```
% Creating vectors and matrices
octave> A = [1 5 2; -3 2 0; 4 -1 -2]      % matrix; semicolons indicate next row
A =

   1   5   2
  -3   2   0
   4  -1  -2

octave> A(2,3)     % grabs entry from matrix A in 2nd row, 3rd column
ans = 0

octave> A(:,3)     % grabs all rows but just 3rd column
ans =

   2
   0
  -2

octave> A(2, [1 3])    % grabs entries in A's 2nd row, 1st and 3rd cols
ans =

  -3   0

octave> A(1:2, 3)      % entries in both 1st and 2nd rows, only 3rd col
ans =

   2
   0

octave> A(1:2, end)     % same as above, since 3rd row is end row
ans =

   2
   0

octave> A(2:end, 2)     % helpful if you don't know how many rows A has
ans =

   2
  -1

octave> x = [2 3 -1 4 7]    % a 1-by-5 matrix, called a row vector
ans =

   2   3  -1   4   7

octave> 10:17     % row vector (unnamed) with sequential entries
x =

   10   11   12   13   14   15   16   17

octave> x = 10:17;     % same x as above, but semicolon suppresses output

octave> x(3)        % displays 3rd entry in row vector x
ans =  -1

octave> x = 1:.25:2     % entry between colons alters stepsize from 1
x =

    1.0000    1.2500    1.5000    1.7500    2.0000

octave> A'      % prime exchanges rows for columns when A has real entries
ans =

   1  -3   4
   5   2  -1
```

```
   2   0  -2

octave> x = (1:.25:2)'    % recreates x from above, now as column vector
x =

   1.0000
   1.2500
   1.5000
   1.7500
   2.0000


% Mathematical functions built into Octave
octave> cos(pi)     % Octave has full collection of mathematical functions
ans = -1

octave> log(2)     % log is logarithm base e (i.e., natural logarithm)
ans =  0.69315

octave> sin(1:.25:2)   % built-in fns can work on vector of inputs
ans =

   0.84147   0.94898   0.99749   0.98399   0.90930

octave> sin(x)     % shape of output depends on shape of input
ans =

   0.84147
   0.94898
   0.99749
   0.98399
   0.90930

octave> exp(A)     % inputs can even be matrices
ans =

   2.7183e+00   1.4841e+02   7.3891e+00
   4.9787e-02   7.3891e+00   1.0000e+00
   5.4598e+01   3.6788e-01   1.3534e-01


% Plotting in the plane
octave> xVals = 0:.5:2*pi    % "pixel" locations in interval [0, 2*pi]
xVals =

 Columns 1 through 7:

    0.00000    0.50000    1.00000    1.50000    2.00000    2.50000    3.00000

 Columns 8 through 13:

    3.50000    4.00000    4.50000    5.00000    5.50000    6.00000

octave> plot(xVals, cos(xVals))   % inputs are vectors specifying points

octave> plot(xVals, cos(xVals), '.')   % same points now disconnected
octave> plot(xVals, cos(xVals), 'o')   % bigger symbol for points
octave> plot(xVals, cos(xVals), '.', 'MarkerSize', 18)   % bigger dots

octave> plot(cos(xVals), sin(xVals), '.', 'MarkerSize', 18)   % circle?
octave> plot(cos(xVals), sin(xVals))     % same points connected up
octave> axis equal     % aspect ratio set to 1:1

octave> xVals = 0:.1:2*pi;     % smaller stepsize ==> better resolution
octave> plot(xVals, cos(xVals))     % resulting cosine graph
```

```
octave> plot(cos(xVals), sin(xVals))     % circle?
octave> axis equal


% Defining your own mathematical functions
octave> f = @(x) x^2    % way of defining own math function f(x) = x^2
f =

@(x) x ^ 2

octave> f(-2)
ans = 4

% But the function created by the definition above can't handle vector input
octave> f(1:5)
error: for x^y, only square matrix arguments are permitted and one argument must
 be scalar.  Use .^ for elementwise power.
error: called from
    @<anonymous> at line 1 column 11

% Built-in functions CAN handle vector inputs.  To make our own capable of
% handling vector input, operations like multiplication, division, and
% exponentiation must be "vectorized".  This is achieved with an extra ".":
%   instead of * use .*
%   instead of / use ./
%   instead of ^ use .^
octave> f = @(x) x.^2
octave> f(1:5)
ans =

    1    4    9   16   25

octave> xVals = -2:.1:2;      % vector of x-values for displayed points
octave> plot(xVals, f(xVals))    % now produces desired plot

octave> xVals = -2:.5:2     % coarser mesh so outputs aren't so long
xVals =

 Columns 1 through 8:

