

NumPy for MATLAB users

Help

MATLAB/Octave	Python	Description
doc	help()	Browse help interactively
<code>help -i % browse with Info</code>		
help help <i>or</i> doc doc	help	Help on using help
help plot	help(plot) <i>or</i> ?plot	Help for a function
help splines <i>or</i> doc splines	help(pylab)	Help for a toolbox/library package
demo		Demonstration examples

Searching available documentation

MATLAB/Octave	Python	Description
lookfor plot		Search help files
help	help(); modules [Numeric]	List available packages
which plot	help(plot)	Locate functions

Using interactively

MATLAB/Octave	Python	Description
<code>octave -q</code>	ipython -pylab	Start session
<code>TAB <i>or</i> M-?</code>	TAB	Auto completion
foo(.m)	execfile('foo.py') <i>or</i> run foo.py	Run code from file
history	hist -n	Command history
diary on [...] diary off		Save command history
exit <i>or</i> quit	CTRL-D	End session
	CTRL-Z # windows	
	sys.exit()	

Operators

MATLAB/Octave	Python	Description
help -		Help on operator syntax

Arithmetic operators

MATLAB/Octave	Python	Description
a=1; b=2;	a=1; b=1	Assignment; defining a number
a + b	a + b <i>or</i> add(a,b)	Addition
a - b	a - b <i>or</i> subtract(a,b)	Subtraction
a * b	a * b <i>or</i> multiply(a,b)	Multiplication
a / b	a / b <i>or</i> divide(a,b)	Division

<code>a .^ b</code>	<code>a ** b</code> <code>power(a,b)</code> <code>pow(a,b)</code>	Power, a^b
<code>rem(a,b)</code>	<code>a % b</code> <code>remainder(a,b)</code> <code>fmod(a,b)</code>	Remainder
<code>a+=1</code>	<code>a+=b</code> <i>OR</i> <code>add(a,b,a)</code>	In place operation to save array creation overhead
<code>factorial(a)</code>		Factorial, $n!$

Relational operators

MATLAB/Octave	Python	Description
<code>a == b</code>	<code>a == b</code> <i>OR</i> <code>equal(a,b)</code>	Equal
<code>a < b</code>	<code>a < b</code> <i>OR</i> <code>less(a,b)</code>	Less than
<code>a > b</code>	<code>a > b</code> <i>OR</i> <code>greater(a,b)</code>	Greater than
<code>a <= b</code>	<code>a <= b</code> <i>OR</i> <code>less_equal(a,b)</code>	Less than or equal
<code>a >= b</code>	<code>a >= b</code> <i>OR</i> <code>greater_equal(a,b)</code>	Greater than or equal
<code>a ~= b</code>	<code>a != b</code> <i>OR</i> <code>not_equal(a,b)</code>	Not Equal

Logical operators

MATLAB/Octave	Python	Description
<code>a && b</code>	<code>a and b</code>	Short-circuit logical AND
<code>a b</code>	<code>a or b</code>	Short-circuit logical OR
<code>a & b</code> <i>OR</i> <code>and(a,b)</code>	<code>logical_and(a,b)</code> <i>OR</i> <code>a and b</code>	Element-wise logical AND
<code>a b</code> <i>OR</i> <code>or(a,b)</code>	<code>logical_or(a,b)</code> <i>OR</i> <code>a or b</code>	Element-wise logical OR
<code>xor(a, b)</code>	<code>logical_xor(a,b)</code>	Logical EXCLUSIVE OR
<code>~a</code> <i>OR</i> <code>not(a)</code>	<code>logical_not(a)</code> <i>OR</i> <code>not a</code>	Logical NOT
<code>~a</code> <i>OR</i> <code>!a</code>		
<code>any(a)</code>		True if any element is nonzero
<code>all(a)</code>		True if all elements are nonzero

root and logarithm

MATLAB/Octave	Python	Description
<code>sqrt(a)</code>	<code>math.sqrt(a)</code>	Square root
<code>log(a)</code>	<code>math.log(a)</code>	Logarithm, base e (natural)
<code>log10(a)</code>	<code>math.log10(a)</code>	Logarithm, base 10
<code>log2(a)</code>	<code>math.log(a, 2)</code>	Logarithm, base 2 (binary)
<code>exp(a)</code>	<code>math.exp(a)</code>	Exponential function

