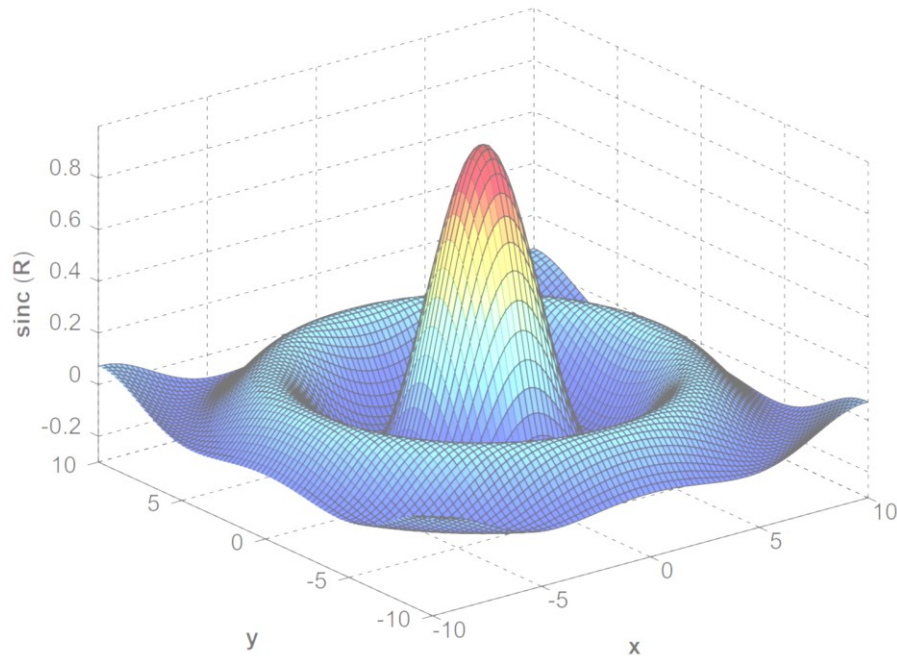


응용전산및실습 I

(강의자료 #3)



