

# 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

<code>a / b</code>	<code>a / b</code> <i>OR</i> <code>divide(a,b)</code>	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 &lt; b</code>	<code>a &lt; b</code> <i>OR</i> <code>less(a,b)</code>	Less than
<code>a &gt; b</code>	<code>a &gt; b</code> <i>OR</i> <code>greater(a,b)</code>	Greater than
<code>a &lt;= b</code>	<code>a &lt;= b</code> <i>OR</i> <code>less_equal(a,b)</code>	Less than or equal
<code>a &gt;= b</code>	<code>a &gt;= 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 &amp;&amp; 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 &amp; 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>around(a)</code> <i>OR</i> <code>math.round(a)</code>	Round
<code>ceil(a)</code>	<code>ceil(a)</code>	Round up
<code>floor(a)</code>	<code>floor(a)</code>	Round down
<code>fix(a)</code>	<code>fix(a)</code>	Round towards zero

## Mathematical constants

MATLAB/Octave	Python	Description
<code>pi</code>	<code>math.pi</code>	$\pi=3.141592$
<code>exp(1)</code>	<code>math.e</code> <i>OR</i> <code>math.exp(1)</code>	$e=2.718281$

## Missing values; IEEE-754 floating point status flags

MATLAB/Octave	Python	Description
<code>NaN</code>	<code>nan</code>	Not a Number
<code>Inf</code>	<code>inf</code>	Infinity, $\infty$
	<code>plus_inf</code>	Infinity, $+\infty$
	<code>minus_inf</code>	Infinity, $-\infty$
	<code>plus_zero</code>	Plus zero, $+0$
	<code>minus_zero</code>	Minus zero, $-0$

## Complex numbers

MATLAB/Octave	Python	Description
<code>i</code>	<code>z = 1j</code>	Imaginary unit
<code>z = 3+4i</code>	<code>z = 3+4j</code> <i>OR</i> <code>z = complex(3,4)</code>	A complex number, $3+4i$
<code>abs(z)</code>	<code>abs(3+4j)</code>	Absolute value (modulus)
<code>real(z)</code>	<code>z.real</code>	Real part
<code>imag(z)</code>	<code>z.imag</code>	Imaginary part
<code>arg(z)</code>		Argument
<code>conj(z)</code>	<code>z.conj()</code> ; <code>z.conjugate()</code>	Complex conjugate

## Trigonometry

MATLAB/Octave	Python	Description
<code>atan(a,b)</code>	<code>atan2(b,a)</code>	Arctangent, $\arctan(b/a)$
	<code>hypot(x,y)</code>	Hypotenuse; Euclidean distance

## Generate random numbers

MATLAB/Octave	Python	Description
<code>rand(1,10)</code>	<code>random.random((10,))</code> <code>random.uniform((10,))</code>	Uniform distribution
<code>2+5*rand(1,10)</code>	<code>random.uniform(2,7,(10,))</code>	Uniform: Numbers between 2 and 7
<code>rand(6)</code>	<code>random.uniform(0,1,(6,6))</code>	Uniform: 6,6 array
<code>randn(1,10)</code>	<code>random.standard_normal((10,))</code>	Normal distribution

## Vectors

MATLAB/Octave	Python	Description
<code>a=[2 3 4 5];</code> <code>adash=[2 3 4 5]';</code>	<code>a=array([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>	Row vector, $1 \times n$ -matrix 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)</code> , <code>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 comics)
<code>a(:)</code>	<code>a.flatten(1)</code>	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(:, [1 3 4])</code>	<code>a[:,... ,2]</code>	Third in last dimension (axis)
	<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&gt;90) = 90;</code>	<code>(a&gt;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.' <i>or</i> transpose(a)</code>	<code>a.transpose()</code>	Non-conjugate transpose
<code>det(a)</code>	<code>linalg.det(a) <i>or</i></code>	Determinant
<code>inv(a)</code>	<code>linalg.inv(a) <i>or</i></code>	Inverse
<code>pinv(a)</code>	<code>linalg.pinv(a)</code>	Pseudo-inverse

```
norm(a)
eig(a)
svd(a)
chol(a)
[v,l] = eig(a)
rank(a)
```

```
norm(a)
linalg.eig(a)[0]
linalg.svd(a)
linalg.cholesky(a)
linalg.eig(a)[1]
rank(a)
```

Norms  
Eigenvalues  
Singular values  
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)
```

### Python

