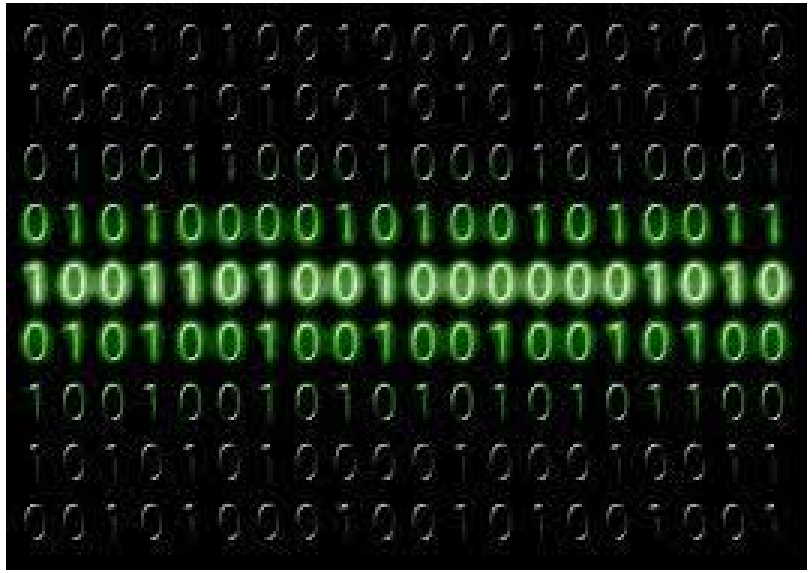
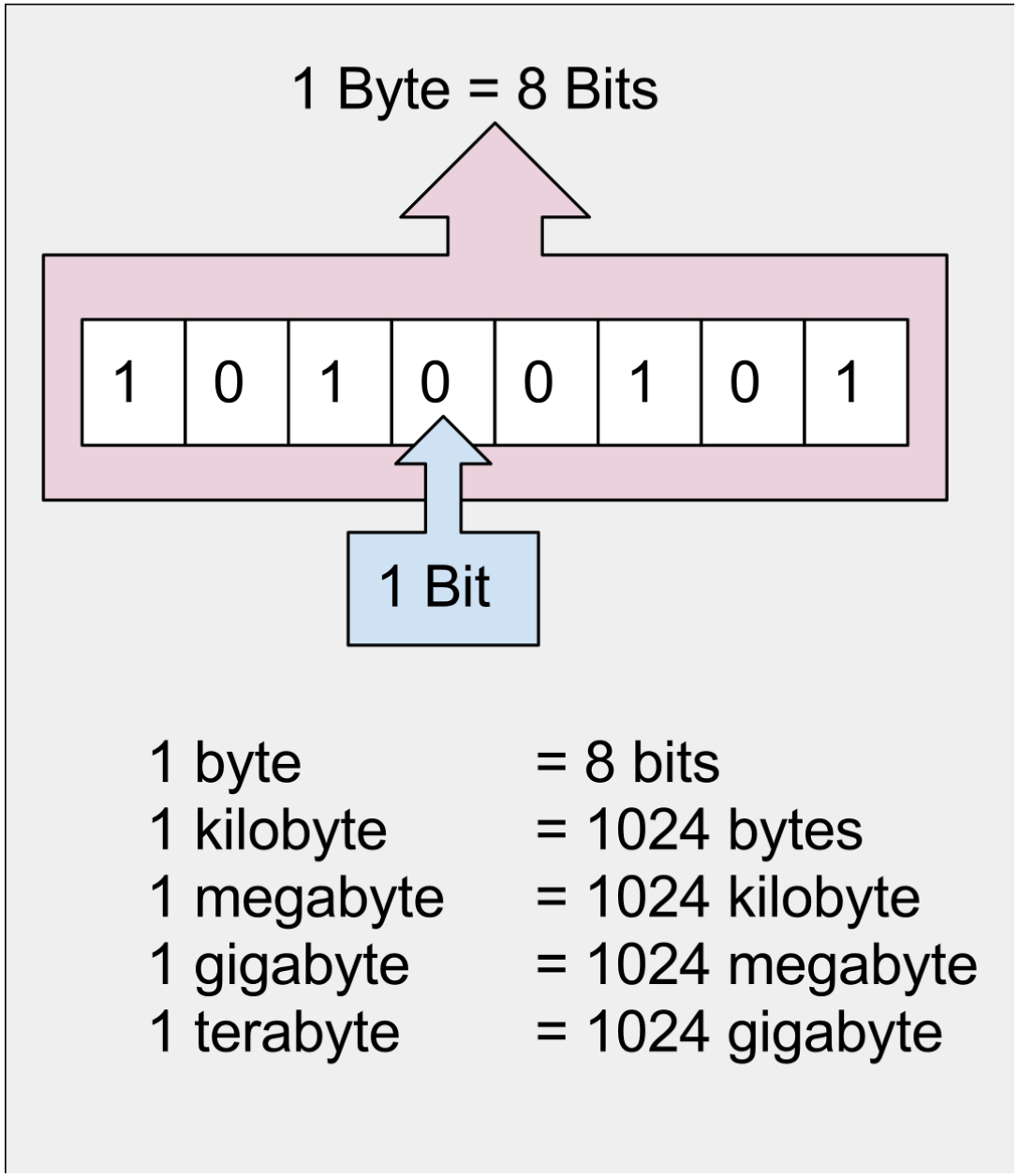
دانشکده روانشناسی و علوم تربیتی

