**Assignment 3**

**In brief:**

Using a supplied dataset in the form of a CSV file, you are tasked to create a Shiny App that does end-to-end analysis of the data resulting in a predictive model for its **Y** variable. In order to keep the assignment simple, the modelling algorithm will be confined to **stats::lm**.

This assignment will demonstrate your end-to-end skills such a

* + discovering the structure of the dataset
  + developing a cleaning strategy
  + developing a missing value strategy
  + producing test-train splits
  + developing a dimensional reduction strategy
  + producing a set of candidate linear models
  + quantifying the performance of the best model

Create a Shiny App that will serve as a showcase for this dataset and your fitted model by communicating each of the above skills through appropriate visualisations and statistics. The only limitation is that you need to work within the range of packages already loaded onto the workstations.

**In detail:**

Load the data using ***read.csv()*** and perform any data tidying required: ID variables to **rownames**, date-strings to dates, strings to factors. Decide whether factors should be ordered. The Shiny app can be organised as a set of tabs. The following is suggested:

**Tab 1 Data Structure**

Discover what you can about the training data and provide some **Shiny** visualisations of the dataset that communicate its structure e.g. str(data), head(data), matplot(y = data)

**Tab 2 Outliers**

Develop a strategy for detecting outliers. Drive this cleaning strategy with **Shiny** controls so that the effect of cleaning can be explored visually. Hint: ***boxplots***, using the ***range*** parameter and ***boxplot.stats*** using the ***coeff***parameter are two possibilities. *Cook’s Distance* is a third.

**Tab 3 Missing Values**

Visualise the missing value structure throughout the data set e.g. ***vis\_miss()***. The choices for Missing values are to a) omit, b) ignore i.e. let the downstream algorithms deal with them, c) impute them by some algorithm. Decide and develop a strategy appropriate to this data. Use **Shiny** controls to signify your decision and any alternatives worth exploring.

**Tab 4 Data Split**

Split the cleaned data into a test set and a train set. Use a Shiny slider to control the test-train split ratio and preset it with an appropriate value.

**Tab 5 Dim Reduce**

Develop a strategy for dimensional reduction. Consider the numeric variables; is there benefit in reducing the number of these? What is the potential *obs/parameters* ratio, with and without, dimensional reduction? If you choose to allow dimensional reduction, use a slider to choose the number of principle components.

Since the test cases come from the same dataset as the training cases, we can assume they will not be outliers to the Dimensional Reduction process and no test for this will be required.

Hint: to isolate the numeric variables

*numeric <- sapply(data, is.numeric)*

*numData <- data[,numeric]*

**Tab 6 Train**

Using **stats::lm,** manually develop a formula that is optimal with respect to the training data. The formula should be able to be typed in to a ***text*** or ***textarea*** field in **Shiny** and the training results shown below. Progressively remove variables from the formula that have no or little modelling power.

**Tab 7 Test**

Use the optimised formula to calculate the unbiased performance statistics for the model. i.e. how well it should work on unseen data. Also include some visualisations of its performance, for example Predicted v Actual. Do not forget to report a) the number of training observations, b) the number of test observations, c) the number of removed observations (due to cleaning), d) the number of parameters used during modelling [ hint:

*mod <- lm(formula=input$formula, data=dat, x=TRUE)*

*mod.matrix <- mod$x*

*obs <- dim(mod.matrix)[1]*

*params <- dim(mod.matrix)[2]*

to account for any interaction terms]

When you are finished test it out on a fresh instance of R, (use *Session / Restart R*and *Session / Clear Workspace*in the **Rstudio**menu). Make sure all your **library()**calls are in place.

Use the course **LEARN**website to upload the **ui.R**, **server.R**, (**global.R**) files. You should ensure that the ***Ass3Data.csv*** file is sought in the same folder as these files.

**Marking:**

You must work alone. Extra marks will go to deliverables that communicate well and allow the critical decisions be modifiable via Shiny controls. Use on-screen text to communicate the reasons for your various critical decisions. Try, where possible, to allow alternatives to your decisions to also be explored.

**Overdue Work:**

Overdue work will incur a penalty of 10% for each day it is late.