G3/M3 Assessment 3 — Instructions

1. For this assessment, you are required to write a single R function. The code for this function should be saved in a file named `Surname.r`, where `Surname` is your surname. For example, if your name is K Pearson, your code should be saved in the file `Pearson.r`.

2. Your function should be **thoroughly commented**. It should consist of a header section summarising the logical structure, followed by the main body of the function. The main body should itself contain comments.

3. Your function should *not* create any output files.

4. You are **required** to submit the following:

   - A printout of your R function.
   - An electronic copy of your R function (see below).
   - A brief explanation of how your function works, along with a summary of its output. The explanation should include, for example, details of any mathematical calculations that you carried out before implementing the IWLS algorithm. Where you have made decisions regarding what to produce by way of output, you should justify these decisions. As a rough guide, this explanation/summary should be no more than 2 pages long. You will be penalised if it takes more than 3 pages.

   You are *not* required to submit any output from your function. However, *optionally* you are allowed to submit a second script file, which should be named `Surname_example.r`, where `Surname` is your surname. This script should contain an example of usage of your function. Please read the question for further details and purpose.

5. Printouts and explanations should be handed in to the Statistical Science departmental office. This should be submitted in a **single stapled document** (not in loose pages) that is clearly identified with your name. **Remember to complete a plagiarism declaration, and to attach it to your work**.

6. Electronic copies of your script(s) should be submitted via the Moodle page for the course – use the link “ICA3: Click here to submit your assignment”.

PLAGIARISM AND COLLUSION – EXTRACT FROM DEPARTMENTAL STUDENT HANDBOOKS

Plagiarism means attempting to pass off someone else's work as your own, while collusion means passing off joint work as your own unaided effort. Both are unacceptable, particularly in material submitted for examination purposes including exercises done in your own time for in-course assessment. Plagiarism and collusion are regarded by the College as examination irregularities (i.e. cheating) and are taken extremely seriously. UCL uses a sophisticated detection system (Turnitin®) to scan work for evidence of plagiarism and collusion, and the Department reserves the right to use this for assessed coursework. This system gives access to billions of sources worldwide, including websites and journals, as well as other work submitted to the Department, UCL and other universities. It is therefore able to detect similarities between scripts that indicate unacceptable levels of collusion, as well as material taken from other sources without attribution.

If plagiarism or collusion are suspected, on the basis either of the Turnitin® software or other evidence, it can be dealt with informally only in the case of first offences committed by first year students. All other cases must be dealt with formally, which involves adjudication by a departmental panel and/or College Examinations Irregularities panel. If the panel finds that an offence of plagiarism or collusion has been committed, a penalty will be imposed. Penalties depend on the severity of the offence, and range from being awarded zero marks for the work in question up to exclusion from all further examinations. They can also include a formal reprimand, which will be entered on the student's departmental and College records.

What isn't acceptable?

Students sometimes find it difficult to know what counts as plagiarism or collusion. The following list is not exhaustive, but gives some indication of what to avoid. It is based on guidelines developed by Nick Hayes of the UCL Pharmacology Department. You may **NOT**:

- Create a piece of work by cutting and pasting material from other sources (including websites, books, lecture notes and other students' work).
- Use someone else's work as your own. This includes, but is not limited to:
  - Making notes while discussing an assessment with a friend, and subsequently using these as the basis for all or part of your submission.
  - Telephoning another student to discuss how best to carry out a particular piece of analysis.
  - Employing a professional ghostwriting firm or anyone else to produce work for you.
- Use somebody else's ideas in your work without citing them.
- Ask a lecturer in the department for help with assessed work, unless you make it clear to them that the work is assessed.
- Help another student with their assessed work. If you do this, you will be deemed to be guilty of an examination irregularity.

What is acceptable?

The following practices do not constitute plagiarism / collusion:

- Quoting from other people's work, with the source (e.g. book, lecture notes, website) clearly identified and the quotation enclosed in quotation marks.
- Summarising or paraphrasing other people's work, providing they are acknowledged as the source of the ideas (again, usually this will be via a reference to the book, journal or website from which the information was obtained).
- Asking the course lecturer for help with difficult material, providing it is clear that the question is in connection with the assessment. The lecturer will be able to judge for him or herself what is an appropriate level of assistance.

