## MATH6157 – Applied Statistical Modelling
## Assessed Assignment 2

This assignment should be submitted by **26th May 2017 at 3pm**. It counts for 50% of your final mark for this module.

Your solutions should take the form of annotated `R` code (within a Jupyter notebook, if you like) that can be successfully run using a standard `R` installation (that includes the necessary packages). Your annotations should fully explain any modelling choices you make, and any conclusions you draw, as asked for by the question. You **do not** need to write a separate report. The code should produce any plots that are requested and/or that you discuss. I suggest setting the random number seed (`set.seed`) to ensure reproducibility of results.

Please submit your assignment through the "Assignments" page on Blackboard. There is **no need** to submit a paper copy through the student office.

1. **(Mixed models)** The `bacteria` data set (available from the `MASS` library in `R`) gives the results of tests for the presence (`y`) or absence (`n`) of the bacteria *H. influenzae* in a group of 50 children (`ID`) in Australia. Each child was given one of three treatments (`trt`), `placebo`, `drug`, `drug+`, and measurements were taken at `weeks` 0, 2, 4, 6 and 11.

   (a) Fit various generalised linear models (GLMs) to the data using `week`, `trt` and `ID` as explanatory variables, and comment on their fit, e.g. using summary statistics and/or graphical summaries. Produce a normal probability plot for the residuals from at least one of the models you fit, and comment on its usefulness for binary (ungrouped) data, giving reasons for your comments. [3 marks]

   (b) Now fit various generalised linear mixed models (GLMMs), including random effects. Assess the fit of each model you try. Qualitatively compare your models to the best GLM fit (i.e. informally compare them). [5]

   (c) Perform an appropriate hypothesis test to compare models with and without a random intercept (i.e. models including `trt`, `week` and with/without a random effect for `ID`). [6]

   (d) One advantage of the random intercept model is that you can predict for unseen subjects (new levels of `ID`). Using this model, predict the response for all five weeks for three new subjects, one who has been given the `placebo`, one who has been given the `drug`, and one who has been given `drug+`. [2]

   (e) If we ignore the variable `week`, we can create a binomial response for each child. Aggregate the data across weeks to create a new binomial response for each child (level of `ID`) and fit a GLMM with random intercept to this new data set. Do the two treatments differ from `placebo` for this grouped data? [4]

2. **(Smooth regression)** The `aatemp` data set (available from the `faraway` library in `R`) gives the annual mean temperature (in degrees Fahrenheit) in Ann Arbor, Michigan, for 115 years between 1854 and 2000. We will fit various smooth models to the subset of the data which excludes the first three observations (from years 1854, 1855 and 1871). From 1881 onwards, we have mean temperatures for every year.

   (a) Start by creating two new variables that contain the year and temperature data that we wish to model. Plot the data. [1]

   (b) Fit a simple linear model with `temp∼year`. Add the fitted regression line to your plot. Is there evidence for a non-zero slope parameter? What is a main restriction of this modelling approach? [1]

(c) A smoothing method not discussed in lectures is "local weighted regression" (often called Loess or Lowess). For prediction at a point $x$, this smoother fits polynomial regression models using data "near" $x$, weighted by the distance from $x$ (similar in principle to the weighting in a kernel smoother). It can be thought of as a generalisation of a "moving average" smoother, with simple averaging of data local to $x$ replaced by a local least squares fit. It is implemented in the package `loess` in `R`. The proportion of data points used for each local fit is set by the parameter `span` ($> 0$).

i. Read the help file on the `loess` function and then apply the function to the data using the default `span` $= 0.75$. Add the fitted smoother to your plot. [1]

ii. The `bootstrap` package in `R` contains function `crossval`, which can be used for cross-validation with a reasonably general class of models (see also Worksheet 5). Read the help file for this function, and use it to perform replicated cross-validation to choose the value of `span`. Add the best fitted Loess smoother to your plot. Comment on the differences between the three smoothers you have fitted. [4]

(d) Also fit a kernel smoother to this data, using cross-validation to choose the smoothing parameter. You can use the `cv` function from lectures. Add the best smoother to your plot. [1]

(e) For each smoother (the linear model, the two Loess smoothers and the kernel smoother), work out the $R^2$ value. By comparing the $R^2$ values and the plots, comment on the differences between the various smoothers (e.g., which appears to have the smallest/largest equivalent degrees of freedom?). [2]

3. **(Design and smoothing non-normal data)** In this question, you will generate an experimental design, collect some (simulated) data and fit generalised additive models. To ensure I get the same results as you when I run your code, **you must set the random number seed using your student ID, e.g. `set.seed(12345678)`, at the start of your answer for this question**. After you are satisfied with your answer, I suggest you run all the code again, to ensure it is reproducible.

(a) Generate a maximin Latin hypercube design with $n = 100$ in 10 variables. Plot the one-dimensional projections of your design, and comment on their distributions. [2]

(b) Generate 1000 **random** Latin hypercube designs and 1000 designs with points randomly sampled from independent uniform distributions for each variable. For each design, calculate the minimum inter-point Euclidean distance. For each set of designs, plot the distribution of these distances and then graphically compare them to the minimum inter-point distance of your maximin design. [5]

The `R` workspace `math6157Cwk2.RData` can be downloaded from Blackboard. It can be loaded into `R` using the command `load("math6157Cwk2.RData")`. The workspace contains one function,

`math6157Cwk2Data()`,

which takes one argument, a dataframe with 10 columns. It returns a vector of binomial responses, in the form of the number of successes from 10 trials.

(c) Use the function `math6157Cwk2Data()` to generate data from your maximin design. Note that the function returns `random` data, so different responses will be generated each time you run the function. Therefore, only run this function once to get your data. [1]

Additive models, introduced in lectures, can be generalised to accommodate non-normally distributed responses, in much the same way as linear models can be extended. So-called *generalised additive models* (GAMs) are defined as

$$\begin{aligned}
&y \sim \text{Exponential family}\,; \qquad E(y) = \mu\,; \qquad g(\mu) = \eta\,; \qquad \text{Var}(y) = V(y)\,; \\
&\eta = \beta_0 + \sum_{l=1}^{p} f_l(x_l)\,.
\end{aligned}$$

Hence, we have extended the generalised linear model by allowing the linear predictor $\eta$ to contain arbitrary smooth functions.

For example, a model for binomial data might have:

$$y \sim \text{Binomial}(m, \pi(x)); \qquad E(y) = \mu = m\pi(x); \qquad \log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \eta; \qquad \text{Var}(y) = m\pi(x)(1-\pi(x))\,.$$

The `gam` function in the `mgcv` package in `R` can fit these GAMs.

(d) For your data, plot each of the 10 variables against the response. Comment on any trends. [2]

(e) Read the help file for `gam` and then use the function to fit a GAM that includes all 10 variables in the training data. Produce effect plots for this model, including partial residuals. From the `summary` of the fitted model object and the plots, comment on which terms seem most important. [6]

(f) Use the `anova` function (and `test = "Chisq"`) to find a reduced model that provides an adequate fit for the data. Produce effect plots for this model. [4]