Getting Started with MATLAB, Python and R

AAEC 6305: Dynamic Economic Optimization - Fall 2019

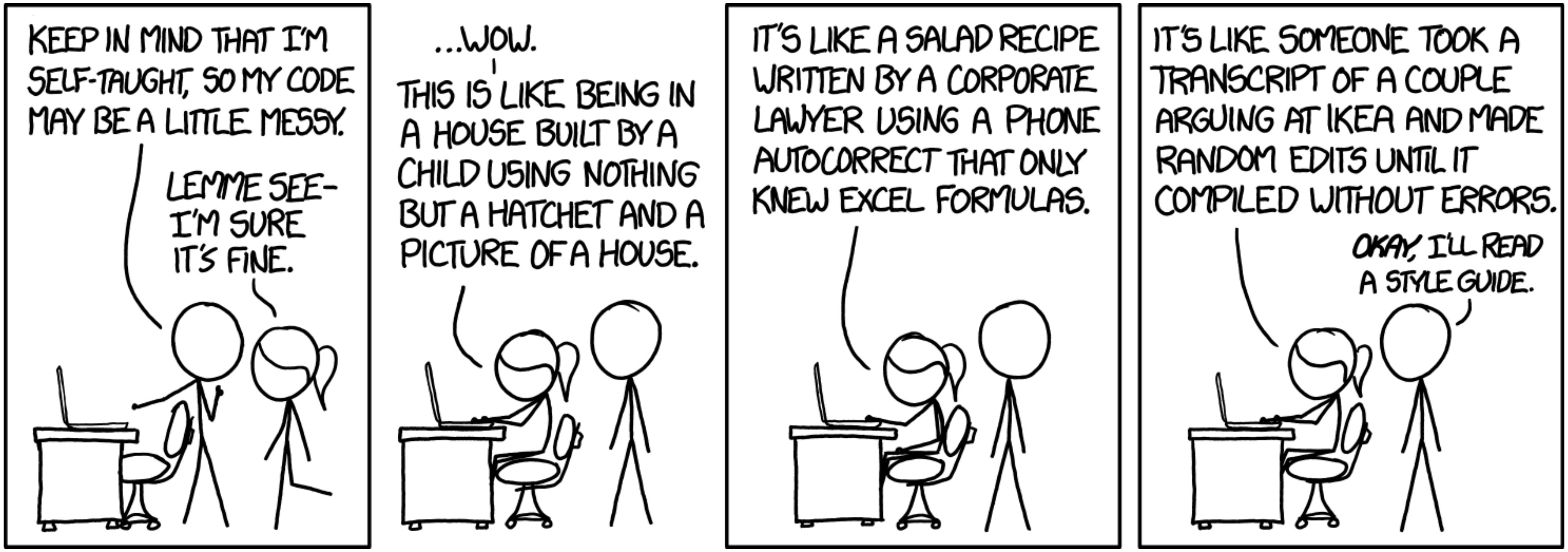
Matthew Aaron Looney

8/6/2019

Contents

1 Introduction

This document is intended to give you a quick reference guide on how to perform simple tasks in MatLab, Python and R. The document is by no means a complete list of commands. Your best friend when trying to learn the syntax of a new programming language is to use the built-in *Help* guides contained within the languge documentation (see Section 2 of this document). The second best place to find information about syntax is from Google. Chances are if you are trying to perform a specific task and are having trouble, Google the problem and you are very likely to find a solution developed by someone else trying to do the same thing. This is also a very good way to learn more complex coding techniques. Learn by seeing, doing and making mistakes!


KEEP IN MIND THAT I'M SELF-TAUGHT, SO MY CODE MAY BE A LITTLE MESSY.
LEMME SEE- I'M SURE IT'S FINE.
...WOW.
THIS IS LIKE BEING IN A HOUSE BUILT BY A CHILD USING NOTHING BUT A HATCHET AND A PICTURE OF A HOUSE.
IT'S LIKE A SALAD RECIPE WRITTEN BY A CORPORATE LAWYER USING A PHONE AUTOCORRECT THAT ONLY KNEW EXCEL FORMULAS.
IT'S LIKE SOMEONE TOOK A TRANSCRIPT OF A COUPLE ARGUING AT IKEA AND MADE RANDOM EDITS UNTIL IT COMPILED WITHOUT ERRORS.
OKAY, I'LL READ A STYLE GUIDE.

2 Help

2.1 Getting Help

Language	MATLAB/Octave	Python	R
Browse help interactively	`doc` Octave: `help -i % browse with Info`	`help()`	`help.start()`
Help on using help	`help help` or `doc doc`	`help`	`help()`
Help for a function	`help plot`	`help(plot)` or `?plot`	`help(plot)` or `?plot`
Help for a toolbox/library package	`help splines` or `doc splines`	`help(pylab)`	`help(package=’splines’)`
Demonstration examples	`demo`		`demo()`
Example using a function			`example(plot)`

2.2 Searching available help documentation

Language	MATLAB/Octave	Python	R
Search help files	`lookfor plot`		`help.search(’plot’)`
Find objects by partial name			`apropos(’plot’)`
List available packages	`help`	`help(); modules [Numeric]`	`library()`
Locate functions	`which plot`	`help(plot)`	`find(plot)`
List available methods for a function			`methods(plot)`

