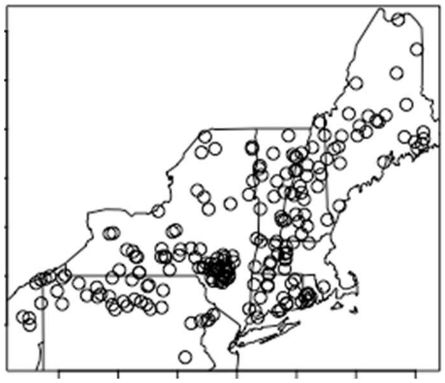
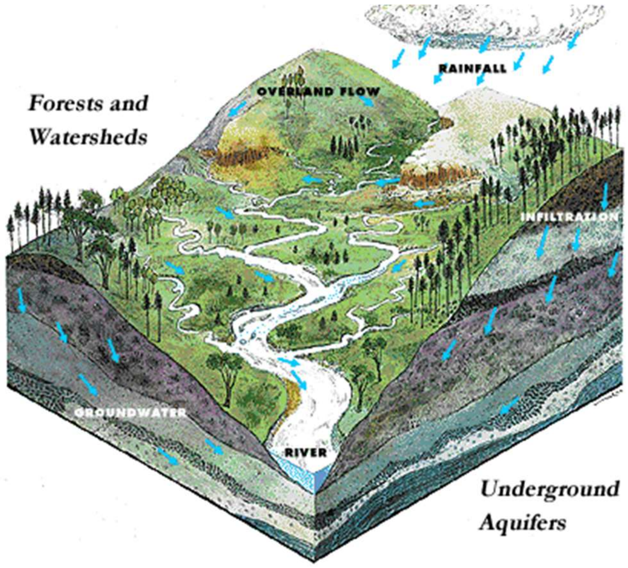
BEE 4310/6310: Multivariate Statistics for Environmental Applications
Assignment #2 (10 points)

Include responses to everything below in bold, and make sure your final assignment is well organized (i.e., numbered responses, full sentences, code attached at the end). This makes it easier to grade (and easier to give partial credit).

This assignment will walk through some of the practicalities of applying the ridge and lasso regression methods. We will fit and test regular (OLS), ridge, and lasso regressions for average annual runoff vs. many other characteristics for 211 catchments across the Northeast US. A map of the watershed centroids is shown, along with a picture of a characteristic watershed.





a) Download the gages2.data.txt file from Canvas under Assignment 2. This file contains:
   - the annual average runoff in the first column, and
   - 36 other catchment characteristics in the following columns

   These data are all from the Gages II database developed by the USGS. A description of the code for each covariate can be found in the excel file "gagesII_sept30_2011_var_desc". The dependent variable for prediction will be annual average runoff, and all other variables are independent covariates to be used to make the predictions.

b) Load the data into a variable called Gages2 using read.table().

c) Create a function called scale2 that takes as input one argument (a vector x) and returns a standardized version of x, i.e., by subtracting off the mean and then dividing by the standard deviation.

d) Using the apply function and your scale2 function, standardize each variable in Gages2, and put these results into a new data frame called Gages2.scale. You can use the function data.frame() to turn the results of the apply function into a data frame.

e) (**1 pt**) Using the data in Gages2.scale, fit a standard linear regression for annual average runoff vs. the other covariates for the 211 catchments. Do not include an intercept in this regression (since you already centered the runoff data round 0). **Using the summary() command on your regression object, create a regression table and include this table in your assignment. Which of the different predictors have a statistically significant relationship to flow based on a standard t-test framework? At what significance level? Finally, how would you interpret the magnitude of the regression coefficients for those covariates, i.e., how would you articulate how much runoff changes per change in the covariates? Think about the standardization you did in step c for your answer here.**

f) **(1 pts)** Using your fitted regression and the predict() function, create a vector of runoff predictions based on your model. **Calculate and report the root mean squared error (RMSE) of these predictions. Plot your predicted values against the observed values. Be sure to include a 1:1 line and to label your axes appropriately (what exactly is being plotted on each axis?).**

