



COGS 119/219

MATLAB for Experimental Research

Fall 2014 – Week 3
Cell Arrays

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);
38
```

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.

Cell array

- Recall:
Vectors and matrices hold elements of a single data type.
- Cell array is a data structure in MATLAB that holds elements of **multiple data types**.

Cell array example

```
>> mycell{1} = 'this is a string';  
>> mycell{2} = 'October 9, 2013';  
>> mycell{3} = date;  
>> mycell{4} = 7;
```

Cell array example (cont.)

- Let's add more elements to our cell array *mycell*:

```
for i = 1:8
```

```
    randoms(i) = round(rand*10);
```

```
end
```

```
randoms = reshape(randoms, 2, 4);
```

```
mycell{5} = randoms;
```

```
mycell{6} = diag([pi pi pi]);
```

```
mycell{7} = {'you' ; 'get' ; 'the' ; 'idea'};
```

Indexing cell arrays: note the parentheses

```
>> a = mycell{1}(3);
```

```
>> b = mycell{1}(4:13);
```

```
>> c = mycell{7}{3};
```

```
>> d = mycell{6}(2,3);
```

```
>> e = mycell{5}(2,4);
```

```
>> f = mycell{5}(:,2);
```

```
>> g = mycell{5}(4,2);
```

Exercise for home –
understand how to access
subfields/subsubelements of
cell arrays