ENME 303 LAB

Week 9: Functions

Nameless Lab

Week 9: Functions

I. User-defined Functions

II. Defining: User-defined Functions

III. Using: User-defined Functions

IV. Vector Norm

TEST YOUR CODE

Before Submitting, Make sure to …

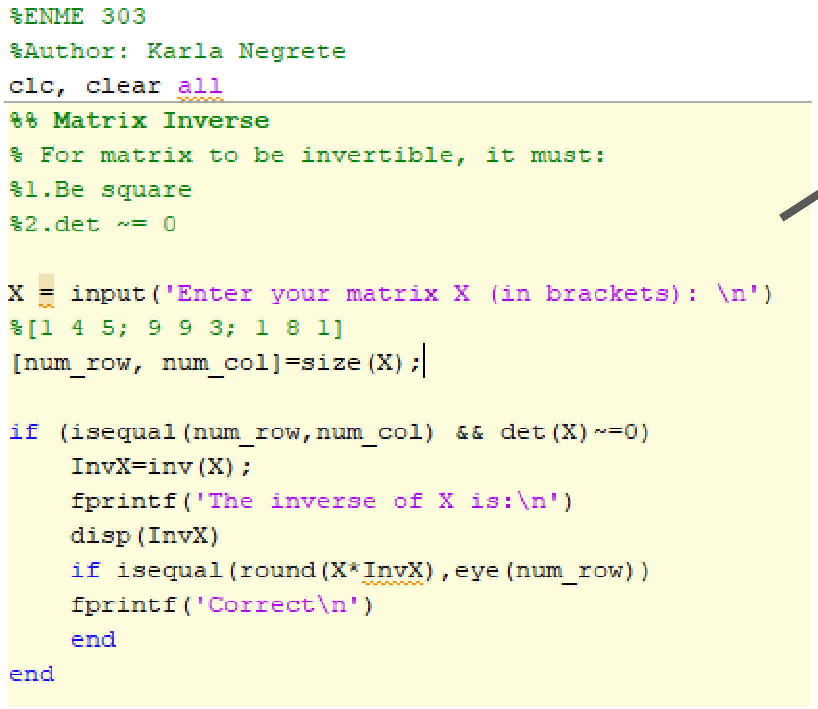
1. **Run your code**
    - Verify that your code does not result in any errors.
    - Verify all variables are suppressed.
    - Verify that the number of outputs and inputs are consistent with the prompt.
2. **Verify the outputs**
    - Inspect the outputs. Does the answer make sense? Use your knowledge of linear algebra from lecture to check your answers.

I. User-defined Functions

Most non-trivial programs are written as a set of independent functions bc:

1. Breaking a large problem into smaller subproblems is **easier to tackle**
2. Duplicate code can be **eliminated**
3. Isolating the number of places a variable can change **simplifies debugging**

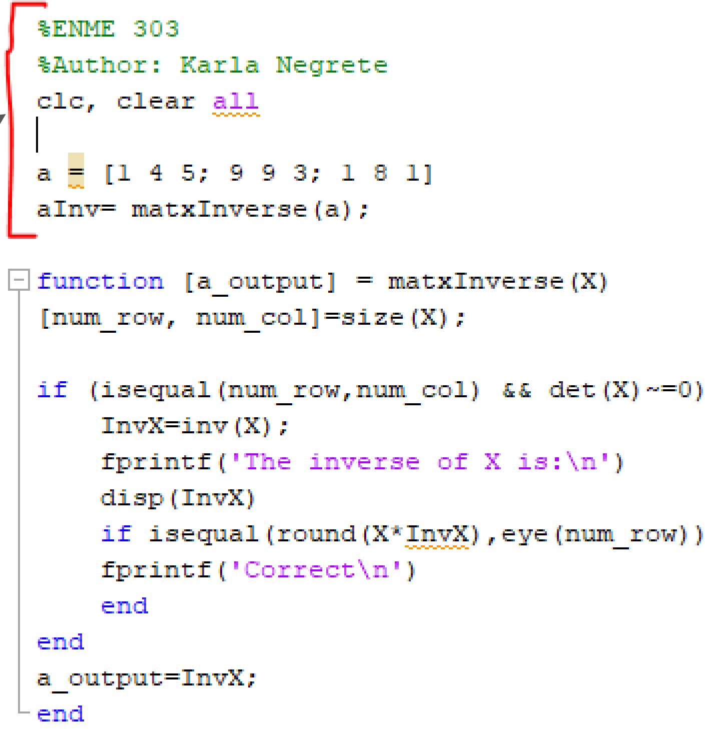
Simplify the main: Neater code!



```
%ENME 303
%Author: Karla Negrete
clc, clear all
%% Matrix Inverse
% For matrix to be invertible, it must:
%1.Be square
%2.det ~= 0

X = input('Enter your matrix X (in brackets): \n')
%[1 4 5; 9 9 3; 1 8 1]
[num_row, num_col]=size(X);

if (isequal(num_row,num_col) && det(X)~=0)
    InvX=inv(X);
    fprintf('The inverse of X is:\n')
    disp(InvX)
    if isequal(round(X*InvX),eye(num_row))
    fprintf('Correct\n')
    end
end
```



