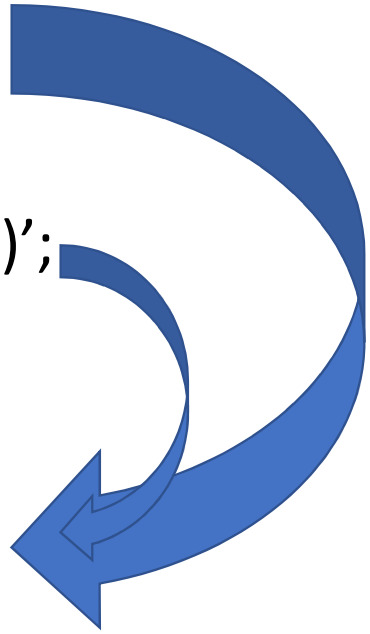
Elementary Functions
Statistics (sum)

```
>>X=0:10:100;
>>sum(X)

>>X=(0:10:100)';
>>sum(X)

ans=
 550
```



```
>>M=rand(3,5)
    0.14    0.79    0.04    0.68    0.39
    0.42    0.96    0.85    0.76    0.66
    0.92    0.66    0.93    0.74    0.17
>>sum(M)
    1.48    2.41    1.82    2.18    1.22
>>sum(M,1)

>>sum(M,2)
    2.04
    3.64
    3.42
```

Elementary Functions Statistics

- sum (M)
- sum (M,1)
- sum(M,2)

- mean(X)
- mean (X,1)
- mean(X,2)

- prod (X)
- prod (X,1)
- prod(X,2)

M=rand(3,5)

0.14	0.79	0.04	0.68	0.39
0.42	0.96	0.85	0.76	0.66
0.92	0.66	0.93	0.74	0.17

- sum(sum(M))
- all(prod(prod(M))==prod(M(:)))
- mean(mean(M))==mean(M(:))

- ans =1 (logical)

Elementary Functions Statistics

X =randi(10,1,5)%X=rand(5,1)

X = 8.00 1.00 3.00 1.00 1.00

M =randi(10,5,5)

M = 9.00 5.00 5.00 3.00 5.00
7.00 4.00 5.00 7.00 10.00
4.00 8.00 7.00 7.00 4.00
10.00 8.00 8.00 2.00 6.00
1.00 2.00 8.00 2.00 3.00

Same for max

min (X)	**ans= 1**
min(M) %by column	ans = 1.00 2.00 5.00 2.00 3.00
~~min(M,1)~~ ~~min(M,2)~~	
[val,pos]=min(M) val = 1.00 2.00 5.00 2.00 3.00 pos = 5.00 5.00 1.00 4.00 5.00 Pos = the row index that corresponds to the minimum value of M on each column. (according to the operating dimension)	

Elementary Functions
Statistics

- variance

V = var(A)

returns the variance of the elements of A :

If A is a vector of observations, the variance is a scalar.

If A is a matrix, V is a row vector containing the variances corresponding to each column.

<u>Mathematically:</u>

$$V = \frac{1}{N-1}\sum_{i=1}^{N}|A_i - \mu|^2 \qquad \mu = \frac{1}{N}\sum_{i=1}^{N}A_i.$$

Elementary Functions
Statistics

- Standard deviation

S = std(A)

returns the standard deviation of the elements of A.

If A is a vector of observations, then the standard deviation is a scalar.

If A is a matrix then S is a row vector containing the standard deviations corresponding to each column.

Mathematically:

$$S = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}|A_i - \mu|^2}, \qquad \mu = \frac{1}{N}\sum_{i=1}^{N}A_i.$$

Elementary Functions
Statistics

- **R= sort(X)**
- For X a vector, it sorts the elements of X in ascending order.
- for matrices, it sorts each column of X in ascending order.

Example
m=[28 3 5 -10 0 6 4 3];

```
>> sort(m)
ans =
-10 0 3 3 5 6 28
```

```
>> sort(m,'descend')
ans =
28 6 5 3 3 0 -10
```

```
>>[n ind]=sort(m)
n =
-10 0 3 3 5 6 28
ind =
4 5 2 7 3 6 1
```

```
A = [3 6 5; 7 -2 4; 1 0 -9]
A=
   3    6    5
   7   -2    4
   1    0   -9
```

```
>>sort(A)
  ans =
       1.00   -2.00   -9.00
       3.00       0    4.00
       7.00    6.00    5.00
```

```
[n,ind]=sort(A)
n =
       1.00   -2.00   -9.00
       3.00       0    4.00
       7.00    6.00    5.00

ind =

       3.00    2.00    3.00
       1.00    3.00    2.00
       2.00    1.00    1.00
```

Elementary Functions Algebria/shape

- reshape (M,row?,column?)

