In MATLAB terminology, both vectors and matrices are arrays of numerical values.

Technically, a vector is a form of a matrix having one row or one column (read about isvector and ismatrix commands for more information).

Practically, however, matrices are usually referred to in the documentation as having at least 2 columns and 2 rows, so I would rather call xL and xU vectors.

A vector is a one-dimensional array of numbers. MATLAB allows creating two types of vectors −

* Row vectors
* Column vectors

Column Vectors

**Column vectors** are created by enclosing the set of elements in square brackets, using semicolon to delimit the elements.

c = [7; 8; 9; 10; 11]

c

c(3) Ref

c(:)

Row Vectors

C = [1 2 3 4 5 6 7 8 9];

Vector Add, and Subtract

A = [7, 11, 15, 23, 9];

B = [2, 5, 13, 16, 20];

C = A + B;

D = A - B;

disp(C);

disp(D);

Scaler

v = [ 12 34 10 8];

m = 5 \* v

r = [r1,r2]

r = [r1;r2]

Matrix:

a = [ 1 2 3 4 5; 2 3 4 5 6; 3 4 5 6 7; 4 5 6 7 8]

mx(m, n);

Array

zeros(5)

ones(4,3)

rand(3, 5)

magic(4)

v = [ 23 45 12 9 5 0 19 17] % horizontal vector

sort(v)

c = cell(2, 5);

c = {'Red', 'Blue', 'Green', 'Yellow', 'White'; 1 2 3 4 5}

String functions

A complete reference of all of MATLAB's string functions can be obtained by typing 'help strfun' at the MATLAB prompt.

| Function/Syntax | Description | Example |
| --- | --- | --- |
| t = [s1 s2 s3] | String concatenation | s1 = 'A dumb'; s2 = ' ';  s3 = 'example';  t = [s1 s2]  ⇒ t = 'A dumb example' |
| s(n)  s(n:m) | String addressing/sectioning | s = 'knowhow';  t1 = s(4) ⇒ t1 = 'w'  t2 = s(4:6) ⇒ t2 = 'who' |
| t = strcat(s1,s2,...) | String concatenation | s1, s2, s3 as above  t = strcat(s1,s2,s3)  ⇒ t = 'A dumb example' |
| t = char(x) | Create string from numeric array. | t = char([77 65 84 76 65 66])  ⇒ t = 'MATLAB' |
| t = num2str(x) | Convert numbers to a string. | t = num2str(3.141516)  ⇒ t = '3.141516' |
| t = sprintf(FORMAT, A,B,...) | Create a string by formating data arguments A,B,... according to the (c-language-like style) FORMAT string | t = sprintf('%0.5g',(1+sqrt(5))/2)  ⇒ t = '1.618' |
| t = lower(s) | Convert string to lowercase. | t =lower('Do not SHOUT!')  ⇒ t = 'do not shout!' |
| t = upper(s) | Convert string to uppercase | t = upper('I will say it louder')  ⇒ t = 'I WILL SAY IT LOUDER' |
| yn = strcmp(s1,s2) | returns logical 1 (true) if strings S1 and S2 are the same and logical 0 (false) otherwise | strcmp('not','Not') &rArr 0  strcmp('the same', ['the ', 'same']) &rArr 1 |
| t = findstr(s1,s2) | returns the starting indices of any occurrences of the shorter of the two strings in the longer. | s = 'How much wood would a woodchuck chuck?';  t = findstr(s,'wood') ⇒ t= [10 23] |
| S = strvcat(s1,s2,s3,..) | Vertically concatenate strings. Automatically pads each string with blanks in order to form a valid matrix. | S = strvcat('several','different', 'strs');  ⇒ S = ['several '  'different'  'strs ']; |
| t = ischar(x) | returns 1 if S is a character array and 0 otherwise. | ischar(123) ⇒ 0  ischar('123') ⇒ 1 |

function max = mymax(n1, n2, n3, n4, n5)

%This function calculates the maximum of the

% five numbers given as input

max = n1;

if(n2 > max)

max = n2;

end

if(n3 > max)

max = n3;

end

if(n4 > max)

max = n4;

end

if(n5 > max)

max = n5;

end

## Example 1

Let us load and display an image file. Create a script file and type the following code in it −

filename = 'smile.jpg';

A = importdata(filename);

image(A);

## Example 2

In this example, we import a text file and specify Delimiter and Column Header. Let us create a space-delimited ASCII file with column headers, named *weeklydata.txt*.

Our text file weeklydata.txt looks like this −

SunDay MonDay TuesDay WednesDay ThursDay FriDay SaturDay

95.01 76.21 61.54 40.57 55.79 70.28 81.53

73.11 45.65 79.19 93.55 75.29 69.87 74.68

60.68 41.85 92.18 91.69 81.32 90.38 74.51

48.60 82.14 73.82 41.03 0.99 67.22 93.18

89.13 44.47 57.63 89.36 13.89 19.88 46.60

Create a script file and type the following code in it −

filename = 'weeklydata.txt';

delimiterIn = ' ';

headerlinesIn = 1;

A = importdata(filename,delimiterIn,headerlinesIn);

% View data

for k = [1:7]

disp(A.colheaders{1, k})

disp(A.data(:, k))

disp(' ')

end

filename = '/data/myfile.txt';

rows = 7;

cols = 5;

% open the file

fid = fopen(filename);

% read the file headers, find M (number of months)

M = fscanf(fid, '%\*s %\*s\n%\*s %\*s %\*s %\*s\nM=%d\n\n', 1);

% read each set of measurements

for n = 1:M

mydata(n).time = fscanf(fid, '%s', 1);

mydata(n).month = fscanf(fid, '%s', 1);

% fscanf fills the array in column order,

% so transpose the results

mydata(n).raindata = ...

fscanf(fid, '%f', [rows, cols]);

end

for n = 1:M

disp(mydata(n).time), disp(mydata(n).month)

disp(mydata(n).raindata)

end

% close the file

fclose(fid)