Round off

MATLAB/Octave	Python	Description
<code>round(a)</code>	<code>round(a)</code> <i>OR</i> <code>math.round(a)</code>	Round
<code>ceil(a)</code>	<code>ceil(a)</code>	Round up

`floor(a)`
`fix(a)`

`floor(a)`
`fix(a)`

Round down
Round towards zero

Mathematical constants

MATLAB/Octave

`pi`
`exp(1)`

Python

`math.pi`
`math.e` *or* `math.exp(1)`

Description

$\pi=3.141592$
 $e=2.718281$

Missing values; IEEE-754 floating point status flags

MATLAB/Octave

`NaN`
`Inf`

Python

`nan`
`inf`
`plus_inf`
`minus_inf`
`plus_zero`
`minus_zero`

Description

Not a Number
Infinity, $+\infty$
Infinity, $+\infty$
Infinity, $-\infty$
Plus zero, $+0$
Minus zero, -0

Complex numbers

MATLAB/Octave

`i`
`z = 3+4i`
`abs(z)`
`real(z)`
`imag(z)`
`arg(z)`
`conj(z)`

Python

`z = 1j`
`z = 3+4j` *or* `z = complex(3,4)`
`abs(3+4j)`
`z.real`
`z.imag`

`z.conj()`; `z.conjugate()`

Description

Imaginary unit
A complex number, $3+4i$
Absolute value (modulus)
Real part
Imaginary part
Argument
Complex conjugate

Trigonometry

MATLAB/Octave

`atan(a,b)`

Python

`atan2(b,a)`
`hypot(x,y)`

Description

Arctangent, $\arctan(b/a)$
Hypotenuse; Euclidean distance

Generate random numbers

MATLAB/Octave

`rand(1,10)`

`2+5*rand(1,10)`

`rand(6)`
`randn(1,10)`

Python

`random.random((10,))`
`random.uniform((10,))`

`random.uniform(2,7,(10,))`

`random.uniform(0,1,(6,6))`
`random.standard_normal((10,))`

Description

Uniform distribution

Uniform: Numbers between 2 and 7
Uniform: 6,6 array
Normal distribution

Vectors

MATLAB/Octave	Python	Description
<code>a=[2 3 4 5];</code>	<code>a=array([2,3,4,5])</code>	Row vector, $1 \times n$ -matrix
<code>adash=[2 3 4 5]';</code>	<code>array([2,3,4,5])[:,NewAxis]</code> <code>array([2,3,4,5]).reshape(-1,1)</code> <code>r_[1:10,'c']</code>	Column vector, $m \times 1$ -matrix

Sequences

MATLAB/Octave	Python	Description
<code>1:10</code>	<code>arange(1,11, dtype=Float)</code> <code>range(1,11)</code>	1,2,3, ... ,10
<code>0:9</code>	<code>arange(10.)</code>	0.0,1.0,2.0, ... ,9.0
<code>1:3:10</code>	<code>arange(1,11,3)</code>	1,4,7,10
<code>10:-1:1</code>	<code>arange(10,0,-1)</code>	10,9,8, ... ,1
<code>10:-3:1</code>	<code>arange(10,0,-3)</code>	10,7,4,1
<code>linspace(1,10,7)</code>	<code>linspace(1,10,7)</code>	Linearly spaced vector of n=7 points
<code>reverse(a)</code>	<code>a[::-1]</code> <i>or</i>	Reverse
<code>a(:) = 3</code>	<code>a.fill(3), a[:] = 3</code>	Set all values to same scalar value

Concatenation (vectors)

MATLAB/Octave	Python	Description
<code>[a a]</code>	<code>concatenate((a,a))</code>	Concatenate two vectors
<code>[1:4 a]</code>	<code>concatenate((range(1,5),a), axis=1)</code>	

Repeating

MATLAB/Octave	Python	Description
<code>[a a]</code>	<code>concatenate((a,a))</code>	1 2 3, 1 2 3
	<code>a.repeat(3)</code> <i>or</i>	1 1 1, 2 2 2, 3 3 3
	<code>a.repeat(a)</code> <i>or</i>	1, 2 2, 3 3 3

