

Elementary Functions

Rounding and remainder functions

- `fix` , rounds each element of X to the nearest integer toward zero. This operation effectively truncates the numbers in X to integers by removing the decimal portion of each number

```
>>fix(3.5) → ans=3
```

- `round` , rounds each element of X to the nearest integer.

```
>>round(3.5) → ans=4
```

Elementary Functions

Rounding and remainder functions

- `floor` - Round towards minus infinity.
- `ceil` - Round towards plus infinity.

```
>> ceil(3.1)
ans =
    4
>> floor(3.9)
ans =
    3
```

Elementary Functions

Rounding and remainder functions

- **$r = \text{rem}(a,b)$** returns the remainder after division of a by b , where a is the dividend and b is the divisor.

This function is often called the remainder operation, which can be expressed as $r = a - b \cdot \text{fix}(a./b)$.

$$r = \text{rem}(27,5) \rightarrow r=2$$

- **$w = \text{mod}(a,m)$** returns the remainder after division of a by m , where a is the dividend and m is the divisor.

This function is often called the modulo operation, which can be expressed as $w = a - m \cdot \text{floor}(a./m)$.

$$w = \text{mod}(27,-5) \rightarrow w = -3$$

$$W = \text{mod}(-27,5) \rightarrow W = 3$$

Elementary Functions

Discrete math functions

- Discrete math functions perform operations on integers ($\dots, -2, -1, 0, 1, 2, \dots$), or return discrete output as integers.
- You can use these functions to factor large numbers, calculate factorials, find permutations and combinations, or find greatest common denominators.

Elementary Functions

Discrete math functions

<code>factor</code>	Prime factors
<code>factorial</code>	Factorial of input
<code>gcd</code>	Greatest common divisor
<code>isprime</code>	Determine which array elements are prime
<code>lcm</code>	Least common multiple
<code>nchoosek</code>	Binomial coefficient or all combinations
<code>perms</code>	All possible permutations
<code>matchpairs</code>	Solve linear assignment problem
<code>primes</code>	Prime numbers less than or equal to input value
<code>rat</code>	Rational fraction approximation
<code>rats</code>	Rational output

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

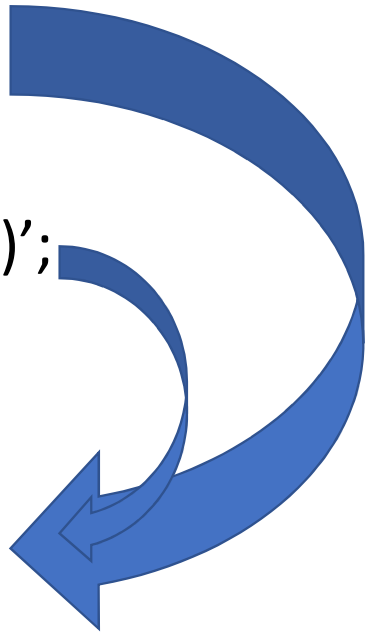
```
>>sum(M)
```

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

M=rand(3,5)

- | | | | | | | |
|--------------|--------------|------|------|------|------|------|
| • sum (M) | | 0.14 | 0.79 | 0.04 | 0.68 | 0.39 |
| • sum (M,1) | | 0.42 | 0.96 | 0.85 | 0.76 | 0.66 |
| • sum(M,2) | • mean(X) | 0.92 | 0.66 | 0.93 | 0.74 | 0.17 |
| | • mean (X,1) | | | | | |
| | • mean(X,2) | | | | | |
| • prod (X) | | | | | | |
| • prod (X,1) | | | | | | |
| • prod(X,2) | | | | | | |
- 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

<code>min (X)</code>	<code>ans= 1</code>				
<code>min(M)</code> <code>%by column</code>	<code>ans =</code> 1.00	2.00	5.00	2.00	3.00
<code>min(M,1)</code>	<code>min(M,2)</code>				
<code>[val,pos]=min(M)</code> <code>val =</code>	1.00	2.00	5.00	2.00	3.00
<code>pos =</code>	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

$V = \text{var}(A)$

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.

