Multiple Regressions

Here we work with the same professor rating data set we used in OI Lab 8. In the interest of time, we set aside assumptions tests, but we approach analyses more completely, based on two philosophies: 1) let the machine do the thinking vs. 2) think for ourselves. *As you might guess, option 2 is preferred*!

1. Let the Machine Think for You This is the lazy way to run a multiple regression - throw a bunch of predictor variables at a response and see what sticks, based on the idea that significant factors must be important (never mind explaining them). As you might guess, this may be appropriate for *exploratory* work if we have no idea what to expect, but bears little resemblance to hypothesis-based research. So it has its [limited] place, but does not replace thinking.

1. Load and attach the professor grading data set. The data set is at: http://www.openintro.org/stat/data/evals.RData

   Also install (if needed) and load the "car" package – Companion to Applied Regression

2. We first make a model where every potential factor is listed to describe the response (score). Copy and run this command (you're welcome – I typed all these in for you...):

   ```
   fullmodel <- lm(score ~ rank + ethnicity + gender + language + age +
   cls_perc_eval + cls_did_eval + cls_students + cls_level + cls_profs +
   cls_credits + bty_avg + pic_outfit + pic_color)
   ```

3. You may notice only bty-avg was used here; other bty- variables made that average and I considered them redundant. Ask for the summary output of the model:

   ```
   summary(fullmodel)
   ```

4. First we check to see if some of these predictor variables are closely correlated – if so, this *multicollinearity* violates the assumption of independence. Closely correlated variables artificially ***inflate*** the explained variance of each other. The measure we use is the generalized Variable Inflation Factor (GVIF): values above 10 indicate strong correlation and you should omit those variables from subsequent models. So run this command (in the `car` package):

   ```
   vif(fullmodel)
   ```

5. A rule of thumb: VIF > 10 means strong multicollinearity. If you get two highly correlated (collinear) variables, omit from the model the one variable that has least to do with the evaluation scores. Call the new model "fullmodel2," run it, and check again the VIF values for that new model.
6. A few factors look like they significantly affect score in fullmodel2, but many do not. So we let the machine weed them out. First we do a forward-selection – this adds factors one-at-a-time, keeps significant ones, and builds a model. *The order in which factors are entered matters*, and added variables may render already-included variables non-

significant (*a problem with this method!*)

```
stepfor <- step(fullmodel2, direction="forward")
summary(stepfor)
```

7. Now we start with the full model and work <u>backwards</u> to remove nonsignificant factors one-at-a-time, ending with a more efficient model. This is less prone to problems of order because least significant factors are removed first. *But!* variables may be dropped that could be significant when added to the final reduced model.

```
stepback <- step(fullmodel2, direction="backward")
summary(stepback)
```

Do the forward-selection and backward-selection models differ? *This is a big problem! Beware of papers that use one or the other, and <u>especially beware of forward-selection</u>*. One key difference between former approaches and this current approach in R: analyses used to be based on R^2, whereas this modern approach is based on AIC scores, which works better. Also, R let's us run stepwise selection in *both* directions, which is better and default:

8. Run in both directions:

```
stepboth <- step(fullmodel2) #note: direction="both" is the default
summary(stepboth)
```

9. Let's compare the alternative models. Load the bbmle package and then run this command:

```
AICctab(stepback, stepfor, stepboth, fullmodel2, weights = TRUE, delta =
TRUE, base = TRUE, sort=TRUE)
```

Which model(s) are most plausible? Does this make sense? Notice that having more variables does not necessarily make a model “better” by this approach.

Importantly, *these analyses deferred all important thinking to the machine*. That means we abdicated hypotheses and inference to whatever we obtained with the step algorithm. *<u>It would be far better to hypothesize, then test</u>*. So let's walk through the second, preferred approach. First some hypotheses.

2. <u>Hypothesis Testing</u> Look at the data collected for this study – there are hypotheses implicit in the variables collected. Here is a *partial* list (in the interest of time), where score depends on … :

A) a professor's **`rank, ethnicity, gender,`** accent (**`language`**)**,** and **`age,`** because older, white males are perceived as “authority” figures relative to others (says the old white guy...). If so, then old white guys would have greater, positive coefficients; others would have lesser, even negative coefficients.

B) a professor's looks, including:
   1. their average “beauty” rating (**`bty_avg`**) because people respond positively to beauty
   2. **`age,`** which should have a negative coefficient while **`bty-avg`** should be positive (we set aside other bty_ variables here – they made bty_avg)

3. the professor's clothes (**`pic_outfit`**) in the picture, and
4. whether it was color or B&W (**`pic_color`**) may also matter.

C) demographics of students (**`cls_perc_eval, cls_did_eval`**) because larger classes tend to score lower (less personal interaction). Here we just use `cls_did_eval` because we already learned it is closely correlated with `cls_students` but is more directly related to evaluation scores

D) class structure (**`cls_level, cls_profs, cls_credits`**) because upper-level, multi-instructor, multi-credit courses tend to score lower (perceived as harder)

E) The full model used at #2 above, because both professors and students make a class work well (or not). Again, simplify to retain only significant variables.

Model A is set up here – run that and then follow with models B-E.

10. Run this pair of commands:

```
modelA <- lm(score ~ rank + ethnicity + gender + language + age)
summary(modelA)
```

11. Now use this model A template for models B-E by substituting terms on the right side of the "~" and renaming the models.

12. Here comes the punchline: compute an AIC table for the models.

Which model is the most plausible? What can you say about the hypotheses (A-E)?

After all this, how well does our best model represent the total variation in the data? Does it differ from the stepboth model at #8?

Did the hypothesis-testing approach reveal more to you about professors' evaluation scores than the machine-based-analyze-everything approach?

Notice what we did –

- we tested hypotheses. Ahhhhh, Science!
- we used AIC-based model comparison to help decide which of our hypotheses was most plausible.
- The results of the machine-based stepwise approach and our hypothesis testing end with similar conclusions, but we thought it all through more carefully with hypotheses. And that is the point of hypothesis testing!

Finally, how much variation in professor scores did we represent with our overall model? And notice that this had nothing to do with the actual course content and delivery! Presumably, information on the course and effectiveness of teaching would account for substantial variation here. But analyses here showed biases matter, too.

Let's wrap up with a useful tool. A multiple regression predicts the response variable. But which predictor variable is most important? That is a common question, but there is much debate on a correct answer. A reasonable way to evaluate among predictor variables is a classification and regression tree (aka CART), which recursively partitions the response variable into subsets based on its relationship to predictor variables. The predictor variable at the first split yields the greatest change in explained deviance (like minimizing SSE in an ANOVA).

14. Install and load the `tree` package and then run this command on our final, “best” model:

```
G2tree <- tree(score ~ gender + language + age + cls_perc_eval +
cls_credits + bty_avg + pic_color)
plot(G2tree)
text(G2tree)
```

This makes a tree, plots it and labels it. The predictor variables appear in decreasing order of importance to the model. Which predictor variable is most important to our results?