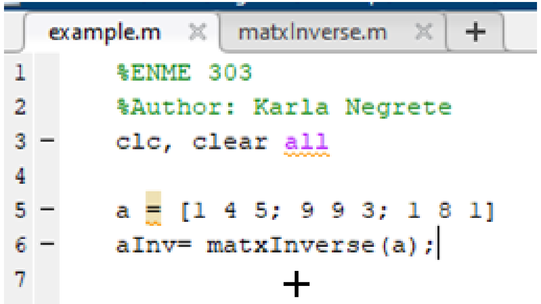
```
%ENME 303
%Author: Karla Negrete
clc, clear all

a = [1 4 5; 9 9 3; 1 8 1]
aInv= matxInverse(a);

function [a_output] = matxInverse(X)
[num_row, num_col]=size(X);

if (isequal(num_row,num_col) && det(X)~=0)
    InvX=inv(X);
    fprintf('The inverse of X is:\n')
    disp(InvX)
    if isequal(round(X*InvX),eye(num_row))
    fprintf('Correct\n')
    end
end
a_output=InvX;
end
```

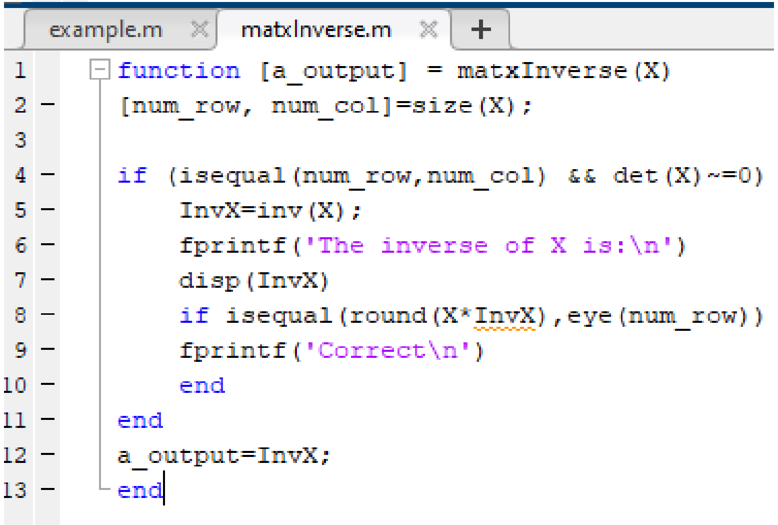
Functions defined either as separate files **or** at the end of script files



```
%ENME 303
%Author: Karla Negrete
clc, clear all

a = [1 4 5; 9 9 3; 1 8 1]
aInv= matxInverse(a);
```