Miss those elements out

MATLAB/Octave	Python	Description
<code>a(2:end)</code>	<code>a[1:]</code>	miss the first element
<code>a([1:9])</code>		miss the tenth element
<code>a(end)</code>	<code>a[-1]</code>	last element
<code>a(end-1:end)</code>	<code>a[-2:]</code>	last two elements

Maximum and minimum

MATLAB/Octave	Python	Description
<code>max(a,b)</code>	<code>maximum(a,b)</code>	pairwise max
<code>max([a b])</code>	<code>concatenate((a,b)).max()</code>	max of all values in two vectors
<code>[v,i] = max(a)</code>	<code>v,i = a.max(0),a.argmax(0)</code>	

Vector multiplication

MATLAB/Octave	Python	Description
<code>a.*a</code>	<code>a*a</code>	Multiply two vectors
<code>dot(u,v)</code>	<code>dot(u,v)</code>	Vector dot product, $u \cdot v$

Matrices

MATLAB/Octave	Python	Description
<code>a = [2 3;4 5]</code>	<code>a = array([[2,3],[4,5]])</code>	Define a matrix

Concatenation (matrices); rbind and cbind

MATLAB/Octave	Python	Description
<code>[a ; b]</code>	<code>concatenate((a,b), axis=0)</code> <code>vstack((a,b))</code>	Bind rows
<code>[a , b]</code>	<code>concatenate((a,b), axis=1)</code> <code>hstack((a,b))</code>	Bind columns
<code>[a(:), b(:)]</code>	<code>concatenate((a,b), axis=2)</code> <code>dstack((a,b))</code>	Bind slices (three-way arrays)
<code>[1:4 ; 1:4]</code>	<code>concatenate((r_[1:5],r_[1:5])).reshape(2,-1)</code> <code>vstack((r_[1:5],r_[1:5]))</code>	Concatenate matrices into one vector
<code>[1:4 ; 1:4]'</code>		Bind rows (from vectors)
		Bind columns (from vectors)

Array creation

MATLAB/Octave	Python	Description
<code>zeros(3,5)</code>	<code>zeros((3,5),Float)</code>	0 filled array
	<code>zeros((3,5))</code>	0 filled array of integers
<code>ones(3,5)</code>	<code>ones((3,5),Float)</code>	1 filled array
<code>ones(3,5)*9</code>		Any number filled array
<code>eye(3)</code>	<code>identity(3)</code>	Identity matrix
<code>diag([4 5 6])</code>	<code>diag((4,5,6))</code>	Diagonal
<code>magic(3)</code>		Magic squares; Lo Shu
	<code>a = empty((3,3))</code>	Empty array

Reshape and flatten matrices

MATLAB/Octave	Python	Description
<code>reshape(1:6,3,2)';</code>	<code>arange(1,7).reshape(2,-1)</code> <code>a.setshape(2,3)</code>	Reshaping (rows first)
<code>reshape(1:6,2,3);</code>	<code>arange(1,7).reshape(-1,2).transpose()</code>	Reshaping (columns first)
<code>a'(:)</code>	<code>a.flatten()</code> <i>or</i>	Flatten to vector (by rows, like

<code>a(:)</code>	<code>a.flatten(1)</code>	comics) Flatten to vector (by columns)
<code>vech(a)</code>		Flatten upper triangle (by columns)

Shared data (slicing)

MATLAB/Octave	Python	Description
<code>b = a</code>	<code>b = a.copy()</code>	Copy of a

Indexing and accessing elements (Python: slicing)

MATLAB/Octave	Python	Description
<code>a = [11 12 13 14 ... 21 22 23 24 ... 31 32 33 34]</code>	<code>a = array([[11, 12, 13, 14], [21, 22, 23, 24], [31, 32, 33, 34]])</code>	Input is a 3,4 array
<code>a(2,3)</code>	<code>a[1,2]</code>	Element 2,3 (row,col)
<code>a(1,:)</code>	<code>a[0,]</code>	First row
<code>a(:,1)</code>	<code>a[:,0]</code>	First column
<code>a([1 3],[1 4]);</code>	<code>a.take([0,2]).take([0,3], axis=1)</code>	Array as indices
<code>a(2:end,:)</code>	<code>a[1:,:]</code>	All, except first row
<code>a(end-1:end,:)</code>	<code>a[-2:,:]</code>	Last two rows
<code>a(1:2:end,:)</code>	<code>a[:,2:,:]</code>	Strides: Every other row
	<code>a[... ,2]</code>	Third in last dimension (axis)
<code>a(:, [1 3 4])</code>	<code>a.take([0,2,3],axis=1)</code>	Remove one column
	<code>a.diagonal(offset=0)</code>	Diagonal

