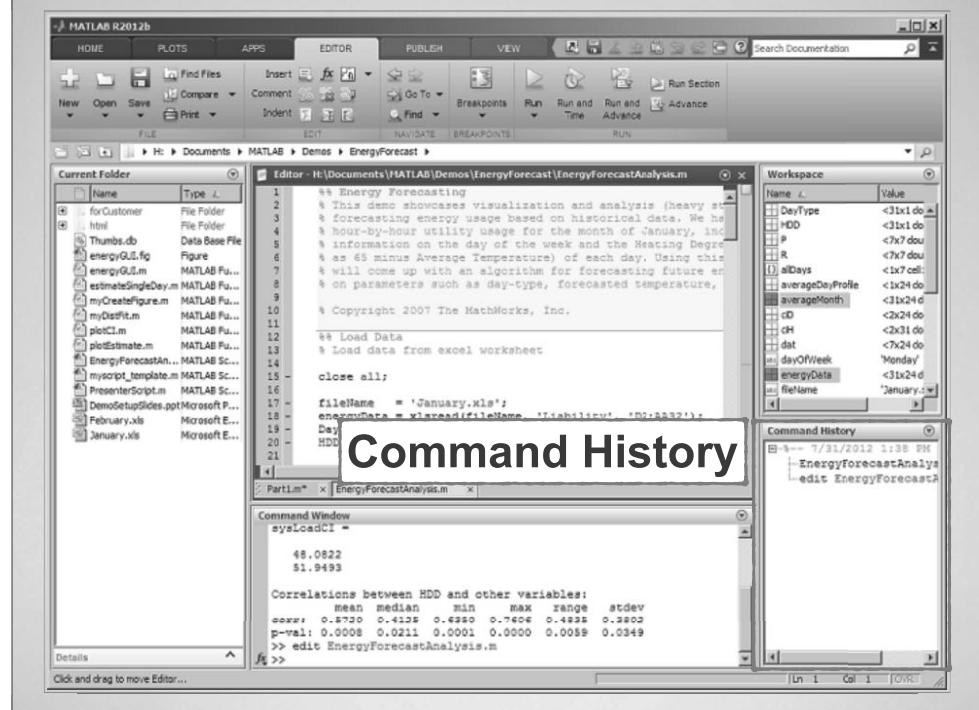
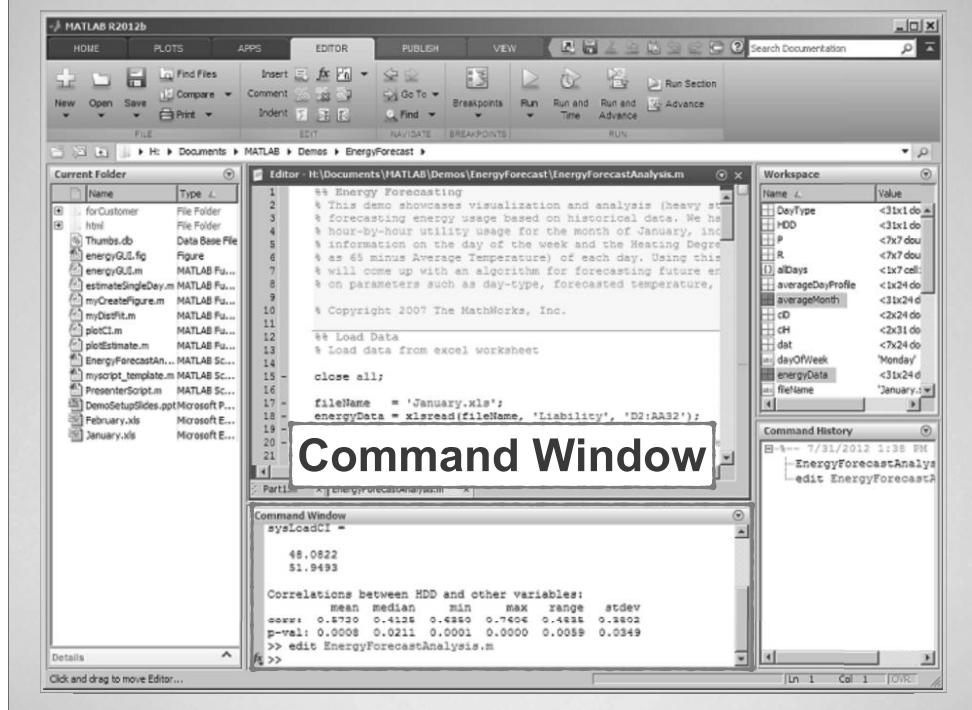
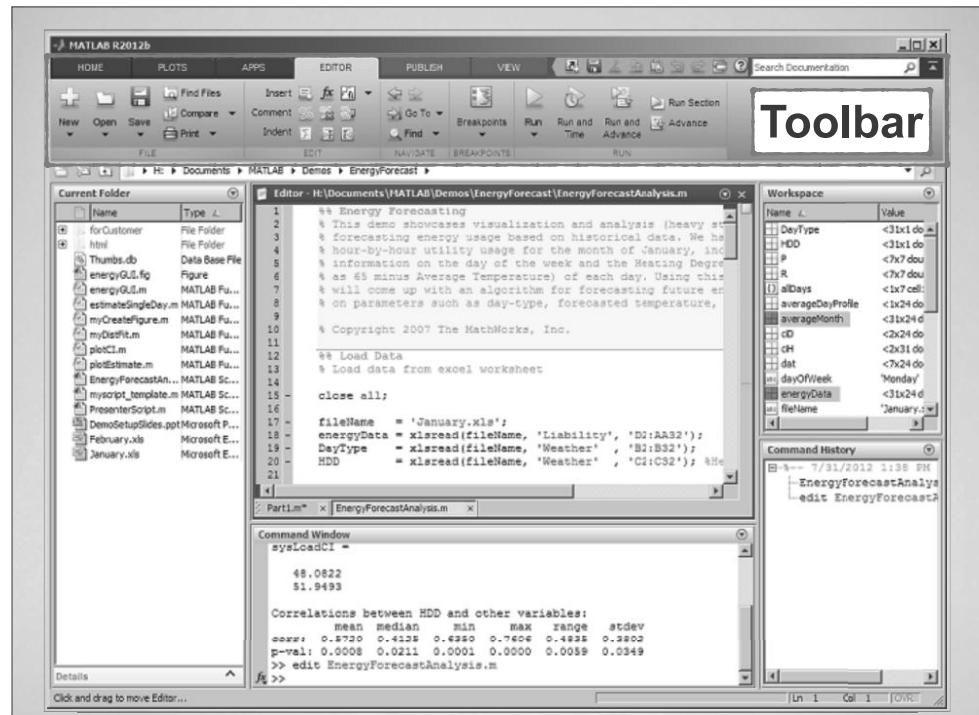


