

# Octave Tutorial 2: Vectors and Matrices

with extracts from *Introduction to Octave*, by P.J.G. Long

In this tutorial you will learn how to

- build vectors and matrices using different notations and appropriate built-in functions;
- construct new vectors and matrices from existing ones;
- extract and change single elements or subsets of vectors and matrices.

Octave is a program specially designed for manipulating matrices. Simply speaking, an  $n$  by  $m$  matrix is a box of numbers with  $n$  rows and  $m$  columns. Vectors are a special case of a matrix where there is only one row (a row vector) or only one column (a column vector). Numbers are also a special case of a matrix as Octave regards them as 1 by 1 matrices. So building and manipulating matrices is very important in Octave.

As in Tutorial 1, here we will refer to `octave#:#>` as the Octave prompt. To execute a command, type it at the prompt and hit return.

## 1 Building vectors and matrices

### 1.1 The square brackets

There are a lot of ways for defining vectors and matrices in Octave. Usually the easiest thing to do is to type the elements inside *square* brackets `[]`. Try it! Type

```
octave#:#> A = [1 2 3]
```

at the prompt and hit enter. Octave has assigned the row vector `[1 2 3]` to the variable `A`. Recall that the equal sign `=` is the operator that assigns a value to a variable (if you don't specify the variable name, Octave will assign the result to the variable `ans`). Try it! Simply type `[1 2 3]` at the prompt, followed by enter, and observe the output on the screen).

Vector elements can also be entered separated by commas. For example, the command

```
octave#:#> B = [0.1,2,5]
```

will create the row vector `B=[0.1 2 5]`.

Now try typing

```
octave#:#> A = [1 2 3; 4 5 6]
```

followed by a carriage return. What has happened here? The variable `A` has been re-assigned.

Its value is now the 2 by 3 matrix  $\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$ . The semicolon `;` after the third entry marked the end of the first row of elements.

Try another example. Type

```
octave#:#> C = [4;7.5;-1]
```

and hit return. This time Octave has assigned the column vector  $\begin{bmatrix} 4.0 \\ 7.5 \\ -1.0 \end{bmatrix}$  to the variable `C`.

In summary, a list of numbers separated by spaces or commas, inside square brackets, defines a row vector; numbers separated by semicolons (or carriage returns), define a column vector. Matrices are entered row by row using the same syntax as for vectors.

## 1.2 The transpose

To interchange rows with columns, that is, to find the transpose of a vector or a matrix, use the apostrophe. For example, the command

```
octave#:#> C = [4 7.5 -1]'
```

will transform the row vector  $C = [4 \ 7.5 \ -1]$  into a column vector. This is another way to build column vectors.

## 1.3 The colon notation

A useful shortcut for constructing row vectors with evenly spaced entries is using the colon symbol. Try typing

```
octave#:#> X = [1:10]
```

followed by a carriage return. Got the idea? The colon `:` tells Octave to create a (row) vector of numbers starting from the first number and counting up to (and including) the second number, incrementing by 1 each time. Note that when using this colon notation, you don't need to write the numbers inside square brackets. Typing `X=1:10` would have produced the same output as typing `X = [1:10]`.

Now try

```
octave#:#> X = 1:2:10
```

What has happened? The number in the middle — in this case 2 — is the increment step. The step can be any size, even negative, in which case it will count down to the end number.

The colon notation can also be used in the definition of multi-row matrices. For example,

```
octave#:#> X = [1:3; 8:-2:4]
X =
  1 2 3
  8 6 4
```

---

*Exercise 1:* What command line would you type to assign the vector  $[1.0 \ 0.8 \ 0.6 \ 0.4 \ 0.2]$  to the variable `X` using the colon notation?

*Exercise 2:* Using the colon notation, create a vector containing the numbers starting with 2 and ending with 6 with steps of 0.25.

## 1.4 Constructing vectors and matrices from existing ones

You can construct a vector from an existing vector by including it in the definition. For example, above you defined the variable `B` to be the vector  $[0.1 \ 2 \ 5]$ , now you can construct a new vector `Y` to contain the entries  $[0.1 \ 2 \ 5 \ 6]$  using the vector `B` by typing

```
octave#:#> Y = [B 6]
```

Existing vectors can be used to construct individual rows of matrices. For example,

```
octave#:#> D = [3 4 5];
octave#:#> T = [D; 6 7 8]
T=
  3 4 5
  6 7 8
```

Note that the semicolon ; at the end of the first command line has suppressed the output of that command.