Assignment

MATLAB/Octave	Python	Description
<code>a(:,1) = 99</code>	<code>a[:,0] = 99</code>	
<code>a(:,1) = [99 98 97]'</code>	<code>a[:,0] = array([99,98,97])</code>	
<code>a(a>90) = 90;</code>	<code>(a>90).choose(a,90)</code>	Clipping: Replace all elements over 90
	<code>a.clip(min=None, max=90)</code>	
	<code>a.clip(min=2, max=5)</code>	Clip upper and lower values

Transpose and inverse

MATLAB/Octave	Python	Description
<code>a'</code>	<code>a.conj().transpose()</code>	Transpose
<code>a.' or transpose(a)</code>	<code>a.transpose()</code>	Non-conjugate transpose
<code>det(a)</code>	<code>linalg.det(a) or</code>	Determinant
<code>inv(a)</code>	<code>linalg.inv(a) or</code>	Inverse
<code>pinv(a)</code>	<code>linalg.pinv(a)</code>	Pseudo-inverse
<code>norm(a)</code>	<code>norm(a)</code>	Norms
<code>eig(a)</code>	<code>linalg.eig(a)[0]</code>	Eigenvalues
<code>svd(a)</code>	<code>linalg.svd(a)</code>	Singular values

`chol(a)`
`[V,L] = eig(a)`
`rank(a)`

`linalg.cholesky(a)`
`linalg.eig(a)[1]`
`rank(a)`

Cholesky factorization
Eigenvectors
Rank

Sum

MATLAB/Octave

`sum(a)`
`sum(a')`
`sum(sum(a))`

`cumsum(a)`

Python

`a.sum(axis=0)`
`a.sum(axis=1)`
`a.sum()`
`a.trace(offset=0)`
`a.cumsum(axis=0)`

Description

Sum of each column
Sum of each row
Sum of all elements
Sum along diagonal
Cumulative sum (columns)

Sorting

MATLAB/Octave

`a = [4 3 2 ; 2 8 6 ; 1 4 7]`

`sort(a(:))`
`sort(a)`
`sort(a')`
`sortrows(a,1)`

Python

`a = array([[4,3,2],[2,8,6],
[1,4,7]])`
`a.ravel().sort()` *or*
`a.sort(axis=0)` *or* `msort(a)`
`a.sort(axis=1)`
`a[a[:,0].argsort(),:]`
`a.ravel().argsort()`
`a.argsort(axis=0)`
`a.argsort(axis=1)`

Description

Example data

Flat and sorted
Sort each column
Sort each row
Sort rows (by first row)
Sort, return indices
Sort each column, return indices
Sort each row, return indices

Maximum and minimum

MATLAB/Octave

`max(a)`
`max(a')`
`max(max(a))`
`[v i] = max(a)`
`max(b,c)`
`cummax(a)`

Python

`a.max(0)` *or* `amax(a [,axis=0])`
`a.max(1)` *or* `amax(a, axis=1)`
`a.max()` *or*

`maximum(b,c)`

`a.ptp(); a.ptp(0)`

Description

max in each column
max in each row
max in array
return indices, i
pairwise max

max-to-min range

Matrix manipulation

MATLAB/Octave

`fliplr(a)`
`flipud(a)`
`rot90(a)`
`repmat(a,2,3)`
`kron(ones(2,3),a)`

`triu(a)`
`tril(a)`

Python

`fliplr(a)` *or* `a[:,::-1]`
`flipud(a)` *or* `a[::-1,:]`
`rot90(a)`
`kron(ones((2,3)),a)`

`triu(a)`
`tril(a)`

Description

Flip left-right
Flip up-down
Rotate 90 degrees
Repeat matrix: [a a a ; a a a]

Triangular, upper
Triangular, lower

Equivalents to "size"