2.3 Using interactively

Language	MATLAB/Octave	Python	R
Start session	Octave: `octave -q`	`ipython -pylab` or JupyterLab	RStudio
Auto completion	Octave: `TAB` or `M-?`	`TAB`	
Run code from file	`foo(.m)`	`execfile(’foo.py’)` or `run foo.py`	`source(’foo.R’)`
Command history	Octave: `history`	`hist -n`	`history()`
Save command history	`diary on [..] diary off`		`savehistory(file=".Rhistory")`

End session	`exit` or `quit`	`CTRL-D`
		`CTRL-Z # windows`
		`sys.exit()`
			`q(save=’no’)`

3 Basic programming

3.1 Loading packages

Language	MATLAB/Octave	Python	R
Script file extension	`.m`	`.py`	`.R`
Comment symbol (rest of line)	`%` Octave: `%` or `#`	`#`	`#`
Import library functions	`% must be in MATLABPATH` Octave: `% must be in LOADPATH`	`from pylab import *`	`library(RSvgDevice)`

Eval	`string='a=234';`
	`eval(string)`
		`string="a=234"`
		`eval(string)`
			`string <- "a <- 234"`
			`eval(parse(text=string))`

3.2 Working directory and OS

Language	MATLAB/Octave	Python	R
List files in directory	`dir` or `ls`	`os.listdir(".")`	`list.files()` or `dir()`
List script files in directory	`what`	`grep.grep("*.py")`	`list.files(pattern="\.r$")`
Displays the current working directory	`pwd`	`os.getcwd()`	`getwd()`
Change working directory	`cd foo`	`os.chdir('foo')`	`setwd('foo')`

Invoke a System Command	`!notepad`
	Octave: `system("notepad")`
		`os.system('notepad')`
		`os.popen('notepad')`
			`system("notepad")`

3.3 Debugging and profiling code

Language	MATLAB/Octave	Python	R
Most recent evaluated expression	`ans`		`.Last.value`
List variables loaded into memory	`whos` or `who`		`objects()`
Clear variable x from memory	`clear x` or `clear [all]`		`rm(x)`
Print	`disp(a)`	`print a`	`print(a)`

3.4 Conditionals

Language	MATLAB/Octave	Python	R	
if-statement	`if 1>0 a=100; end`	`if 1>0: a=100`	`if (1>0) a <- 100`	
if-else-statement	`if 1>0 a=100; else a=0; end`			
Ternary operator (if?true:false)			`ifelse(a>0,a,0)`	$a > 0?a : 0$

3.5 Loops

Language	MATLAB/Octave	Python	R
for-statement	`for i=1:5; disp(i); end`	`for i in range(1,6): print(i)`	`for(i in 1:5) print(i)`

Multiline for statements	`for i=1:5`
	`   disp(i)`
	`   disp(i*2)`
	`end`
		`for i in range(1,6):`
		`    print(i)`
		`    print(i*2)`
			`for(i in 1:5) {`
			`    print(i)`
			`    print(i*2)`
			`}`

---

4 File and Data input/output

Language	MATLAB/Octave	Python	R

Reading from a file (2d)	`f = load('data.txt')`	`f = fromfile("data.txt")`
		`f = load("data.txt")`
			`f <- read.table("data.txt")`

Reading from a file (2d)	`f = load('data.txt')`	`f = load("data.txt")`	`f <- read.table("data.txt")`
Reading fram a CSV file (2d)	`x = dlmread('data.csv', ';')`	`f = load('data.csv', delimiter=';')`	`f <- read.table(file="data.csv", sep=";")`

Writing to a file (2d)	`save -ascii data.txt f`	`save('data.csv', f, fmt='%.6f',`
		`delimitwrier='te(f;'),file="data.txt")`	

Writing to a file (1d)		`f.tofile(file='data.csv', format='%.6f', sep=';')`	
Reading from a file (1d)		`f = fromfile(file='data.csv', sep=';')`	

---

5 Basic Operators

5.1 Getting help on operator syntax

Language	MATLAB/Octave	Python	R
Help on operator syntax	`help -`		`help(Syntax)`

5.2 Arithmetic operators

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

Power, a^b	a .^ b	a ** b
		power(a,b)
		pow(a,b)
			a ^ b

Remainder	rem(a,b)	a % b
		remainder(a,b)
		fmod(a,b)
			a %% b

Integer division			a %/% b
In place operation to save array creation overhead	Octave: a+=1	a+=b or add(a,b,a)	
Factorial, $n!$	factorial(a)		factorial(a)

5.3 Relational operators

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

5.4 Logical operators

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

5.5 Roots and logarithms

Language	MATLAB/Octave	Python	R	
Square root	sqrt(a)	math.sqrt(a)	sqrt(a)	$\sqrt{a}$
Logarithm, base e (natural)	log(a)	math.log(a)	log(a)	$\ln a = \log_e a$
Logarithm, base 10	log10(a)	math.log10(a)	log10(a)	$\log_{10} a$
Logarithm, base 2 (binary)	log2(a)	math.log(a, 2)	log2(a)	$\log_2 a$
Exponential function	exp(a)	math.exp(a)	exp(a)	e^a

5.6 Rounding

Language	MATLAB/Octave	Python	R
Round	`round(a)`	`around(a)` or `math.round(a)`	`round(a)`
Round up	`ceil(a)`	`ceil(a)`	`ceil(a)`
Round down	`floor(a)`	`floor(a)`	`floor(a)`
Round towards zero	`fix(a)`	`fix(a)`	