  Example

  A=1:10

  reshape(A,5,2)

  reshape(A,[5,2])

ans =

1.00	6.00
2.00	7.00
3.00	8.00
4.00	9.00
5.00	10.00

- Repmat (M,rows, columns)

  Example

  M=1:4

  A=M’

  repmat(M,3,2)

  ans =

1.00	2.00	3.00	4.00	1.00	2.00	3.00	4.00
1.00	2.00	3.00	4.00	1.00	2.00	3.00	4.00
1.00	2.00	3.00	4.00	1.00	2.00	3.00	4.00

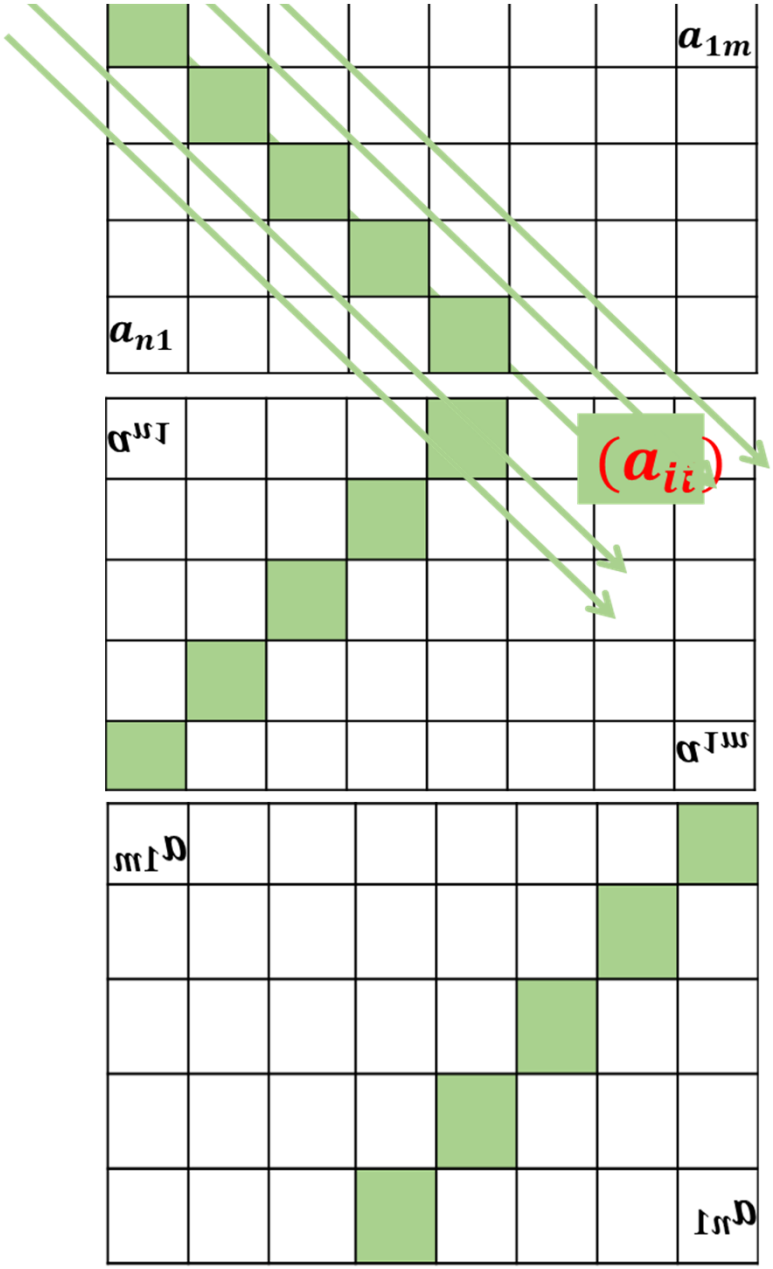
Repmat(A,3,2)

ans =

1.00	1.00
2.00	2.00
3.00	3.00
4.00	4.00
1.00	1.00
2.00	2.00
3.00	3.00
4.00	4.00
1.00	1.00
2.00	2.00
3.00	3.00
4.00	4.00

Elementary Functions Algebria/shape

- diag(A)
- diag(A,k) %k>0 ou k<0
- Flipud
- fliplr



Elementary Functions Algebria/shape

- sub2ind (size, row, column)
- ind2sub (size, ind)

