Table of Contents

LECTURE 20 Nov

```
%----------------------------------------------

% Load a table from the saved workspace of last lecture
load filename Tablibor
```

Extracting from a Table

Extract chfd from the table

```
Tablechfd =Tablibor(1:end,{'chfd'});
```

```
Tablechfd1 =Tablibor(1:end,6);
```

```
Tablechfd2 =Tablibor.chfd;
```

Adding variables

Add new variable to Tablibor

```
Tablibor.newvariable= Tablibor.chfd.^2;
```

Modifying Tables

Move newvariable using indexing

```
Tablibor = Tablibor(:,[1:4 7 6 5]);

% alternative
Tablibor(:,[4 7])=Tablibor(:,[7 4]);

% Remove newvariable from the table
Tablibor.newvariable = [];
```

Applying functions to table

Create a function

```
avg = @mean
% The average value of all the variables for the last 50 observation
average=varfun(avg,Tablibor(50:end,[1:6]))


% compute mean
average2= mean(Tablibor.chfd(not(isnan(Tablibor.chfd))));
average22 = nanmean(Tablibor.chfd);
%
average3 = nanmean(Tablibor{:,2:end});



% Add another variable to Tablechfd
Tablechfd.var2= Tablechfd.chfd.^2;

% Add another variable toTablibor
Tablibor.var2= Tablibor.chfd.^2;
```

```
avg =

    @mean


average =

    mean_datedd     mean_usdd     mean_gbpd     mean_eurd     mean_chfd
 mean_jpyd
    ___________     _________     _________     _________     _________
 _________

    7.3381e+05      NaN           NaN           NaN           NaN
 NaN
```

Testing, Comparing Elements of a Table

```
% Check whether each element of chfd is less than 1.5.
% Store the result in I.
I = Tablibor.chfd<1.5;

% Compare usdd with gbpd using == and
% Store the result in I2.

I2 = Tablibor.usdd == Tablibor.gbpd;
```

Some Logical Operations

```
% Determine if any array elements of A are nonzero
A = [0.5 0.17 0.04 0.28 0.17 0.24 0.96]
B = (A < 0.3)
```

```
% Test matrix
A = [0 0 0.2;0 0 0.4;0 0 0.3]
B = any(A)

% see what happens with
B =any(A,1)
B =any(A,2)

% Logical AND of two matrices.
% The result contains logical 1 (true)
% Only where both matrices contain nonzero values.

A = [1 3 0; 0 2 9; 5 0 0]
B = [2 6 0; -4 2 7; 2 0 0]
A & B

% Find logical NOT
A = eye(4)
B = ~A

% Find the logical OR of two matrices.
% The result contains logical 1 (true)
% Where either matrix contains a nonzero value.
A = [4 8 0; 0 7 8; 6 0 0]
B = [7 2 0; 1 4 7; -4 0 0]
A | B
```

```

A =

    0.5000    0.1700    0.0400    0.2800    0.1700    0.2400    0.9600


B =

     0     1     1     1     1     1     0


A =

         0         0    0.2000
         0         0    0.4000
         0         0    0.3000


B =

     0     0     1


B =

     0     0     1
```

```
B =

     1
     1
     1


A =

     1     3     0
     0     2     9
     5     0     0


B =

     2     6     0
    -4     2     7
     2     0     0


ans =

     1     1     0
     0     1     1
     1     0     0


A =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1


B =

     0     1     1     1
     1     0     1     1
     1     1     0     1
     1     1     1     0


A =

     4     8     0
     0     7     8
     6     0     0


B =
     7     2     0
     1     4     7
    -4     0     0


ans =

     1     1     0
     1     1     1
     1     0     0
```

Published with MATLAB® R2015b