# Matlab



**Current Folder**

The screenshot shows the MATLAB R2012b interface. The Current Folder browser on the left lists files like 'forCustomer', 'Thumbs.db', 'energyGUI.fig', etc. The Editor window in the center displays the code for 'EnergyForecastAnalysis.m'. The Command Window at the bottom shows the results of running the script.

```
% Energy Forecasting
% This demo showcases visualization and analysis (heavy str
% forecasting energy usage based on historical data. We ha
% hour-by-hour utility usage for the month of January, inc
% information on the day of the week and the Heating Degree
% as 65 minus Average Temperature) of each day. Using this
% will come up with an algorithm for forecasting future en
% on parameters such as day-type, forecasted temperature,
%
% Copyright 2007 The MathWorks, Inc.
%
% Load Data
% Load data from excel worksheet
%
close all;
%
fileName = 'January.xls';
energyData = xlsread(fileName, 'Liability', 'D2:AA32');
DayType = xlsread(fileName, 'Weather', 'B2:B32');
HDD = xlsread(fileName, 'Weather', 'C2:C32'); % He
%
Correlations between HDD and other variables:
mean median min max range stdev
corr: 0.1730 0.4122 0.4220 0.7404 0.4122 0.3202
p-val: 0.0008 0.0211 0.0001 0.0000 0.0059 0.0349
>> edit EnergyForecastAnalysis.m
```

Click and drag to move Editor...