교과목명 : 응용전산 및 실습 I

담당교수 : 이 수 형

E-mail : soohyong@uu.ac.kr

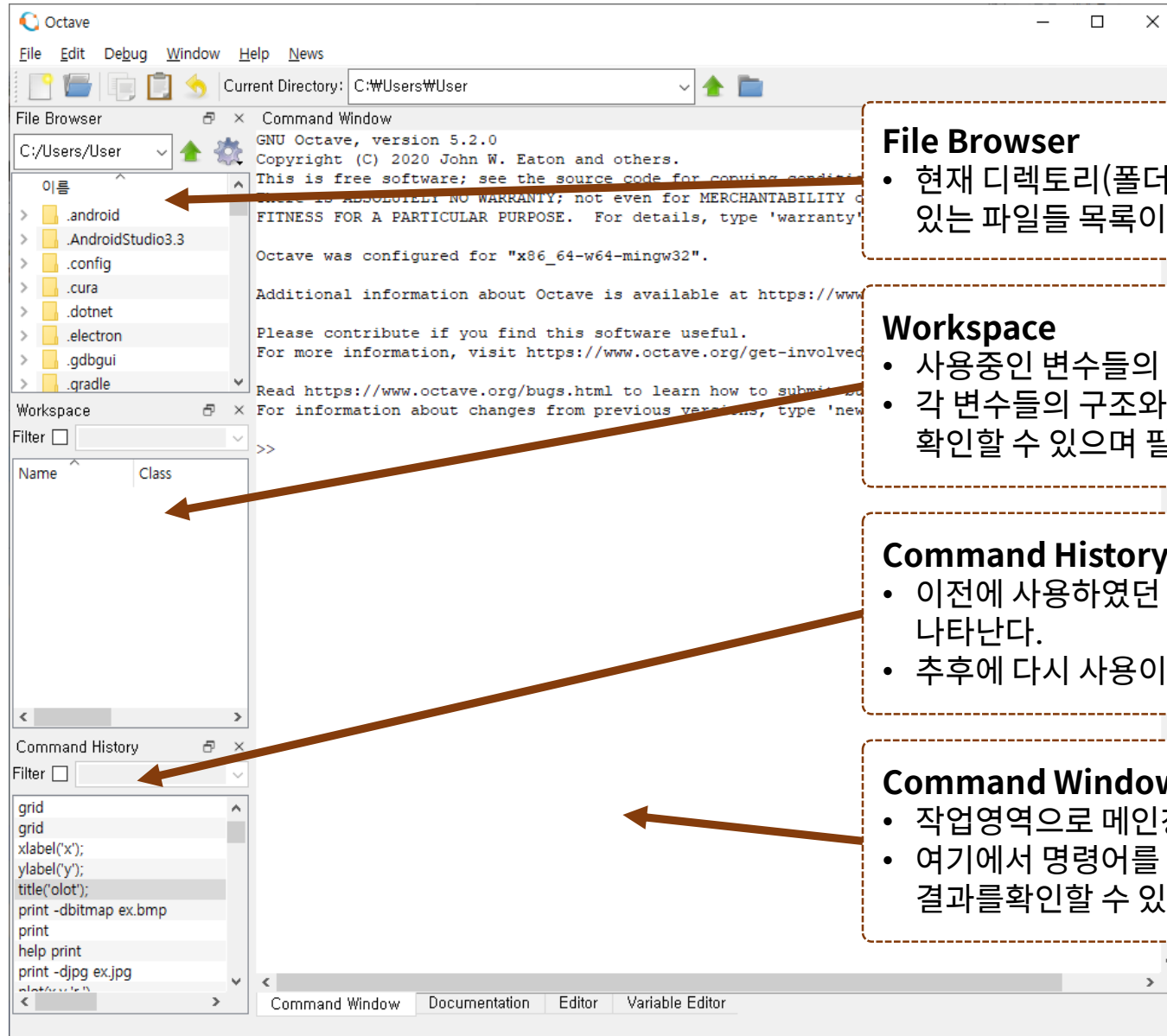
교재명 : 유인물

수강생 안내

- 본 교과목은 기본적으로 “대면”수업입니다.
- ‘코로나바이러스감염증-19’ 등의 이유로 대면수업을 받지 못하는 학생들을 위해 ‘비대면 실시간 수업’을 병행합니다. 기본적으로 대면 수업이므로 대면 수업에 참석한 학생들은 교실에서 출석을 체크하면 되며, 참석하지 못하고 ‘비대면 수업’을 듣는 학생들은 ‘실시간 온라인 강의’에 참석하면 출석이 반영됩니다.
- 수강생 여러분들은 강의 자료를 온라인 배포 및 게재 등의 행위를 할 시 저작권 문제가 발생할 수 있사오니 이에 유념하여 강의 자료를 공유하는 행위는 삼가 바랍니다.
- 교재는 따로 없으며, 유인물로 수업을 진행하니 각자 다운로드하여 보기 바랍니다.

지난시간에는?

- Matlab이란?
 - Matlab은 Matworks사의 테크니컬 컴퓨팅 (공학계산용) 언어.
 - MATLAB = MATriX + LABoratory : 행렬연산을 손쉽게
 - M-file이라는 형태로 일반적인 프로그래밍 언어로서의 사용이 가능.
- Matlab의 대안
 - 비대면 수업으로 인하여 고가의 상용제품인 Matlab대신 호환성이 뛰어난 Octave를 사용
 - Octave : 자유소프트웨어재단(FSF)에서 배포하는 무료 수치해석용 S/W



File Browser

- 현재 디렉토리(폴더)와 여기에 포함되어 있는 파일들 목록이 나타난다.

Workspace

- 사용중인 변수들의 목록이 나타난다.
- 각 변수들의 구조와 값을 포함한 정보들을 확인할 수 있으며 필터를 사용할 수 있다.