MATLAB for Brain and Cognitive Psychology (Matrix algebra)

Presented by:
Ehsan Rezayat, Ph.D.
Faculty of Psychology and Education, University of Tehran,
Institute for Research in Fundamental Sciences (IPM), School of Cognitive Sciences,
emails: rezayat@ut.ac.ir, rezayat@ipm.ir, erezayat.er@gmail.com

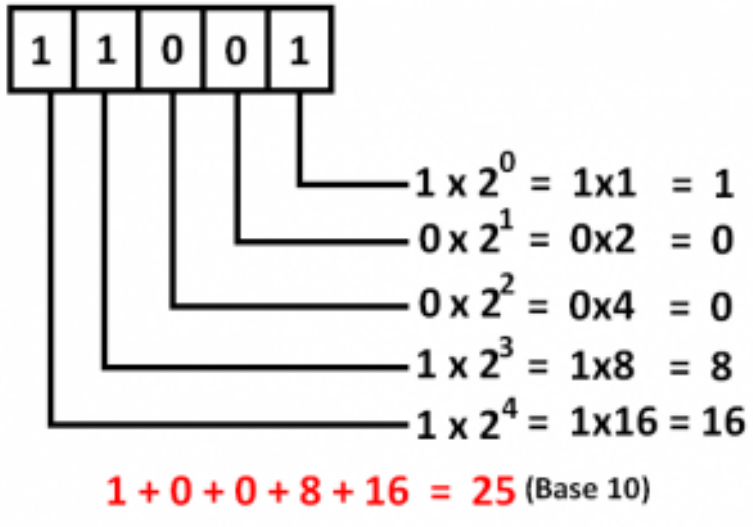
Bit &Byte





1 byte	= 8 bits
1 kilobyte	= 1024 bytes
1 megabyte	= 1024 kilobyte
1 gigabyte	= 1024 megabyte
1 terabyte	= 1024 gigabyte

Bit &Byte



Bit-Depths Converted To Potential Gray Tones and Colors

Bits Per Color	Log Formula (power of 2)	Monochrome Grayscale Values	Potential R,G,B Color Values
1-bit	$2^1 =$	2	8
2-bit	$2^2 =$	4	64
3-bit	$2^3 =$	8	512
4-bit	$2^4 =$	16	4096
6-bit	$2^6 =$	64	262144
8-bit	$2^8 =$	256	16.77 Million
10-bit	$2^{10} =$	1024	1.07 Billion
11-bit	$2^{11} =$	2048	8.59 Billion
12-bit	$2^{12} =$	4096	68.72 Billion

Variable types

Numbers

- integer: no decimal places
  345
- double: floating point number
  3.24
- Boolean : true false

```
>> N  = 64;
N1 = 385;
N2 = 1276;


>> Flt    = 54.97;
Fn123J = .7;
A_d2   = 3.2;

>> T = true;
F = false;
```

Some example of wrong Variable Name

- 13f = 34
- _temp = 3.7
- $fs = 3
- ?var = 100

```
>>13f   = 34
_temp = 3.7
$fs   = 3
?var  = 100

>>
 13f   = 34
   |
Error: Unexpected MATLAB expression.
```

Vectors and matrices

- Vectors are like lists

  Confidence_level = [1,2,3,4,5]

- Matrices are like lists of lists

  Confidence_level = [1,3,5,7;
  2,4,6,8]

- Matrices can have many dimensions

```
>> Confidence_level = [1 2 3 4 5]
Confidence_level =
     1     2     3     4     5

>> Confidence_level = [1,2,3,4,5]
Confidence_level =
     1     2     3     4     5

>> Confidence_level = [1:5]
Confidence_level =
     1     2     3     4     5
```

Also try:
x = [1:0.1:10]

Creating matrices

- Matrix with 1 number
- Matrix with 0 number
- Matrix with random number
- Matrix with null number