New matrices can be built from old ones. If A and B are two matrices with the same number of rows, then [A B] is the matrix obtained by placing them side by side. Try it!

## 1.5 Vector and matrix creation functions

Octave provides a set of built-in functions for creating particular vectors and matrices. A list of these functions is provided below. Refer to the Octave help for more information on each function.

<code>zeros(M,N)</code>	Create an $M$ by $N$ matrix where every element is zero.
<code>ones(M,N)</code>	Create an $M$ by $N$ matrix where every element is one.
<code>eye(M)</code>	Create the $M$ by $M$ identity matrix.
<code>rand(M)</code>	Create an $M$ by $M$ matrix with random elements.
<code>diag(X)</code>	Create a diagonal matrix with vector $X$ on the main diagonal.
<code>linspace(x1,x2,N)</code>	Create a row vector of $N$ elements, evenly spaced between $x1$ and $x2$ .

---

*Exercise 3:* Repeat Exercises 1 and 2 using the `linspace` function.

*Exercise 4:* Using the `zeros` and `ones` functions, create a 6 element column vector where all the entries are 0 and a 4 element row vector where all the entries are 1.

*Exercise 5:* Without typing element by element, build the following matrix

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 2 & 3 & 4 & 5 \end{bmatrix}.$$

## 2 Extracting and changing elements

Individual elements of a vector or matrix are referred to by using normal brackets () (don't confuse them with square brackets used in defining a matrix!), and they are numbered starting at one (unlike in C++ where they start from zero). For example, define the following six-element row vector

```
octave#:#> F = [3 4 5 6 7 8]
F =
  3 4 5 6 7 8
```

You can extract the third element of F by typing

```
octave#:#> F(3)
ans = 5
```

You can change a vector entry by simply reassigning the new value. For example, by executing

```
octave#:#> F(3)=6
```

Octave has replaced the third element of F with the number 6.

For a matrix, you need to specify the row co-ordinate first and then the column co-ordinate. For example, consider the following 2 by 4 matrix  $X = [3 \ 4 \ 5 \ 1; \ 6 \ 7 \ 8 \ 2]$ , you can extract the element in the first row and third column by typing

```
octave#:#> X(1,3)
ans = 5
```

To extract a subset of elements, you can use the colon `:` operator. For example, the command `octave#:#> X(2,1:3)` extracts, from the second row, all the elements between the first and the third column (included). Try it!

To extract an entire row or column, use the colon `:` operator like this,

```
octave#:#> X(1,:)
ans = 3 4 5 1
```

This will extract the first row of the matrix  $X$ . In this notation, the `:` operator refers to all the elements in the specified row or column.

To change some elements of a matrix, simply reassign their values. For example, if you have followed this tutorial closely, you should have a vector  $F=[3 \ 4 \ 6 \ 6 \ 7 \ 8]$  saved in memory (if not, define it). Can you tell how the matrix  $X$  changes if we execute the following command?

```
octave#:#> X(1,:)=F(3:6)
```

The first row of  $X$  has been replaced with the last 4 elements of vector  $F$ .

---

*Exercise 6:* Create a matrix by typing `M=rand(5)`. What is the output of the following commands: `M(:,3)`, `M(1:2,4)`, `M(3,:)`, `M(3,4)`? Answer this question without using the software, then check your answers with Octave.

### 3 More practice problems

- Given the matrix  $A=[2 \ 4 \ 1; \ 6 \ 7 \ 2; \ 3 \ 5 \ 9]$  answer the following questions without using Octave (you can then check your answer using the software):
  - what is the Octave command(s) that assigns the first row of  $A$  to a vector called  $x1$ ?
  - what is the output of the Octave command `A(:,2)`?
  - what is the output of the Octave command `A(1:2,2)`?
  - what is the output of the Octave command `[A(1:2,1) A(1:2,2)]`?
  - what other Octave command results in the same output obtained in part d?
  - what is the output of the Octave command `[A ; A(2:3,:)]`?
- The Octave function `sum(X)` returns the sum of the elements in vector  $X$ . Use this function to compute the sum over the rows of the matrix  $A$  defined in Exercise 1.
- Enter the vector `[0 2 4 6 6 4 2 0]` in Octave without typing element by element.
- Let  $a=[1 \ 2 \ 3 \ 4 \ 5 \ 6]$ ,  $b=[6 \ 5 \ 4 \ 3 \ 2 \ 1]$ , and  $c=[2 \ 4 \ 6 \ 8 \ 10 \ 12]$ . Define the matrix  $X = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$ . Then set all the entries of  $X$  in the first and second row and second and third column to zero.