Command History

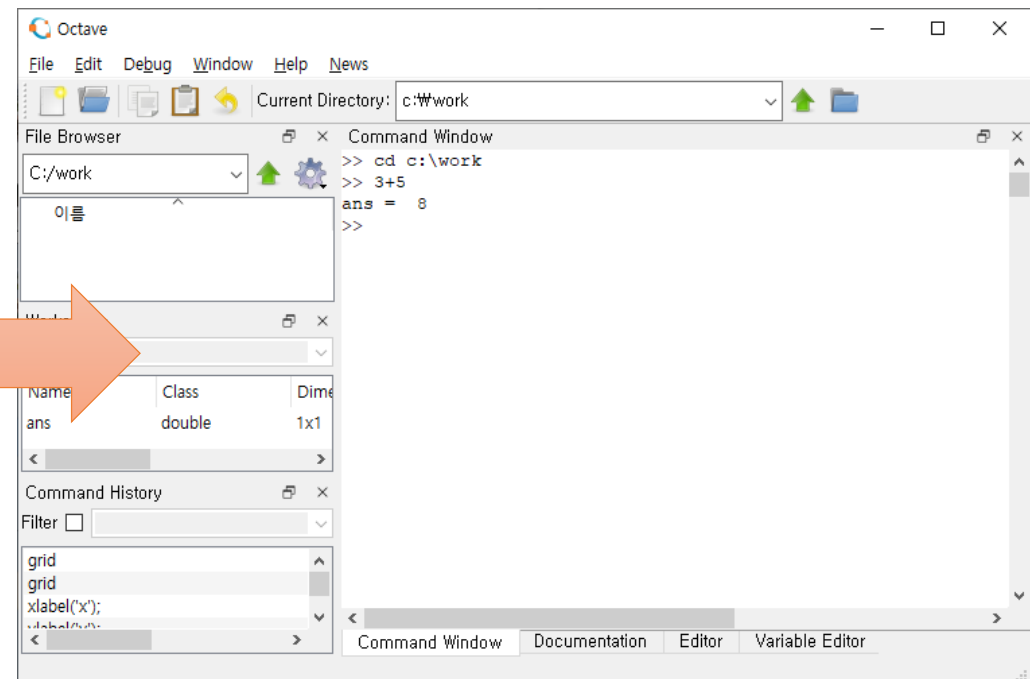
- 이전에 사용하였던 명령어/계산식들이 나타난다.
- 추후에 다시 사용이 가능하다.

Command Window

- 작업영역으로 메인창에 해당한다.
- 여기에서 명령어를 입력하고 수행 결과를 확인할 수 있다.

- 실습시에 작업영역은 C:\work로 지정한다.
 - `cd c:\work` → c:\work 디렉토리(폴더)로 현재의 폴더를 변경한다.
- 명령창(Command Window)에서 명령어 입력
 - `[>>]` 기호 : 명령어를 입력할 수 있는 대기상태 → 명령어 입력
 - `clc` (clear console screen) : 화면 지우기
 - `3+5` : 수식을 입력하면 수식을 계산

```
>> clc
>> cd c:\work
>> 3 + 5
ans = 8
>>
```



- 사칙연산자

Operation	Symbol	Example
Addition, $a + b$	+	$3 + 22$
Subtraction, $a - b$	-	$90 - 54$
Multiplication, $a \cdot b$	*	$3.14 * 0.85$
Division, $a \div b$	/ or \	$56/8 = 8 \backslash 56$
Exponentiation, a^b	^	2^8

- 일반적인 프로그래밍 언어와 유사

- 나누기 연산자

- / : right division (오른쪽 나눗셈) → 일반적인 실수에서는 동일한 결과, 행렬 연산에서 차이
- \ : left division (왼쪽 나눗셈)

- 변수(variable)란?

- 변수 : 하나의 이름을 가지는 기억장소, '=' 기호를 사용하여 값을 저장 (**a = 5**)
- 기억된 변수는 그 자체로 기억된 숫자로서 수식, 함수 등에 바로 사용
- 숫자 하나를 저장할 수도 있고, 벡터 또는 배열을 저장할 수도 있음

- 변수 이름 규칙

- 대소문자 구별 (Cost, cost, CoST, COST 는 서로 다른 변수)
- 최대 31글자까지 허용. 단, 최근 Matlab과 Octave에서는 63글자까지 허용
- 변수 이름은 문자, 숫자, 밑줄(_)이 사용된 하나의 단어로 구성되며, 문자로 시작
- 키워드는 변수로 사용할 수 없으며, 함수 이름은 변수로 사용하지 않도록 주의.

- 변수의 사용법

- 변수에 값을 지정하고자 할 때는 = 기호를 사용한다.

- 변수 a를 만들고 5를 대입한다.

- 변수 b를 만들고 4를 대입한다.

```
>> a = 5
a =
    5
>> b = 4
b =
    4
```

- 수식에 변수를 적으면 변수에 저장되어 있는 값이 사용된다.