5.7 Mathematical constants

Language	MATLAB/Octave	Python	R
$\pi = 3.141592$	`pi`	`math.pi`	`pi`
$e = 2.718281$	`exp(1)`	`math.e` or `math.exp(1)`	`exp(1)`

5.8 Pseudo-random number generator

Language	MATLAB/Octave	Python	R

Uniform distribution	`rand(1,10)`	`random.random((10,))`
		`random.uniform((10,))`
			`runif(10)`

Uniform: Numbers between 2 and 7	`2+5*rand(1,10)`	`random.uniform(2,7,(10,))`	`runif(10, min=2, max=7)`
Uniform: 6,6 array	`rand(6)`	`random.uniform(0,1,(6,6))`	`matrix(runif(36),6)`
Normal distribution	`randn(1,10)`	`random.standard_normal((10,))`	`rnorm(10)`

6 Basic vector construction

6.1 Vectors

Language	MATLAB/Octave	Python	R
Row vector, $1 \times n$-matrix	`a=[2 3 4 5];`	`a=array([2,3,4,5])`	`a <- c(2,3,4,5)`

Column vector, $m \times 1$-matrix	`adash=[2 3 4 5]’;`	`array([2,3,4,5])[:,NewAxis]`
		`array([2,3,4,5]).reshape(-1,1)`
		`r_[1:10,’c’]`
			`adash <- t(c(2,3,4,5))`

6.2 Sequences

Language	MATLAB/Octave	Python	R

1,2,3, ... ,10	1:10	arange(1,11, dtype=Float)
		range(1,11)
			seq(10) or 1:10

0.0,1.0,2.0, ... ,9.0	0:9	arange(10.)	seq(0,length=10)
1,4,7,10	1:3:10	arange(1,11,3)	seq(1,10,by=3)
10,9,8, ... ,1	10:-1:1	arange(10,0,-1)	seq(10,1) or 10:1
10,7,4,1	10:-3:1	arange(10,0,-3)	seq(from=10,to=1,by=-3)
Linearly spaced vector of n=7 points	linspace(1,10,7)	linspace(1,10,7)	seq(1,10,length=7)
Reverse	reverse(a)	a[::-1] or	rev(a)
Set all values to same scalar value	a(:) = 3	a.fill(3), a[:] = 3	

6.3 Vector concatenation

Language	MATLAB/Octave	Python	R
Concatenate two vectors	[a a]	concatenate((a,a))	c(a,a)
	[1:4 a]	concatenate((range(1,5),a), axis=1)	c(1:4,a)

6.4 Repeating

Language	MATLAB/Octave	Python	R
1 2 3, 1 2 3	[a a]	concatenate((a,a))	rep(a,times=2)
1 1 1, 2 2 2, 3 3 3		a.repeat(3) or	rep(a,each=3)
1, 2 2, 3 3 3		a.repeat(a) or	rep(a,a)

6.5 Leave out elements

Language	MATLAB/Octave	Python	R
miss the first element	a(2:end)	a[1:]	a[-1]
miss the tenth element	a([1:9])		a[-10]
miss 1,4,7, ...			a[-seq(1,50,3)]
last element	a(end)	a[-1]	
last two elements	a(end-1:end)	a[-2:]	

6.6 Vector minimum and maximum

Language	MATLAB/Octave	Python	R
pairwise max	max(a,b)	maximum(a,b)	pmax(a,b)
max of all values in two vectors	max([a b])	concatenate((a,b)).max()	max(a,b)
	[v,i] = max(a)	v,i = a.max(0),a.argmax(0)	v <- max(a) ; i <- which.max(a)

6.7 Vector Multiplication

Language	MATLAB/Octave	Python	R
Multiply two vectors	`a.*a`	`a*a`	`a*a`
Vector dot product, $u \cdot v$	`dot(u,v)`	`dot(u,v)`	

7 Basic matrix operations

7.1 Matrix construction

Language	MATLAB/Octave	Python	R	

Define a matrix	`a = [2 3;4 5]`	`a = array([[2,3],[4,5]])`	`rbind(c(2,3),c(4,5))`
			`array(c(2,3,4,5), dim=c(2,2))`
				$\begin{bmatrix} 2 & 3 \\ 4 & 5 \end{bmatrix}$

7.2 Matrix concatenation

Language	MATLAB/Octave	Python	R

Bind rows	`[a ; b]`	`concatenate((a,b), axis=0)`
		`vstack((a,b))`
			`rbind(a,b)`

Bind columns	`[a , b]`	`concatenate((a,b), axis=1)`
		`hstack((a,b))`
			`cbind(a,b)`

Bind slices (three-way arrays)		`concatenate((a,b), axis=2)`
		`dstack((a,b))`	

Concatenate matrices into one vector	`[a(:), b(:)]`	`concatenate((a,b), axis=None)`	

Bind rows (from vectors)	`[1:4 ; 1:4]`	`concatenate((r_[1:5],r_[1:5])).reshape(2,-1)`
		`vstack((r_[1:5],r_[1:5]))`
			`rbind(1:4,1:4)`