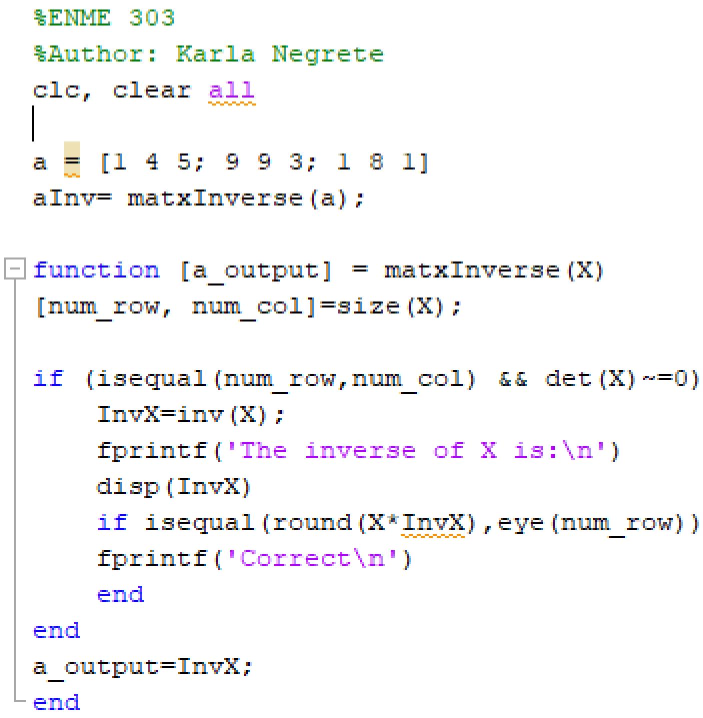
+



```
function [a_output] = matxInverse(X)
[num_row, num_col]=size(X);

if (isequal(num_row,num_col) && det(X)~=0)
    InvX=inv(X);
    fprintf('The inverse of X is:\n')
    disp(InvX)
    if isequal(round(X*InvX),eye(num_row))
    fprintf('Correct\n')
    end
end
a_output=InvX;
end
```

or



```
%ENME 303
%Author: Karla Negrete
clc, clear all

a = [1 4 5; 9 9 3; 1 8 1]
aInv= matxInverse(a);

function [a_output] = matxInverse(X)
[num_row, num_col]=size(X);

if (isequal(num_row,num_col) && det(X)~=0)
    InvX=inv(X);
    fprintf('The inverse of X is:\n')
    disp(InvX)
    if isequal(round(X*InvX),eye(num_row))
    fprintf('Correct\n')
    end
end
a_output=InvX;
end
```

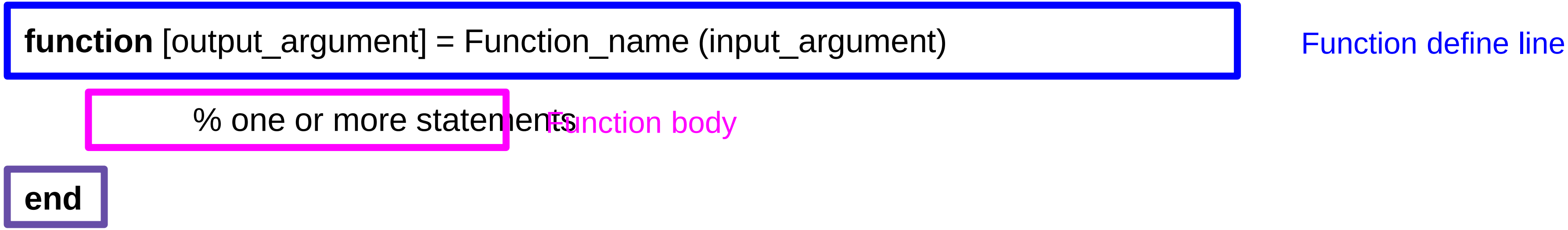
User-defined Function

Structure is key:

