MATLAB Cheat Sheet

César Zamudio

Assistant Professor, Marketing and Entrepreneurship, Kent State University

August 10, 2013

Some useful statements

- **help X**: See help file for function X. Always use it!
- **clear**: Deletes every variable in Workspace.
- **clc**: Clears all commands shown in Command Window.

Matrix generation

- **X=zeros(M,N)**: Creates X, a MxN matrix of zeros.
- **X=ones(M,N)**: Creates X, a MxN matrix of ones.
- **X=eye(N)**: Creates X, an identity matrix of size N.
- **X=rand(MxN)**: Creates X, a MxN matrix with [0,1] uniformly distributed numbers.
- **X=mu+sigma*randn(MxN)**: Creates X, a MxN matrix with normally distributed random numbers $N \sim (\mu, \sigma^2)$.
- **X= [A B ...; Z Y X ...]**: Concatenates the matrices shown in a larger matrix, X. Note that X itself can be part of the concatenation. The dimensions of the matrices must be consistent for concatenation to occur. New elements of a vector, or new rows/columns of a matrix can be added in this way.

Matrix operations and data manipulation

- **X(i,j)**: Refers to element (i, j) of a matrix X.
- **colon :**: The colon can be used in several ways:
  1. Create a continuous array (i.e. 1:5 is [1 2 3 4 5]).
  2. Nonunit spacing can be done through the form a:space:b, where a is the initial value, b the final value, and the term ”space” represents the unit jumps, i.e. 1:0.5:3 = [1 1.5 2 2.5 3].
  3. Placeholder for all rows or columns of a matrix. For example, X(:,1:3) returns a matrix composed of all rows of X and columns 1, 2 and 3 of X.

- **end**: Placeholder for last column (or row) of the matrix, e.g. X(1, end) accesses the cell located in X's first row and last column.
- **size(X,i)**: Returns the number of rows and columns of X. If $i = 1$, only the number of rows is returned; if $i = 2$, the number of columns is returned. If i is not specified, a vector with both rows and columns is returned.
- **X(:,[a b c])**: There are two uses for this expression. Assume X is a 4x4 matrix. Then:
  1. If the brackets are exhaustive, then X's columns or rows are reordered. For example, X(:,[2 1 3 4]) reorders all four of X's columns.
  2. If the brackets are NOT exhaustive, then a reduced matrix is created with the specified number of columns or rows. E.g. Y=X(:,[1 2]) creates Y, a 4x2 matrix with X's first two columns and all rows.
- **X'**: The apostrophe transposes the matrix X.
- **flipdim(X,i)**: Flips a matrix. If $i = 1$ the flip is vertical; if $i = 2$ flip is horizontal. See **fliplr** also.
- **X(i,j)= [ ]**: Deletes element (i, j) in matrix X. Including a colon instead deletes a full row or column.
- **repmat(X,i,j)**: Copies or "repeats" matrix X in i rows and j columns. Useful to create a large matrix consisting of blocks of identical matrices.
- Note that operations beginning with a dot (.) indicate element-by-element operation (such as .*, or ./, etc.). Operations without a dot are matrix algebra operations. For data manipulation, dots should be used in most cases.
- Summation or subtraction of a scalar from a vector or matrix operates on *all* elements of said vector or matrix. For instance, the operation X-5 subtracts 5 from *every* element of X.

Descriptives

- **min(X)**: For vectors, returns the minimum among all vector elements. For matrices, reports the minimum of each column. Use min(min(X)) to obtain the minimum of a matrix.
- **max(X)**: Returns maximum. Works in the same way as min(X).
- **mean(X)**: Computes mean. Work as min(X).
- **std(X)**: Computes standard deviation. Work as min(X).
- **sum(X,i)**: Returns the sum of the rows and columns of X. Works in a similar way to size(X,i).

Matrix algebra

- **diag(X)**: Returns the diagonal elements of a matrix X.
- **triu(X,i)**: Returns the upper triangular matrix of X. If i=0, then the diagonal is returned. If i=1, then the diagonal is omitted. If i=2, the diagonal and the next diagonal are omitted, and so forth.
- **det(X)**: Returns the determinant of a matrix X.
- **rref(X)**: Returns the reduced echelon form of a matrix X.
- **inv(X)**: Returns the inverse of a matrix X. Singular matrices cannot be inverted.
- **eig(X)**: Returns the eigenvalues of a matrix X.