- 기억되어 있는 a, b 값이 계산에 사용되어
a + b 는 5 + 4로 계산된다.

```
>> a + b
ans =
    9
>> a * b
ans =
   20
```

• 미리정의된 특별한 변수들

- Matlab에서는 시작할 때 미리 정의되어 있는 특별한 변수
- 값의 변경이 가능하나, 변경 후에는 원래의 의미가 사라짐

Special Variable	Description
ans	결과를 나타내는 기본 변수, 가장 최근에 계산된 결과를 저장하고 있음
pi	원주율 (π)
eps	인접한 두 수 사이의 최소값으로 이 숫자 이하로 가까이 있는 숫자들은 구분하지 못함 : 2.2204×10^{-16}
inf	무한대 ($1 / 0$)
i, j	허수를 표현하는데 사용되는 상수, $i = j = \sqrt{-1}$
realmin	저장할 수 있는 최소의 + 숫자 : 2.2251×10^{-308}
realmax	저장할 수 있는 최대의 + 숫자 : 1.7977×10^{308}

```
>> pi
ans =
    3.1416
>> eps
ans =
    2.2204e-16
>> inf
ans =
    Inf
>> i
ans =
    0.0000 + 1.0000i
>> j
ans =
    0.0000 + 1.0000i
>> realmin
ans =
    2.2251e-308
>> realmax
ans =
    1.7977e+308
```

복소수 (Complex Number)

- 복소수 사용법

Declaration
$a \pm bi, a \pm bj$

Related Function	Description
<code>abs(x)</code>	Magnitude of $x (= a \pm bi) = \sqrt{a^2 + b^2}$
<code>angle(x)</code>	Angle in radians of $x (= a \pm bi) = \tan^{-1}(b/a)$
<code>real(x)</code>	Real part of $x (= a \pm bi) = a$
<code>imag(x)</code>	Imaginary part of $x (= a \pm bi) = b$
<code>conj(x)</code>	Complex conjugate of $x (= a + bi) = a - bi$

```
>> a = sqrt(3) + 1j
a = 1.7321 + 1.0000i

>> m = abs(a)
m = 2.0000

>> theta = angle(a)
theta = 0.52360

>> degree = theta * 180 / pi
degree = 30.000

>> r = real(a)
r = 1.7321

>> im = imag(a)
im = 1
```

Exponential Function	Description
$\exp(x)$	Exponential
$\log(x)$	Natural logarithm
$\log_{10}(x)$	Base 10 logarithm
$\log_2(x)$	Base 2 logarithm and floating-point dissection
$\text{pow}_2(x)$	2^x
$\text{sqrt}(x)$	Square root

Trigonometric Function	Description
$\sin(x)$	Sine
$\cos(x)$	Cosine
$\tan(x)$	Tangent
$\csc(x)$	Cosecant of $x = 1 / \sin(x)$
$\sec(x)$	Secant of $x = 1 / \cos(x)$
$\cot(x)$	Cotangent of $x = 1 / \tan(x)$
$\text{asin}(x)$	Inverse sine of $x = \arcsin(x) = \sin^{-1}(x)$
$\text{acos}(x)$	Inverse cosine of $x = \arccos(x) = \cos^{-1}(x)$
$\text{atan}(x)$	Inverse tangent of $x = \arctan(x) = \tan^{-1}(x)$

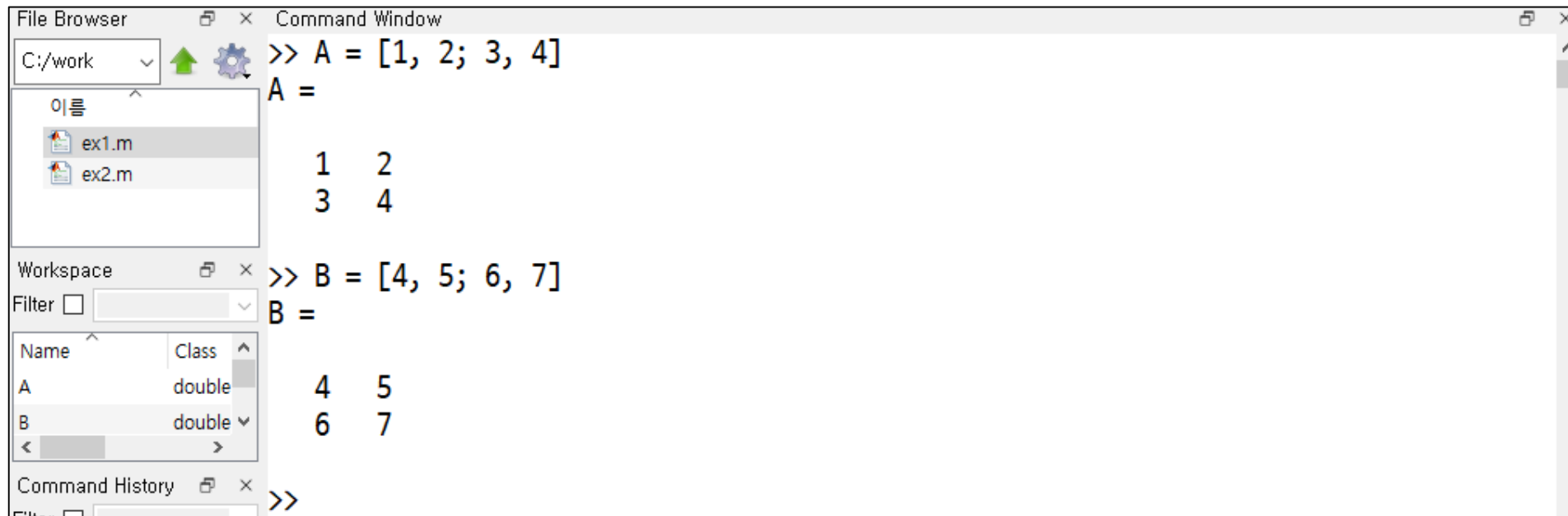
- M파일 : Matlab에서 사용하는 프로그램 저장용 파일
- 스크립트 파일
 - 명령어들을 미리 모아놓은 파일
 - 명령창에서 파일 이름 지정 → 저장된 명령어들을 순차적으로 실행