---

g) **(1 pts)** Calculate the VIF for each covariate. To make things easy, there is a vif() function in the “car” package, which you may choose to use here. Using the sort() function, sort the VIF values from smallest to largest. **Report the sorted VIF values, and comment on some of the largest VIF values you see in the database. Based on these results, why might you distrust the p-values you see based on the t-tests above, and do you think they underestimate or overestimate the significance of certain covariates? Explain why.**

h) You will now fit ridge and lasso regressions in R using the glmnet package. First, use the cv.glmnet function to calculate the best lambda value for the regression based on a K-fold cross validation (CV). The cv.glmnet function will automatically select a 10-fold CV and average the cross-validated mean square error for predictions over the 10 out-of-sample prediction sets (see documentation for more detail).

Within this function, set alpha=1 for lasso and alpha = 0 for ridge. Conduct the cross-validation for 100 lambda values between 0 and 0.5 (use sequence() to make these lambda values). Finally, you’ll need to change the variable types from a data.frame to a data matrix to use in glmnet. You can do this as:

```
X <- data.matrix(Gages2.scale)      # turns the data frame into a data matrix
x <- X[,2:ncol(X)]                  # pulls out the independent variables
y <- X[,1]                          # pulls out the dependent variable
cv.lasso <- cv.glmnet(x,y,…         #the start of the cv.glmnet function.
```

i) **(2 pts) In a two-panel, labeled figure (axes and title), plot the mean cross-validated error values again the lambda values for both ridge and lasso cross-validations.** The lambda values and the cross-validated mean error values are stored in the objects returned by cv.glmnet. Select the lambda with the smallest mean cross-validated error for both ridge and lasso regression (also stored in the cv.glmnet object). **Report the selected lambda values.**

j) **(2 pts)** Fit a final lasso and ridge model using the glmnet() function and the selected lambda values. **Report the fitted regression coefficients for the OLS, ridge, and lasso regressions and all covariates in a table. Compare the coefficient values, focusing on variables with high VIF values from (g). What sign and magnitude of coefficients does OLS regression assign to these variables? How do the lasso and ridge approaches differ in how they assign the coefficients compared to the OLS regression?**

k) We will now test which of these regression approaches provide the best out-of-sample predictions. Randomly split the database into 2 equal size and mutually exclusive subsets (subset 1 and subset 2). You can use the “sample()” function to select half of the 211 catchments at random without replacement for subset 1, and put the remaining catchments into subset 2:

```
subset1 <-sample(1:nrow(X),size=round(nrow(X)/2),replace=F)
subset2 <- (1:nrow(X))[-subset1]
```

l) Fit a linear regression via OLS to the data for subset 1.

m) Predict annual runoff using the OLS-based model for subset 2, and calculate the root mean square error (RMSE) of these predictions (hint: use the newdata argument in the predict() function).

n) Select lambda values for both lasso and ridge regressions based on the cv.glmnet function and the data in subset 1.

o) Fit lasso and ridge regression models to the subset1 data using the optimized lambda values.

p) Predict annual runoff using the fitted ridge and lasso models for the data in subset 2 (hint: for models from the glmnet package, use the newx argument in the predict() function to make predictions with a new set of data). Calculate the mean square error of these predictions.

q) Repeat this process 100 times, using different random splits of the data each time. You can do this by repeating the steps above (k-p) within a *for* loop, and saving the RMSE for all models in each iteration.

r) **(2 pts) Plot boxplots of the RMSE for predictions of subset 2 data under the OLS-based, ridge, and lasso regressions across the 100 iterations. Discuss how the prediction errors differ between the three models, what this implies about the utility of regularized regression for prediction, and why you think you see these prediction differences.**

s) **(1 pts)** Read the Dormann et al. (2013) article on a comparison of methods for multicollinearity. The article can be found on Canvas under the Assignment 2 folder. **In one or two lines, please describe how the lasso and ridge regression approaches compared to other methods, in terms of their out-of-sample prediction skill.**