Bind columns (from vectors)	`[1:4 ; 1:4]'`		`cbind(1:4,1:4)`

7.3 Array construction

Language	MATLAB/Octave	Python	R	
0 filled array	`zeros(3,5)`	`zeros((3,5),Float)`	`matrix(0,3,5)` or `array(0,c(3,5))`	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
0 filled array of integers		`zeros((3,5))`		
1 filled array	`ones(3,5)`	`ones((3,5),Float)`	`matrix(1,3,5)` or `array(1,c(3,5))`	$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$
Any number filled array	`ones(3,5)*9`		`matrix(9,3,5)` or `array(9,c(3,5))`	$\begin{bmatrix} 9 & 9 & 9 & 9 & 9 \\ 9 & 9 & 9 & 9 & 9 \\ 9 & 9 & 9 & 9 & 9 \end{bmatrix}$
Identity matrix	`eye(3)`	`identity(3)`	`diag(1,3)`	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$
Diagonal	`diag([4 5 6])`	`diag((4,5,6))`	`diag(c(4,5,6))`	$\begin{bmatrix} 4 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 6 \end{bmatrix}$
Magic squares; Lo Shu	`magic(3)`			$\begin{bmatrix} 8 & 1 & 6 \\ 3 & 5 & 7 \\ 4 & 9 & 2 \end{bmatrix}$
Empty array		`a = empty((3,3))`		

7.4 Reshape matricies

Language	MATLAB/Octave	Python	R	

Reshaping (rows first)	`reshape(1:6,3,2)’;`	`arange(1,7).reshape(2,-1)`
		`a.setshape(2,3)`
			`matrix(1:6,nrow=3,byrow=T)`	$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$

Reshaping (columns first)	`reshape(1:6,2,3);`	`arange(1,7).reshape(-1,2).transpose()`	`matrix(1:6,nrow=2)`
			`array(1:6,c(2,3))`
				$\begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$

Flatten to vector (by rows, like comics)	`a’(:)`	`a.flatten()` or	`as.vector(t(a))`	$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}$
Flatten to vector (by columns)	`a(:)`	`a.flatten(1)`	`as.vector(a)`	$\begin{bmatrix} 1 & 4 & 2 & 5 & 3 & 6 \end{bmatrix}$
Flatten upper triangle (by columns)	`vech(a)`		`a[row(a) <= col(a)]`	

7.5 Copy (slicing) data

Language	MATLAB/Octave	Python	R
Copy of a	`b = a`	`b = a.copy()`	`b = a`

7.6 Indexing and accessing elements inside a matrix

Language	MATLAB/Octave	Python	R	

Input is a 3,4 array	`a = [ 11 12 13 14 ...`
	`21 22 23 24 ...`
	`31 32 33 34 ]`
		`a = array([[ 11, 12, 13, 14 ],`
		`[ 21, 22, 23, 24 ],`
		`[ 31, 32, 33, 34 ]])`
			`a <- rbind(c(11, 12, 13, 14),`
			`c(21, 22, 23, 24),`
			`c(31, 32, 33, 34))`
				$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$

Element 2,3 (row,col)	`a(2,3)`	`a[1,2]`	`a[2,3]`	a_{23}
First row	`a(1,:)`	`a[0,]`	`a[1,]`	$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \end{bmatrix}$
First column	`a(:,1)`	`a[:,0]`	`a[,1]`	$\begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \end{bmatrix}$
Array as indices	`a([1 3],[1 4]);`	`a.take([0,2]).take([0,3], axis=1)`		$\begin{bmatrix} a_{11} & a_{14} \\ a_{31} & a_{34} \end{bmatrix}$
All, except first row	`a(2:end,:)`	`a[1:,]`	`a[-1,]`	$\begin{bmatrix} a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$
Last two rows	`a(end-1:end,:)`	`a[-2:,]`		$\begin{bmatrix} a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$
Strides: Every other row	`a(1:2:end,:)`	`a[::2,:]`		$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}$
Third in last dimension (axis)		`a[...,2]`		
All, except row,column (2,3)			`a[-2,-3]`	$\begin{bmatrix} a_{11} & a_{13} & a_{14} \\ a_{31} & a_{33} & a_{34} \end{bmatrix}$
Remove one column	`a(:,[1 3 4])`	`a.take([0,2,3],axis=1)`	`a[,-2]`	$\begin{bmatrix} a_{11} & a_{13} & a_{14} \\ a_{21} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} \end{bmatrix}$
Diagonal		`a.diagonal(offset=0)`		$\begin{bmatrix} a_{11} & a_{22} & a_{33} & a_{44} \end{bmatrix}$

7.7 Element assignment

Language	MATLAB/Octave	Python	R
	`a(:,1) = 99`	`a[:,0] = 99`	`a[,1] <- 99`
	`a(:,1) = [99 98 97]’`	`a[:,0] = array([99,98,97])`	`a[,1] <- c(99,98,97)`

Clipping: Replace all elements over 90	`a(a>90) = 90;`	`(a>90).choose(a,90)`
		`a.clip(min=None, max=90)`
			`a[a>90] <- 90`

