Introduction to Computers for Engineers:

Recitation #4

Learning Objectives

- Learn how to create and modify arrays in MATLAB
- Learn the difference between the index and the element of an array
- Experience using the built-in sort command
- Learn how for and while loops work

Activity 1 – Plotting in MATLAB

Suppose you were given the following data:

Exam Grade	# of Hours Studied
85	20
47.5	3
68	7
99.5	32
70	15
15	1
90	27

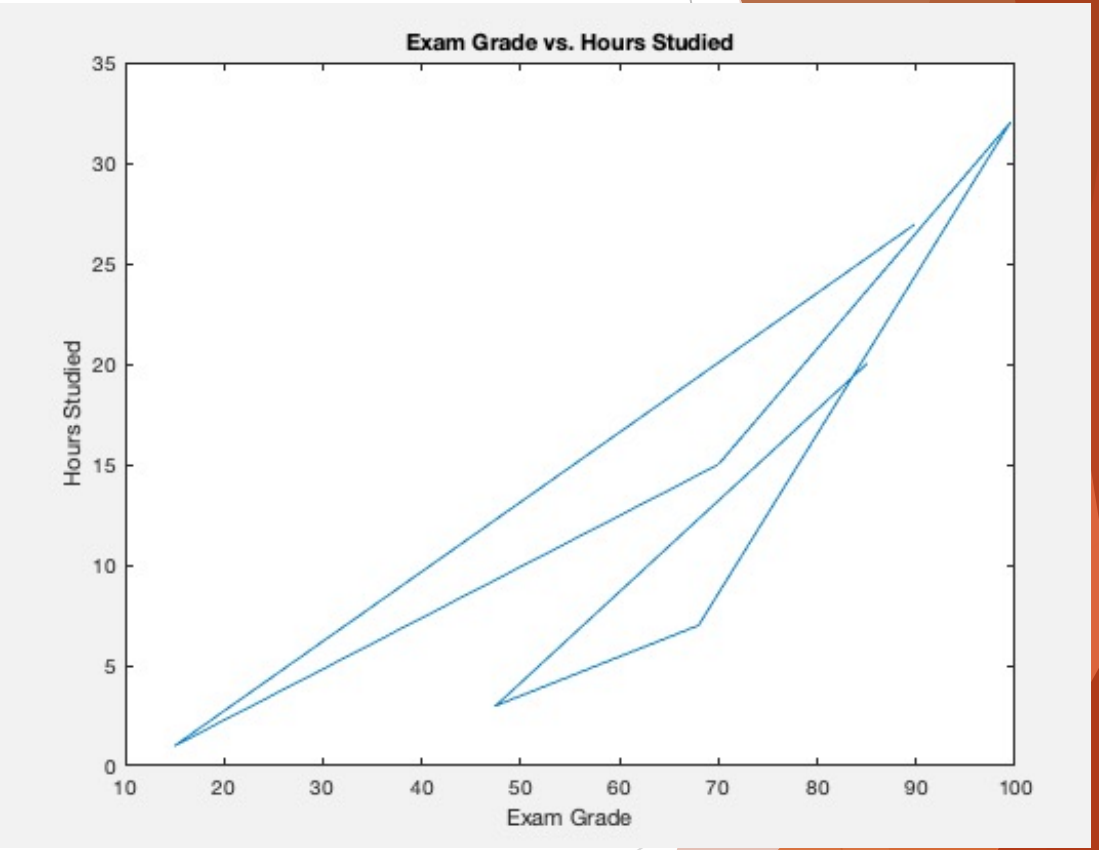
- In a **script**, create two arrays, **exam_grade** and **hours_studied** with the values given in the table above.
- Plot the two arrays in MATLAB using
  - plot(exam_grade, hours_studied)

  Discuss: Does your plot tell you anything useful about the data?

Activity 1 - Solution

```
exam_grade = [85, 47.5, 68, 99.5, 70, 15, 90];
hours_studied = [20, 3, 7, 32, 15, 1, 27];

figure
plot(exam_grade, hours_studied)
title('Exam Grade vs. Hours Studied')
xlabel('Exam Grade')
ylabel('Hours Studied')
```



Activity 2 – More Plotting in MATLAB

- It turns out that we should sort our data before plotting!
- How can we sort two arrays simultaneously in MATLAB?
- We can use the built-in command sort in MATLAB:
  - [sorted_grades, indices] = sort(exam_grade)

Discuss: What is the output of the variable sorted_grades? What is the output of indices?

Activity 2 – More Plotting in MATLAB

- It turns out that we should sort our data before plotting!
- How can we sort two arrays simultaneously in MATLAB?
- We can use the built-in command sort in MATLAB:
  - [sorted_grades, indices] = sort(exam_grade)

Discuss: What is the output of the variable sorted_grades? What is the output of indices?

- We can rearrange the array hours_studied using the indices given by the sort function.
- We can do this by defining a new variable:
  - sorted_hours = hours_studied(indices)

Plot the two sorted arrays and label your plot!