**Workspace**

The screenshot shows the MATLAB R2012b interface. The Workspace browser on the right lists variables like 'DayType', 'HDD', 'P', 'R', 'allDays', 'averageDayProfile', etc. The Editor window in the center displays the code for 'EnergyForecastAnalysis.m'. The Command Window at the bottom shows the results of running the script.

```
% Energy Forecasting
% This demo showcases visualization and analysis (heavy str
% forecasting energy usage based on historical data. We ha
% hour-by-hour utility usage for the month of January, inc
% information on the day of the week and the Heating Degree
% as 65 minus Average Temperature) of each day. Using this
% will come up with an algorithm for forecasting future en
% on parameters such as day-type, forecasted temperature,
%
% Copyright 2007 The MathWorks, Inc.
%
% Load Data
% Load data from excel worksheet
%
close all;
%
fileName = 'January.xls';
energyData = xlsread(fileName, 'Liability', 'D2:AA32');
DayType = xlsread(fileName, 'Weather', 'B2:B32');
HDD = xlsread(fileName, 'Weather', 'C2:C32'); % He
%
Correlations between HDD and other variables:
mean median min max range stdev
corr: 0.1730 0.4122 0.4220 0.7404 0.4122 0.3202
p-val: 0.0008 0.0211 0.0001 0.0000 0.0059 0.0349
>> edit EnergyForecastAnalysis.m
```

Click and drag to move Editor...

**Editor**

The screenshot shows the MATLAB R2012b interface. The Current Folder browser on the left lists files like 'forCustomer', 'Thumbs.db', 'energyGUI.fig', etc. The Editor window in the center displays the code for 'EnergyForecastAnalysis.m'. The Command Window at the bottom shows the results of running the script.

```
% Energy Forecasting
% This demo showcases visualization and analysis (heavy str
% forecasting energy usage based on historical data. We ha
% hour-by-hour utility usage for the month of January, inc
% information on the day of the week and the Heating Degree
% as 65 minus Average Temperature) of each day. Using this
% will come up with an algorithm for forecasting future en
% on parameters such as day-type, forecasted temperature,
%
% Copyright 2007 The MathWorks, Inc.
%
% Load Data
% Load data from excel worksheet
%
close all;
%
fileName = 'January.xls';
energyData = xlsread(fileName, 'Liability', 'D2:AA32');
DayType = xlsread(fileName, 'Weather', 'B2:B32');
HDD = xlsread(fileName, 'Weather', 'C2:C32'); % He
%
Correlations between HDD and other variables:
mean median min max range stdev
corr: 0.1730 0.4122 0.4220 0.7404 0.4122 0.3202
p-val: 0.0008 0.0211 0.0001 0.0000 0.0059 0.0349
>> edit EnergyForecastAnalysis.m
```

Click and drag to move Editor...