Clip upper and lower values		`a.clip(min=2, max=5)`	

7.8 Transpose and inverse

Language	MATLAB/Octave	Python	R
Transpose	`a'`	`a.conj().transpose()`	`t(a)`
Non-conjugate transpose	`a.'` or `transpose(a)`	`a.transpose()`	
Determinant	`det(a)`	`linalg.det(a)` or	`det(a)`
Inverse	`inv(a)`	`linalg.inv(a)` or	`solve(a)`
Pseudo-inverse	`pinv(a)`	`linalg.pinv(a)`	`ginv(a)`
Norms	`norm(a)`	`norm(a)`	
Eigenvalues	`eig(a)`	`linalg.eig(a)[0]`	`eigen(a)$values`
Singular values	`svd(a)`	`linalg.svd(a)`	`svd(a)$d`
Cholesky factorization	`chol(a)`	`linalg.cholesky(a)`	
Eigenvectors	`[v,l] = eig(a)`	`linalg.eig(a)[1]`	`eigen(a)$vectors`
Rank	`rank(a)`	`rank(a)`	`rank(a)`

7.9 Matrix sum

Language	MATLAB/Octave	Python	R
Sum of each column	`sum(a)`	`a.sum(axis=0)`	`apply(a,2,sum)`
Sum of each row	`sum(a')`	`a.sum(axis=1)`	`apply(a,1,sum)`
Sum of all elements	`sum(sum(a))`	`a.sum()`	`sum(a)`
Sum along diagonal		`a.trace(offset=0)`	
Cumulative sum (columns)	`cumsum(a)`	`a.cumsum(axis=0)`	`apply(a,2,cumsum)`

7.10 Matrix sorting

Language	MATLAB/Octave	Python	R	
Example data	`a = [ 4 3 2 ; 2 8 6 ; 1 4 7 ]`	`a = array([[4,3,2],[2,8,6],[1,4,7]])`		$\begin{bmatrix} 4 & 3 & 2 \\ 2 & 8 & 6 \\ 1 & 4 & 7 \end{bmatrix}$
Flat and sorted	`sort(a(:))`	`a.ravel().sort()` or	`t(sort(a))`	$\begin{bmatrix} 1 & 2 & 2 \\ 3 & 4 & 4 \\ 6 & 7 & 8 \end{bmatrix}$
Sort each column	`sort(a)`	`a.sort(axis=0)` or `msort(a)`	`apply(a,2,sort)`	$\begin{bmatrix} 1 & 3 & 2 \\ 2 & 4 & 6 \\ 4 & 8 & 7 \end{bmatrix}$
Sort each row	`sort(a')'`	`a.sort(axis=1)`	`t(apply(a,1,sort))`	$\begin{bmatrix} 2 & 3 & 4 \\ 2 & 6 & 8 \\ 1 & 4 & 7 \end{bmatrix}$
Sort rows (by first row)	`sortrows(a,1)`	`a[a[:,0].argsort(),]`		$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 8 & 6 \\ 4 & 3 & 2 \end{bmatrix}$
Sort, return indices		`a.ravel().argsort()`	`order(a)`	
Sort each column, return indices		`a.argsort(axis=0)`		
Sort each row, return indices		`a.argsort(axis=1)`		

7.11 Matrix minimum and maximum

Language	MATLAB/Octave	Python	R
max in each column	max(a)	a.max(0) or amax(a [,axis=0])	apply(a,2,max)
max in each row	max(a')	a.max(1) or amax(a, axis=1)	apply(a,1,max)
max in array	max(max(a))	a.max() or	max(a)
return indices, i	[v i] = max(a)		i <- apply(a,1,which.max)
pairwise max	max(b,c)	maximum(b,c)	pmax(b,c)
	cummax(a)		apply(a,2,cummax)
max-to-min range		a.ptp(); a.ptp(0)	

7.12 Matrix manipulation

Language	MATLAB/Octave	Python	R
Flip left-right	fliplr(a)	fliplr(a) or a[:,::-1]	a[,4:1]
Flip up-down	flipud(a)	flipud(a) or a[::-1,]	a[3:1,]
Rotate 90 degrees	rot90(a)	rot90(a)	
Repeat matrix: [a a a ; a a a]	repmat(a,2,3)	kron(ones((2,3)),a)	kronecker(matrix(1,2,3),a)
	Octave: kron(ones(2,3),a)		
Triangular, upper	triu(a)	triu(a)	a[lower.tri(a)] <- 0
Triangular, lower	tril(a)	tril(a)	a[upper.tri(a)] <- 0

7.13 Matrix dimension

Language	MATLAB/Octave	Python	R
Matrix dimensions	size(a)	a.shape or a.getshape()	dim(a)
Number of columns	size(a,2) or length(a)	a.shape[1] or size(a, axis=1)	ncol(a)
Number of elements	length(a(:))	a.size or size(a[, axis=None])	prod(dim(a))
Number of dimensions	ndims(a)	a.ndim	
Number of bytes used in memory		a.nbytes	object.size(a)