Comparison, search and logical operators

- **i==j**: Determine if two matrices or scalars are equal. If i is equal to j, a 1 is returned. Otherwise, a 0 is returned.
- **i∼=j**: Determine if two matrices or scalars are NOT equal. If i is NOT equal to j, a 1 is returned. Otherwise, a 0 is returned.
- **strcmp(a,b)**: Compares two strings. If a is the same as b, a 1 is returned. Otherwise, a 0 is returned.
- **find(X==i)**: Function used to find which elements in vector X satisfy condition i. Inequalities can also be used. Returns the *position* of the elements within the vector that satisfy condition i.

File input and output

- **fid=fopen('filename.txt')**: Opens a file for reading. Used mostly for text files or very large files. The object "fid" is used to view and access the elements of 'filename.txt'.
- **d=textscan(fid, cols, 'Delimiter', delimiter)**: Scans the text contained in the file opened with **fopen** and saves it into object d. In the term "cols", input the type of data contained in every column. For example, if you have three string columns, cols= %s%s%s. If you have three integer columns, cols=%d%d%d. If the data is space delimited, do not write 'Delimiter'. Otherwise, write 'Delimiter' followed by a comma and the data delimiter (e,g, '\t' for tab-delimited files, ',' for comma-delimited files).
- **fclose('filename.txt')**: Closes a file previously opened using **fopen**. Mostly used for text files.
- **d=load('filename.dat')**: Imports a textfile into MATLAB into object d. Mostly used for clean, numeric data or MATLAB .mat files.

Statistics Toolbox

- **randtool** calls the Random Number Generation tool.
- **disttool** calls the Distribution GUI tool.
- **dfittool** calls the Distribution Fitting tool.

Graphics

- **figure**: Opens a new, blank figure in MATLAB. Not very used.
- **hold on; hold off**: The hold on statement makes each new plot request to be graphed on top of the other; hold off eliminates overlapping and, instead, graphics are replaced. Useful to plot multiple lines or scatters in a single figure.
- **axis([xlower xupper ylower yupper])**: This statement sets the values for a graph's axes.
- **xlabel/ylabel/title**: xlabel('foo') renames X's label to 'foo'. The functions ylabel and title are used in the same manner.
- **plot(X,Y)**: Plots line of vector Y versus vector X.

- **subplot(M,N,i)**: Creates a set of plots of size MxN, and allows to plot on quadrant i. The quadrant is determined by counting from left to right and top to bottom. For example, subplot(3,3,4) creates a 3x3 plot set, and accesses the subplot located at (2,1). This is useful for displaying multiple plots simultaneously.
- **bar(X)**: If X is MxN, draws M groups of N bars with the data in X.
- **scatter(X,Y)**: Creates a scatterplot of vectors X and Y.
- **qqplot(X)**: Creates a Q-Q plot of vector X. Useful to visually test for normality.
- **hist(X)**: Creates a histogram of vector X.
- **[y,z] = hist(X)**: Saves data of vector X's histogram. y contains the frequency distribution and z contains the histogram bins.
- **ksdensity(X)**: Creates a kernel density plot of vector X.
- **[y,z] = ksdensity(X)**: Saves X's kernel density distribution data. y contains the densities at each of the 100 points specified in b.

Filetype conversions

- **cell2mat, mat2cell**: Converts cell objects to matrices and viceversa. Remember that items within a cell object are accessed with curly brackets { }, not regular brackets [ ].
- **num2str, str2num**: Converts numbers to strings and viceversa.
- **double2str, str2double**: Converts double-numbers to strings and viceversa. Useful if str2num or num2str do not work. See also int2str.

Other

- **length(find(X==a))**: Gives the number of times a appears in the vector X.
- **rem(x, div)**: Gives the remainder of the division $\frac{x}{div}$. Useful to discern between odd and even values.
- **floor(X)**: Truncates the values in X (removes the mantissa), that is, rounds towards$-\infty$.
- **ceil(X)**: Same as floor(X) but rounds towards$+\infty$.
- **round(X)**: Rounds the values in X to the nearest integers.
- **xlswrite('A.xls', X)**: Writes matrix X in Excel file "A.xls".
- **randperm(N)**: Generates a 1xN vector of permutations 1:N. This is useful to rearrange a matrix in a random manner.