Keyword / Function	Meaning	Example
;	Suppress output	radius = 5;
%	Comment	radius = 5; % 반지름 값
disp(variable)	Display results without identifying variable names	disp(radius)
input	Prompt user for input	radius = input('반지름을 입력하세요> ');
pause	Pause until user presses any key	pause
fprintf(format, var1, var2,...)	Print msg	fprintf('계산이 완료되었습니다.');
		fprintf('반지름은 %f 입니다.\n', radius);

행렬(Matrix)의 사용

- 행렬의 정의

Keyword	Meaning
[]	배열의 시작과 끝
;	행 구분
,	열 구분



- 벡터(vector)
 - 1차원 행렬, 행벡터(row vector), 열벡터(column vector)

```
>> x = [1, 2, 3, 5]
```

```
x =
```

```
    1    2    3    5
```

```
>> y = [1; 3; 2]
```

```
y =
```

```
    1
```

```
    3
```

```
    2
```

```
>> y = [1, 3, 2]'
```

```
y =
```

```
    1
```

```
    3
```

```
    2
```


- 행렬의 계산

Operation	Symbol	Example
Addition, $a + b$	$+$	$A + B$
Subtraction, $a - b$	$-$	$A - B$
Multiplication, $a \cdot b$	$*$	$A * B$
Division, $a \div b$	$/$ or $\text{inv}()$	$A / B = A * \text{inv}(B)$
Exponentiation, a^b	$^$	$A ^ 2$
Transposition, a^T	$'$	A'

• 행렬의 계산

```
>> A = [1,2; 3,4]
```

```
A =
```

```
     1     2
     3     4
```

```
>> B = [5,6; 7,8]
```

```
B =
```

```
     5     6
     7     8
```

```
>> C = A + B
```

```
C =
```

```
     6     8
    10    12
```

```
>> D = A - B
```

```
D =
```

```
    -4    -4
    -4    -4
```

```
>> H = A * B
```

```
H =
```

```
    19    22
    43    50
```

```
>> G = A ^ 2
```

```
G =
```

```
     7    10
    15    22
```

```
>> E = A / B
```

```
E =
```

```
    3.0000   -2.0000
    2.0000   -1.0000
```

```
>> F = A * inv(B)
```

```
F =
```

```
    3.0000   -2.0000
    2.0000   -1.0000
```

```
>> J = A - 3
```

```
J =
```

```
    -2    -1
     0     1
```

행렬연산의 응용

- 선형방정식의 해 구하기

$$\begin{aligned}x_1 - x_2 + 2x_3 &= 10 \\ 3x_1 + 2x_2 + 9x_3 &= 9 \\ x_2 - 4x_3 &= -3\end{aligned}$$



$$\begin{bmatrix} 1 & -1 & 2 \\ 3 & 2 & 9 \\ 0 & 1 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 10 \\ 9 \\ -3 \end{bmatrix}$$



$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 & -1 & 2 \\ 3 & 2 & 9 \\ 0 & 1 & -4 \end{bmatrix}^{-1} \begin{bmatrix} 10 \\ 9 \\ -3 \end{bmatrix}$$

```
>> A = [1, -1, 2; 3, 2, 9; 0, 1, -4]
```

```
A =
```

```
     1     -1      2
     3      2      9
     0      1     -4
```

```
>> B = [10; 9; -3]
```

```
B =
```

```
    10
     9
    -3
```

```
>> x = inv(A) * B
```

```
x =
```

```
    6.4783
   -4.0435
   -0.2609
```