**help**

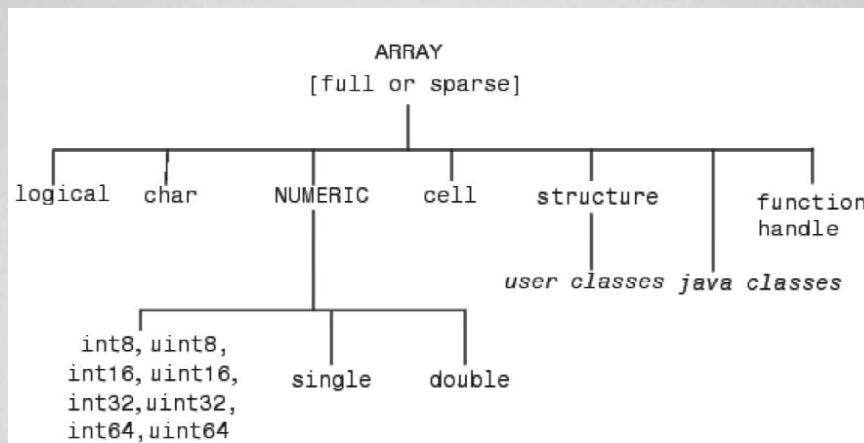
**Fx**

The screenshot shows the MATLAB R2012b interface. The help icon is visible in the toolbar. The Current Folder browser on the left lists files like 'forCustomer', 'Thumbs.db', 'energyGUI.fig', etc. The Editor window in the center displays the code for 'EnergyForecastAnalysis.m'. The Command Window at the bottom shows the results of running the script.

```
% Energy Forecasting
% This demo showcases visualization and analysis (heavy str
% forecasting energy usage based on historical data. We ha
% hour-by-hour utility usage for the month of January, inc
% information on the day of the week and the Heating Degree
% as 65 minus Average Temperature) of each day. Using this
% will come up with an algorithm for forecasting future en
% on parameters such as day-type, forecasted temperature,
%
% Copyright 2007 The MathWorks, Inc.
%
% Load Data
% Load data from excel worksheet
%
close all;
%
fileName = 'January.xls';
energyData = xlsread(fileName, 'Liability', 'D2:AA32');
DayType = xlsread(fileName, 'Weather', 'B2:B32');
HDD = xlsread(fileName, 'Weather', 'C2:C32'); % He
%
Correlations between HDD and other variables:
mean median min max range stdev
corr: 0.1730 0.4122 0.4220 0.7404 0.4122 0.3202
p-val: 0.0008 0.0211 0.0001 0.0000 0.0059 0.0349
>> edit EnergyForecastAnalysis.m
```

Click and drag to move Editor...

## ชนิดของข้อมูล



## การสร้างอาร์เรย์ 1 มิติ

```
>> x = [1 2 3 4]
```

```
x =  
1 2 3 4
```

```
>> x = [1,2,3,4]
```

```
x =  
1 2 3 4
```

```
>> x = [1;2;3;4]
```

```
x =  
1  
2  
3  
4
```

## การสร้างตัวแปร

```
>> a = 5
```

```
a =
```

```
5
```

```
>> c = uint8(a)
```

```
a =
```

```
5
```

```
>> b = 'text'
```

```
a =
```

```
text
```

```
>> d = uint8('a')
```

```
a =
```

```
97
```

>> whos % แสดงรายละเอียดตัวแปรใน Workspace

## การสร้างอาร์เรย์ 2 มิติ

```
>> y = [1,2,3; 4,5,6]
```

```
y =  
1 2 3  
4 5 6
```

```
>> y(3) = 7
```

```
y =  
1 7 3  
4 5 6
```

```
>> y(3)
```

```
ans =  
2
```

```
>> y(2,3) = 10
```

```
y =  
1 7 3  
4 5 10
```

\* จะใช้ commas (,) และ semicolons (;)

## การสร้างอาร์เรย์ 3 มิติ

```
>> z(:,:,1) = [1,2; 3,4];      z(:,:,1) =  
>> z(:,:,2) = [5,6; 7,8];      1     2  
>> z(:,:,3) = [9,10; 11,12];   3     4  
                                z(:,:,2) =  
                                5     6  
                                7     8  
                                z(:,:,3) =  
                                9     10  
                                11    12
```

## Arithmetic Operators

Assignment =	$a = b$ (assign $b$ to $a$ )
Addition +	$a+b$
Subtraction -	$a-b$
Multiplication * or .*	$a*b$ or $a.*b$
Division / or ./	$a/b$ or $a./b$
Power ^ or .^	$a^b$ or $a.^b$

## ขนาดของมิติ

```
>> a = [1 2 3; 4 5 6]
```

```
a =
```

```
1     2     3  
4     5     6
```

```
>> [x, y] = size(a)
```

```
x =
```

```
2
```

```
y =
```

```
3
```

```
>> size(a,1)
```

```
ans =
```

```
2
```

```
>> size(a,2)
```

```
ans =
```

```
3
```

## การคูณและการหารในเวกเตอร์

```
b = [1 2; 3 4]
```

```
b =
```

```
1     2  
3     4
```

```
c = b*b
```

```
c =
```

```
7     10  
15    22
```

```
d = b.*b
```

```
d =
```

```
1     4  
9     16
```

```
e = b./b
```

```
e =
```

```
1     1  
1     1
```

## Comparison Operators

Less Than	<
Less Than or Equal	<=
Greater Than	>
Greater Than or Equal	>=
Equal To	==
Not Equal To	~=

## Logical Operators

not	~
and	&&
or	

## Conditional Structures

```
if expression cond.  
    sentences  
elseif expr. cond.  
    sentences  
else  
    sentences  
end
```

```
a = input('a = ');\n'b = input('b = ');\nif a == b,\n    fprintf('a is equal to b\n');\nelseif a > 0 && b > 0\n    fprintf('both positive\n');\nelse\n    fprintf('other case\n');\nend
```