Mathematically:

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

Elementary Functions

Statistics

- Standard deviation

$S = \text{std}(A)$

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:

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

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
```

```
>>[n ind]=sort(m)
```

```
n =  
-10 0 3 3 5 6 28
```

```
ind =  
4 5 2 7 3 6 1
```

```
>> sort(m,'descend')
```

```
ans =  
28 6 5 3 3 0 -10
```

```
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

Algebra/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)

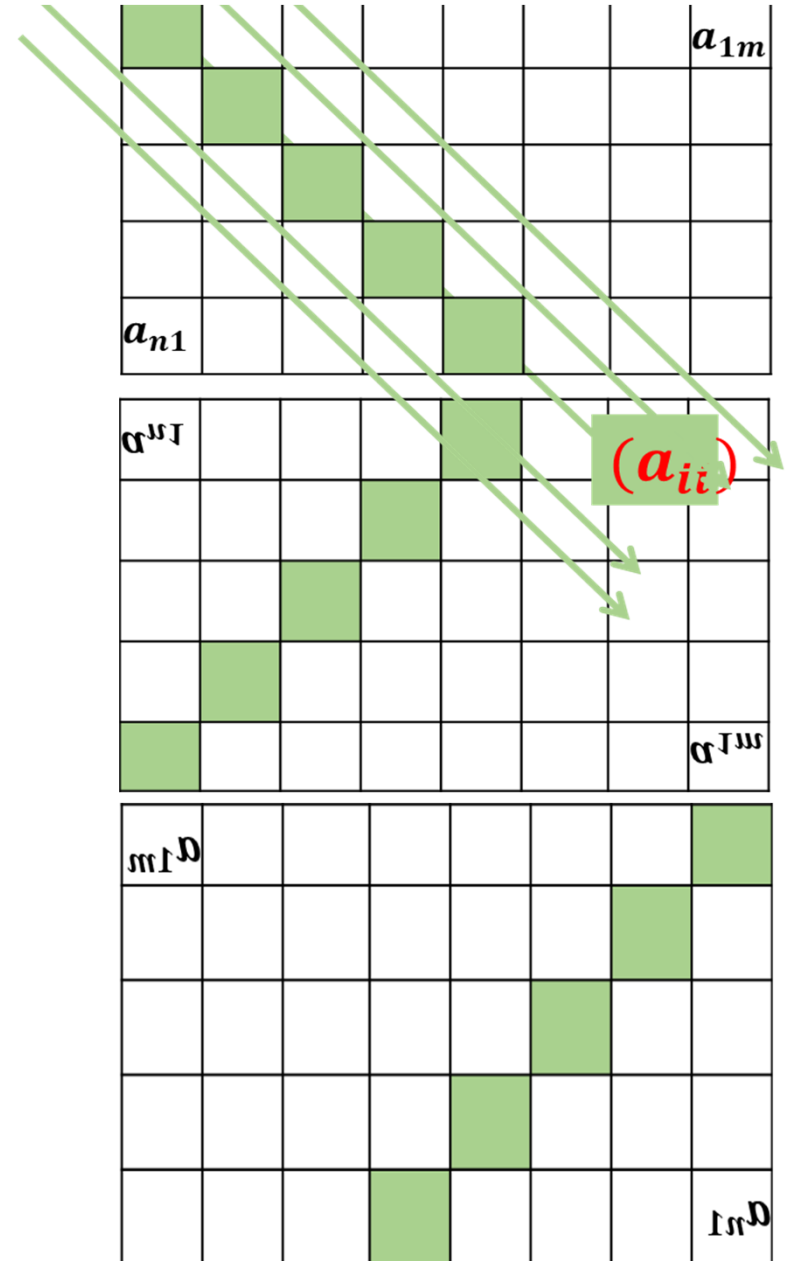
ans =

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

Algebra/shape

- `diag(A)`
- `diag(A,k)` % $k > 0$ ou $k < 0$
- `flipud`
- `fliplr`



Elementary Functions

Algebra/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
```

l=Find(A>0.5)

```
l = 1.00
     3.00
     4.00
     5.00
     6.00
```

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

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 =
```