**WEEK 2 Octave/Matlab Tutorial**

1. **Moving Data Around**

|  |  |  |
| --- | --- | --- |
| command | 설명 | 예시 & 부가설명 |
| size(A) | row column size |  |
| size(A,1) | row size |  |
| size(A,2) | column size |  |
| length(A) | longer dimension size | confusing하므로 vector에서만 사용 |
| pwd | 현재 디렉토리 |  |
| cd ‘경로’ | 경로 변경 | cd ‘C:\Users\dltndls’ |
| ls | 현재 디렉토리에 있는 디렉토리&파일 |  |
| load 파일명 | featuresX variable에 파일 불러옴 | load featuresX.dat |
| who | variables 목록 in the current scope |  |
| whos | Attr/Name/Size/Bytes/Class |  |
| clear | get rid of variable  모든 변수 삭제 | clear featuresX  clear |
| v = priceY(1:10) | 10개의 column만으로 새로운 변수 구성 | priceY는 47x1 dimension |
| save | hello 파일에 v 변수 저장  save as text (ASCII) | save hello.mat v;  save hello.txt v -ascii |
| A(3,2) | 요소 추출 | A=[1 2; 3 4; 5 6]이면 ans=6 |
| A(2,:)  A(:,2) | “:” means every element along that row/col | ans=3 4  ans=2 / 4 / 6 |
| A([1 3], :) | 1 row의 모든 col, 3 row의 모든 col | ans = 1 2 / 5 6 |
| A(:,2)=[10;11;12] | assign new values | ans = 1 10/ 3 11/ 5 12 |
| A=[A, [100;101;102]] | append another col vector to right | A= 1 10 100/ 3 11 101/ 5 12 102 |
| A(:) | put all elements of A into a single vector | A= 1/3/5/10/11/12/100/101/102 |
| C=[A B] | A 옆에 B 붙이기 | A=1 2/ 3 4, B=11 12/ 13 14  C= 1 2 11 12/ 3 4 13 14 |
| C=[A; B] | A 밑에 B 붙이기  “;” means bottom of that | C=1 2/ 3 4/ 11 12/ 13 14 |

1. **Computing on Data**

|  |  |  |
| --- | --- | --- |
| command | 설명 | 예시 & 부가설명 |
| A\*B | matrix multiplication |  |
| A .\* B | 각 요소를 곱하기 | A= 1 2/ 3 4, B=11 12/ 13 14  ans= 11 24/ 39 56 |
| A .^ 2 | 각 요소 모두 제곱 | ans= 1 4/ 9 16/ 25 36 |
| 1 ./ v | 1에 각 요소를 나누기 | v=[1; 2; 3], ans= 1.0/ 0.5/ 0.33333 |
| log(v) |  |  |
| exp(v) |  |  |
| -v | -1\*v |  |
| abs(v) |  |  |
| A’ | transpose of A | A= 1 2/ 3 4/ 5 6, A’= 1 3 5/ 2 4 6 |
| (A’)’ | A |  |
| val=max(A) | A 요소 중 최대값 | a= [1 15 2 0.5], ans=15  A=[1 2; 3 4; 5 6;], ans=5 6 |
| [val, ind]=max(A) | value, index | val=15, ind=2 |
| max(A,[],1) | column의 max | ans=5 6 |
| max(A,[],2) | Per row maximum | ans=2;4;6 |
| max(max(A)) | 다 합쳐서 최대값= max(A(:)) | ans=6 |
| a < 3 |  | ans= 1 0 1 1 |
| find(a<3) |  | ans=1 3 4 |
| A=magic(n) | n by n matrix 생성  (대각선,row,col 합 모두 같다) | 모르면 help magic 쳐 볼 것 |
| [r,c]=find(A>=7) | r은 row, c는 col return  (help find 참조) | r=1;3;2, c=1;2;3  [1,1], [3,2], [2,3]이 7보다 크다. |
| sum(a)  sum(A,1)  sum(A,2) | 모든 요소의 합  모든 col의 합  모든 row의 합 | ans= 18.5  ans=369 369 369  ans = 369; 369; 369 |
| prod(a) |  |  |
| floor(a) | 버림 | ans=1 15 2 0 |
| ceil(a) | 올림 | 1 15 2 1 |
| rand(3) | random 3x3 matrix (0-1사이) |  |
| eye(n) | nxn의 I matrix | ans=1 0 0; 0 1 0; 0 0 1 |
| flipud(eye(3)) | 위아래 뒤집기 |  |
| pinv(A) | A의 inverse 구하기 |  |