1. Function **define line**
2. Function **body**
3. **End**-command
   a. If function is within the script

II. Defining: User-defined Functions

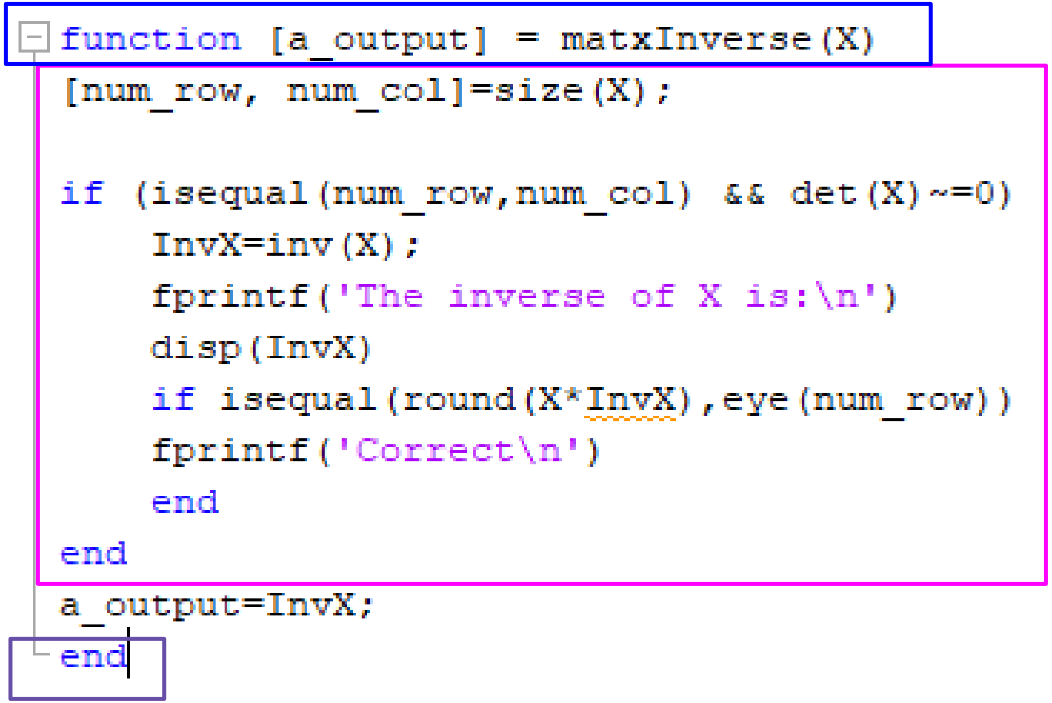
- Functions must be defined before they can be used
    - General syntax is:



```
function [output_argument] = Function_name (input_argument)
        % one or more statements
end
```

- Function input and output arguments are used to pass information into and out of the function

Let's look at that example again



```
function [a_output] = matxInverse(X)
[num_row, num_col]=size(X);

if (isequal(num_row,num_col) && det(X)~=0)
    InvX=inv(X);
    fprintf('The inverse of X is:\n')
    disp(InvX)
    if isequal(round(X*InvX),eye(num_row))
    fprintf('Correct\n')
    end
end
a_output=InvX;
end
```

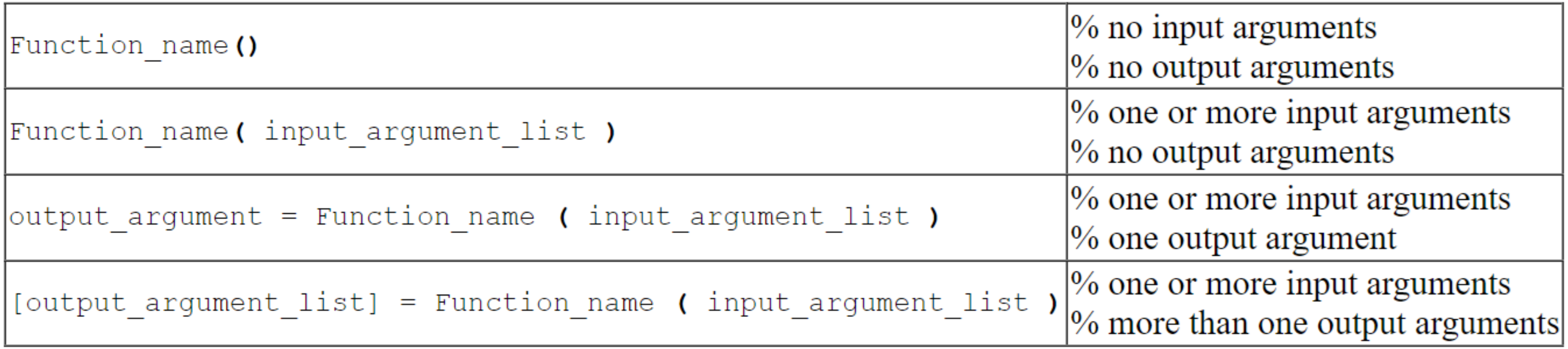
- The function’s name is **matxInverse**
- The function’s **input argument** is **X**
- The function’s **output argument** is **a_output**

