

Creating Matrixes in Matlab

back to [Fan's Intro Math for Econ](#), [Matlab Examples](#), or [Dynamic Asset Repositories](#)

Matlab Define Row and Column Vectors (Matrix)

```
% A column vector 4 by 1, with three numbers you fill in by yourself
colVec = [5;2;3;10]
```

```
colVec = 4x1
     5
     2
     3
    10
```

```
% Another column vector with 4 random numbers
colVecRand = rand(4,1)
```

```
colVecRand = 4x1
    0.4899
    0.1679
    0.9787
    0.7127
```

```
% A row vector 1 by 4
rowVec = [3,2,4,5]
```

```
rowVec = 1x4
     3     2     4     5
```

```
% A row vector 1 by 4 with random number
rowVecRand = rand(1,4)
```

```
rowVecRand = 1x4
    0.5005    0.4711    0.0596    0.6820
```

Matlab Define a Matrix

```
% A 2 by 3 matrix by hand
matA = [1,2,1;
        3,4,10]
```

```
matA = 2x3
     1     2     1
     3     4    10
```

```
% Another 2 by 3 matrix, now with random numbers
matRand = rand(2,3)
```

```
matRand = 2x3
    0.0424    0.5216    0.8181
    0.0714    0.0967    0.8175
```

```
% Another 2 by 3 matrix, now with random integers between 1 and 10
% rand draws between 0 and 1, ceil converts 0.1 to 1, 1.1 to 2, etc
matRand = ceil(rand(2,3)*10)
```

```
matRand = 2x3
```

8	7	10
2	6	7

Matlab Define a Square Matrix

```
% A 4 by 4 square matrix
matSquare = rand(4)
```

```
matSquare = 4x4
    0.8003    0.0835    0.8314    0.5269
    0.4538    0.1332    0.8034    0.4168
    0.4324    0.1734    0.0605    0.6569
    0.8253    0.3909    0.3993    0.6280
```

```
% or can define 4 by 4
matSquare = rand(4, 4)
```

```
matSquare = 4x4
    0.2920    0.1672    0.4897    0.0527
    0.4317    0.1062    0.3395    0.7379
    0.0155    0.3724    0.9516    0.2691
    0.9841    0.1981    0.9203    0.4228
```

```
% or can define 4 by 4, between 1 and 5 each number
matSquare = ceil(rand(4, 4)*5)
```

```
matSquare = 4x4
     3     2     4     5
     5     4     4     1
     3     4     1     1
     5     3     1     3
```

Identity Matrix

If a matrix A is square matrix with the same number of rows and columns, and all diagonal elements are 1 and non-diagonal elements are 0, then A is an identity matrix:

- $A_{i,j}$ are the value in the i th row and j th column of the matrix A
- A is an identity matrix, when: $A_{i,j} = 0$ if $i \neq j$, $A_{i,j} = 1$ if $i = j$

```
% 4 by 4 identity matrix
identity4by4 = eye(4)
```

```
identity4by4 = 4x4
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
```

When a matrix is multiplied by the identity matrix, you get the same matrix back, for example, multiplying random integer 4 by 4 matrix by the 4 by 4 identity matrix:

```
matSquare
```

```
matSquare = 4x4
     3     2     4     5
```

```

5    4    4    1
3    4    1    1
5    3    1    3

```

```
matSquareTimesIdentity = matSquare*identity4by4
```

```

matSquareTimesIdentity = 4x4
3    2    4    5
5    4    4    1
3    4    1    1
5    3    1    3

```

When a row vector is multiplied by the identity matrix, you get the same vector back, for example, multiplying random integer 1 by 4 row vector by the 4 by 4 identity matrix:

```
rowVec
```

```

rowVec = 1x4
3    2    4    5

```

```
rowVecTimesIdentity = rowVec*identity4by4
```

```

rowVecTimesIdentity = 1x4
3    2    4    5

```

When an identity matrix is multiplied by a column vector, you get the same vector back, for example, multiplying 4 by 4 identity matrix by random integer 4 by 1 column vector by the :

```
colVec
```

```

colVec = 4x1
5
2
3
10

```

```
colVecTimesIdentity = identity4by4*colVec
```

```

colVecTimesIdentity = 4x1
5
2
3
10

```

Lower-Triangular Matrix and Upper-Triangular Matrix

A lower triangular matrix is a square matrix where:

- Square matrix A is a **lower triangular** matrix, when: $A_{i,j} = 0$ if $i < j$
- Square matrix A is a **upper triangular** matrix, when: $A_{i,j} = 0$ if $i > j$

```

% lower triangular matrix of matA
lowerTriangular = tril(matSquare)

```

```

lowerTriangular = 4x4
3    0    0    0
5    4    0    0

```

3	4	1	0
5	3	1	3

```
% upper triangular matrix of matA
upperTriangular = triu(matSquare)
```

```
upperTriangular = 4x4
```

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

Three Dimensions Matrix (Tensor)

```
% 3 by 3 by 2, storing multiple matrixes together in tenA
```

```
tenA = zeros(3,3,2);
tenA(:,:,1) = rand(3,3);
tenA(:,:,2) = rand(3,3);
disp(tenA);
```

```
(:,:,1) =
```

0.8819	0.3689	0.1564
0.6692	0.4607	0.8555
0.1904	0.9816	0.6448

```
(:,:,2) =
```

0.3763	0.4820	0.2262
0.1909	0.1206	0.3846
0.4283	0.5895	0.5830

```
% Creating four 2 by 3 matrixes
```

```
matRand = rand(2,3,4)
```

```
matRand =
```

```
matRand(:,:,1) =
```

0.2518	0.6171	0.8244
0.2904	0.2653	0.9827

```
matRand(:,:,2) =
```

0.7302	0.5841	0.9063
0.3439	0.1078	0.8797

```
matRand(:,:,3) =
```

0.8178	0.5944	0.4253
0.2607	0.0225	0.3127

```
matRand(:,:,4) =
```

0.1615	0.4229	0.5985
0.1788	0.0942	0.4709

```
disp(matRand);
```

```
(:,:,1) =
```

0.2518	0.6171	0.8244
0.2904	0.2653	0.9827

```
(:,:,2) =
```

0.7302	0.5841	0.9063
0.3439	0.1078	0.8797

```
(:,:,3) =
```

0.8178	0.5944	0.4253
0.2607	0.0225	0.3127

```
(:,:,4) =
```

0.1615	0.4229	0.5985
0.1788	0.0942	0.4709