7.14 Matrix and elementwise multiplication

Language	MATLAB/Octave	Python	R	
Elementwise operations	`a .* b`	`a * b` or `multiply(a,b)`	`a * b`	$\begin{bmatrix} 1 & 5 \\ 9 & 16 \end{bmatrix}$
Matrix product (dot product)	`a * b`	`matrixmultiply(a,b)`	`a %*% b`	$\begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix}$
Inner matrix vector multiplication $a \cdot b'$		`inner(a,b)` or		$\begin{bmatrix} 5 & 11 \\ 11 & 25 \end{bmatrix}$
Outer product		`outer(a,b)` or	`outer(a,b)` or `a %o% b`	$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 6 & 8 \\ 3 & 6 & 9 & 12 \\ 4 & 8 & 12 & 16 \end{bmatrix}$
Cross product			`crossprod(a,b)` or `t(a) %*% b`	$\begin{bmatrix} 10 & 14 \\ 14 & 20 \end{bmatrix}$
Kronecker product	`kron(a,b)`	`kron(a,b)`	`kronecker(a,b)`	$\begin{bmatrix} 1 & 2 & 2 & 4 \\ 3 & 4 & 6 & 8 \\ 3 & 6 & 4 & 8 \\ 9 & 12 & 12 & 16 \end{bmatrix}$
Matrix division, $b{\cdot}a^{-1}$	`a / b`			
Left matrix division, $b^{-1}{\cdot}a$ (solve linear equations)	`a \ b`	`linalg.solve(a,b)`	`solve(a,b)`	$Ax = b$
Vector dot product		`vdot(a,b)`		
Cross product		`cross(a,b)`		

7.15 Conditional indexing

Language	MATLAB/Octave	Python	R
Non-zero elements, indices	`find(a)`	`a.ravel().nonzero()`	`which(a != 0)`

Non-zero elements, array indices	`[i j] = find(a)`	`(i,j) = a.nonzero()`
		`(i,j) = where(a!=0)`
			`which(a != 0, arr.ind=T)`

Vector of non-zero values	`[i j v] = find(a)`	`v = a.compress((a!=0).flat)`
		`v = extract(a!=0,a)`
			`ij <- which(a != 0, arr.ind=T); v <- a[ij]`

Condition, indices	`find(a>5.5)`	`(a>5.5).nonzero()`	`which(a>5.5)`
Return values		`a.compress((a>5.5).flat)`	`ij <- which(a>5.5, arr.ind=T); v <- a[ij]`
Zero out elements above 5.5	`a .* (a>5.5)`	`where(a>5.5,0,a)` or `a * (a>5.5)`	
Replace values		`a.put(2,indices)`	

8 Multi-way array

Language	MATLAB/Octave	Python	R

Define a 3-way array	`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,...]`	

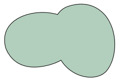
9 Data analysis

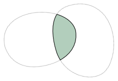
9.1 Set theory

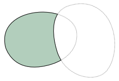
Language	MATLAB/Octave	Python	R	

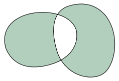
Create sets	`a = [ 1 2 2 5 2 ];`
	`b = [ 2 3 4 ];`
		`a = array([1,2,2,5,2])`
		`b = array([2,3,4])`
		`a = set([1,2,2,5,2])`
		`b = set([2,3,4])`
			`a <- c(1,2,2,5,2)`
			`b <- c(2,3,4)`	

Set unique	`unique(a)`	`unique1d(a)`
		`unique(a)`
		`set(a)`
			`unique(a)`	$\begin{bmatrix} 1 & 2 & 5 \end{bmatrix}$

Set union	`union(a,b)`	`union1d(a,b)`
		`a.union(b)`
			`union(a,b)`	

Set intersection	`intersect(a,b)`	`intersect1d(a)`
		`a.intersection(b)`
			`intersect(a,b)`	

Set difference	`setdiff(a,b)`	`setdiff1d(a,b)`
		`a.difference(b)`
			`setdiff(a,b)`	

Set exclusion	`setxor(a,b)`	`setxor1d(a,b)`
		`a.symmetric_difference(b)`
			`setdiff(union(a,b),intersect(a,b))`	

True for set member	`ismember(2,a)`	`2 in a`
		`setmember1d(2,a)`
		`contains(a,2)`
			`is.element(2,a)` or `2 %in% a`	

9.2 Satistics

Language	MATLAB/Octave	Python	R

Average	`mean(a)`	`a.mean(axis=0)`
		`mean(a [,axis=0])`
			`apply(a,2,mean)`

Median	`median(a)`	`median(a)` or `median(a [,axis=0])`	`apply(a,2,median)`
Standard deviation	`std(a)`	`a.std(axis=0)` or `std(a [,axis=0])`	`apply(a,2,sd)`
Variance	`var(a)`	`a.var(axis=0)` or `var(a)`	`apply(a,2,var)`
Correlation coefficient	`corr(x,y)`	`correlate(x,y)` or `corrcoef(x,y)`	`cor(x,y)`
Covariance	`cov(x,y)`	`cov(x,y)`	`cov(x,y)`

9.3 Basic interpolation and regression

Language	MATLAB/Octave	Python	R

Straight line fit	`z = polyval(polyfit(x,y,1),x)`
	`plot(x,y,'o', x,z ,'-')`
		`(a,b) = polyfit(x,y,1)`
		`plot(x,y,'o', x,a*x+b,'-')`
			`z <- lm(y~x)`
			`plot(x,y)`
			`abline(z)`

Linear least squares $y = ax + b$	`a = x\y`	`linalg.lstsq(x,y)`	`solve(a,b)`
Polynomial fit	`polyfit(x,y,3)`	`polyfit(x,y,3)`	

