The SMS Spam data set

==========

The dataset contains spam and non-spam ('ham') text messages accumulated from several sources. The description is at the UCI archive (Links to an external site.).
We have pre-processed the data to make life easier for you. The

file spam.rda Download spam.rdais an R binary workspace. When you load it into R with

load("spam.rda")

three objects will appear in your R workspace:

df: data frame with three columns: **text** containing the original text for each message, **is_spam** with TRUE for spam and FALSE for non-spam messages, and **words** containing a character vector of words for each message. Words have been created by dividing the text at spaces at the characters **.,:()?!** There are some zero-length words; these occur when two of the dividing characters are adjacent.

common_words: all the words that occur more than 20 times

word_matrix: a matrix with a row for each message and a column for each common word, indicating how often the word occurs in the message.

Assignment 3 Tasks:

==========

1. Use **rpart** to fit and prune a tree predicting spam/non-spam from the common word counts in the **wordmatrix** matrix. Produce a confusion matrix and report its accuracy. Plot the fitted tree (without all the text labels) and comment on its shape.

2. Build a $\mathrm{NaiveBayes}$NaiveBayes classifer. For each common word

in **wordmatrix**, compute the number $y_i$yiand $n_i$ni, which respectively gives the

counts of spam and non-spam messages. Then the overall evidence provided by having this word in a message can be approximated by

$$e_i = \log(y_i + 1) - \log(n_i + 1) ei=log(yi+1)−log(ni+1).$$

A Naive Bayes classifier then sums up the e_i for every common word in the message to get an overall score for each message. It then splits this at some threshold to get a classification. (FYI -- it is called naive Bayes because it would be a Bayesian predictor if the words were all independently chosen, which they obviously won't be in this case).

Construct a naive Bayes classifiers and choose the threshold so that the proportion of spam predicted is the same as the proportion observed. Produce a confusion matrix and report its accuracy.

3. Thoroughly read the description at the UCI archive of how the dataset was constructed. Is the spam/non-spam accuracy likely to be higher with this dataset than in real life? Why or why not? What can you say about the generalisability of the classifier to particular populations of text users?