   -2.0000   -1.5000   -1.0000   -0.5000    0.0000    0.5000    1.0000    1.5000

 Column 9:

    2.0000

octave> f(xVals)    % a 1-by-8 vector of outputs from f given inputs of xVals
ans =

 Columns 1 through 8:

   4.00000   2.25000   1.00000   0.25000   0.00000   0.25000   1.00000   2.25000

 Column 9:

   4.00000

octave> g = @(t) t.^3
octave> g(xVals)     % another 1-by-8 vector
ans =

 Columns 1 through 8:

  -8.00000  -3.37500  -1.00000  -0.12500   0.00000   0.12500   1.00000   3.37500

 Column 9:
```

```
   8.00000

octave> [f(xVals); g(xVals)]    % previous row vectors in single matrix
ans =

 Columns 1 through 8:

   4.00000   2.25000   1.00000   0.25000   0.00000   0.25000   1.00000   2.25000
  -8.00000  -3.37500  -1.00000  -0.12500   0.00000   0.12500   1.00000   3.37500

 Column 9:

   4.00000
   8.00000


% More plotting: 2 functions at once
octave> plot(xVals, [f(xVals); g(xVals)])    % plot of both power fns

octave> xVals = -2:.1:2;     % back to finer mesh
octave> plot(xVals, [f(xVals); g(xVals)])    % plot with finer mesh

octave> plot(xVals, [xVals; xVals.^2; xVals.^3; xVals.^4])


% surface (3D) plotting
octave> f = @(x,y) x.^2 + y.^2     % the function  f(x, y) = x.^2 + y.^2

octave> [X,Y] = meshgrid(-1:.5:1, 0:3)    % mesh on [-1,1] x [0,3]
X =

  -1.00000  -0.50000   0.00000   0.50000   1.00000
  -1.00000  -0.50000   0.00000   0.50000   1.00000
  -1.00000  -0.50000   0.00000   0.50000   1.00000
  -1.00000  -0.50000   0.00000   0.50000   1.00000

Y =

   0   0   0   0   0
   1   1   1   1   1
   2   2   2   2   2
   3   3   3   3   3

octave> mesh(X, Y, f(X,Y))     % plots surface f over our mesh

octave> [X,Y] = meshgrid(-1:.1:1, 0:.1:2);     % finer mesh
octave> mesh(X, Y, f(X,Y))     % plots surface f over our mesh


% Case-sensitivy of the software
octave> X
X =

 Columns 1 through 8:

  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
```

```
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000
  -1.00000  -0.90000  -0.80000  -0.70000  -0.60000  -0.50000  -0.40000  -0.30000

 Columns 9 through 16:

  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000
  -0.20000  -0.10000   0.00000   0.10000   0.20000   0.30000   0.40000   0.50000

 Columns 17 through 21:

   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000
   0.60000   0.70000   0.80000   0.90000   1.00000

octave> x
x =

   1.0000
   1.2500
   1.5000
```

```
   1.7500
   2.0000

octave> x == X;   % Octave is case sensitive; these are not equal
error: mx_el_eq: nonconformant arguments (op1 is 5x1, op2 is 21x21)


% More creating vectors and matrices
octave> A = zeros(5,4)     % generates 5-by-4 matrix full of zeros
A =

   0   0   0   0
   0   0   0   0
   0   0   0   0
   0   0   0   0
   0   0   0   0

octave> A = zeros(5)     % generates 5-by-5 matrix full of zeros
A =

   0   0   0   0   0
   0   0   0   0   0
   0   0   0   0   0
   0   0   0   0   0
   0   0   0   0   0

octave> x = ones(5,1)     % generates 5-by-1 matrix/vector full of ones
x =

   1
   1
   1
   1
   1

octave> diag(x)      % makes square matrix with diagonal entries from x
ans =

Diagonal Matrix

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

octave> diag(3*x)      % makes square matrix with diagonal entries from x
ans =

Diagonal Matrix

   3   0   0   0   0
   0   3   0   0   0
   0   0   3   0   0
   0   0   0   3   0
   0   0   0   0   3

octave> diag(x, -1)      % entries from x placed on 1st subdiagonal
ans =

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

```
octave> B = diag(x,-1) - 2*diag(ones(6,1)) + diag(x,1)
B =

  -2   1   0   0   0   0
   1  -2   1   0   0   0
   0   1  -2   1   0   0
   0   0   1  -2   1   0
   0   0   0   1  -2   1
   0   0   0   0   1  -2

octave> [B [1;2;3;4;5;6]; ones(2,7)]  % 8-by-7 matrix built from other matrices
ans =

  -2   1   0   0   0   0   1
   1  -2   1   0   0   0   2
   0   1  -2   1   0   0   3
   0   0   1  -2   1   0   4
   0   0   0   1  -2   1   5
   0   0   0   0   1  -2   6
   1   1   1   1   1   1   1
   1   1   1   1   1   1   1
```