MATLAB/Octave	Python	Description
size(a)	a.shape <i>or</i> a.getshape()	Matrix dimensions
size(a,2) <i>or</i> length(a)	a.shape[1] <i>or</i> size(a, axis=1)	Number of columns
length(a(:))	a.size <i>or</i> size(a[, axis=None])	Number of elements
ndims(a)	a.ndim	Number of dimensions
	a.nbytes	Number of bytes used in memory

Matrix- and elementwise- multiplication

MATLAB/Octave	Python	Description
a .* b	a * b <i>or</i> multiply(a,b)	Elementwise operations
a * b	matrixmultiply(a,b)	Matrix product (dot product)
	inner(a,b) <i>or</i>	Inner matrix vector multiplication $a \cdot b'$
	outer(a,b) <i>or</i>	Outer product
kron(a,b)	kron(a,b)	Kronecker product
a / b		Matrix division, $b \cdot a^{-1}$
a \ b	linalg.solve(a,b)	Left matrix division, $a^{-1} \cdot b$ (solve linear equations)
	vdot(a,b)	Vector dot product
	cross(a,b)	Cross product

Find; conditional indexing

MATLAB/Octave	Python	Description
find(a)	a.ravel().nonzero()	Non-zero elements, indices
[i j] = find(a)	(i,j) = a.nonzero() (i,j) = where(a!=0)	Non-zero elements, array indices
[i j v] = find(a)	v = a.compress((a!=0).flat) v = extract(a!=0,a)	Vector of non-zero values
find(a>5.5)	(a>5.5).nonzero() a.compress((a>5.5).flat)	Condition, indices Return values
a .* (a>5.5)	where(a>5.5,0,a) <i>or</i> a * (a>5.5) a.put(2,indices)	Zero out elements above 5.5 Replace values

Multi-way arrays

MATLAB/Octave	Python	Description
a = cat(3, [1 2; 1 2],[3 4; 3 4]); a(1,:,:)	a = array([[[1,2],[1,2]], [[3,4],[3,4]]]) a[0,...]	Define a 3-way array

File input and output

MATLAB/Octave	Python	Description
<code>f = load('data.txt')</code>	<code>f = fromfile("data.txt")</code>	Reading from a file (2d)
<code>f = load('data.txt')</code>	<code>f = load("data.txt")</code>	Reading from a file (2d)
<code>x = dlmread('data.csv', ';')</code>	<code>f = load('data.csv', delimiter=';')</code>	Reading from a CSV file (2d)
<code>save -ascii data.txt f</code>	<code>save('data.csv', f, fmt='%.6f', delimiter=';')</code>	Writing to a file (2d)
	<code>f.tofile(file='data.csv', format='%.6f', sep=';')</code>	Writing to a file (1d)
	<code>f = fromfile(file='data.csv', sep=';')</code>	Reading from a file (1d)

Plotting

Basic x-y plots

MATLAB/Octave	Python	Description
<code>plot(a)</code>	<code>plot(a)</code>	1d line plot
<code>plot(x(:,1),x(:,2),'o')</code>	<code>plot(x[:,0],x[:,1],'o')</code>	2d scatter plot
<code>plot(x1,y1, x2,y2)</code>	<code>plot(x1,y1,'bo', x2,y2,'go')</code>	Two graphs in one plot
<code>plot(x1,y1)</code>	<code>plot(x1,y1,'o')</code>	Overplotting: Add new plots to current
<code>hold on</code>	<code>plot(x2,y2,'o')</code>	
<code>plot(x2,y2)</code>	<code>show() # as normal</code>	
<code>subplot(211)</code>	<code>subplot(211)</code>	subplots
<code>plot(x,y,'ro-')</code>	<code>plot(x,y,'ro-')</code>	Plotting symbols and color

Axes and titles

MATLAB/Octave	Python	Description
<code>grid on</code>	<code>grid()</code>	Turn on grid lines
<code>axis equal</code>	<code>figure(figsize=(6,6))</code>	1:1 aspect ratio
<code>axis('equal')</code>		
<code>replot</code>		
<code>axis([0 10 0 5])</code>	<code>axis([0, 10, 0, 5])</code>	Set axes manually
<code>title('title')</code>		Axis labels and titles
<code>xlabel('x-axis')</code>		
<code>ylabel('y-axis')</code>		
	<code>text(2,25,'hello')</code>	Insert text