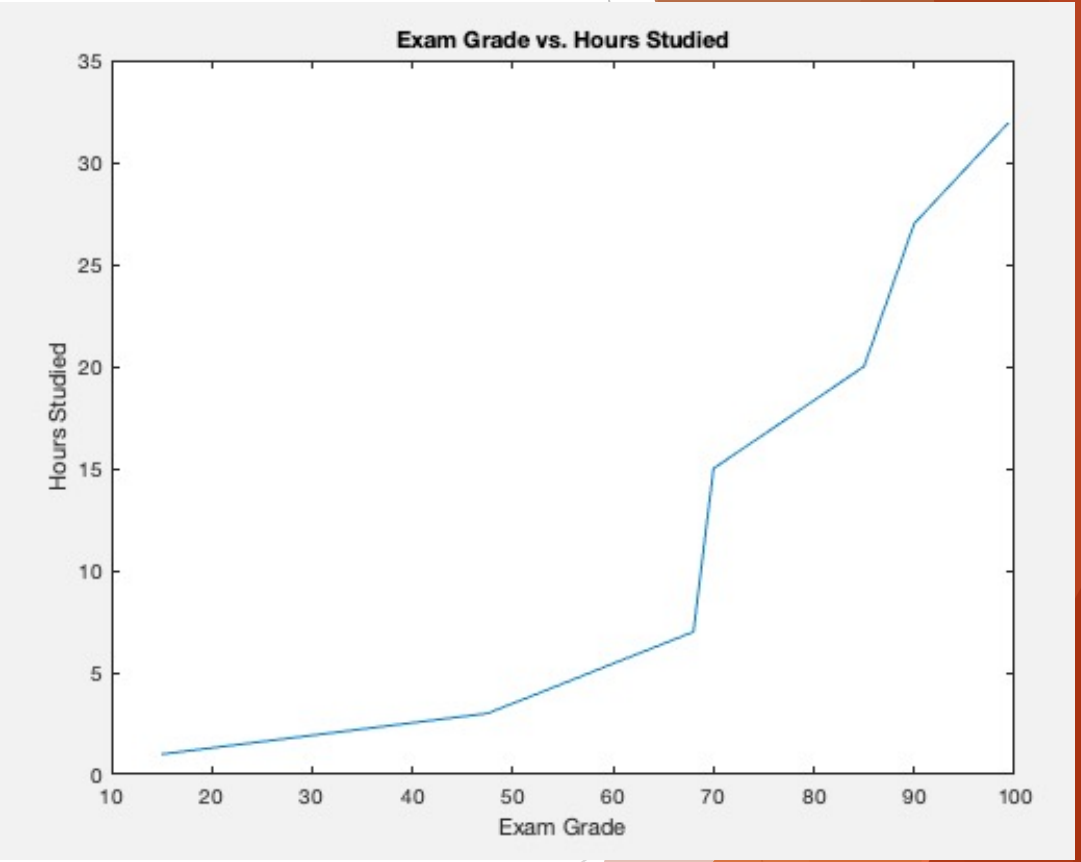
Activity 2 – Solution

```
exam_grade = [85, 47.5, 68, 99.5, 70, 15, 90];
hours_studied = [20, 3, 7, 32, 15, 1, 27];

[sorted_grades, indices] = sort(exam_grade);

sorted_hours = hours_studied(indices);

figure
plot(sorted_grades, sorted_hours)
title('Exam Grade vs. Hours Studied')
xlabel('Exam Grade')
ylabel('Hours Studied')
```



Looks a lot better!

Activity 2 – Even More Plotting!

Suppose we were given two additional data points:

Exam Grade	# of Hours Studied
100	40
35	5

- Add these two points to your existing array without explicitly putting them in:
- DO NOT DO: exam_grade = [85, 47.5, 68, 99.5, 70, 15, 90, 100, 35];

- Sort this new array and plot the two sorted arrays as before.