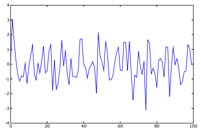
10 Plotting

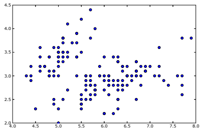
10.1 Basic x-y plots

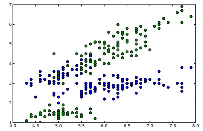
Language	MATLAB/Octave	Python	R
1d line plot	`plot(a)`	`plot(a)`	`plot(a, type="l")`
2d scatter plot	`plot(x(:,1),x(:,2),'o')`	`plot(x[:,0],x[:,1],'o')`	`plot(x[,1],x[,2])`
Two graphs in one plot	`plot(x1,y1, x2,y2)`	`plot(x1,y1,'bo', x2,y2,'go')`	

Overplotting: Add new plots to current	`plot(x1,y1)`
	`hold on`
	`plot(x2,y2)`
		`plot(x1,y1,'o')`
		`plot(x2,y2,'o')`
		`show() # as normal`
			`plot(x1,y1)`
			`matplot(x2,y2,add=T)`

subplots	`subplot(211)`	`subplot(211)`	
Plotting symbols and color	`plot(x,y,'ro-')`	`plot(x,y,'ro-')`	`plot(x,y,type="b",col="red")`







10.2 Titles and axes

Language	MATLAB/Octave	Python	R
Turn on grid lines	`grid on`	`grid()`	`grid()`

1:1 aspect ratio	`axis equal`
	Octave:
	`axis('equal')`
	`replot`
		`figure(figsize=(6,6))`	`plot(c(1:10,10:1), asp=1)`

Set axes manually	`axis([ 0 10 0 5 ])`	`axis([ 0, 10, 0, 5 ])`	`plot(x,y, xlim=c(0,10), ylim=c(0,5))`

Axis labels and titles	`title('title')`
	`xlabel('x-axis')`
	`ylabel('y-axis')`
			`plot(1:10, main="title",`
			`  xlab="x-axis", ylab="y-axis")`

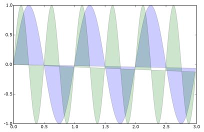
Insert text		`text(2,25,'hello')`	

10.3 Log plots

Language	MATLAB/Octave	Python	R
logarithmic y-axis	`semilogy(a)`	`semilogy(a)`	`plot(x,y, log="y")`
logarithmic x-axis	`semilogx(a)`	`semilogx(a)`	`plot(x,y, log="x")`
logarithmic x and y axes	`loglog(a)`	`loglog(a)`	`plot(x,y, log="xy")`

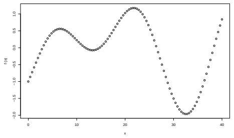
10.4 Fill and bar plots

Language	MATLAB/Octave	Python	R	

Filled plot	`fill(t,s,'b', t,c,'g')`
	Octave: `% fill has a bug?`
		`fill(t,s,'b', t,c,'g', alpha=0.2)`	`plot(t,s, type="n", xlab="", ylab="")`
			`polygon(t,s, col="lightblue")`
			`polygon(t,c, col="lightgreen")`
				

Stem-and-Leaf plot			`stem(x[,3])`	5 5 6 71 7 033 8 00113345567889 9 0133566677788 10 32674

10.5 Plotting functions

Language	MATLAB/Octave	Python	R	
Defining functions	`f = inline('sin(x/3) - cos(x/5)')`		`f <- function(x) sin(x/3) - cos(x/5)`	$f(x) = \sin\left(\frac{x}{3}\right) - \cos\left(\frac{x}{5}\right)$ 

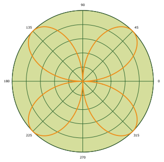
Plot a function for given range	`ezplot(f,[0,40])`
	`fplot('sin(x/3) - cos(x/5)',[0,40])`
	Octave: `% no ezplot`
		`x = arrayrange(0,40,.5)`
		`y = sin(x/3) - cos(x/5)`
		`plot(x,y, 'o')`
			`plot(f, xlim=c(0,40), type='p')`	

10.6 Histogram plots

Language	MATLAB/Octave	Python	R
	`hist(randn(1000,1))`		`hist(rnorm(1000))`
	`hist(randn(1000,1), -4:4)`		`hist(rnorm(1000), breaks= -4:4)`
			`hist(rnorm(1000), breaks=c(seq(-5,0,0.25), seq(0.5,5,0.5)), freq=F)`
	`plot(sort(a))`		`plot(apply(a,1,sort),type="l")`

10.7 Polar coordinate plots

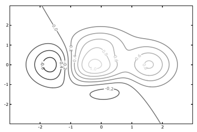
Language	MATLAB/Octave	Python	R	

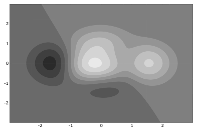
	`theta = 0:.001:2*pi;`
	`r = sin(2*theta);`
		`theta = arange(0,2*pi,0.001)`
		`r = sin(2*theta)`
				$\rho(\theta) = \sin(2\theta)$
				

	`polar(theta, rho)`	`polar(theta, rho)`		

10.8 Contour plots

Language	MATLAB/Octave	Python	R	

Contour plot	`contour(z)`	`levels, colls = contour(Z, V,`
		`    origin='lower', extent=(-3,3,-3,3))`
		`clabel(colls, levels, inline=1,`
		`    fmt='%1.1f', fontsize=10)`
			`contour(z)`	

Filled contour plot	`contourf(z); colormap(gray)`	`contourf(Z, V,`
		`    cmap=cm.gray,`
		`    origin='lower',`
		`    extent=(-3,3,-3,3))`
			`filled.contour(x,y,z,`
			`    nlevels=7, color=gray.colors)`
				

