STA9750
More regression with SAS

Topics

1) Transformations
2) Categorical predictors
3) Model building

TRANSFORMATIONS

Sometimes our models don't fit the data (the residuals don't look good). When that happens, transforming the response or some of the predictors might help.

In lecture, we covered log transformations, which are probably the most commonly used ones. I think that the following handout explains log transformations very well:

https://www.cscu.cornell.edu/news/statnews/stnews83.pdf

You can transform variables in DATA steps.

For example, suppose you have loaded the dataset "evals_sas.csv" (on the course website) into SAS and named it "evals". If you want to take the log of "bty_avg", the square-root of "age", and the square of "score", the following code will do that for you:

```
DATA evals2;
      SET evals;
      logbty = LOG(bty_avg);
      sqrtage = SQRT(age);
      scoresq = score**2;
RUN;
```

CATEGORICAL PREDICTORS

As we saw in lecture, categorical predictors must be treated with care. If you code them as numeric and blindly put them in the model, you're making some linearity assumptions that may not make sense [For example, if you had coded the variable into different numeric labels, your predictions for the categories would change].

We saw that "dummy" variables avoid this problem. We also discussed that if we have a categorical and a quantitative variable in the model, we have some flexibility on how to model the way they affect the response. If we put in an interaction (product) term, we assume that the effect of a quantitative variable depends on the values of the categories (different slopes). If we don't put an interaction term, the effect of the quantitative variable is the same across categories (parallel lines).

The following section of the STA501 course at Penn State explains all of this very well:

https://onlinecourses.science.psu.edu/stat501/node/301/

There are at least 2 ways of dealing with categorical predictors in SAS:

1) Let PROC GLM deal with the dummies.
2) Hand-code the dummies in a DATA step and use PROC REG instead.

Option 1) PROC GLM

Suppose we're still working with "evals_sas.csv", and assume we want to model "score" as a function of "bty_avg" (which is quantitative) and "gender" (which is categorical).

We can fit a model with the additive affects and an interaction term (we model the relationship between "score" and "bty_avg" with different intercepts and slopes for men and women):

```
PROC GLM data=evals;
   CLASS gender;
   MODEL score=bty_avg gender bty_avg*gender / SOLUTION ;
RUN;
```

If we only want an additive model (parallel lines for men and women)

```
PROC GLM data=evals;
   CLASS gender;
   MODEL score=bty_avg gender / SOLUTION ;
RUN;
```

If we only want an interaction (I generally recommend against doing this; in this case, it means that the lines for men and women have different slopes and must meet at bty_avg = 0, which is odd):

```
PROC GLM data=evals;
   CLASS gender;
   MODEL score= bty_avg bty_avg*gender / SOLUTION ;
RUN;
```

The part where we write "`/ SOLUTION`" is needed so that SAS computes the regression coefficients. Otherwise, SAS will output some plots and sums of squares, but not the regression coefficients!

PROC GLM doesn't give us diagnostic plots by default. We force SAS to show them by adding "plots = all" right after "data = <dataset>". We can also create a new dataset with leverages, Cook's distances, intervals, etc. using the same commands we used for PROC REG. We can also change the significance level of the intervals with the option "alpha = <significance_level>". For example, the code below will output plots with regression diagnostics and a dataset named "diag" which includes residuals, Cook's distances, and 99% confidence intervals for the mean and prediction intervals.

```
PROC GLM data=evals alpha =0.01 plots= all;
      class gender;
      model score = bty_avg gender;
      output out = diag
                  r = resid
                  cookd = cooks
```

```
                  lclm = lomean
                  uclm = upmean
                  lcl = lopred
                  ucl = uppred;
RUN;
```

Option 2) Hand-coded dummies

You can hand-code dummies, too. Suppose you want to model "score" as a function of "bty_avg" (quantitative) and "rank" (categorical, with categories "tenure-track", "tenured", and "teaching"). If we take tenured faculty as the baseline and create indicators (dummies) for tenure-track and teaching faculty:

```
DATA dumb;
      SET evals;
      IF rank = 'tenured' THEN
            do;
                  track = 0;
                  teach = 0;
            end;
      ELSE IF rank = 'tenure track' THEN
            do;
                  track = 1;
                  teach = 0;
            end;
      ELSE
            do;
                  track = 0;
                  teach = 1;
            end;
RUN;
```