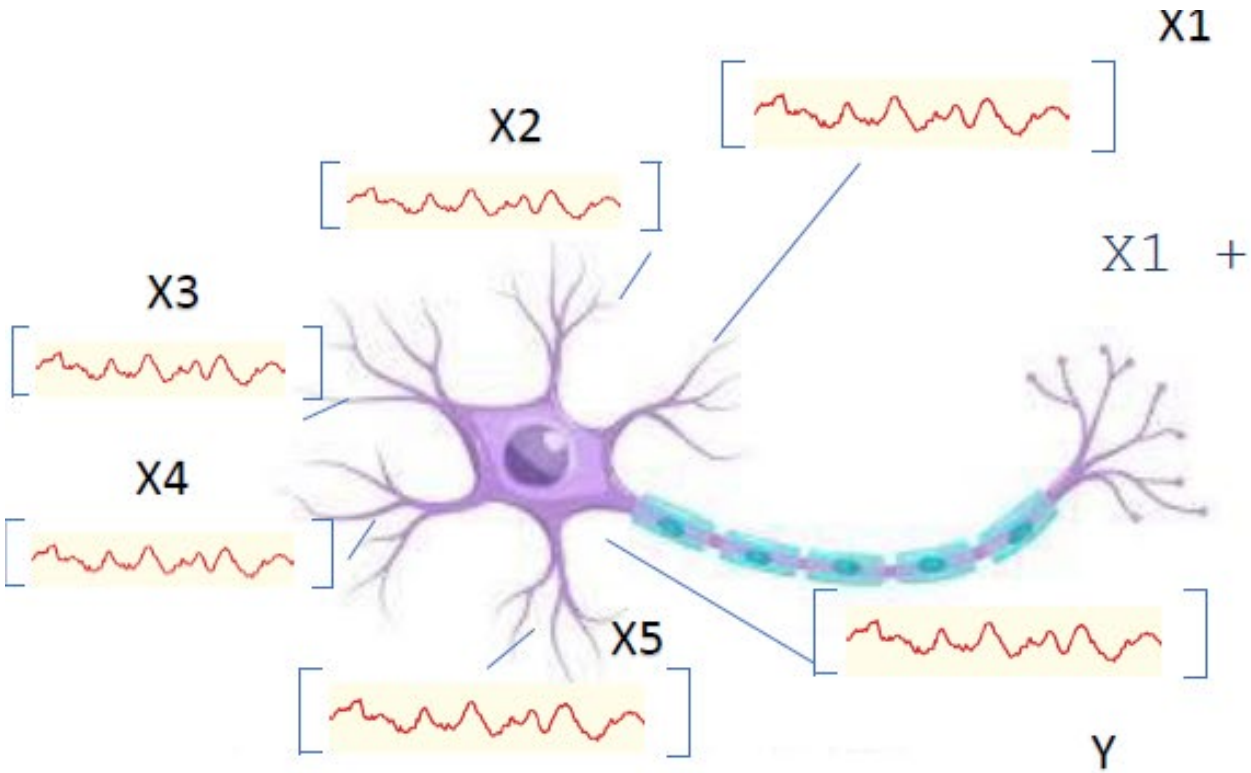
```
>> ones(3)
ans =
     1     1     1
     1     1     1
     1     1     1

>> ones(2,3)
ans =
     1     1     1
     1     1     1

>> zeros(3,4)
ans =
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

Describing matrices

- size() will tell you the dimensions of a matrix
- length() will tell you the length of a vector

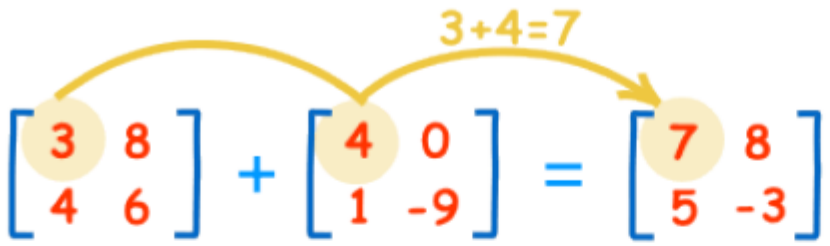
- Accessing elements in a matrix

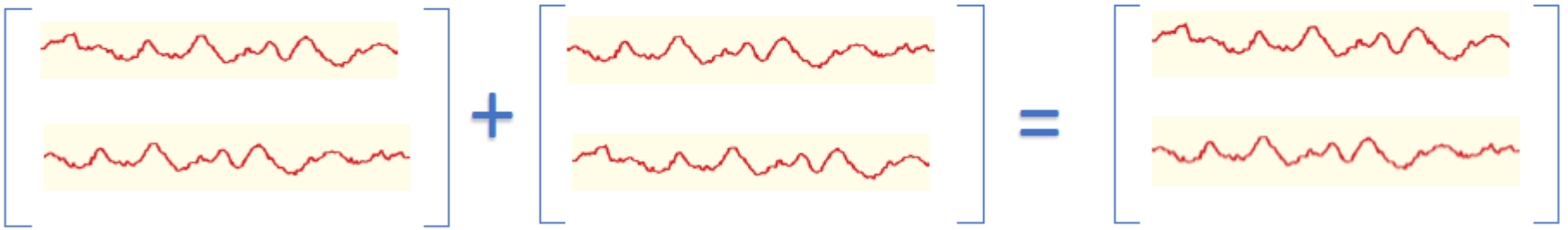
Matrix math: Addition



$$X1 + X2 + X3 + X4 + X5 = Y$$

Matrix math: Addition

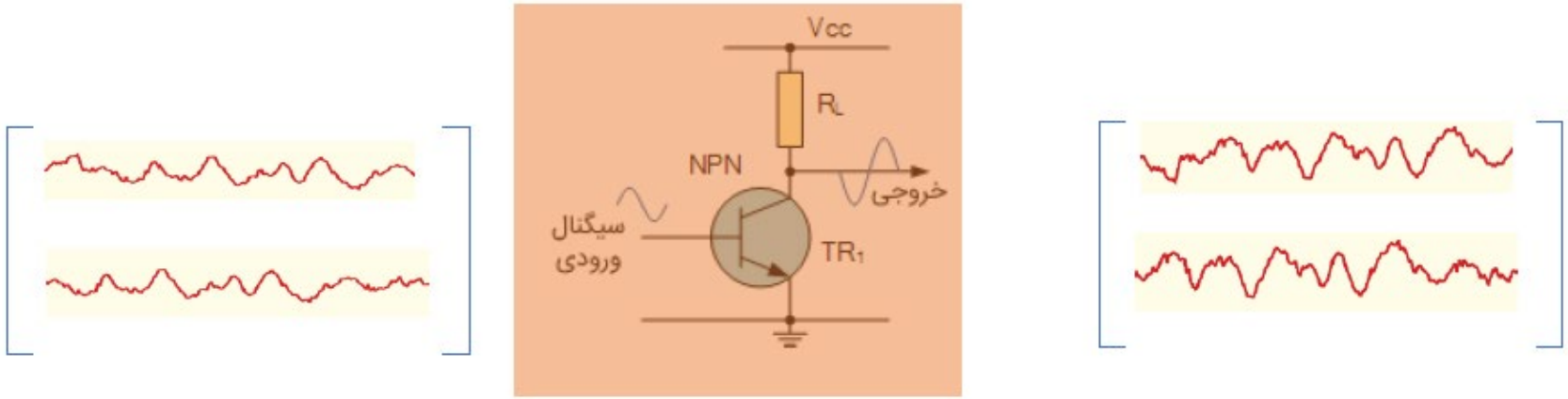




A + B = C