연습

- 원소가 $8, 10/4, 12 \times 1.4, 0.15, \sin 60^\circ, \sqrt{12}$ 인 행벡터를 생성하라.
- 다음 행렬 A 와 B 를 정의하고, $A + B, A - B, AB, A \cdot B^{-1}, A^2$ 를 계산하는 스크립트를 작성하라.

$$\mathbf{A} = \begin{bmatrix} 3 & 2 & 9 \\ 1 & -1 & 0 \\ 2 & 1 & 5 \end{bmatrix}, \mathbf{B} = \begin{bmatrix} -1 & 3 & 9 \\ 2 & 6 & -2 \\ 6 & 8 & 5 \end{bmatrix}$$

행렬연산자

- 스칼라와 행렬의 연산

Element-By-Element Operation	Representative Data $a = [a_1, a_2, \dots, a_n], b = [b_1, b_2, \dots, b_n], c = \langle a \text{ scalar} \rangle$
Scalar Addition	$a + c = [a_1 + c, a_2 + c, \dots, a_n + c]$
Scalar Multiplication	$a * c = [a_1 * c, a_2 * c, \dots, a_n * c]$
Array Addition	$a + b = [a_1 + b_1, a_2 + b_2, \dots, a_n + b_n]$
Array Multiplication	$a .* b = [a_1 * b_1, a_2 * b_2, \dots, a_n * b_n]$
Array Division	$a ./ b = \left[\frac{a_1}{b_1}, \frac{a_2}{b_2}, \dots, \frac{a_n}{b_n} \right]$
Array Exponentiation	$a .^c = [a_1^c, a_2^c, \dots, a_n^c]$
	$c .^a = [c^{a_1}, c^{a_2}, \dots, c^{a_n}]$
	$a .^b = [a_1^{b_1}, a_2^{b_2}, \dots, a_n^{b_n}]$

행렬연산자

- 스칼라와 행렬 연산

$$- A = \begin{bmatrix} 2 & 4 \\ 5 & 6 \end{bmatrix}, B = \begin{bmatrix} 3 & 4 \\ 5 & 7 \end{bmatrix}, C = 4$$

$$- \begin{bmatrix} 2 * 3 & 4 * 4 \\ 5 * 5 & 6 * 7 \end{bmatrix}$$

$$- \begin{bmatrix} 2/3 & 4/4 \\ 5/5 & 6/7 \end{bmatrix}$$

$$- \begin{bmatrix} 2 * 4 & 4 * 4 \\ 5 * 4 & 6 * 4 \end{bmatrix}$$

특수 행렬 만들기

- 자주 사용되는 특수 행렬의 생성

Keyword	Description
ones(n) ones(m, n)	Create n x n arrays containing all ones Create m x n arrays containing all ones
zeros(n) zeros(m, n)	Create n x n arrays containing all zeros Create m x n arrays containing all zeros
eye(n, n) eye(m, n)	Produce n x n identity matrices Produce m x n identity matrices
rand(n) rand(m, n)	Uniformly distributed n x n random arrays Uniformly distributed m x n random arrays
randn(n) randn(m, n)	Zero-mean, unit-variance normal distribution n x n, m x n random arrays
size(A)	Return row & column size

특수 행렬

- 특수 행렬 만들기 연습

```
>> ones(3)
ans =
     1     1     1
     1     1     1
     1     1     1
```

```
>> zeros(2)
ans =
     0     0
     0     0
```

```
>> zeros(2,5)
ans =
     0     0     0     0     0
     0     0     0     0     0
```

```
>>
```

```
>> eye(2)
ans =

Diagonal Matrix

     1     0
     0     1
```

```
>> eye(2,4)
ans =

Diagonal Matrix

     1     0     0     0
     0     1     0     0
```

```
>>
```

```
>> rand(2)
ans =
     0.22906     0.12572
     0.84617     0.42582
```

```
>> randn(2,4)
ans =
    -1.25628     0.55138     0.91693    -1.39879
    -0.49975     1.60643     1.50290    -1.12196
```

```
>> A=[1,2,3; 4,5,6]
A =
     1     2     3
     4     5     6
```

```
>> size(A)
ans =
     2     3
```


출석 과제

- 강의중에 Octave를 이용하여 연습했던 내용들을 따라하고, 수행한 화면들을 캡처해서 제출하시오.