#homework (3 questions, A-C. Make sure to expand all bullets!)

- A: Regularized regression
  - The goal of this exercise is to familiarize yourself with the `glmnet` package, and ridge regression (L2-regularlization) and lasso (L1-regularization), by applying it to some simple simulated data.
  - i) Install the `glmnet` package and run the code in https://github.com/stephens999/stat34800/blob/master/analysis/glmnet_intro.Rmd. Add some additional code chunks of your own to investigate further and check your understanding. For example, you could:
    - Simulate some independent test data from the same model and check that the prediction error of different methods is comparable with the CV results.
    - Plot the (non-intercept) coefficients obtained from ridge regression and lasso against the true values used in the simulation and discuss the "shrinkage" that is occurring.
    - Plot the estimated (non-intercept) coefficients against the "theoretical" expectations you would expect if the predictors were orthogonal. Eg the "soft thresholding" property for the Lasso. (Note that the predictors here are not orthogonal, so the theory will certainly not hold precisely - does it hold approximately?)
    - Check that indeed the sum of absolute values of the coefficients is decreasing along the lasso path.
    - When you have finished, write a brief summary of what the code is doing, what you examined, and what you learned.
  - ii) Note that the simulation in i) involves a non-sparse setting: every predictor has an effect on Y. This might be expected to favor ridge regression over Lasso since ridge regression tends to produce non-sparse solutions, whereas Lasso tends to produce sparse solutions. So now modify the simulation in i) to simulate a sparse scenario, where only 10 of the 100 predictors actually affect Y. [Note that you may or may not have to modify the residual variance to make the problem "not too easy" and "not too hard"]. Investigate whether ridge regression or lasso provide better predictions in this setting.
- B: Conjugate Bayesian inference problems
  - Preparation: read through the introductory vignettes on fiveMinuteStat:
    - https://stephens999.github.io/fiveMinuteStats/bayes_beta_binomial.html
    - https://stephens999.github.io/fiveMinuteStats/bayes_conjugate.html

- i) Do the exercise at the end of https://stephens999.github.io/fiveMinuteStats/bayes_conjugate.html
  - ie show that the Gamma distribution is conjugate for estimating a Poisson rate.
- ii) Suppose you observe data $X \sim N(\theta, 1/\tau)$ [so τ is the inverse of the variance, also known as the "precision"]. Assume a prior distribution for θ of $\theta \sim N(\mu_0, 1/\tau_0)$.
  - Show that the posterior distribution for θ is $\theta|X \sim N(\mu_1, 1/\tau_1)$ where
    - $\mu_1 = wX + (1-w)\mu_0$
    - $\tau_1 = \tau + \tau_0$
    - $w = \tau/\tau_1$
  - Explain how this can be interpreted as providing a "shrinkage" estimate of θ, with shrinkage towards the prior mean.
  - Notes:
    - The posterior precision is the sum of the prior precision and data precision - so collecting data always increases the precision of your information about θ.
    - The posterior mean, μ_1, has a very intuitive form, which is the weighted sum of the prior mean and the data, where the weight depends on the precisions. If the data are very precise compared with the prior then the data dominates. If the data are very imprecise then the prior dominates.
    - Working with the precision, instead of the variance, is a standard trick in Bayesian inference to make the algebra a bit easier to manage.

- C: Empirical Bayes shrinkage
  - The goal here is to implement Empirical Bayes shrinkage for the normal means problem with normal prior: $X_j|\theta_j, s_j \sim N(\theta_j, s_j^2)$ with $\theta_j|\mu, \sigma \sim N(\mu, \sigma^2)$, $j = 1, \ldots, n$. known $l(g)$
    - a) derive an expression for the log-likelihood $l(\mu, \sigma) = \sum_j \log p(X_j|\mu, \sigma, s_j)$.
    - b) write down the expression for the posterior mean $E(\theta_j|X_j, \mu, \sigma)$.

- c) use these expressions to implement a function `ebnm_normal` to do EB shrinkage. Your function should input $x = (x_1, \ldots, x_n)$ and $s = (s_1, \ldots, s_n)$ and output the maximum likelihood estimates $(\hat{\mu}, \hat{\sigma})$ for μ, σ and the vector of posterior means $E(\theta_j | X_j, \hat{\mu}, \hat{\sigma})$. [You will need to use a numerical optimizer, like the R function `optim` , to optimize $l(\mu, \sigma)$; it may be better to optimize over $\eta := \log \sigma$ to avoid the non-negative constraint on σ.]
- d) Demonstrate the code on a simulation to show that a) the estimates of μ and σ are sensible compared with the truth; b) the posterior means provide better accuracy than the maximum likelihood estimates.
- e) Apply your code to the "8-schools problem" at http://andrewgelman.com/2014/01/21/everything-need-know-bayesian-statistics-learned-eight-schools/