```
>> b = [5 1 5]
b =
     5     1     5
>> a + b
ans =
     6     3     8
```

Matrix math



A × a = B

```
>> a = [1 2 3]
a =
     1     2     3
>> a * 10
ans =
    20    30    40
```

Vector multiplication

- The * sign refers to *matrix multiplication:*

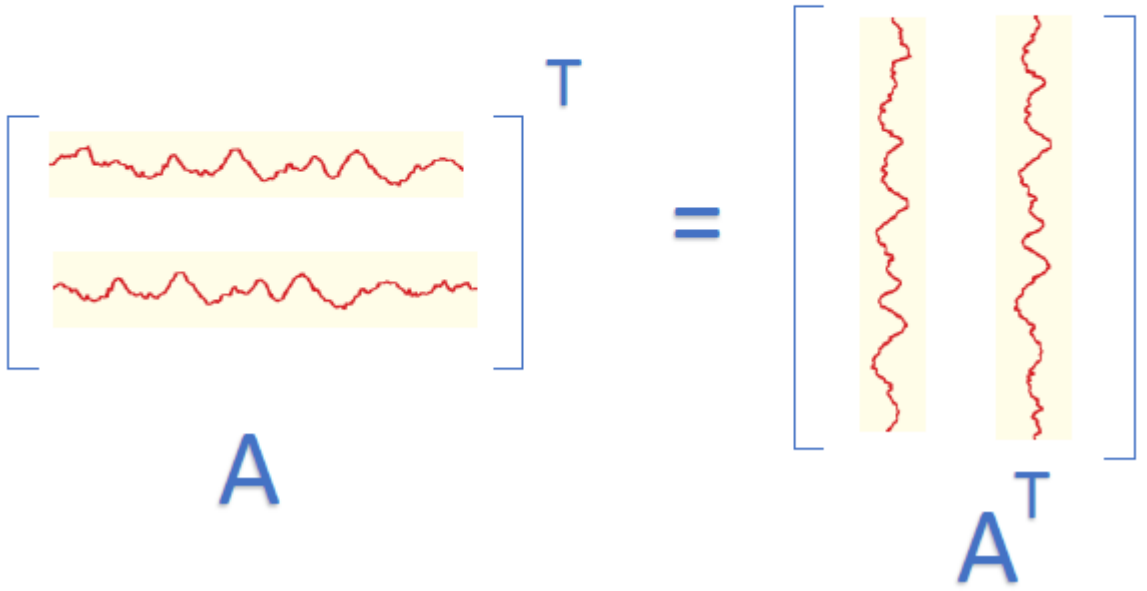
```
>> a = [1 2 3]
a =
     1     2     3
>> b = [2 2 4]
b =
     2     2     4
>> a * b
Error using  *
Inner matrix dimensions m
>> b = b'
b =
     2
     2
     4
>> a * b
ans =
    18
```

transposing a matrix:
use ‘ to transpose, i.e. flip rows and columns

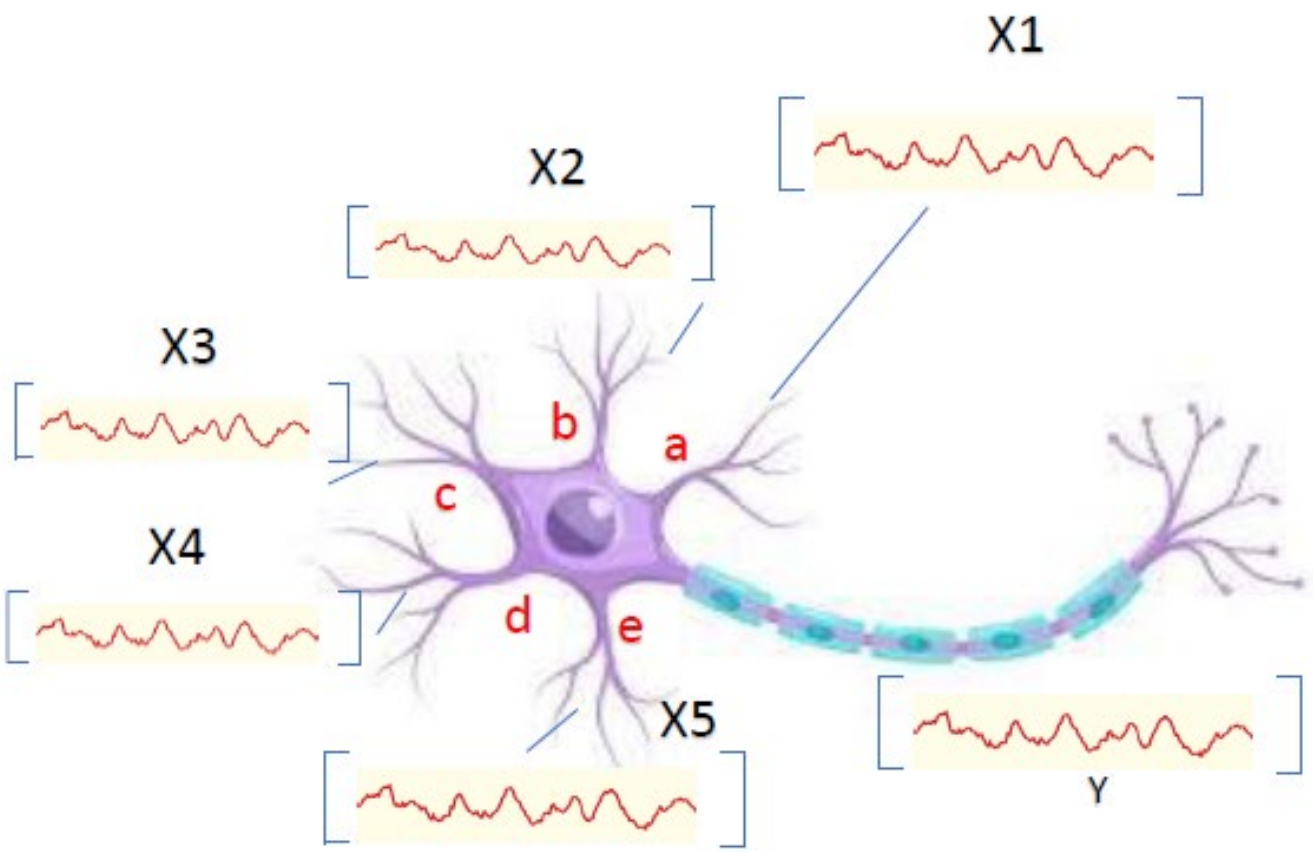
Matrix math

