Back to Main Page

Supervision problems

The aim of the supervision exercises is to get some practice approaching the types of questions you'll get in the exam.

Problem 1a

Take a look at the function below:

```
function [output1] = Function1(input)

if input == 1
    outpu1 = 'small';
else
    output1 = 'big';
end
```

What outputs would be produced if we run the following commands

```
>> A = Function1(1);
>> B = Function1(-10)
>> C = Function1('one')
```

Would the code run correctly with these inputs, and if no errors are produced what would be the values of `A`, `B`, `C`. Provide an explanation for why the function produced the given value.

Problem 1b

Take a look at the function below. It is a slight variation on `Function1.m`.

```
function [output1] = Function1(input)

if input == 1
    output1 = 'small';
elseif input == 'one'
    output1 = 'big';
end
```

What outputs would be produced if we run the following commands

```
>> A = Function1(1);
>> B = Function1(-10)
>> C = Function1('one')
```

Would the code run correctly with these inputs, and if no errors are produced what would be the values of `A`, `B`, `C`. Provide an explanation for why the function produced the given value.

Problem 2a

Write a simple function that produces the output shown given the input `x` with the value shown. You might find it helpful to add comments to your code. e

```
>> x = [1:3];
>> CountFun(x)
one
two
three
```

```
>> x = [3:-1:1];
>> CountFun(x)
three
two
one
```

Write it such a way so that it will produce a sensible output for any input vector that contains the number `1` to `10`. You might find it helpful to add comments to your code.

Hints

1. You will to loop over the input values.
    2. You can use a `for` loop, or use something like `arrayfun()`
2. And you'll have to write an string to the console (by using `disp()`).
    3. The string should be a number word (e.g., `'one'`) that corresponds to the number (e.g., `1`).

Problem 2b

Write a function that does the opposite to the function you wrote for *Problem 1a*. It must take `cell` array of number words (see a below) and print out numbers

```
>> x = {'one','two','three'}
```

Problem 3

Write out a flow diagram for a basic computerised experiment measuring reaction times to a stimulus. You'll need to generate stimuli, present instructions, randomise the trials, collect responses, and save the results.

Take home problem

Problem 1

As a take home problem try to write a function that does the following.

- Takes two `vector`s of the same length numbers as input
- Calculates the pair-wise difference between the two `vector`s
- Calculated the mean of each vector and the mean of the pair-wise difference score.
- Generates two dot plots (with 95% confidence interval error bars). One figure with the two means and one figure with the difference score
- Performs a one-sample t-test
- Writes a message to the **Command Window** saying whether the difference is significant (i.e., whether $p < 0.05$). This message should read `'There is a significant difference between x and y'` or `'There is NOT a significant difference between x and y'`
- Produces the output `1` it is significant and `0` if it is not

To do this, you'll need to use the following functions:

1. `mean()`: calculate a mean
2. `sqrt()`: calculate a square root
3. `std()`: calculate a standard deviation (σ)
4. `scatter()`: a scatter plot/dot plot
5. `errorbar()`: draw an error bar plot
6. `[h, p] = ttest()`: a one-sample t-test on a vector input (i.e., the difference scores).
    - The output `h` is `1` if the difference is significant and `0` if it is not. The output `h` is `0` if the difference is not significant.
    - The output `p` is the p-value.

The formula for the 95% confidence interval is:

$$CI = 1.96 \times \frac{\sigma}{\sqrt{n}}$$

Problem 2

If we list all the natural numbers below 10 that are multiples of 3 or 5, we get 3, 5, 6 and 9. The sum of these multiples is 23.

Find the sum of all the multiples of 3 or 5 below 1000.

Hint: If you construct two `vector`s with one containing the the multiple of `3` and one containing the multiples of `5`, then you can concatenate the two `vector`s together and use the `unique()` function to get ride of the duplicate values (e.g.,

15 would be in both vectors. If the input to unique() contains two (or more) values of 15 the output will contain only 1 value of 15.

Matlab Cheat-Sheet

Defining/changing variables

```
x = 3       % define variable x to be 3

x = [1 2 3] % set x to the 1 x 3 row-vector (1,2,3)

x = [1; 2; 3] % set x to be the 3 x 1 column vector (1,2,3)

x(2) = 7  % change x from (1,2,3) to (1,7,3)

s = 'a string' % make the varibale s equal the string 'a string'

c = {'one','two','three'} % define a cell aray (c) containing three strings
```

Constructing simple matrices

```
rand(12,1) % a 12 x 1 matrix of numbers in uniform distribution of [0, 1)

randn(12,1) % a 12 x 1 matrix of numbers from a normal distribution with mean 0 and varianc

randi(10,12,1) % a 12 x 1 matrix of random integers between 1 and 10

randperm(12,1) % 10 unique random numbers betweeen 1 and 12

zeros(12,1) % a 12 x 1 matrix filled with zeros
```

Portions of matrices and vectors

```
x(2:12)  % the 2nd to the 12 elements of x

x(2:end) % the 2nd to the last element of x

x(1:2:end) % every 2nd element of x from the first to the last

x(:) % all the elements of x

x(5,:) % the fifth row of x
```

```
x(:,2) % the 2nd column of x
```

Mathematical and statistical operations

```
sum(x) % calculate the sum of the elements in x
mean(x) % calculate the mean of x
x .^2 % calculate the square of each element in x if x is a vector
x ^2 % calculate the square of x if x is a single number
sqrt(x) % calculate the square root of the number x
```

Simple loops and flow control

```
if x == 1 % if variable x is equal to 1
    FunctionA % then run Function A
elseif x == 2 % or if x is equal to 2
    FunctionB % then run Function B
else          % otherwise
    FunctionC % run Function C
end
```

```
for i = 1 : 4 % loop 4 times
    FunctionA(i) % over Function A using i as the input
end
```

```
k = 0; % set the variable k to zero
while (k < 5) % loop until k is equal to 5
    FunctionA % run Function A
    k = k + 1; % increase the value of k by 1
end
```

```
if strcmp(s,'a string') == 1 % is s contains the string 'a string'
    FunctionA % run Function A
end
```

```
arrayfun(@(x) FunctionA(x),TheArray) % run the function FunctionA using each element of the
```

```
cellfun(@(x) FunctionA(x),TheCells) % run the function FunctionA using each element of the
```

Simple plotting

```
plot(x) % plot y on the y axis with 1, 2, 3,... as the x axis
```

```
plot(x,y) % plot y vs x (x and y must be the same length)
```

```
axis equal % force x and y axes to be scaled equally
```

```
title('A title') % change the plot title to A title

xlabel('x units') % change the x axis label to x units

ylabel('y units') % change the y axis label to y units
```