Ls % list directory

Pwd % current directory

Eye(4) % identity matrix of size 4x4

[4 4 4; 5 5 5] % matrix of 2 rows by 3 columns

Load feature.dat % load file in data directory

Or Load(‘feature.dat’)

Size % dimensions of matrix

Length % the largest dimension of matrix

Who % details about a variable

Whos % all variables in the current scope

Clear featuresX

V = priceY(1:10)

Save hello.mat v;

Clear % delete all variables from workspace

Load hello.mat % load the previous saved variables in hello.mat (in binary)

Save hello.txt v – ascii % save as text

A(3,2) – the value in the element third row and second column of matrix A

A([1,3], :) % first and second row of A  
A(:,2) % second column of A

A(:,2) = [10;11;12] %assign the column vector to the second column of A

A(:) % put all elements of A into a single vector

C = [A B] % concatenate A and B horizontally

C = [A; B] % concatenate A and B vertically

% [A B] and [A,B] gives the same result

A.\*B % Element-wise multiplication

A.^2 % Element-wise power

1./A

Log(v) % Element-wise

Exp(v) % Element-wise

Abs(v) % Element-wise

V+1 % element-wise matrix addition

A’ % transpose of A

Val = max(a)

[val,ind] = max(a) % index

Find(a<3) % list all elements with 3 or less

A = magic(3) % fun way to generate 3x3 matrix, each row column and diagonal sum to the same

Sum(a) % add all elements of a

Floor(a) % round to lowest int

Ceil(a) % round to highest int

Rand(3) % generate a 3x3 random matrix

Max(A,[],1) % per row maximum

Max(A,[],2) % per column maximum

Max(max(A)) % max of element A of…

Sum(A,1) % per column sum

Sum(A,2) % per row sum

Eye(9)

A.\*eye(9) % diagonalize A

Sum(sum(A.\*eye(9)) % sum of diagonal

Sum(sum(A.\*flipud(9)) % sum of diagonal of a flipped up down matrix

Pinv(A) %pseudo inverse matrix

Plotting Data

T=[0:0.01:0.98]

Y1 = sin(2\*pi\*4\*t)

Y2= cos(2\*pi\*4\*)

Plot(t,y1)

Hold on; % hold on plot one on top of another

Plot(t,y2)

Xlabel(‘time’)

Ylabel(‘value’)

Legend(‘sin’,’cos’)

Title(‘my plot’)

Cd ‘C:\user\...: print -dpng ‘myPlot.png’

Close %close any open figure

Figure (1); plot(t,y1)

Figure (2); plot(t,x1)

Subplot(1,2,1); %divide plot into a 1x2 grid, access first element

Subplot(1,2,1); %divide plot into a 1x2 grid, access second element

Axis([0.5 1 -1 1]) % label axis

Imgesc(A) % plot the value of A in colors

Imgesc(A), colorbar, colormap gray;

A=1,b=2,c=3 %Comma chaining function call, will print, whereas ; won’t

Control Statements: For, while, if statements, and functions

v = zeros (10,1)

for i =1:10,

v(1) = 2^i

end;