```
fliplr(a) or a[:,::-1]
flipud(a) or a[::-1,]
```

### Description

Flip left-right  
Flip up-down

rot90(a)	rot90(a)	Rotate 90 degrees
repmat(a,2,3)	kron(ones((2,3)),a)	Repeat matrix: [ a a a ; a a a ]
kron(ones(2,3),a)		
triu(a)	triu(a)	Triangular, upper
tril(a)	tril(a)	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()	Condition, indices
	a.compress((a>5.5).flat)	Return values
a .* (a>5.5)	where(a>5.5,0,a) <i>or</i> a * (a>5.5)	Zero out elements above 5.5
	a.put(2,indices)	Replace values

## Multi-way arrays



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

## File input and output

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

## Plotting

### Basic x-y plots

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

### Axes and titles

MATLAB/Octave	Python	Description
<pre>grid on</pre>	<pre>grid()</pre>	Turn on grid lines
<pre>axis equal</pre>	<pre>figure(figsize=(6,6))</pre>	1:1 aspect ratio
<pre>axis('equal')</pre>		
<pre>replot</pre>		
<pre>axis([ 0 10 0 5 ])</pre>	<pre>axis([ 0, 10, 0, 5 ])</pre>	Set axes manually
<pre>title('title')</pre>		Axis labels and titles
<pre>xlabel('x-axis')</pre>		
<pre>ylabel('y-axis')</pre>		
	<pre>text(2,25,'hello')</pre>	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
<code>fill(t,s,'b', t,c,'g')</code> <code>% fill has a bug?</code>	<code>fill(t,s,'b', t,c,'g', alpha=0.2)</code>	Filled plot

## Functions

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

## Polar plots

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

## Histogram plots

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

## 3d data

## Contour and image plots

MATLAB/Octave	Python	Description
<code>contour(z)</code>	<code>levels, colls = contour(Z, V,</code> <code>origin='lower', extent=</code> <code>(-3,3,-3,3))</code> <code>clabel(colls, levels, inline=1,</code> <code>fmt='%1.1f', fontsize=10)</code>	Contour plot
<code>contourf(z); colormap(gray)</code>	<code>contourf(Z, V,</code> <code>cmap=cm.gray,</code>	Filled contour plot

image(z)	origin='lower', extent=(-3,3,-3,3))	Plot image data
colormap(gray)	im = imshow(Z, interpolation='bilinear', origin='lower', extent=(-3,3,-3,3))	
	# imshow() and contour() as above	Image with contours
quiver()	quiver()	Direction field vectors

## Perspective plots of surfaces over the x-y plane

MATLAB/Octave	Python	Description
n=-2:.1:2; [x,y] = meshgrid(n,n); z=x.*exp(-x.^2-y.^2); mesh(z) surf(x,y,z) <i>or</i> surf1(x,y,z) <i>% no surf1()</i>	n=arrayrange(-2,2,.1) [x,y] = meshgrid(n,n) z = x*power(math.e,-x**2-y**2)	Mesh plot Surface plot

## Scatter (cloud) plots

MATLAB/Octave	Python	Description
plot3(x,y,z,'k')		3d scatter plot

## Save plot to a graphics file

MATLAB/Octave	Python	Description
plot(1:10) print -depsc2 foo.eps <i>gset output "foo.eps"</i> <i>gset terminal postscript eps</i> <i>plot(1:10)</i>	savefig('foo.eps')	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	Python	Description
a = [ 1 2 2 5 2 ]; b = [ 2 3 4 ];	a = array([1,2,2,5,2]) b = array([2,3,4]) a = set([1,2,2,5,2]) b = set([2,3,4])	Create sets
unique(a)	unique1d(a) unique(a) set(a)	Set unique

<code>union(a,b)</code>	<code>union1d(a,b)</code> <code>a.union(b)</code>	Set union
<code>intersect(a,b)</code>	<code>intersect1d(a)</code> <code>a.intersection(b)</code>	Set intersection
<code>setdiff(a,b)</code>	<code>setdiff1d(a,b)</code> <code>a.difference(b)</code>	Set difference
<code>setxor(a,b)</code>	<code>setxor1d(a,b)</code> <code>a.symmetric_difference(b)</code>	Set exclusion
<code>ismember(2,a)</code>	<code>2 in a</code> <code>setmember1d(2,a)</code> <code>contains(a,2)</code>	True for set member

## 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>polyfit(x,y,3)</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> <code>polyfit(x,y,3)</code>	Straight line fit  Linear least squares $y = ax + b$ Polynomial fit

## Non-linear methods

### Polynomials, root finding

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

`fft(a)`  
`ifft(a)`

### Python

`fft(a)` *or*  
`ifft(a)` *or*  
`convolve(x,y)`

### Description

Fast fourier transform  
Inverse fourier transform  
Linear convolution

## Symbolic algebra; calculus

### MATLAB/Octave

`factor()`

### Python

### Description

Factorization

## Programming

### MATLAB/Octave

`.m`  
`%`  
*% or #*  
`% must be in MATLABPATH`  
*% must be in LOADPATH*  
`string='a=234';`  
`eval(string)`

### Python

`.py`  
`#`  
  
`from pylab import *`  
  
`string="a=234"`  
`eval(string)`

### Description

Script file extension  
Comment symbol (rest of line)

Import library functions

Eval

## Loops

### MATLAB/Octave

`for i=1:5; disp(i); end`  
`for i=1:5`  
`disp(i)`  
`disp(i*2)`  
`end`

### Python

`for i in range(1,6): print(i)`  
`for i in range(1,6):`  
`print(i)`  
`print(i*2)`

### Description

for-statement  
Multiline for statements

## Conditionals

### MATLAB/Octave

`if 1>0 a=100; end`  
`if 1>0 a=100; else a=0; end`

### Python

`if 1>0: a=100`

### Description

if-statement  
if-else-statement

## Debugging

### MATLAB/Octave

`ans`  
`whos` *or* `who`  
`clear x` *or* `clear [all]`

### Python

### Description

Most recent evaluated expression  
List variables loaded into memory  
Clear variable \$x\$ from memory

disp(a)

print a

Print

## Working directory and OS

### MATLAB/Octave

dir *or* ls

what

pwd

cd foo

!notepad

`system("notepad")`

### Python

os.listdir(".")

grep.grep("\*.py")

os.getcwd()

os.chdir('foo')

os.system('notepad')

os.popen('notepad')

### Description

List files in directory

List script files in directory

Displays the current working directory

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.