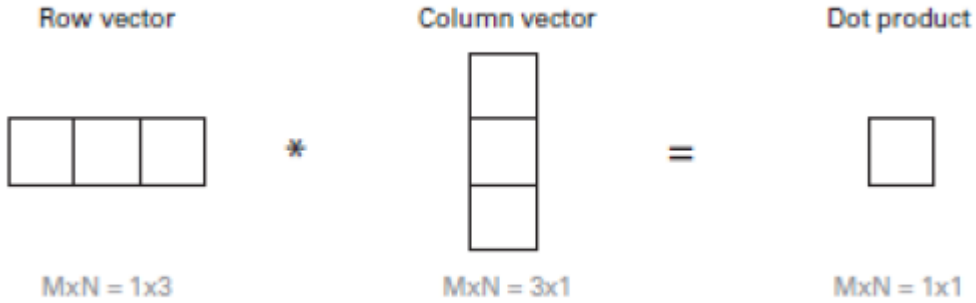
- Transpose

$$\begin{bmatrix} 6 & 4 & 24 \\ 1 & -9 & 8 \end{bmatrix}^T = \begin{bmatrix} 6 & 1 \\ 4 & -9 \\ 24 & 8 \end{bmatrix}$$



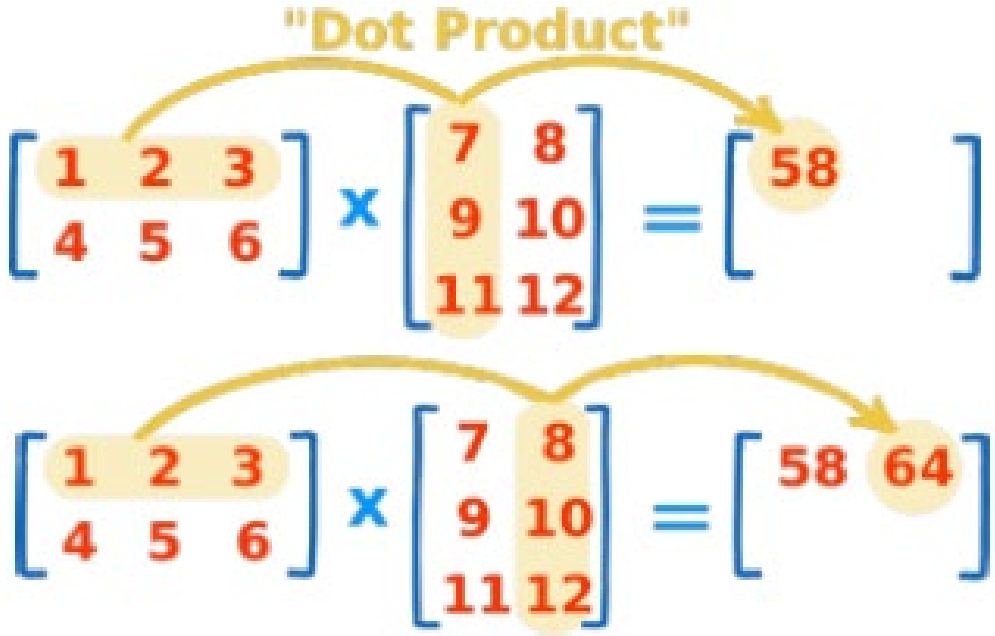
Matrix math: Dot product

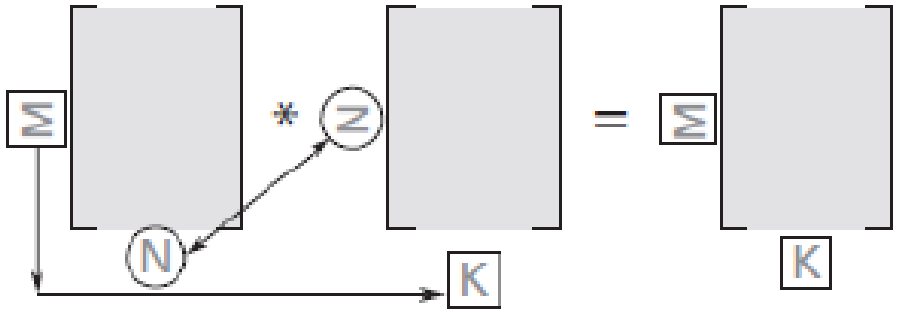




$$[X1 \quad X2 \quad X3 \quad X4 \quad X5] \times \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} = Y$$

Matrix math: Multiplication





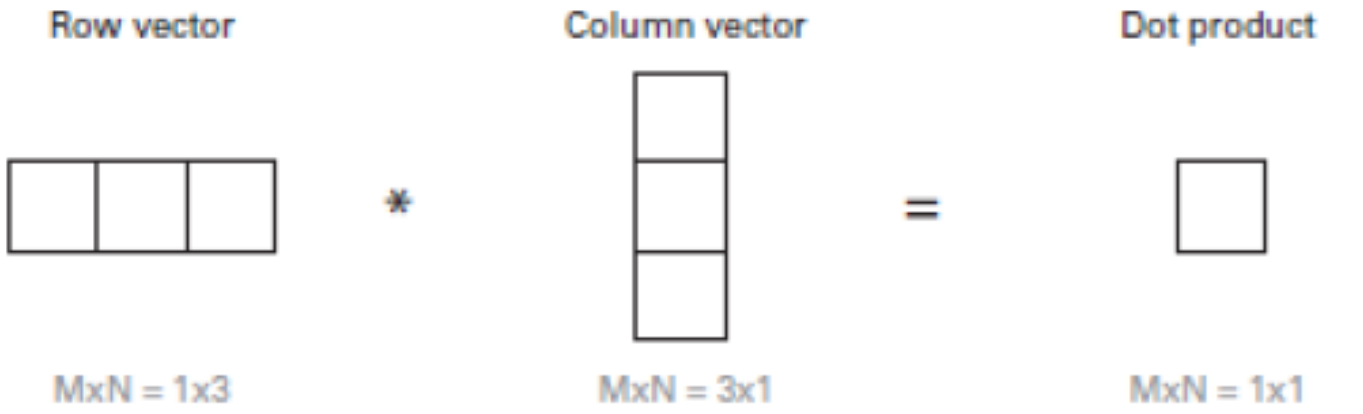
$$\begin{bmatrix} a & e \\ b & f \\ c & g \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} = x \begin{bmatrix} a \\ b \\ c \end{bmatrix} + y \begin{bmatrix} e \\ f \\ g \end{bmatrix} = \begin{bmatrix} xa + ye \\ xb + yf \\ xc + yg \end{bmatrix}$$

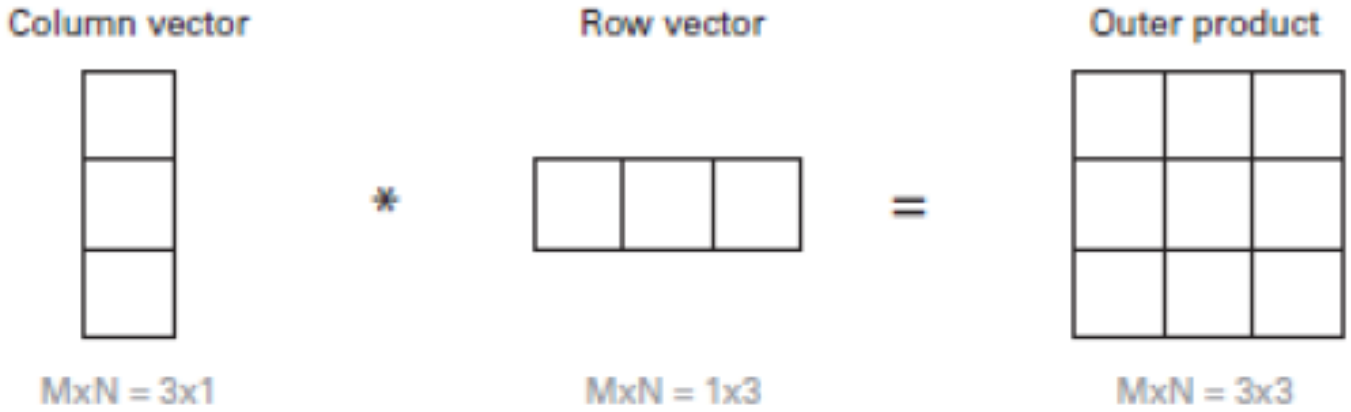
3x2 2x1 3x1

$$\begin{bmatrix} a & e \\ b & f \\ c & g \end{bmatrix} * \begin{bmatrix} x & w \\ y & z \end{bmatrix} = \begin{bmatrix} xa + ye & wa + ze \\ xb + yf & wb + zf \\ xc + yg & wc + zg \end{bmatrix}$$

3x2 2x2 3x2

Matrix math: Multiplication





Operators

- Element-wise operators:
  - .* multiplication
  - ./ division
  - .^ exponentiation
- Many other functions work element-wise, e.g.:

Vector multiplication

- The .* sign refers to *element-wise multiplication:*

