Week 8 MATLAB Activities

Contents

Time required: 120 minutes

1. Create a MATLAB script named **Wk08Lastname.m**
2. Save all programs and tutorials in this script. Functions can go in separate files or at the top of the file.
3. Include your name and date at the top of the script file as comments.
4. Put a Section Break between each program.

Reading

Matlab A Practical Introduction to Programming and Problem Solving (Stormy Attaway)

Sections 4.4, 4.5

Tutorial 1: User Defined Functions

User-defined functions, also known as user-created functions or custom functions, are powerful tools in MATLAB that allow you to encapsulate a specific task or set of calculations into reusable blocks.

```
function [output1, output2, ...] = function_name(input1, input2, ...)

  % Function body:
  % Contains statements that perform the desired operations

  % Return statements (optional):
  % Assign values to the output variables
end
```

Explanation:

1. **Function -** Keyword that defines the beginning of a function.
2. **[output1, output2, ...] -** Comma-separated list of output variables (optional). If no outputs are specified, use empty square brackets [].
3. **function_name -** Name of the function, which must follow naming rules for variables (start with a letter, contain letters, numbers, or underscores).
4. **(input1, input2, ...) -** Comma-separated list of input variables (optional). If no inputs are required, use empty parentheses ().
5. **Function body -** This section contains the MATLAB statements that perform the desired calculations or operations on the input variables.
6. **Return statements (optional) -** Use the return statement to assign values to the output variables. If no explicit return statement is used, the function returns the value of the last evaluated expression.

Creating your first function.

The function name must be the same as the file name. The function file must be in the same folder as the MATLAB program. The function is called **circle_area** → the file must be named **circle_area.m**

- In the same folder as your Wk08Activity file → Create a MATLAB file named: **circle_area.m**
- Insert the following function code.

```
function area = circle_area(radius)

  % Calculate the area using the formula pi * radius^2
  area = pi * radius^2;

  % Return the calculated area
  return

end
```

Create the code that calls the function in your Wk08Activity file.

```
% Example usage: Calculate the area of a circle
radius = input("Enter the radius of a circle: ");

% Call the user defined function with a radius argument
% Return the area if the circle to the circleArea variable
circleArea = circle_area(radius);

fprintf('The area of the circle is: %.2f\n', circleArea)
```

Example run:

```
Enter the radius of a circle:
2.3
The area of the circle is: 16.62
```

Tutorial 2: Percentile and Quartile

A percentile is a statistical measure indicating the relative position of a particular value within a dataset. It represents the percentage of data points below or equal to a specific value in a given set of data.

For example, the 25th percentile (also known as the first quartile) signifies that 25% of the data points in the dataset are equal to or below that particular value. Similarly, the 50th percentile (median) indicates the middle point of the data set, with 50% of the values falling below it. The 75th percentile (third quartile) shows that 75% of the data points are equal to or below that specific value. Percentiles are useful for understanding the distribution and central tendency of a dataset.

A quartile is a statistical measure that divides a dataset into four equal parts, each representing 25% of the data. There are three quartiles, commonly denoted as Q1, Q2, and Q3:

1. **First Quartile (Q1)**: Also known as the lower quartile, it is the value below which 25% of the data falls.
2. **Second Quartile (Q2)**: This is the median of the dataset, dividing it into two equal halves. It represents the middle value, with 50% of the data falling below and 50% above.
3. **Third Quartile (Q3):** Also called the upper quartile, it is the value below which 75% of the data falls.

Quartiles are particularly useful in understanding the spread and distribution of a dataset, providing insights into the central tendency and variability of the values.

```
%% Percentile and Quartiles
% Define a vector
vect = [2 5 9 10 1 7 3];

% Calculate 25th, 50th, and 75th percentiles using the prctile function
p25 = prctile(vect, 25);
p50 = prctile(vect, 50);
p75 = prctile(vect, 75);

% Get the quartiles (Q1, Q2 and Q3) using the quantile function
q1 = quantile(vect, 0.25);
q2 = quantile(vect, 0.50);
q3 = quantile(vect, 0.75);

% Display the results using fprintf function
fprintf('25th percentile: %f\n', p25);
fprintf('50th percentile (median): %f\n', p50);
fprintf('75th percentile: %f\n', p75);
fprintf('Q1: %f\n', q1);
fprintf('Q2: %f\n', q2);
fprintf('Q3: %f\n', q3);
```

Example run:

```
25th percentile: 2.250000
50th percentile (median): 5.000000
75th percentile: 8.500000
Q1: 2.250000
Q2: 5.000000
Q3: 8.500000
```

Assignment 1: Calculate the Area of a Rectangle with a Function

For a project, we need some material to form a rectangle.

Write a function **calcRectArea** that will receive the length and width of a rectangle in inches as input arguments and will return the area of the rectangle.

This function will be called from the **Wk08Activities** file. For example, the function could be called as shown, in which the result is stored in a variable and then the amount of material required is printed, rounded up to the nearest square inch.

```
ra = calcRectArea(3.1, 4.4)
ra = 13.6400
fprintf('We need %d sq in.\n', ceil(ra))
We need 14 sq in.
```

Assignment 2: Calculate Mach with a Function

In aerodynamics, the Mach number is a critical quantity. It is defined as the ratio of the speed of an object (e.g., an aircraft) to the speed of sound.

1. If the Mach number is less than 1, the flow is subsonic.
2. if the Mach number is equal to 1, the flow is transonic.
3. if the Mach number is greater than 1, the flow is supersonic.
4. Write a script that will prompt the user for the speed of an aircraft and the speed of sound at the aircraft's current altitude.
5. Create a function **mach** that will return whether the condition is subsonic, transonic, or supersonic.
6. Call the mach function from the Wk08 Activities file.

Example run

```
Enter the speed of the aircraft: 500
Enter the speed of sound: 450
Supersonic
Enter the speed of the aircraft: 400
Enter the speed of sound: 450
Subsonic
Enter the speed of the aircraft: 450
Enter the speed of sound: 450
Transonic
```

Assignment 3: Pythagorean Theorem with a Function

The Pythagorean theorem formula.

$$a^2 + b^2 = c^2$$

1) Get input for side1 and side2 from the user.
2) Create a MATLAB function that takes in two inputs.
3) Call the function from the Wk08Activities file.
4) Calculate and return the hypotenuse.
5) Display the results.

Example run:

```
Enter side1:
2.5
Enter side2:
3
Length of side 1: 2.500000
Length of side 2: 3.000000
Length of the hypotenuse: 3.905125
```

Assignment Submission

1. Submit properly named and commented script files.
2. Attach a screenshot of the Command Window showing the successful execution of each script.
3. Attach all to the assignment in Blackboard.