Example

A=rand(3,2)

```
A =   0.75   0.70
      0.26   0.89
      0.51   0.96
```

I=Find(A>0.5)

```
I =   1.00
      3.00
      4.00
      5.00
      6.00
```

[i,j]=ind2sub(size(A),I)

```
i =
      1.00
      3.00
      1.00
      2.00
      3.00
j =
      1.00
      1.00
      2.00
      2.00
      2.00
```

sub2ind(size(A),i,j)

```
ans =
      1.00
      3.00
      4.00
      5.00
      6.00
```

[i,j]=find(A>0.5)

```
i =
      1.00
      3.00
      1.00
      2.00
      3.00
j =
      1.00
      1.00
      2.00
      2.00
      2.00
```

Format command

- MATLAB always attempts to display integers (whole numbers) exactly. However, if the integer is too large, it is displayed in scientific notation with five significant digits,

Example:

first enter 123456789 at the command prompt, and then 1234567890.

```
>> 123456789

ans =

   123456789

>> 1234567890

ans =

   1.2346e+09
```

Format command

- Numbers with decimal parts are displayed with four significant digits. This is what is called the default format, i.e. what normally happens.
- However, you can change from the default with variations on the format command, as follows.
- Also, All output from subsequent display statements will be in

examples

>> format short e % scientific notation (floating point form)

>> format long e %for more accurate output.A scientific notation, but with 15 significant digits

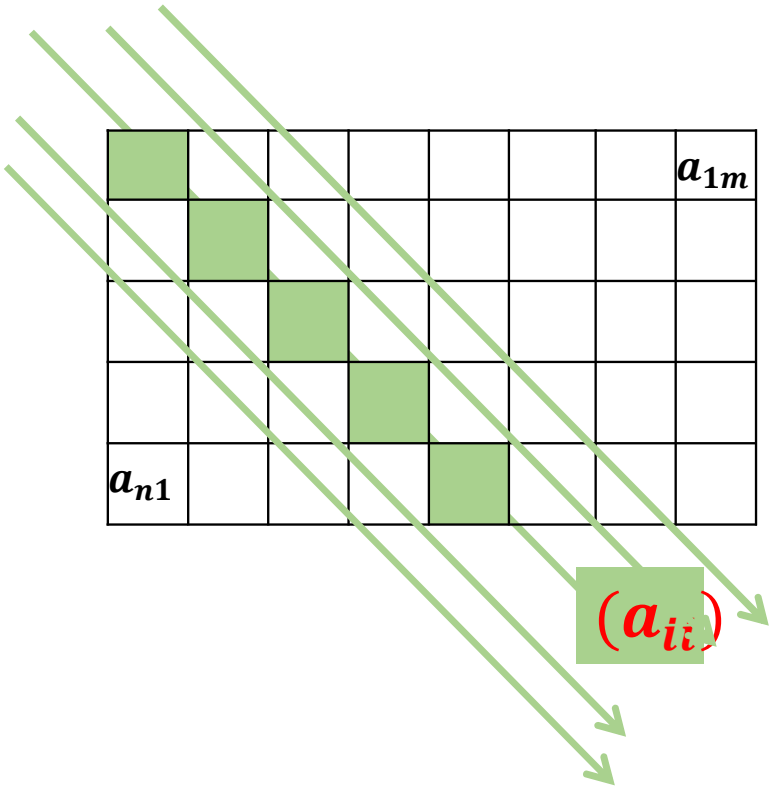
Format command

```
>> format short e
>> 1/7
ans =
  1.4286e-01

>> format long e
>> 1/7
ans =
   1.428571428571428e-01

>> format short
>> 1/7
ans =
```

- Gardé en cas ou



Format command for me

Numbers with decimal parts are displayed with four significant digits. If the value x is in the range $0{:}001 < x \leq 1000$ it is displayed in fixed point form, otherwise scientific (floating point) notation is used, in which case the mantissa is between 1and 9.9999, e.g. 1000.1 is displayed as 1.0001e+003. Check this by entering following numbers at the prompt (on separate lines): 0.0011, 0.0009, 1/3, 5/3, 2999/3, 3001/3

Elementary Functions
Algebria/shape

- [row,col] = ind2sub(sz,ind) returns the arrays row and col containing the equivalent row and column subscripts corresponding to the linear indices ind for a matrix of size sz. Here sz is a vector with two elements, where sz(1) specifies the number of rows and sz(2) specifies the number of columns.