```
>> a = [1 2 3]
a =
     1     2     3
>> b = [2 2 4]
b =
     2     2     4
>> a .* b
ans =
     2     4    12

>> a * 4
ans =
     4     8    12
>> a .* 4
ans =
     4     8    12
```

Working with strings

- Strings in MATLAB are vectors of characters
- Always use single quotes to define strings

```
>> name = 'Jonas'
name =
Jonas
>> name(1)
ans =
J
>> name(1:3)
ans =
Jon
```

Working with strings

```
>> x = 'abc'
x =
abc
>> y = 'def'
y =
def
>> x + y
ans =
   197   199   201
>> double('a')
ans =
    97
>> double('d')
ans =
   100
>> char(97)
ans =
a
```

Working with strings

```
>> strcat(x,y)
ans =
abcdef
>> newstring = strcat(x,y)
newstring =
abcdef
>> newstring = strcat(x,y);
>>
```

results stored in new variable

semicolon suppresses output of results

Formatting strings

- What do we typically do with strings?
  - Printing out messages to the workspace
  - Printing out data or messages to files
  - Using them as stimuli in an experiment
  - Using them as filenames, codes, or identifiers

Formatting strings

- Several ways to print a string out to the workspace:
  - type the name of the variable w/o a trailing semicolon
  - disp() is almost the same as above, except it does not print out the variable name
  - fprintf() is for formatting text and printing out to a file or other device, such as the workspace
  - sprintf() is for formatting text in order to create new string variables

Working with strings

```
>> name = 'Fred';
>> name
name =
Fred
>> disp(name)
Fred
>> fprintf(name)
Fred>> sprintf(name)
ans =
Fred
```

notice the lack of newline character

Formatting strings

- fprintf() is a very powerful command for formatting strings, combining them, and printing them out

```
>> help fprintf
 fprintf Write formatted data to text file.
    fprintf(FID, FORMAT, A, ...) applies the FORMAT to all elements of
    array A and any additional array arguments in column order, and writes
    the data to a text file.  FID is an integer file identifier.  Obtain
    FID from FOPEN, or set it to 1 (for standard output, the screen) or 2
    (standard error). fprintf uses the encoding scheme specified in the
    call to FOPEN.

    fprintf(FORMAT, A, ...) formats data and displays the results on the
    screen.
```

Formatting strings

```
>> employee = 'Fred';
>> age = 32;
>> score = 88.432;
>> fprintf('Employee: %s is %d years old and scored
%f',employee,age,score);
Employee: Fred is 32 years old and scored 88.432000>>
```

These symbols that start with % are substitution points ('conversion characters'). Matlab will insert the subsequent variables into the text, in order. The number of variables listed must match the number of conversion characters.

%s	string
%d	integer/digit
%i	integer/digit
%f	floating point number
%c	single character

Formatting strings

```
>> >> fprintf('%s\t%d\n',employee,age)
Fred    32
```

There are many special characters to control formatting that begin with the backslash:

\t	tab
\n	newline
\v	vertical tab

Working with numbers in strings

```
>> fprintf('Score: %f\n',score);
Score: 88.432000
>> fprintf('Score: %.2f\n',score);
Score: 88.43
>> fprintf('Score: %.0f\n',score);
Score: 88
>> fprintf('Score: %.5f\n',score);
Score: 88.43200
```

Specifies the number of decimal places in a floating point number

```
>> fprintf('Age: %d\n',age)
Age: 32
>> fprintf('Age: %.4d\n',age)
Age: 0032
```

Or the number of total digits in an integer

Special characters