Some examples

Unfortunately, each year there are some students in the Department of Statistical Science who submit work that contravenes the regulations. The consequences can be severe.

Example 1: Final-year student A had a lot of coursework deadlines in the same week as an important job interview. One of the coursework deadlines was for an extended piece of data analysis, set two weeks previously. Because of his other commitments, student A did not start this piece of coursework until shortly before the deadline, at which point he discovered that he did not have enough time to do it. He asked student B for help. The result was that both students submitted essentially identical work using exactly the same computer output. A departmental panel was convened to investigate the matter. The panel suggested that student B had passed electronic material (computer output and graphics files) to student A, who had pasted this material straight into his own submission. Although student A admitted asking student B for help, both students denied exchanging electronic material. They were, however, unable to explain how the same electronic files came to appear in both submissions. As a result, the allegation was upheld and both students were penalised. Student A was recorded as "non-complete" for the course in question (this meant that he had no possibility of passing it that year), and student B was given a mark of zero for the coursework component.

Example 2: Students C and D both had to submit some computer code for an assessment, which was worth one third of the total mark for a course. There was considerable flexibility in how to go about the assessment. Although the students submitted code that looked very different, closer inspection revealed that they were carrying out the same procedures in more or less the same order, and that the methods they used to carry out these procedures were essentially the same. Further, these procedures and methods were not used by other students in the class. On investigation, it transpired that the students had discussed the assessment over the phone while sitting in front of their computers. This is unacceptable, and as a result the marks of both students for this piece of assessment were halved.

Example 3: The in-course assessment for a particular module was organised as a multiple choice exam taken via Moodle outside of lessons. Each student could attempt the one-hour exam at any time of their choosing within a ten day window, but were clearly advised that they must work alone. After the exams had been graded, it was noticed that students E and F had given identical answers to every question (including incorrect answers). Inspection of the Moodle logs revealed that the students had started and finished their attempts at exactly the same time, using IP addresses that were traced to adjacent PCs in the same computer cluster. Students E and F admitted colluding on the in-course assessment and were both given a mark of zero.

How to avoid plagiarism and collusion

If you are found to have committed an offence of plagiarism or collusion, it makes no difference whether or not you intended to do so. Ignorance is no excuse. To avoid committing an offence, a useful rule of thumb is: if in doubt, don't do it. Make sure that any work you submit is your own unaided effort. More specific guidance is as follows:

- Plan your work schedule carefully, to allow enough time to complete each piece of assessment.
- If you have genuine problems in meeting a deadline, don't take the easy way out and borrow a friend's work. Discuss your difficulty with the course lecturer in the first instance.
- If you are stuck with an assessment, don't ask another student for help. Discuss it with the course lecturer.
- If another student asks you for help with an assessment, or asks to see your work, suggest that they approach the course lecturer instead. Remember: if somebody else copies or uses your work, you will be penalised as well, even if you didn't expect them to use your work in this way.

G3/M3 Assessment 3 — Hints

1. There is no single 'right answer' to this question. To obtain a good mark you need to approach the problem sensibly, and to provide a clear justification of what you're doing. Credit will be given for code that is *clear* and *readable*. In particular, code that is inadequately commented will be penalised.

2. You should ensure that your function produces output that is clearly and appropriately labelled and formatted.

3. You are not required to analyse any data here; however, when marking this assessment, your function will be tested on one or more datasets to ensure that it works correctly. You may therefore wish to test your function on a simple dataset before submission, and optionally submit your test script along with your function as described in paragraph 4 of the instructions

4. If desired, you may use the `IWLS` function from Lab 8 as a starting point for this assessment.

5. To explain how your function works, you will probably need to use quite a lot of mathematical notation. Therefore, unless you are familiar with LaTeX, you should probably *not* attempt to word-process this explanation! A legible handwritten explanation is perfectly acceptable.

6. Your scripts will be tested by calling your function from a program that assumes that you have done *exactly* what the question asks for. This means, for example, that you must specify your function's arguments in the order given above, and that the names of respective elements of the list result must be the same as those given above. If you do not do this, your function will fail when called, and you will lose marks.

7. R has some built-in routines relating to the exponential distribution. You may use these if you think they would be useful.

G3/M3 Assessment 3 — Marking guidelines

This assessment is marked out of 50. The marks are **roughly** subdivided into the following components.

