COGS 119/219
MATLAB for Experimental Research

Fall 2014 – Week 2

Flow Control: Example with Loops and Conditional Expressions and

Cell Arrays

Problem

1. Create an array of 10 random integers between 1 and 20, and save it into an array called trials.
2. Determine a random number between 1 and 20 (call it threshold)
3. Check whether each element in trials is greater than threshold. Create a new row in trials: If the element is greater than threshold, put a 1 in the new row. Otherwise, put a 0.
4. For each element in the first row of the array, ask user whether that element is greater than threshold. Get a 'y' for yes or 'n' for no response. If the user inputs a response other than 'y' or 'n', give a warning. Create a new row that contains the user responses for each element. If yes, enter a 1; if no, enter a 0; if another response is given, enter a 9.
5. Calculate the accuracy of the user response by comparing the second and third rows of trials element-by-element. Write the accuracy in a new row: 1 for correct, 0 for incorrect.
6. Assign the current trials matrix to a new variable called finalTrials, and display it.

Problem

1. Create an array of 10 random integers between 1 and 20, and save it into an array called trials.

```
trials = randi(20, [1,10]);
```

Problem

2. Determine a random number between 1 and 20 (call it threshold)

```
threshold = randi(20, [1,1]);
```

Problem

3. Check whether each element in trials is greater than threshold. Create a new row in trials: If the element is greater than threshold, put a 1 in the new row. Otherwise, put a 0.

```
for i = 1:length(trials)
    if trials(1,i) > threshold
        trials(2,i) = 1;
    else
        trials(2,i) = 0;
    end
end
```

Problem

4. For each element in the first row of the array, ask user whether that element is greater than threshold. Get a ‘y’ for yes or ‘n’ for no response. If the user inputs a response other than ‘y’ or ‘n’, give a warning. Create a new row that contains the user responses for each element. If yes, enter a 1; if no, enter a 0; if another response is given, enter a 9.

```
response = input(['Is ' num2str(trials(1,j)) '>' num2str(threshold)...
                  '? Enter y for yes, n for no: '],  's');
```

Problem

4. For each element in the first row of the array, ask user whether that element is greater than threshold. Get a 'y' for yes or 'n' for no response. If the user inputs a response other than 'y' or 'n', give a warning. Create a new row that contains the user responses for each element. If yes, enter a 1; if no, enter a 0; if another response is given, enter a 9.

```
for j = 1:length(trials)
    response = input(['Is ' num2str(trials(1,j)) '>' num2str(threshold)...
                                '? Enter y for yes, n for no: '],  's');
    if strcmp(response, 'y')
      trials(3,j) = 1;
    elseif strcmp(response, 'n')
      trials(3,j) = 0;
    else
      display('You did not enter a valid response.')
      trials(3,j) = 9;
    end
end
```

Problem

5. Calculate the accuracy of the user response by comparing the second and third rows of trials element-by-element. Write the accuracy in a new row: 1 for correct, 0 for incorrect.

```
for k = 1:length(trials)
    if trials(2,k) == trials(3,k)
        trials(4,k) = 1;
    else
        trials(4,k) = 0;
    end
end
```

Problem

6. Assign the current trials matrix to a new variable called finalTrials, and display it.

```
finalTrials = trials;
display(finalTrials);
```

trials.m

```
1
2 -    trials = randi(20, [1,10]);
3 -    threshold = randi(20, [1,1]);
4
5 -    for i = 1:length(trials)
6 -        if trials(1,i) > threshold
7 -            trials(2,i) = 1;
8 -        else
9 -            trials(2,i) = 0;
10 -       end
11 -   end
12
13
14 -   for j = 1:length(trials)
15 -       response = input(['Is ' num2str(trials(1,j)) '>' num2str(threshold)...
16                          '? Enter y for yes, n for no: '],  's');
17 -       if strcmp(response, 'y')
18 -          trials(3,j) = 1;
19 -       elseif strcmp(response, 'n')
20 -          trials(3,j) = 0;
21 -       else
22 -          display('You did not enter a valid response.')
23 -          trials(3,j) = 9;
24 -       end
25 -   end
26
27
28 -   for k = 1:length(trials)
29 -       if trials(2,k) == trials(3,k)
30 -           trials(4,k) = 1;
31 -       else
32 -           trials(4,k) = 0;
33 -       end
34 -   end
35
36 -    finalTrials = trials;
37 -    display(finalTrials);
```

Exercise

Comment trials.m explaining the steps in your comments

Can you quickly calculate this subject's accuracy?

Can you extend the program as follows:

Goal: Run the mini experiment on Nsub number of subjects?
Each subject's trials should ask about different random integers.
Each subject's data should be stored in the same way as in the example in 4 rows and 10 columns.
Each new subject's data should be added on to the matrix finalTrials in a 3rd dimension – Hint: finalTrials should be of size 4 x 10 x Nsub

Now extend the program to run for Nsub = 5. You can pretend to be all of the subjects yourself to input the data (Don't do this in real experiments!).
At the end, calculate each subject's mean accuracy and from this, calculate and display on the screen the mean (MeanAcc, the minimum (MinAcc) and the maximum (MaxAcc) as follows:

The average accuracy for the <Nsub> subjects was <MeanAcc>.
The minimum accuracy was <MinAcc> for subject number <MinAccSubjectNo>
The maximum accuracy was <MaxAcc> for subject number <MaxAccSubjectNo>

Note: The text in the <> should be replaced by the values in the program, hint: use numrstr function.