```
>> fprintf ('Score was %.2f%%\n',score)
Score was 88.43%
>> fprintf('Name is ''%s''\n',name)
Name is 'Fred'
```

If you want to print the actual character instead of invoking its special meaning:

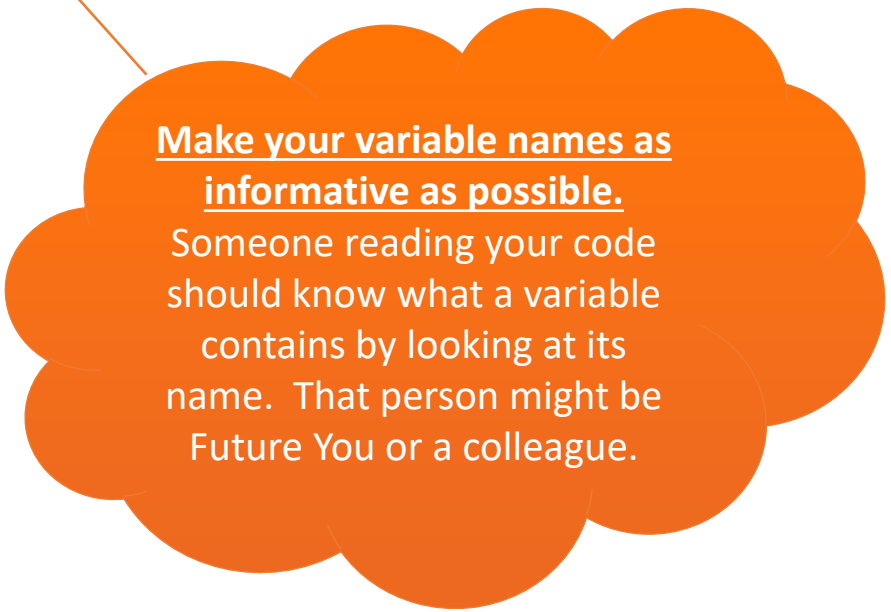
‘’	to print a single-quote
%%	to print a percent sign

Creating string variables

```
>> help sprintf
 sprintf Write formatted data to string.
    STR = sprintf(FORMAT, A, ...) applies the FORMAT to all elements of
    array A and any additional array arguments in column order, and returns
    the results to string STR.

    [STR, ERRMSG] = sprintf(FORMAT, A, ...) returns an error message when
    the operation is unsuccessful.  Otherwise, ERRMSG is empty.

    sprintf is the same as FPRINTF except that it returns the data in a
    MATLAB string rather than writing to a file.
```

Creating string variables

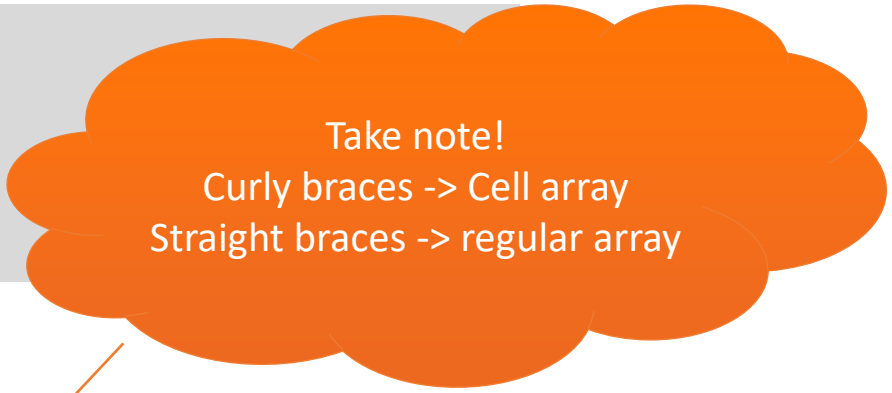
```
>> subject = 'SXF32';
>> logfileName = sprintf('data_%s.txt',subject);
>> logfileName
logfileName =
data_SXF32.txt
```



Collections of strings

Lists of strings

```
>> names = ['Jonas','Fred','John']
names =
JonasFredJohn
```



Introducing *cell arrays*

```
>> names = {'Jonas','Fred','John'}
names =
    'Jonas'    'Fred'    'John'
```

Cell arrays

- Cell arrays can mix and match data types.
- Each cell is its own self-contained variable
- Cell arrays can be arranged in multiple dimensions just like matrices

Using cell arrays

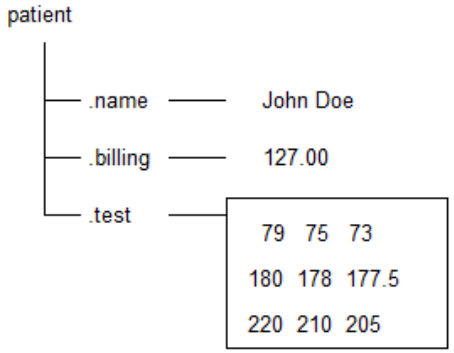
```
>> mycell = {'hello',4,'goodbye',543.43}
mycell =
    'hello'    [4]    'goodbye'    [543.4300]
>> mycell = {[1:5],[6:10]}
mycell =
    [1x5 double]    [1x5 double]
>> mycell(1)
ans =
    [1x5 double]
>> mycell{1}
ans =
     1     2     3     4     5
```

access the cells themselves

access the *contents* of the cells

Structures

- Structures can be used to organize and group information




```
>> patient.name = 'John Doe';
>> patient.billing = 127.00;
>> patient.test = [79, 75, 73; 180, 178,
177.5; 220, 210, 205];
>> patient
patient =
       name: 'John Doe'
    billing: 127
       test: [3x3 double]
```

Arrays of structures

```
>> patient(2).name = 'Jane Doe';
>> patient(2).billing = 156.00;
>> patient(2).test = [71 73, 55; 101, 22, 22; 242,
211, 205];
>> patient
patient =
1x2 struct array with fields:
    name
    billing
    test
>> patient(1)
ans =
       name: 'John Doe'
    billing: 127
       test: [3x3 double]
```

Assignment session #2

- Write a script m.file named “ yourInitials_session2()
  - Define a matrix with in 10 × 3, all values are 0 .
  - Change the value of (5, 5) to 10.

- Define a random matrix 2 × 100.
  - Calculate the mean of this matrix