1. **Plotting Data**

|  |  |  |
| --- | --- | --- |
| command | 설명 | 예시 & 부가설명 |
| PS1(‘>> ‘) | 명령행 >> 앞이 생략되어 표시된다. |  |
| t=[0:0.01:0.98] | 0부터 0.98까지 0.01씩 증가(99개 column) |  |
| y1=sin(2\*pi\*4\*t) |  |  |
| plot(t,y1); | t를 x로 하고 y1을 y로 하는 함수를 그려줌 |  |
| hold on | plot위에 이전 함수 고정 |  |
| plot(t,y2,’r’) | 고정된 함수 위에 다른 색으로 y2를 겹쳐 그림. | y2=cos(2\*pi\*4\*t) |
| xlabel(‘time’) | x, y축에 이름 붙이기 | ylabel(‘value’) |
| legend(‘sin’,’cos’) | 함수 색깔별로 이름 붙이기 | sin 파란선, cos 빨간선 |
| title(‘my plot’) | 표 이름 my plot |  |
| print -dpng ‘myPlot.png’ | 경로 설정하려면 앞에 cd ’path’;를 같은 줄에 | 그래프 png파일로 저장 |
| figure(1); | 창 이름 ’figure 1’ | figure(1); plot(t,y1);  figure(2); plot(t,y2); |
| subplot(1,2,1);  subplot(1,2,2); | divide plot a 1x2 grid, access first element  access 2th element | subplot(1,2,1); plot(t,y1);  subplot(1,2,2); plot(t,y2); |
| axis([0.5 1 -1 1]) | x축 0.5에서 1, y축 -1에서 1 |  |
| clf; | clear figure |  |
| imagesc(A) | matrix를 image로 보여줌  imagesc(A), colorbar, colormap gray; | A=magic(5) |

1. **Control Statements: for while, if statement**

|  |  |  |
| --- | --- | --- |
| command | 설명&결과 | 예시 & 부가설명 |
| v=zeros(10,1) | 10개의 행, 1개의 열에 모두 0 |  |
| for i=1:10,  v(i)=2^I;  end; | v= 2; 4; 8; 16; …; 1024; |  |
| indices=1:10;  for i=indeces,  disp(i);  end; | indicies= 1 2 3 4 … 10  1; 2; 3; 4; 5; … 10 |  |
| i=1;  while i<=5,  v(i)=100;  i=i+1;  end; | v = 100; 100; 100; 100; 100; 64; … 1024 |  |
| i=1;  while true,  v(i)=999;  i=i+1;  if i==6,  break;  end;  end; | v= 999; 999; 999; 999; 999; 64; …; 1024 |  |
| v(1)=2;  if v(1)==1,  disp(‘The value is one’);  elseif v(1)==2,  disp(‘The value is two’);  else  disp(‘The value is other’);  end; | The value is two |  |
| ①바탕화면 m 파일로 function 사용 | function y = squareThisNumber(x)  y = x^2; | squareThisNumber.m  (wordpad) |
| cd ‘내 Desktop 경로’  squreThisNumber(5) | ans= 25 |  |
| addpath(‘C:\Users\dltnd/Desktop’) | %Octave search path (advanced/optional)인데  addpath해주면 다른 pwd여도 사용가능. |  |
| ②파일로 function 사용, return 2개 | function [y1,y2] = squareAndCubeThisNumber(x)  y1=x^2;  y2=x^3; |  |
| [a,b]=squareAndCube…(5); | a=25 b=125 |  |
|  | function J=costFunctionJ(X, y, theta)  m=size(X,1);  predictions=X\*theta;  sqrErrors=(predictions-y) .^2;  J=1/(2\*m) \* sum(sqrErrors); | costFunctionJ.m  % X is the “design matrix” containing our training examples.  % y is the class labels  % m: num of training ex  % predictions of hypothesis on all m  % squared errors |
|  |  |  |
|  |  |  |
|  |  |  |