1. Correct implementation of the IWLS algorithm: 12 marks. The submitted function must return the correct estimates and standard errors.

2. Correct calculation of p-values, degrees of freedom and deviance: 7 marks. As in the first item, the code will be checked for obvious errors/omissions.

3. Correct setting of default values: 3 marks. Defaults should be set in the function definition rather than via `if` statements in the function body.

4. Appropriate, and correct, checks of input values: 5 marks. This should check at least the dimensions of `y` and `X`, among other pieces of information you may find relevant.

5. Appropriate, and correct, diagnostics: 8 marks. For full marks, diagnostics for both systematic structure and distributional assumptions are required.

6. Correct value of function: 2 marks. Output should obey guidelines strictly.

7. Good coding style: 5 marks, for code that is clear, efficient, portable, readable and appropriately commented.

8. Clear (and correct) written explanation: 8 marks. A presentation of the correct expression of the basic components of IWLS, such as the mean and variance function, is expected, along with a clear justification for the diagnostics that have been chosen and the formulas used in them.

Marks will be deducted for code that a user would find 'difficult' to use. The 'user' here can either be (i) someone who cannot code and only knows how to run an R script and expects something meaningful to be produced on their screen or written to file; (ii) a fellow developer who would like to not only run your code but also understand how it works with a view to maybe building some of their own code on top of it. Generally, both of these user types should find your code useful and easy to use in order for you to get good marks.

G3/M3 Assessment 3

Suppose that $\mathbf{Y}$ is a vector of exponential random variables, with $Y_i \sim Exp(\lambda_i)$. Suppose also that $\mathbf{x}_i$ is a vector of covariates, forming the ith row of a matrix $\mathbf{X}$, such that

$$\ln \lambda_i = \mathbf{x}_i^T \boldsymbol{\beta} = \eta_i \text{ , say,}$$

for some coefficient vector $\boldsymbol{\beta}$.

Write an R function to fit such a model using iterative weighted least squares, and to check the fitted model. Your function should be called `erm` ('exponential regression model'). The arguments to the function should be `y`, a vector of responses; `X`, a design matrix of covariates, and `startval`, an initial estimate of the model coefficients. The default value of `startval` should be a vector of zeroes (or any other sensible choice).

The value of your function should be a `list` object containing the following components (you may add more components if you feel that these would be useful):

`y`:	The observed responses.
`fitted`:	The fitted values.
`betahat`:	The estimated regression coefficients.
`sebeta`:	The standard errors of the estimated regression coefficients.
`cov.beta`:	The covariance matrix of the estimated regression coefficients.
`df.model`:	The degrees of freedom for the model.
`df.residual`:	The residual degrees of freedom.
`deviance`:	The deviance for the model.

The structure of your function should be similar to the following:

1. Check that the values of `y` and `X` are compatible, and that the data are suitable for modelling using the exponential distribution — if not, stop with an appropriate error message.

2. Carry out the IWLS procedure to fit the model, and output the results to screen (as described below).

3. Produce residual plots and other appropriate model diagnostics.

4. Assemble the results into a `list` object, and return this as the value of the function.

In step 2, the screen output should consist of: a table showing the estimated coefficients, their standard errors, z-statistics and associated p-values; the degrees of freedom for model and residuals; and the deviance for the fitted model. You may output any other relevant information if you wish.

In step 3, you should use your knowledge of model checking for GLMs to produce an appropriate selection of diagnostics. You do not have to produce the same plots as R does when you plot a glm object.

Your function must *not* use the glm command (nor anything similar such as glm.fit)!

Optional test case script

You are allowed to write a second script which loads a dataset, fits a regression model using your implementation of erm, and outputs a selection of estimates and diagnostics. The choice of data is yours, but the execution must be reproducible by any users of your script. Hence, limit yourself to datasets which can be loaded from a R package, or which can be constructed from R code within the script itself. For the former, we recommend the package 'datasets'.

The choice of data and outputs is yours to make. The goal of this script is for you to demonstrate to us an example of your script working in practice, in case we have any problems running it on our own test cases. For instance, if your script works correctly with the data provided by you but not with all of our test cases, we will be able to give you appropriate credit for demonstrating a situation in which the script works. For that to be possible however, we require that your test case script is clearly written and commented. As long as the code is clear and reproducible, the format is up to you.