

Make and Manipulate Matrices

Conclusion

Summary

Summary: Make and Manipulate Matrices

Manually Enter Arrays

Create a Row Vector

Use square brackets and separate the values using a comma or a space.

```
a = [10 15 20 25]
```

```
a =  
10 15 20 25
```

```
a = [10, 15, 20, 25]
```

```
a =  
10 15 20 25
```

Create a Column Vector

Use square brackets and separate the values using a semicolon.

```
b = [2; 3; 5; 7]
```

```
b =  
2  
3  
5  
7
```

Transpose a Vector

Use the transpose operator.

```
c = b'
```

```
c =  
2 3 5 7
```

Create a Matrix

Use square brackets and enter values row-by-row.
Separate values in a row using a comma or a space, and use a semicolon to start a new row.

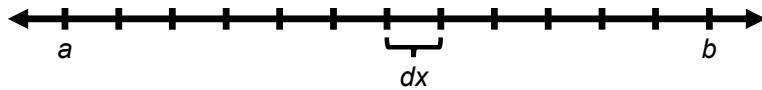
`A = [1 3 5; 2 4 6]`

`A =`

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

Create Evenly Spaced Vectors

Given the Starting Value, Ending Value, and Interval



Use the colon operator to separate the starting value, the interval, and the ending value.

`a = 3:2:7`

`a =`

```
3 5 7
```

When the Interval Is 1

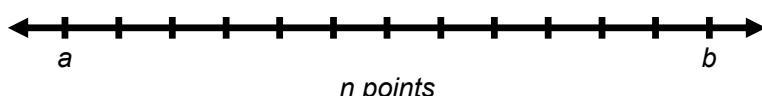
Use the colon operator to separate the starting and the ending value.

`b = 3:7`

`b =`

```
3 4 5 6 7
```

Given the Starting Value, Ending Value, and Number of Elements



Use the function `linspace` when you know the number of elements in the vector.

`c = linspace(3.2,8.1,5)`

`c =`

```
3.2 4.42 5.65 6.87 8.1
```

Concatenate Arrays

Horizontal Concatenation

$$\begin{bmatrix} X & Y \end{bmatrix} = Z$$

The diagram illustrates horizontal concatenation. On the left, two matrices are shown side-by-side: matrix X, which is a 3x3 grid of blue squares, and matrix Y, which is a 3x3 grid of yellow squares. A large bracket above them indicates they are being concatenated horizontally. An equals sign to the right of the bracket points to matrix Z, which is a larger 3x6 grid. The first three columns of Z are filled with the blue squares from matrix X, and the next three columns are filled with the yellow squares from matrix Y.

Separate elements by
using a comma (,) or
space ().

Vertical Concatenation

$$\begin{bmatrix} X \\ Y \end{bmatrix} = Z$$

The diagram illustrates vertical concatenation. On the left, two matrices are stacked vertically: matrix X, which is a 3x3 grid of blue squares, and matrix Y, which is a 3x3 grid of yellow squares. A semicolon between them indicates they are being concatenated vertically. An equals sign to the right of the bracket points to matrix Z, which is a 6x3 grid. The top three rows of Z are filled with the blue squares from matrix X, and the bottom three rows are filled with the yellow squares from matrix Y.

Separate elements by
using a semicolon (;).

Combined Concatenation

$$\begin{bmatrix} X & Y & ; & Z \end{bmatrix} = W$$

The diagram illustrates combined concatenation. On the left, three matrices are shown: matrix X (3x3 blue), matrix Y (3x3 yellow), and matrix Z (3x1 green). They are concatenated into matrix W, which is a 3x7 grid. Matrix X occupies the first three columns, matrix Y occupies the next three columns, and matrix Z occupies the final column. The semicolon between Y and Z indicates they are concatenated vertically, while the commas within X and Y indicate they are concatenated horizontally.

Create each row by separating elements with a
comma (,) or space (), and then separate the
rows with a semicolon (;).

Array Creation Functions

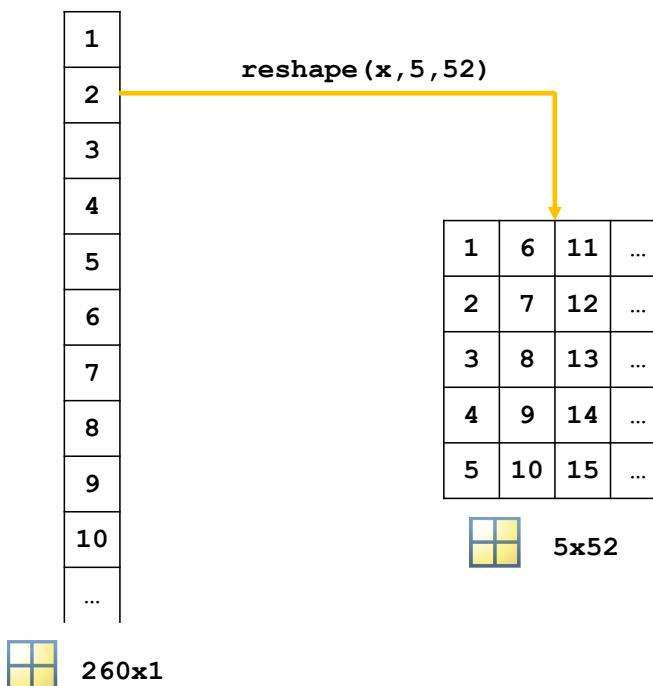
You can use several functions to create arrays.

Most of these functions support the calling syntaxes shown below.

Calling Syntax	Output
<code>fun(m,n)</code>	m-by-n
<code>fun(n)</code>	n-by-n

Reshape Arrays

You can reshape a column into a matrix.



Create a vector of random numbers to reshape.

Specify the dimensions for the new array.

For convenience, you can also leave one of the dimensions blank when you call `reshape` and the function calculates that dimension automatically.

You can create a vector from a matrix by using the colon operator.

```
x = rand(260,1);
y = reshape(x,5,52);
y = reshape(x,5,[]);
y = x(:);
```

Indexing

1	2.3
2	1.5
3	1.3
4	0.9
5	1.3

v

Extract one element from a vector.

`v(2)`

1.5

`v(end)`

1.3

`v([1 end-2:end])`

2.3

1.3

0.9

1.3

Extract multiple elements from a vector.

To extract elements from a matrix, provide two indices, the row and column numbers.

	1	2	3	4
1	1.5	1.1	2.6	0.9
2	1.5	2.4	1.7	1.4
3	2.5	1.6	1.9	0.7
4	2.4	1.1	1.8	2.5
5	1.9	2.8	0.6	0.6

M

Extract one element from a matrix.

`M(2,3)`

1.7

Extract an entire column. Here, it is the last one.

`M(:,end)`

0.9

1.4

0.7

2.5

0.6

Extract multiple elements from a matrix.

`M([1 end],2)`

1.1

2.8

Change Elements in Arrays

Change one element in a vector.

`v(2) = 0`

2.3

0

1.3

0.9

1.3

Change multiple elements in a vector to the same value.

`v(1:3) = 0`

0

0

0

0.9

1.3

Change multiple elements in a vector to different values.

`v(1:3) = [3 5 7]`

3

5

7

0.9

1.3

Remove elements from a vector.

```
v(1:3) = []
```

0.9

1.3

```
v(5) = 42
```

0.9

1.3

0

0

42

Assign a nonexistent value. The vector is padded with zeros.

Changing elements in matrices works the same way as with vectors, but you must specify both rows and columns.
