The QuantEcon MATLAB-Python-Julia Cheat Sheet

Victoria Gregory

September 08, 2016

This document summarizes commonly-used, equivalent commands across MATLAB, Python, and Julia

CREATING VECTORS

Operation	MATLAB	Python	Julia
Create a row vector	`A = [1 2 3]`	`A = np.array([1 2 3])`	`A = [1 2 3]`
Create a column vector	`A = [1; 2; 3]`		`A = [1; 2; 3]`
Sequence starting at j ending at n, with difference k between points	`A = j:k:n`		`A = j:k:n`
Linearly spaced vector of k points	`A = linspace(1, 5, k)`		`A = linspace(1, 5, k)`
Vector dot product	`dot(A, B)`		`dot(A, B)`

CREATING MATRICES

Operation	MATLAB	Python	Julia
Create a matrix	`A = [1 2; 3 4]`		`A = [1 2; 3 4]`
Create a 2 by 2 matrix of zeros	`A = zeros(2, 2)`		`A = zeros(2, 2)`
Create a 2 by 2 matrix of ones	`A = ones(2, 2)`		`A = ones(2, 2)`
Create a 2 by 2 identity matrix	`A = eye(2, 2)`		`A = eye(2, 2)`
Create a diagonal matrix	`A = diag([1 2 3])`		`A = diagm([1; 2; 3])`

MANIPULATING VECTORS AND MATRICES

Operation	MATLAB	Python	Julia
Transpose	`A'`		`A'`
Concatenate horizontally	`A = [[1 2] [1 2]]` `A = horzcat([1 2], [1 2])`		`A = [[1 2] [1 2]]` `A = hcat([1 2], [1 2])`
Concatenate vertically	`A = [[1 2]; [1 2]]` `A = vertcat([1 2], [1 2])`		`A = [[1 2]; [1 2]]` `A = vcat([1 2], [1 2])`

FOUR

INPUT AND OUTPUT

Operation	MATLAB	Python	Julia
Opening a file	`fopen('file')`	`open('file')`	`open('file')`