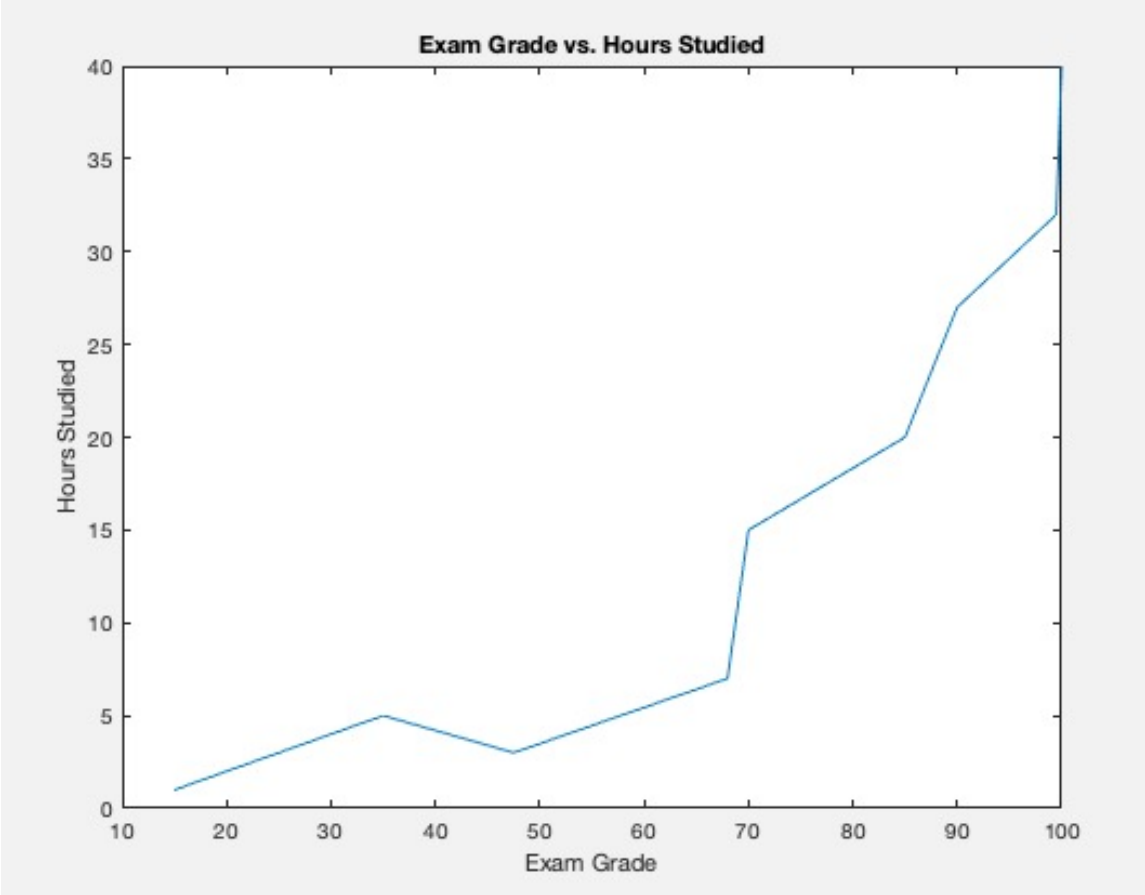
Activity 2 – Solutions

```
exam_grade = [85, 47.5, 68, 99.5, 70, 15, 90];
hours_studied = [20, 3, 7, 32, 15, 1, 27];

exam_grade = [exam_grade, 100, 35];
hours_studied = [hours_studied, 40, 5];

[sorted_grades, indices] = sort(exam_grade);
sorted_hours = hours_studied(indices);

figure
plot(sorted_grades, sorted_hours)
title('Exam Grade vs. Hours Studied')
xlabel('Exam Grade')
ylabel('Hours Studied')
```



Activity 3 – For-Loops

- Suppose that we wanted a code that said “Hello!” 3 times in MATLAB.
- We can do this using a for-loop:

activity_three.m

```
1 -    max_number = 3;
2
3 -    for i=1:max_number
4
5 -        disp('Hello!')
6
7 -    end
```

```
>> activity_three
Hello!
Hello!
Hello!
```

- **Problem:** Write a for-loop that counts up to 10 starting from 5.
- **Problem:** Write a code that outputs the numbers 0, 20, 40, 60, 80, 100 using a for-loop

```
0
20
40
60
80
100
```

Activity 3 – Solutions

```
max_number = 10;

for i=5:max_number

    disp(i)

end
```

```
max_number = 5;

for i=1:max_number
    count_by_20 = i*20;
    disp(count_by_20)

end
```

```
>> activity_three
    20

    40

    60

    80

   100
```

```
max_number = 100;

for i = 0:20:max_number
    disp(i)

end
```

```
>> activity_three
     0

    20

    40

    60

    80

   100
```

Activity 3 – While Loops

- In a script, create the variable
  - **temp_variable** = 10;
- Write a while loop that subtracts 3 from **temp_variable** on each iteration and stops once **temp_variable** becomes negative
  - i.e. **temp_variable** < 0

Activity 3 - Solution

```
temp_variable = 10;

while temp_variable > 0
    temp_variable = temp_variable - 3

end
```

```
>> activity_three

temp_variable =

     7


temp_variable =

     4


temp_variable =

     1


temp_variable =

    -2
```

Activity 4 - Factorial Function

- Create a function called **factorialFunction**, with one input variable **input_number** and one output variable **factorial_value**
- This function will compute the factorial of the input number:
- [factorial_value] = factorialFunction(5)
- factorial_value = 120

Team 1:

Write the function using a for-loop.

Team 2:

Write the function using a while loop.

Activity 4 - Solution

```
function [factorial_value] = factorialFunction(input_number)

factorial_value = 1;

for i=1:input_number
    factorial_value = factorial_value * i;

end
```

```
function [factorial_value] = factorialFunction(input_number)

factorial_value = 1;
counter = 1;

while counter <= input_number
    factorial_value = factorial_value * counter;
    counter = counter + 1;

end
```