# Matrix Addition and Multiplication

**Back to Fan's Intro Math for Economist Table of Content**

## Scalar Multiplication/Division, Addition/Subtraction

If we multiply a matrix by a number, we multiply every element of that matrix by that number. Addition, subtraction, and division of a matrix with a sclar value work the same way

```
c = 10
```

```
c = 10
```

```
mat_a = rand(3,2)
```

```
mat_a = 3×2
    0.8147    0.9134
    0.9058    0.6324
    0.1270    0.0975
```

```
c*mat_a
```

```
ans = 3×2
    8.1472    9.1338
    9.0579    6.3236
    1.2699    0.9754
```

```
mat_a/c
```

```
ans = 3×2
    0.0815    0.0913
    0.0906    0.0632
    0.0127    0.0098
```

```
mat_a - c
```

```
ans = 3×2
   -9.1853   -9.0866
   -9.0942   -9.3676
   -9.8730   -9.9025
```

```
mat_a + c
```

```
ans = 3×2
   10.8147   10.9134
   10.9058   10.6324
   10.1270   10.0975
```

## Addition and Subtraction

You can add/subtract together two matrixes of the same size. We can add up the two 3 by 1 vectors from above, and the two 2 by 3 matrixes from above.

```
col_vec_a = rand(3,1)
```

```
col_vec_a = 3×1
    0.2785
    0.5469
```

```
    0.9575
```

```
col_vec_b = rand(3,1)
```

```
col_vec_b = 3×1
    0.9649
    0.1576
    0.9706
```

```
mat_a = rand(3,2)
```

```
mat_a = 3×2
    0.9572    0.1419
    0.4854    0.4218
    0.8003    0.9157
```

```
mat_b = rand(3,2)
```

```
mat_b = 3×2
    0.7922    0.0357
    0.9595    0.8491
    0.6557    0.9340
```

```
col_vec_a + col_vec_b
```

```
ans = 3×1
    1.2434
    0.7045
    1.9281
```

```
mat_a - mat_b
```

```
ans = 3×2
    0.1650    0.1062
   -0.4741   -0.4274
    0.1445   -0.0183
```

When using matlab, even if you add up to a single column or single row with a matrix that has multiple rows and columns, if the column count or row count matches up, matlab will ***broadcast*** rules, and addition will still be legal. In the example below, mat_a is 3 by 2, and col_vec_a is 3 by 1, matlab duplicate col_vec_a and add it to each column of mat_a (*Broadcast rules are important for efficient storage and computation*):

```
mat_a + col_vec_a
```

```
ans = 3×2
    1.2357    0.4204
    1.0323    0.9686
    1.7578    1.8732
```

## Matrix Multiplication

When we try to multiply two matrixes together: $A \cdot B$ for example, the ***number of columns*** of matrix *A* and the ***number of rows*** of matrix *B* have to match up.

If the matrix *A* is has *L* rows and *M* columns, and the matrix *B* has *M* rows and *N*columns, then the resulting matrix of $C = A \cdot B$ has to have *L* rows and *N* columns.

Each of the $(l, n)$ cell in the product matrix $C = A \cdot B$, is equal to:

- $$C_{l,n} = \sum_{m=1}^{M} A_{l,m} \cdot B_{m,n}$$

Note that we are summing over *M*: row *l* in matrix *A*, and column *n* in matrix *B* both have *M* elements. We multiply each *m* of the *M* element from the row in *A* and column in *B* together one by one, and then sum them up to end up with the value for the *l*th row and *n*th column in matrix *C*.

```
% (3 by 4) times (4 by 2) end up with (3 by 2)
L = 3;
M = 4;
N = 2;
mat_A = rand(L, M)
```

```
mat_A = 3×4
    0.6787    0.3922    0.7060    0.0462
    0.7577    0.6555    0.0318    0.0971
    0.7431    0.1712    0.2769    0.8235
```

```
mat_B = rand(M, N)
```

```
mat_B = 4×2
    0.6948    0.4387
    0.3171    0.3816
    0.9502    0.7655
    0.0344    0.7952
```

```
mat_C = mat_A*mat_B
```

```
mat_C = 3×2
    1.2685    1.0247
    0.7679    0.6842
    0.8621    1.2582
```

```

% (2 by 10) times (10 by 1) end up with (2 by 1)
L = 2;
M = 10;
N = 1;
mat_A = rand(L, M)
```

```
mat_A = 2×10
    0.1869    0.4456    0.7094    0.2760    0.6551    0.1190    0.9597    0.5853 ···
    0.4898    0.6463    0.7547    0.6797    0.1626    0.4984    0.3404    0.2238
```

```
mat_B = rand(M, N)
```

```
mat_B = 10×1
    0.8909
    0.9593
    0.5472
    0.1386
    0.1493
    0.2575
    0.8407
    0.2543
```

```
    0.8143
    0.2435
```

```
mat_C = mat_A*mat_B
```

```
mat_C = 2×1
    2.8395
    2.4372
```