Matlab Level 2: More on matrices

You can create column and row vectors using the ‘:’

```
>>a=1:10
>>b=0:20
```

By default, the increment between positions is ‘1’, but you can change that:

```
>>d=0:0.01:1
>>e=0:0.001:1
>>f=1:2:20
```

Check out the size of your variables

```
>>whos
```

Note that the ‘size’ is listed as ‘rows x columns’

You can change any position within the array

```
>>a(1)=0
>>a(5)=0
>>a(10)=0
```

Note that the notation ‘a(1)’ refers to a particular position in a vector whereas the notation ‘a1’ would be a valid variable name.

You can multiple the entire vector by any scalar (i.e., a single number)

```
>>c=a*2
```

You can also change a vector value by reassigning itself

```
>>a=a*10
```

You can make a ‘row-vector’ into a ‘column-vector’ by transposing

```
>>a=a'
```

Now, clear your workspace and create the following vectors

```
>>clear
>>a=1:10
>>b=11:20
```

Now, combine them into one vector, with ‘a’ in row 1, ‘b’ in row 2:

```
>>c=[a;b]
```

Do the same thing, but change ‘a’ and ‘b’ to column vectors and put ‘a’ in column 1, ‘b’ in column 2

```
>>a=a'
>>b=b'
>>c=[a b]
```

You can change any element in ‘c’ by c(row position, column position)

```
>>c(6,2)=0
>>c(5,1)=0
```

Watch what happens if you assign a position that does not currently exist:

```
>>c(20,2)=1
```

You can also access an entire column by using the ‘:’

```
>>b=c(:,1)
```

What ‘:’ means is: ‘access all the rows in column 1.’
You can also access only part of all the rows or columns:

```
>>b=c(1:5,2)
```

You can also do some neat math using the array notation

```
>>c(:,3)=c(:,1)*2
```

Watch your math expressions. When multiplying vectors, MATLAB performs vector math. In most cases, you will be doing math element by element. That is, you may want to multiply the first element in a vector by the first element in some other vector. To do this, you use the `.*` notation.

```
>>a=1:5;
>>b=2:6;
>>c=a.*b
```

Note that the expression `a.*b` multiplies corresponding elements in `a` and `b` and gives a different answer than `a*b`.

Sometimes you may want to either know how long a vector is or access the entire vector without the knowledge of the length of the vector. To do this, use the `length` function (or, alternatively `size`).

```
>>help length
>>d=a(2:length(a)
>>e=a(2:length(a)-1);
```

Explore some commonly used functions:

```
>>help mean
>>help sum
>>help std
>>help min
>>help max
>>help find
```

Try these problems.

1. Create a vector of the odd whole numbers between 21 and 51.
2. Create a vector from 0 to 1 at an increment of 1/1000. How many elements are there?
3. Given the vector $x=[5 \ 3 \ 2 \ 10 \ 7 \ 4]$
   a. add 10 to each element
   b. add 10 to only elements 1, 3, and 5
   c. Multiply elements 2 and 4 by 2
4. Given the vectors $x_1=[1 \ 2 \ 3 \ 4 \ 5]$ and $x_2=[2 \ 3 \ 4 \ 5 \ 6]$
   a. add the vectors, element by element (i.e., $x_1(1)+x_2(1)$…
   b. multiply each element in $x_1$ by the corresponding element in $x_2$
   c. divide each element in $x_1$ by the corresponding element in $x_2$
5. Given a vector $x=[1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9]$, write commands to
   a. Assign the first row to variable $x_1$, 2nd row to $x_2$ and 3rd row to $x_3$
   b. Assign the last two columns to a variable called ‘$x_4$’.
   c. Use a matlab function to compute sum of all the elements in the first column.
   d. Use a matlab function to compute the mean value of each column.
6. Given a vector $x=[1 \ 3 \ 2 \ 3 \ 1 \ 7 \ 3 \ 2 \ 1 \ 0 \ 3 \ 5]$
   a. Use a matlab function to identify the maximum value within the vector
   b. Use a matlab function to calculate the mean of the vector
   c. Use a matlab function to find all the positions that contain the value ‘3’
   d. Use a matlab function to find all the positions that are greater than ‘3’