Homework 4

Due: Wednesday February 20 at 11:59pm via compass2g

Use RStudio for all exercises. Efficiency is important. Use efficient programming techniques and modular programming as discussed in class, and make use of functions we have already created when possible.

You should provide one script (a `.R` or `.rmd` file) that contains all the code and includes code comments noting which code is for which exercises. You will also need to show and comment on the results, so place the results in a Word (or Open Office or HTML or PDF) document and write sentences to answer the questions, or use `knitr` to programmatically create your document. **Script files must be the actual script files**, not unevaluated code pasted into some other document.

Include your name in the name for each file submitted ('<Your-First-Name> <Your-Last-Name> HW#.R', e.g. 'JaneDoeHW4.R'). Any code based on code from elsewhere (e.g. code provided with the text) **must reference in code comments** the source of the original code.

Some initial setup code is provided in **HW4Setup.R** in the Homework 4 directory in compass.

Exercises for All Students

1) Create a function `computeMsgLLR2` which implements the following log of ratios of products of probabilities formula for the log likelihood ratio statistic:

$$\log\left(\left(\prod_{in\ msg} P(word\ present|\ spam)\right) / \left(\prod_{in\ msg} P(word\ present|\ ham)\right)\right) + \log\left(\left(\prod_{not\ in\ msg} P(word\ absent|\ spam)\right) / \left(\prod_{in\ msg} P(word\ absent|\ ham)\right)\right)$$

   Compare the results from this definition with the results from the `computeMsgLLR` function used in the text which used the sum of differences of log probabilities.
   Specifically, compare accuracy for this formula compared to the one used in class (Hint: to estimate relative accuracy you should look at (observed-expected)/expected, and treat the results from `computeMsgLLR2` as observed and the results from `computeMsgLLR` as expected) and note any issues that arise with non-representable numbers (e.g. very large or very small intermediate results that result in infinite, incorrect 0, or not a number results from your function).
2) Do exercise **Q.13** from page 167 of *Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving*, by Deborah Nolan and Duncan Temple Lang. Within the exercise, construct two functions: one that counts the number of yelling lines, and one that gives the percentage.
3) Check that the `hour` feature in `emailDF` gives valid values for all of the email messages. Then perform descriptive analysis to compare this feature for spam and ham, and comment on the possibility of using this feature to classify email.

Additional Exercises for Graduate Students

4) Do exercise **Q.14** from page 167 of *Data Science in R: A Case Studies Approach to Computational Reasoning and Problem Solving*, by Deborah Nolan and Duncan Temple Lang.