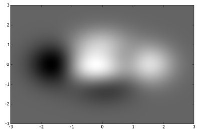
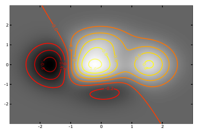
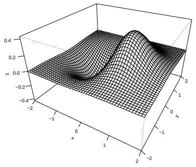
Plot image data	`image(z)`
	`colormap(gray)`
		`im = imshow(Z,`
		`    interpolation='bilinear',`
		`    origin='lower',`
		`    extent=(-3,3,-3,3))`
			`image(z, col=gray.colors(256))`	

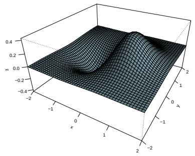
Image with contours		`# imshow() and contour() as above`		
Direction field vectors	`quiver()`	`quiver()`		

10.9 Perspective plots

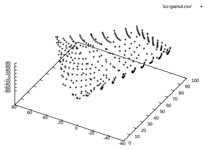
Language	MATLAB/Octave	Python	R	

	`n=-2:.1:2;`
	`[x,y] = meshgrid(n,n);`
	`z=x.*exp(-x.^2-y.^2);`
		`n=arrayrange(-2,2,.1)`
		`[x,y] = meshgrid(n,n)`
		`z = x*power(math.e,-x**2-y**2)`
			`f <- function(x,y) x*exp(-x^2-y^2)`
			`n <- seq(-2,2, length=40)`
			`z <- outer(n,n,f)`
				$f(x,y) = xe^{-x^2-y^2}$

Mesh plot	`mesh(z)`		`persp(x,y,z,`
			`  theta=30, phi=30, expand=0.6,`
			`  ticktype='detailed')`
				

Surface plot	`surf(x,y,z)` or `surfl(x,y,z)`
	Octave: `% no surfl()`
			`persp(x,y,z,`
			`  theta=30, phi=30, expand=0.6,`
			`  col='lightblue', shade=0.75, ltheta=120,`
			`  ticktype='detailed')`
				

10.10 Cloud plots

Language	MATLAB/Octave	Python	R	
3d scatter plot	`plot3(x,y,z,'k+')`		`cloud(z~x*y)`	

10.11 Save plot to file

Language	MATLAB/Octave	Python	R

PostScript	`plot(1:10)`
	`print -depsc2 foo.eps`
	Octave:
	`gset output "foo.eps"`
	`gset terminal postscript eps`
	`plot(1:10)`
		`savefig(’foo.eps’)`	`postscript(file="foo.eps")`
			`plot(1:10)`
			`dev.off()`

PDF		`savefig(’foo.pdf’)`	`pdf(file=’foo.pdf’)`
SVG (vector graphics for www)		`savefig(’foo.svg’)`	`devSVG(file=’foo.svg’)`
PNG (raster graphics)	`print -dpng foo.png`	`savefig(’foo.png’)`	`png(filename = "Rplot%03d.png"`

11 References

11.1 Computer Algebra Systems

Wester, Michael (ed). Computer Algebra Systems: A Practical Guide (1999), available from http://www.math.unm.edu/~wester/cas_review.html (accessed 2019.01.01).

11.2 MatLab

Moler, Cleve. Numerical Computing with MATLAB (MathWorks, 2004), available from http://www.mathworks.com/moler/ (accessed 2019.01.01);

11.3 Octave

Hankin, Robin. R for Octave users (2001), available from http://cran.r-project.org/doc/contrib/R-and-octave.txt (accessed 2019.01.01);

Eaton, John W. Octave Quick Reference (2007);

11.4 Python

Martelli, Alex. Python in a Nutshell (O'Reilly, 2006);

Oliphant, Travis. Guide to NumPy (Trelgol, 2006), available from http://web.mit.edu/dvp/Public/numpybook.pdf (accessed 2019.01.01);

Hunter, John. The Matplotlib User's Guide (2019), available from http://matplotlib.sf.net/ (accessed 2019.01.01);

Langtangen, Hans Petter. Python Scripting for Computational Science (Springer, 2009);

Ascher et al.: Numeric Python manual (2019), available from https://docs.scipy.org/doc/ (accessed 2019.01.01);

Greenfield, Jedrzejewski & Laidler. Using Python for Interactive Data Analysis (2007), pp.125-134, available from https://ssb.stsci.edu/perry/pydatatut.pdf (accessed 2019.01.01);

11.5 R

Venables & Smith: An Introduction to R (2019), available from https://cran.r-project.org/doc/manuals/r-release/R-intro.pdf (accessed 2019.01.01);

Short, Tom. R reference card (2004), available from https://cran.r-project.org/doc/contrib/Short-refcard.pdf (accessed 2019.01.01);

11.6 Miscellaneous

Merrit, Ethan. Demos for gnuplot version 5.2 (2018), available from http://gnuplot.sourceforge.net/demo/ (accessed 2019.01.01);

Woo, Alex. Gnuplot Quick Reference (2004), available from http://www.gnuplot.info/docs_4.0/gpcard.pdf (accessed 2019.01.01);

Brisson, Eric. Using IDL to Manipulate and Visualize Scientific Data, available from http://scv.bu.edu/documentation/tutorials/IDL/ (accessed 2019.01.01);