Queen Mary
University of London

Specific Learning Difficulties Cover Note

Student ID: 170857044

Advice for assessors and examiners

Guidelines for markers assessing coursework and examinations of students diagnosed with Specific Learning Difficulties (SpLDs) –
As far as the learning outcomes for the module allow, examiners are asked to mark exam scripts sympathetically, ignoring the types of errors that students with SpLDs make and to focus on content and the student's understanding of the subject. Specific learning difficulties such as Attention Deficit Disorders, dyslexia and or dyspraxia may affect student performance in the following ways:

- The candidate's spelling, grammar and punctuation may be less accurate than expected
- The candidate's organisation of ideas may be confused, affecting the overall structure of written work
- The candidate's proof reading may be weak with some errors undetected, particularly homophones and homonyms which can avoid spell checkers

Under examination conditions, these difficulties are likely to be exacerbated. Errors are likely to become more marked towards the end of scripts.

Useful approaches can include:

- Reading the passage quickly for content
- Including positive/constructive comments amongst the feedback so that students can work with specialist study skills tutors on developing new coping strategies
- Using clear English and when correcting; explain what is wrong and give examples
- Using non-red coloured pens for comments/corrections

Colleagues in schools are asked to ensure that students with specific learning difficulties access the support provided by the Disability and Dyslexia Service.

For more information regarding marking guidelines see DDS webpage http://www.dds.qmul.ac.uk/staffinfo/index.html and the Institutional Marking Practices for Dyslexic Students

Disability and Dyslexia Service
Room 3.06, Francis Bancroft Building
Queen Mary, University of London
Mile End Road
London
E1 4NS

T: +44 (0) 20 7882 2756
F: +44 (0) 20 7882 5223
E: dds@qmul.ac.uk
W: www.dds.qmul.ac.uk

Alteration or misuse of this document will result in disciplinary action

Assignment 2

1.

Predict

Inputs: Activation threshold and stimulation voltage. The value at which the activation threshold of the axon is set and a list of values that are voltages used to stimulate the axon.

Output: Activation. The point at which the activation threshold is crossed by the stimulating voltage.

Function: -70 is replicated the amount of times equal to the length of the stimulation voltage input. If a value of stimulation voltage is greater than the activation threshold then the output is now equal to 40. Not that this is also replicated the same amount of times as -70 was. This is used to predict when activation will occur.

Calculate Errors

Inputs: Predicted and Observed. The amount of predicted errors and the amount of observed errors.

Output: Total Errors. The total number of errors present.

Function: To calculate the mean squared error present by using the observed and predicted errors.

Fit Threshold

Inputs: Input values stimulation, Input values response and threshold. A list of stimulation values, a list of response values and a value that is used to set a threshold.

Output: fit errors.

Function: to use the two previous function in order to predict the values in which activation occurs, then use these values along with the response values to calculate errors fitted to the model.

2.

The data is modelled via a series of liner functions with stimulation potential being the independent variable and excitation/threshold being the dependant. The response control voltage is the observation within the model.

3.

For the model there is 1 degree of freedom using the n-1 rule. Residuals has 13 degrees of freedom, as there are 14 degrees left and you have to minus one.

4.

A random number is picked from between -100 and 40 within a uniform distribution. Using this number as the threshold value the errors are calculated. This is then repeated another 20 times (therefore 21 times in total). A new random number between -100 and 40 is chosen and a new error calculated. This is then compared against the old error, if the new error is smaller than the old one then both the new threshold and error values are taken. The next run is compared against these values until all 20 runs are complete and the lowest error is achieved. From this know what the best threshold and errors are from those 21 repeats.

Statistically speaking this is ensuring our model is fitted as good as possible by reducing the error as much as possible. This in turn is making the model more statistically viable for further analysis and use.

5.

In order to optimise the model's fit to the data, the script randomly selects 20 values between -100 and 40, that have been uniformly distributed. This value is used as the fitted threshold in order to calculate the error. This error is then compared against the original error, and if it is smaller the new error and threshold values are selected. This new value is considered the best model and used for the remaining comparisons. Ultimately this output will give the best possible error and fitted threshold from the randomly selected values and identify the best possible model compared to the other 20 random models.

6.

To improve the optimisation of the model the number of loops within the script could be increased. This loop is used in the current optimisation and is set at 20 however is this was increased a more reliable model can be randomly chosen, as there is more choice. Another possibility is to remove the random selection element for the fitted threshold and loop for all possibilities. This in turn allows a comparison of all models and have the best possible fitting model be selected.

7.

To fit the data given for the three experimental treatments, modify the code so the control treatment within the fit threshold function was substituted for one of the experimental treatments. Repeat this for all the experimental treatments.

Code:

```
#Please note the name of the variable for each concentration would need to change in order to keep every value.
fitted_threshold_c1 = runif(1, -100, 40)
error_c1 = fit_threshold(stimulation_potential, response_voltage_concN_1, fitted_threshold) #repeat for concN_2 and concN_3.

for (i in 1:20){
        new_threshold_c1 = runif(1, -100, 40)
        new_error_c1 = fit_threshold(stimulation_potential, response_voltage_concN_1, fitted_threshold)
        if (new_error_c1 < error_c1){
                fitted_threshold_c1 = new_threshold_c1
                error_c1 = new_error_c1
}
} #repeat this for concN_2 and concN_3 as well.
```

In order to compare the goodness of fit amongst the fitted models of nicotine concentrations, a Kolmogorov-Smirnov test could be conducted. Once all of the concentrations have got their best fitted model. To modify the code you would have to input the ks.test(x, y,) function, with x being the concentration and y the control.

The Null Hypothesis within this case is as follows; nicotine concentrations have no effect on the action potential of the nervous system.

Part B

1.

The docker file given already sets the working directory and copies the given directory into the container. Using the CMD command it allows the running of R as. Well as the aforementioned script which has been saved as "activation-thresholds.R".

To run this code on a grid first you need to upload the container in order to make it pull-able later on. The create a job script with the .sh extension that can be executed on a grid. Within this script you can include many parameters of your choice depending on what you want you are trying to achieve with the container. Within this file you need to ensure you have loaded the singularity module and run it along with the container which will be an image file.
Finally I would submit this script using the qsub command, this in turn allows the analysis to run on the grid.

2.

To change the output directory you need to add #$ -o ../output:

```
#$ -cwd
#$ -S /bin/bash
#$ -j y
#$ -o ../output # change the output directory an output folder within the directory this is run
#$ -pe smp 1
#$ -l h_vmem=2G
Module load singularity
singularity run activationContainer.img input_data.csv
```

To change random seeds, you need to add the -p command to the singularity run line and add an integer:

```
#$ -cwd
#$ -S /bin/bash
#$ -j y
#$ -pe smp 1
#$ -l h_vmem=2G
Module load singularity
singularity run activationContainer.img input_data.csv -p [integer]
```

To run as an array with 42 replicates you need to add #$ -t 1-42 and #$ -tc 1:

```
#$ -cwd
#$ -S /bin/bash
#$ -j y
#$ -pe smp 1
#$ -l h_vmem=2G
#$ -t 1-42 #adds an array with 42 replicates.
#$ -tc 1 #does 1 run at a time to ensure 42 replicates are achieved.
Module load singularity
singularity run activationContainer.img input_data.csv
```