Statistics 216
Homework 3, due Wednesday Feb 26, 2014.

1. We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain $p+1$ models, containing $0, 1, 2, \ldots, p$ predictors. Explain your answers:

   (a) Which of the three models with k predictors has the smallest *training* RSS?

   (b) Which of the three models with k predictors has the smallest *test* RSS?

   (c) True or False:

      i. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by forward stepwise selection.

      ii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by backward stepwise selection.

      iii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by forward stepwise selection.

      iv. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the $(k+1)$-variable model identified by backward stepwise selection.

      v. The predictors in the k-variable model identified by best subset are a subset of the predictors in the $(k+1)$-variable model identified by best subset selection.

2. Suppose that a curve $\hat{g}$ is computed to smoothly fit a set of n points using the following formula:

$$\hat{g} = \arg\min_{g} \left(\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int \left[g^{(m)}(x) \right]^2 dx \right),$$

   where $g^{(m)}$ represents the mth derivative of g (and $g^{(0)} = g$). Provide example sketches of $\hat{g}$ in each of the following scenarios.

   (a) $\lambda = \infty, m = 0$.

   (b) $\lambda = \infty, m = 1$.

   (c) $\lambda = \infty, m = 2$.

   (d) $\lambda = \infty, m = 3$.

   (e) $\lambda = 0, m = 3$.

3. **You may work in groups up to size 4 on this problem. If you do work in groups, write the names of all your group members on your problem set.**

   This problem works with the `body` dataset used in the in class session from Feb 12. The goal of this problem is to perform and compare Principal Components Regression and Partial Least Squares on the problem of trying to predict someones weight. While you can use any R tools at your disposal to complete the problem, `library(pls)` and `Lab 3` from `ISLR` will probably be very helpful and the problem set was written with these approaches in mind. If you have not already downloaded the data, please go to the class coursework page and do so. More information about this dataset can be found at `http://www.amstat.org/publications/jse/v11n2/datasets.heinz.html`.

   (a) Read the `body` dataset into R using the `load()` function. This dataset contains:

      - `X`: A dataframe containing 21 different types of measurements on the human body.
      - `Y`: A dataframe that contains the age, weight (kg), height (cm), and the gender of each person in the sample.

      Let's say we forgot how the gender is coded in this dataset. Using a simple visualization, explain how you can tell which gender is which.

(b) Reserve 200 observations from your dataset to act as a test set and use the remaining 307 as a training set. On the training set, use both `pcr` and `plsr` to fit models to predict a person's weight based on the variables in `X`. Use the options `scale = TRUE` and `validation='CV'`. Why does it make sense to scale our variables in this case?

(c) Run `summary()` on each of the objects calculated above, and compare the training % variance explained from the `pcr` output to the `plsr` output. Do you notice any consistent patterns (in comparing the two)? Is that pattern surprising? Explain why or why not.

(d) For each of models, pick a number of components that you would use to predict future values of weight from `X`. Please include any further analysis you use to decide on the number of components.

(e) Practically speaking, it might be nice if we could guess a person's weight without measuring 21 different quantities. Do either of the methods performed above allow us to do that? If not, pick another method that will and fit it on the training data.

(f) Compare all 3 methods in terms of performance on the test set. Keep in mind that you should only run one version of each model on the test set. Any necessary selection of parameters should be done only with the training set.

4. **You may work in groups up to size 4 on this problem. If you do work in groups, write the names of all your group members on your problem set.**

In this problem we will build our R function that can be used to fit a cubic spline to data by using a truncated power basis.

(a) Write a function `h` such that `h(x, z)` $= (x - z)_+^3$.

(b) Using `h` and `sapply`, write a function `hs` such that `hs(xs, z) =`
`(h(xs[1],z),h(xs[2],z),..., h(xs[length(xs)],z))`

(c) Using `hs`, write a function `splinebasis` such that `splinebasis(xs, zs)` returns a
`length(xs)`×`(length(zs) + 3)` matrix with the following columns:
`xs`, `xs`^{2}, `xs`^{3}, `hs(xs,zs[1])`, `hs(xs,zs[2])`, ..., `hs(xs,zs[length(zs)])`.

(d) Now, let's generate some data:

```
set.seed(1337)
x = runif(100)
y = sin(10*x)
```

(e) When we fit a cubic spline with knots at the points `knots`, that is the same thing as fitting a linear model predicting `y` from `splinebasis(xs, zs)`. Using `lm()` and your `splinebasis()`, fit a spline to the data we generated using $k = 3$ evenly spaced knots (`knots = c(1/4,2/4,3/4)`). Make a plot that clearly shows the original points, the curve those points were generated from, and a curve that represents your fitted spline. Note that the curves can be generated either using `curve()` or by using `lines()` with many points along the curve. Does the spline do a good job approximating the function?

(f) Repeat the above process for $k = 1, 2, \ldots, 9$. Do NOT copy paste your code.

(g) As we let the number of knots increase, do we expect our fitted curve to continually become a better approximation to the true curve? Why or why not?