Log plots

MATLAB/Octave	Python	Description
<code>semilogy(a)</code>	<code>semilogy(a)</code>	logarithmic y-axis
<code>semilogx(a)</code>	<code>semilogx(a)</code>	logarithmic x-axis
<code>loglog(a)</code>	<code>loglog(a)</code>	logarithmic x and y axes

Filled plots and bar plots

MATLAB/Octave	Python	Description
<pre>fill(t,s,'b', t,c,'g') % fill has a bug?</pre>	<pre>fill(t,s,'b', t,c,'g', alpha=0.2)</pre>	Filled plot

Functions

MATLAB/Octave	Python	Description
<pre>f = inline('sin(x/3) - cos(x/5)') ezplot(f,[0,40]) fplot('sin(x/3) - cos(x/5)', [0,40]) % no ezplot</pre>	<pre>x = arange(0,40,.5) y = sin(x/3) - cos(x/5) plot(x,y, 'o')</pre>	Defining functions Plot a function for given range

Polar plots

MATLAB/Octave	Python	Description
<pre>theta = 0:.001:2*pi; r = sin(2*theta); polar(theta, rho)</pre>	<pre>theta = arange(0,2*pi,0.001) r = sin(2*theta) polar(theta, rho)</pre>	

Histogram plots

MATLAB/Octave	Python	Description
<pre>hist(randn(1000,1)) hist(randn(1000,1), -4:4) plot(sort(a))</pre>		

3d data

Contour and image plots

MATLAB/Octave	Python	Description
<pre>contour(z)</pre>	<pre>levels, colls = contour(Z, V, origin='lower', extent=(-3,3,- 3,3)) clabel(colls, levels, inline=1, fmt='%1.1f', fontsize=10)</pre>	Contour plot
<pre>contourf(z); colormap(gray)</pre>	<pre>contourf(Z, V, cmap=cm.gray, origin='lower', extent=(-3,3,-3,3))</pre>	Filled contour plot
<pre>image(z) colormap(gray)</pre>	<pre>im = imshow(Z, interpolation='bilinear', origin='lower', extent=(-3,3,-3,3)) # imshow() and contour() as above</pre>	Plot image data Image with contours

quiver()

quiver()

Direction field vectors

Perspective plots of surfaces over the x-y plane

MATLAB/Octave

```
n=-2:.1:2;
[x,y] = meshgrid(n,n);
z=x.*exp(-x.^2-y.^2);
mesh(z)
surf(x,y,z) or surf1(x,y,z)
% no surf1()
```

Python

```
n=arrayrange(-2,2,.1)
[x,y] = meshgrid(n,n)
z = x*power(math.e,-x**2-y**2)
```

Description

Mesh plot

Surface plot

Scatter (cloud) plots

MATLAB/Octave

```
plot3(x,y,z,'k')
```

Python

Description

3d scatter plot

Save plot to a graphics file

MATLAB/Octave

```
plot(1:10)
print -depsc2 foo.eps
gset output "foo.eps"
gset terminal postscript eps
plot(1:10)
```

Python

```
savefig('foo.eps')
```

Description

PostScript

```
print -dpng foo.png
```

```
savefig('foo.pdf')
```

```
savefig('foo.svg')
```

```
savefig('foo.png')
```

PDF

SVG (vector graphics for www)

PNG (raster graphics)

Data analysis

Set membership operators

MATLAB/Octave

```
a = [ 1 2 2 5 2 ];
b = [ 2 3 4 ];
```

Python

```
a = array([1,2,2,5,2])
b = array([2,3,4])
a = set([1,2,2,5,2])
b = set([2,3,4])
```

Description

Create sets

unique(a)

unique1d(a)

unique(a)

set(a)

Set unique

union(a,b)

union1d(a,b)

a.union(b)

Set union

intersect(a,b)

intersect1d(a)

a.intersection(b)

Set intersection

setdiff(a,b)

setdiff1d(a,b)

a.difference(b)

Set difference

setxor(a,b)

setxor1d(a,b)

a.symmetric_difference(b)

Set exclusion

<code>ismember(2,a)</code>	<code>2 in a</code>	True for set member
	<code>setmember1d(2,a)</code>	
	<code>contains(a,2)</code>	