## Conditional Structures

```
for variable = expr  
    sentence;  
    ...  
    sentence;  
end
```

```
M = rand(4,4);  
sum = 0;  
  
for i = 1:4  
    for j = 1:4  
        sum = sum + M(i,j);  
    end  
end  
  
fprintf('sum = %d\n',sum);
```

## Conditional Structures

```
while expr  
    sentence;  
    ...  
    sentence;  
end
```

```
M = rand(4,4);  
i = 1; j = 1; suma = 0;  
  
while i <= 4  
    while j <= 4  
        suma = suma + M(i,j);  
        j = j+1;  
    end  
    i = i+1;  
end  
  
fprintf('suma = %f\n',suma);
```

## M-files: Script

- Without input arguments,  
they do not return any value

filename.m

```
x = [4 3 2 10 -1];  
n = length(x);  
suma1 = 0; suma2 = 0;  
for i=1:n  
    suma1 = suma1 + x(i);  
    suma2 = suma2 + x(i)*x(i);  
end  
promig = suma1/n;  
desvia = sqrt(suma2/n - promig*promig);
```

## M-files

Text files containing Matlab programs. Can be called from the command line or from other M-files

- Present “.m” extension
- Two kind of M-files:
  - Scripts
  - Functions

## M-files: Function

```
function [out1, out2, ..., outN] = name-function (par1, par2, ..., parM)  
    sentence;  
    ....  
    sentence;  
end
```

fn\_sum.m

```
function [x,y] = fn_sum(a,n)  
    x = 0; y = 0;  
    for i=1:n  
        x = x + a(i); % sum ai  
        y = y + a(i)^2; % sum ai^2  
    end  
end
```

## Image Processing Toolbox

**imread** Read image from graphics file

```
imread(filename)
```

**imshow** Display image

```
imshow(filename)
```

```
A = imread('tire.tif');  
imshow(A)
```



tire.tif

## Image Processing Toolbox

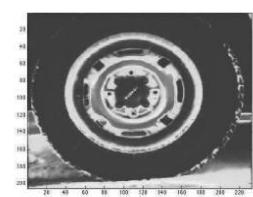
**imagesc** Scale data and display image object

```
imagesc(C)
```

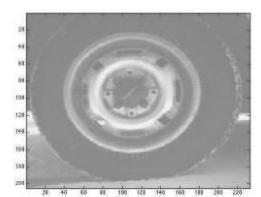
**colormap** Set and get current colormap

```
colormap(map)
```

```
imagesc(A);  
colormap(summer);
```

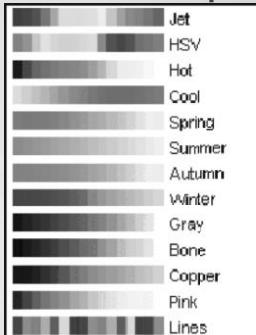


imagesc



colormap

map



## Image Processing Toolbox

**imwrite** Write image to graphics file

```
imwrite(A,filename)
```

**imfinfo** Information about graphics file

```
info = imfinfo(filename)
```

```
imwrite(A, 'tire.jpg');  
imfinfo('tire.jpg');
```



tire.jpg

## Image Processing Toolbox

**rgb2gray** Convert RGB image or colormap to grayscale

```
I = rgb2gray(RGB)
```

**im2bw** Convert image to binary image,  
based on threshold

```
BW = im2bw(I, level)
```

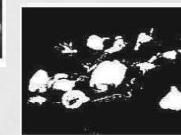
```
Acolor = imread('peppers.png');  
imshow(Acolor);
```



```
Agray = rgb2gray(Acolor);  
imshow(Agray);
```



```
Abw = im2bw(Agray);  
imshow(ABW);
```



# Image Processing Toolbox

**subplot** Create axes in tiled positions

subplot(m,n,p)

**title** Add title to current axes

title(str)

```
subplot(1,3,1);
imshow(Acolor); title('Color Image');
subplot(1,3,2);
imshow(Agray); title('Gray Image');
subplot(1,3,3);
imshow(Abw); title('Binary Image');
```

