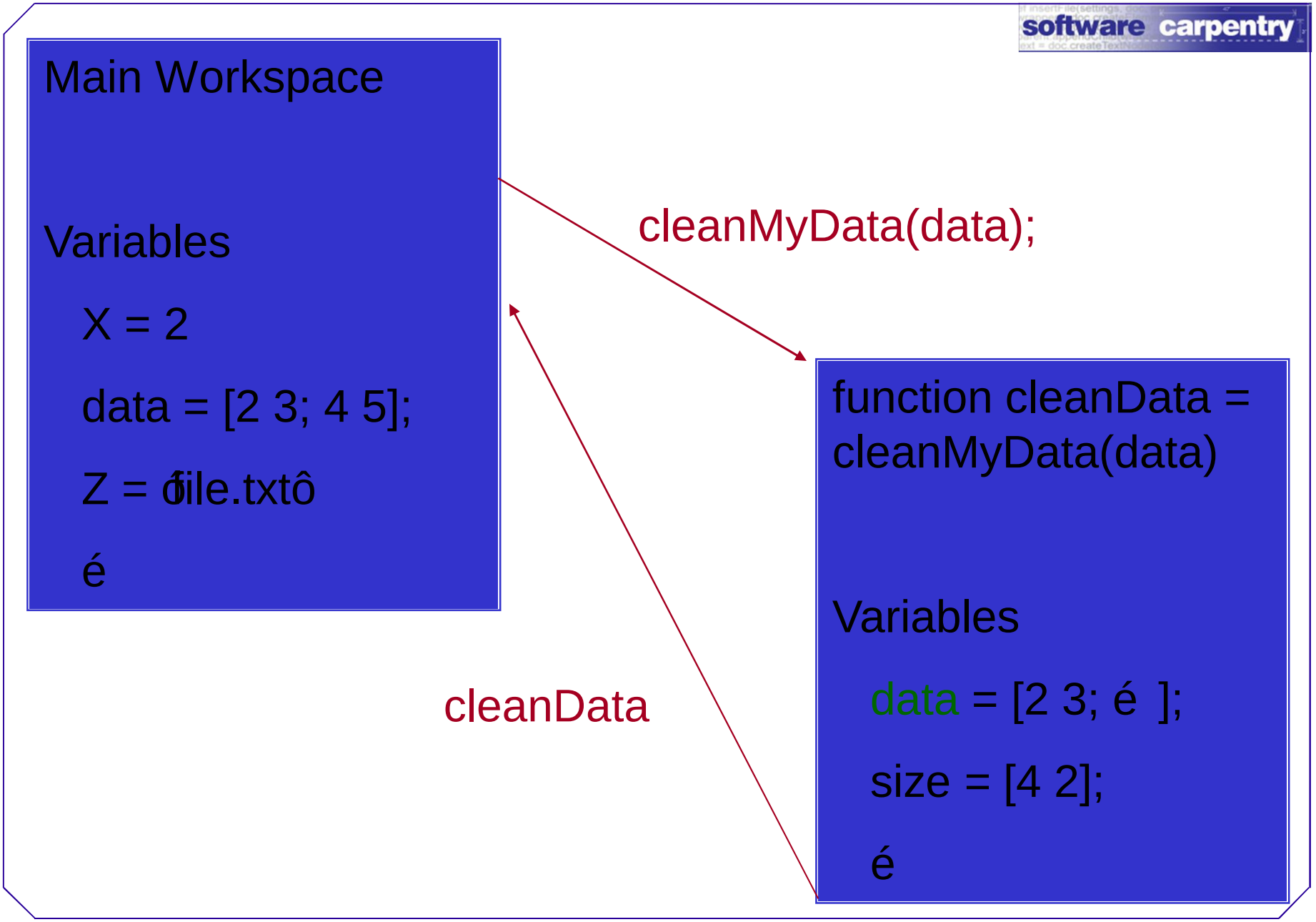
Program is a set of tasks:

- Load data
- Perform calculation
- Display result

Program is a set of tasks:

- Load data
  - Get file name
  - Read
  - Test for correct format
- Perform calculation
  - Prepare for computation
  - Perform several independent subtasks
  - Combine into a result

é


Main Workspace
Variables
X = 2
data = [2 3; 4 5];
Z = 'file.txt'
cleanMyData(data);
cleanData
function cleanData = cleanMyData(data)
Variables
data = [2 3; 4 5];
size = [4 2];

Example:

The eigenvectors and eigenvalues of a square matrix are the set of vectors and associated scalars that satisfy the following equation:

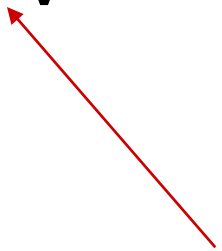
$$M * v = a * v$$

Matrix M times vector v

Example:

The eigenvectors and eigenvalues of a square matrix are the set of vectors and associated scalars that satisfy the following equation:

$$M * v = \lambda * v$$



Scalar λ times vector v

Eigenvectors in MATLAB:

```
A = [1 2; 3 4];
U = eig(A)
U =
   -0.3723
    5.3723
```

Eigenvalues and eigenvectors:

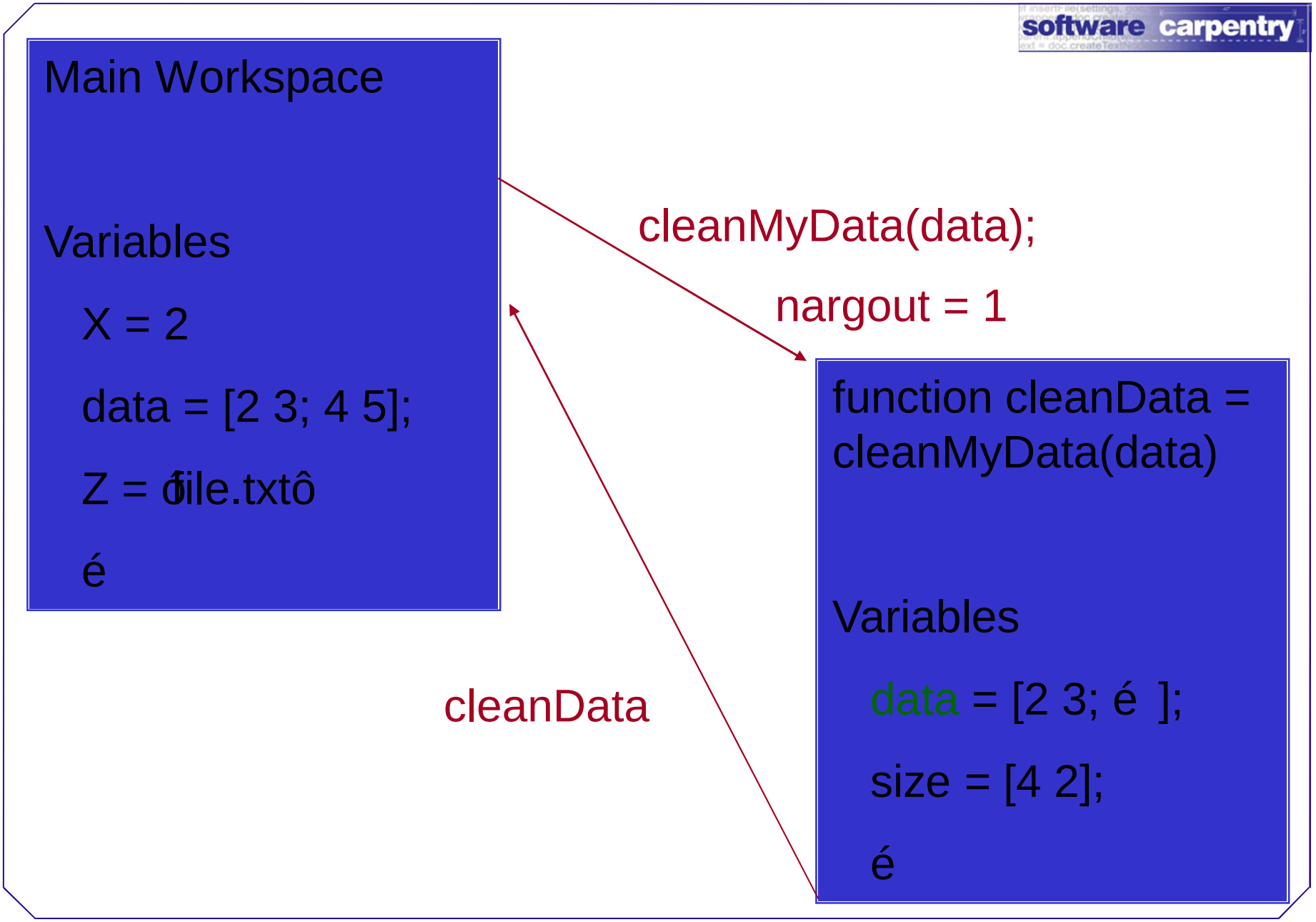
```
[U V] = eig(A)
U
   -0.8246   -0.4160
    0.5658   -0.9094
V
   -0.3723
    5.3723
```

Is this a good idea?

Two special functions:

nargin() # How many parameters were passed?

nargout() # How many return values are expected?


Main Workspace
Variables
X = 2
data = [2 3; 4 5];
Z = 'file.txt'
cleanMyData(data);
nargout = 1
function cleanData = cleanMyData(data)
Variables
data = [2 3; 4 5];
size = [4 2];
cleanData

Interpolation example:

- Time series data...
- With some bad measurements...
- That are 0.0.

Interpolation example:

- Time series data...
- With some bad measurements...
- That are either 0.0 or (optionally) user specifiedé
- é and that could be greater than an optional max.
- Report the number of values that are replaced.
- Report the locations of all values that are replaced.

MATLAB programming convention:

The first return value should be the result of the calculation

Subsequent, optional return values status variables.

Example:

max/min - return the actual max or min and the optional index.

ode45 (differential equation solver) - return the time points and the actual solution then optional trajectory information.

```
function clean_data = interpolate(data)
```

```
function [clean_data, num_replaced,
  replace_locs]
     = interpolate(data, min, max)
```

```
function [clean_data, num_replaced,
  replace_locs]
      = interpolate(data, min, max)


if nargin() < 2
  min = 0;
end
[I, J] = find( data < min );
```

```
function [clean_data, num_replaced,
  replace_locs]
      = interpolate(data, min, max)
if nargin() < 2
  min = 0;
end
[I, J] = find( data < min );
if nargin() == 3
    [Imax, Jmax] = find(data > max);
    I = [I ; Imax];
    J = [J ; Jmax];
end
```

```
( Clean the data and put it in clean_data)


if nargout() >= 2
    num_replaced = length(I);
end
if nargout() == 3
    replaced_locations = [I J];
end


clean_data = data;
```

Functions encapsulate things:

- Separate logical unit from the rest of the code
- Separate variable names to avoid confusion

Multiple parameters and returns.

- nargin()
- nargout()

created by

Richard T. Guy

February 2011