Statistics

MATLAB/Octave	Python	Description
<code>mean(a)</code>	<code>a.mean(axis=0)</code> <code>mean(a [,axis=0])</code>	Average
<code>median(a)</code>	<code>median(a)</code> <i>or</i> <code>median(a [,axis=0])</code>	Median
<code>std(a)</code>	<code>a.std(axis=0)</code> <i>or</i> <code>std(a [,axis=0])</code>	Standard deviation
<code>var(a)</code>	<code>a.var(axis=0)</code> <i>or</i> <code>var(a)</code>	Variance
<code>corr(x,y)</code>	<code>correlate(x,y)</code> <i>or</i> <code>corrcoef(x,y)</code>	Correlation coefficient
<code>cov(x,y)</code>	<code>cov(x,y)</code>	Covariance

Interpolation and regression

MATLAB/Octave	Python	Description
<code>z = polyval(polyfit(x,y,1),x)</code> <code>plot(x,y,'o', x,z ,'-')</code> <code>a = x\y</code>	<code>(a,b) = polyfit(x,y,1)</code> <code>plot(x,y,'o', x,a*x+b,'-')</code> <code>linalg.lstsq(x,y)</code>	Straight line fit Linear least squares $y = ax + b$
<code>polyfit(x,y,3)</code>	<code>polyfit(x,y,3)</code>	Polynomial fit

Non-linear methods

Polynomials, root finding

MATLAB/Octave	Python	Description
<code>polyval([1 -1 -1])</code>	<code>poly()</code>	Polynomial
<code>roots([1 -1 -1])</code>	<code>roots()</code>	Find zeros of polynomial
<code>f = inline('1/x - (x-1)')</code> <code>fzero(f,1)</code>		Find a zero near $x = 1$
<code>solve('1/x = x-1')</code>		Solve symbolic equations
<code>polyval([1 2 1 2],1:10)</code>	<code>polyval(array([1,2,1,2]),arange(1,11))</code>	Evaluate polynomial

Differential equations

MATLAB/Octave	Python	Description
<code>diff(a)</code>	<code>diff(x, n=1, axis=0)</code>	Discrete difference function and approximate derivative
		Solve differential equations

Fourier analysis

MATLAB/Octave	Python	Description
<code>fft(a)</code>	<code>fft(a)</code> <i>or</i>	Fast fourier transform

ifft(a)	ifft(a) <i>or</i> convolve(x,y)	Inverse fourier transform Linear convolution
---------	------------------------------------	---

Symbolic algebra; calculus

MATLAB/Octave	Python	Description
factor()		Factorization

Programming

MATLAB/Octave	Python	Description
.m	.py	Script file extension
%	#	Comment symbol (rest of line)
<i>% or #</i>		
% must be in MATLABPATH <i>% must be in LOADPATH</i>	from pylab import *	Import library functions
string='a=234'; eval(string)	string="a=234" eval(string)	Eval

Loops

MATLAB/Octave	Python	Description
for i=1:5; disp(i); end	for i in range(1,6): print(i)	for-statement
for i=1:5 disp(i) disp(i*2) end	for i in range(1,6): print(i) print(i*2)	Multiline for statements

Conditionals

MATLAB/Octave	Python	Description
if 1>0 a=100; end	if 1>0: a=100	if-statement
if 1>0 a=100; else a=0; end		if-else-statement

Debugging

MATLAB/Octave	Python	Description
ans		Most recent evaluated expression
whos <i>or</i> who		List variables loaded into memory
clear x <i>or</i> clear [all]		Clear variable \$x\$ from memory
disp(a)	print a	Print

Working directory and OS

MATLAB/Octave	Python	Description
dir <i>or</i> ls	os.listdir(".")	List files in directory
what	grep.grep("*.py")	List script files in directory
pwd	os.getcwd()	Displays the current working directory

```
cd foo
!notepad
```

```
system("notepad")
```

```
os.chdir('foo')
os.system('notepad')
os.popen('notepad')
```

Change working directory
Invoke a System Command

Time-stamp: "2007-11-09T16:46:36 vidar"

©2006 Vidar Bronken Gundersen, /mathesaurus.sf.net

Permission is granted to copy, distribute and/or modify this document as long as the above attribution is retained.