User-defined Function: Inputs and Outputs

- Functions may have zero or more input arguments and zero or more output arguments. Ex:
  - function [mpay, tpay] = loan (amount, rate, years)
    - 3 input, 2 output
  - function [A] = RectArea(a,b)
    - 2 input, 1 output
  - function [V,A] = SphereVolArea(r)
    - 1 input, 2 output
  - function trajectory (v, h, g)
    - 3 input, zero output

III. Using: User-defined Functions

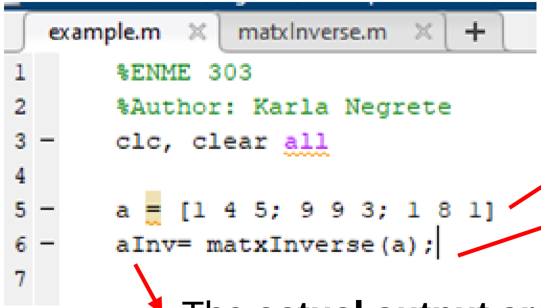
- User-defined functions are called (used) in the same way as Matlab’s built-in functions!
    - General syntax for calling your user-defined function is based on the input and output arguments:



Function_name()	% no input arguments % no output arguments
Function_name(input_argument_list)	% one or more input arguments % no output arguments
output_argument = Function_name (input_argument_list)	% one or more input arguments % one output argument
[output_argument_list] = Function_name (input_argument_list)	% one or more input arguments % more than one output arguments

- The input and output arguments used when a function is called are called **actual arguments**

What happens when I call a user-defined function?



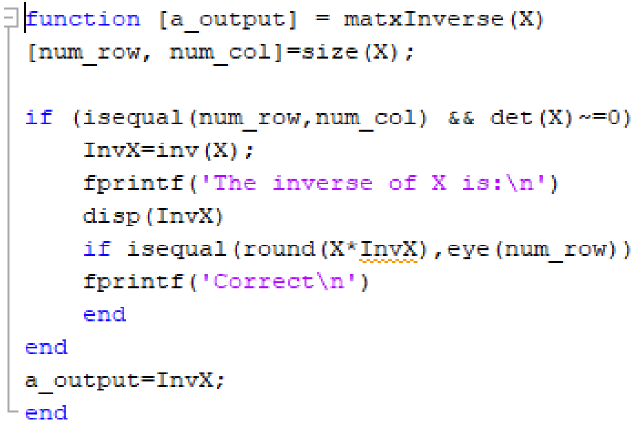
```
%ENME 303
%Author: Karla Negrete
clc, clear all

a = [1 4 5; 9 9 3; 1 8 1]
aInv= matxInverse(a);
```

My **one** **actual input argument** “a”

I pass my **actual input argument** to user-defined function **“matxInverse”**

The **actual output argument** “aInv”



```
function [a_output] = matxInverse(X)
[num_row, num_col]=size(X);

if (isequal(num_row,num_col) && det(X)~=0)
    InvX=inv(X);
    fprintf('The inverse of X is:\n')
    disp(InvX)
    if isequal(round(X*InvX),eye(num_row))
    fprintf('Correct\n')
    end
end
a_output=InvX;
end
```

The **actual input argument** “a” gets assigned to the function input argument “X”

Body of function executed

The function output argument “InvX” gets assigned to the **actual output argument** “aInv”

IV. Vector Norm

- The `norm(a,p)` function is used to find the p norm of the vector a where `p =1,2 or inf.`
- `p = 1,` gives first norm.
- `p = 2,` gives second norm.

  Note: using `norm(a)` gives **second** norm.
- `p = inf,` gives infinity norm

Ex.

```
v = [-2 3 -1];
n = norm(v,1)
```

Output:

```
n = 6
```

Acknowledgement

The lab slides you see are not made by one person. All the TA/TFs served for this course have contributed their effort and time to the slides. Below are the leading TFs for each semester:

- 2021 FA - Karla Negrete (GTA)
- 2022 SP - Justin Grahovac
- 2022 FA - Kelli Boyer, Yisrael Wealcatch, Noelle Ray (GTA)
- 2023 SP - Mahamoudou Bah and Matt Moeller
- 2024 SP - Mohammad Riyaz Ur Rehman & Michael Mullaney