**Assignment 2**

**In brief:**

Choose a Data Science algorithm and use Shiny to communicate it. The communication should reveal something about what makes this algorithm special. Specifically: when the algorithm works well and when the algorithm does not. The deliverable will be a pair of files: ui.R & server.R that I should be able to run and score.

**This assignment will demonstrate**

1. your ability to self-discover algorithmic aspects of R packages
2. your grasp of Shiny
3. your ability to communicate clearly.

The only limitation is that you need to work within the range of packages already loaded onto the workstations. The Internet, Shiny Gallery examples and this course’s examples will serve as the study material for this assignment.

**Steps:**

1. Select an algorithm that interests you. Examples include: glm, k-means, SVM, PCR, logit, decision trees, ANN. Do not be too ambitious – there is plenty to discover, visualise & communicate in even the more mundane algorithms.
2. Research the packages that such algorithms reside in (there may be several). Come to understand what the parameters do / how it works. Come to appreciate what kind of data is going to suit the model. Look at package vignettes and package example-code. Which parameters can be optimised by trial and error? How would you classify the algorithm? (eager/lazy, classification/regression, supervised/unsup...)
3. Choose some data (or maybe several sets of data) that you are going to train the model on. Synthetic data is permitted if you want to try that. Plan what summary text & visualisations reveal something of the **data**.
4. Plan what model visualisations reveal something about how the **model** works. Perhaps you may use input controls to modify the parameters interactively? Are there meta-parameters that need to be independently optimised and can you visualise the optimisation process? Perhaps you can plot its timing as the number of observations and/or variables grows? Perhaps you can show how the algorithm copes when the number of observations is low. Perhaps a user can switch between multiple datasets and thereby come to understand when the algorithm is appropriate?
5. Create a Shiny App that will serve as a showcase for your model. Maybe a “tabset” will be a good way of keeping everything organised on a single screen without scrolling. Will you need any check-boxes, selectors etc to control the parameters?
6. When you are finished test it out on a fresh instance of R, (use *Session / Restart R* in the Rstudio menu). Seriously, do this or your model may not run correctly when the lecturer runs it.
7. Reread these notes, before you upload, to check you have done everything required.
8. Use the course LEARN website to upload the **ui.R**& **server.R** files.