Then, you can use PROC REG to run the regressions you want. For example, if you want a model that models the relationship between "score' and "bty_avg" with parallel lines for different ranks, you can do that with:

```
PROC REG data=dumb;
      model score = track teach bty_avg;
RUN;
```

PROC REG isn't as nice as PROC GLM in dealing with interactions. If you want to have an interaction in the model, you have to code it up (you can't just write "teach*bty_avg" in the "model" statement).

MODEL BUILDING

In lecture, we saw different strategies to model building.

Two general classes are:

1) Go through all subsets, and score models according to some criterion
2) Search for good models (sometimes you can't just go through all models) using some heuristic

This is explained in some detail in a set of slides that you can find on the course website.

If you want to go through all your models and you're using PROC REG, you can do that pretty easily. Suppose you're working with the IQ data ("iq.txt" on the website) and you named it "iq". If you want to find the best model for "PIQ", given "brain" (which was a measure of how big your head is), "height", and "weight", you can use this command:

```
PROC REG data=iq;
      MODEL PIQ = brain height weight/ selection = adjrsq bic cp;
RUN;
```

It will go through all the subsets of variables in and out of the model, and report adjusted R^2, BIC, and Mallows' C_p

You can find more information on the criteria supported by PROC REG here:

https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_reg_sect030.htm

You can tell PROC REG that you one models that have a minimum and a maximum number of variables with the options "start' and "stop". You can also tell SAS that you only want to see the best, say, 100 models according to a criterion with the option "best". For example, if you want to see the best 2 models that have between 2 and 3 variables according to BIC:

```
PROC REG data=iq;
      MODEL PIQ = brain height weight/ selection = bic start = 2 stop = 3
best = 2;
RUN;
```

You can do forward, backward, and stepwise selection (using p-values as the criterion for entry and staying in the model) as well:

```
PROC REG data=iq;
      MODEL PIQ = brain height weight/ selection = forward slentry = 0.05;
RUN;

PROC REG data=iq;
      MODEL PIQ = brain height weight/ selection = backward slstay = 0.05;
RUN;

PROC REG data=iq;
      MODEL PIQ = brain height weight/ selection = stepw    ise slentry =
0.05 slstay = 0.05;
RUN;
```

The parameters "slentry" and "slstay" stand for "significance level for entry" and "significance level for staying".

Model building with categorical predictors

If you have categorical predictors, things become a little tricky. If you're working with dummies, the "variable" isn't just one column: it's all the columns that have dummies related to the variable. PROC

GLMSELECT deals with this issue automatically; unfortunately, it doesn't do "all subsets selection", but it does forward, backward, and stepwise selection.

This is an example with the beauty data, where I do stepwise selection with significance level of entry equal and significance level of staying of 0.05:

```
proc glmselect data = evals;
   class gender;
   model score=bty_avg gender bty_avg*gender / selection = stepwise(select =
SL SLE=0.05 SLS = 0.05);
run;
```

You can also use other criteria that are not p-values quite easily.

For example, if you want to use BIC:

```
proc glmselect data = evals;
   class gender;
   model score=bty_avg gender bty_avg*gender / selection = stepwise(select =
BIC);
run;

proc glmselect data = evals;
   class gender;
   model score=bty_avg gender bty_avg*gender / selection = backward(select =
cp);
run;

proc glmselect data = evals;
   class gender;
   model score=bty_avg gender bty_avg*gender / selection = backward(select =
adjrsq);
run;
```

If you want to know more about the different selection criteria and the options of GLMSELECT, I think that the following section of the documentation of PROC GLMSELECT is useful and accessible:

http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_glmselect_sect016.htm

If you're comfortable working with models that contain strict subsets of dummy variables (for example, a model that has only an indicator for "tenure-track", in the beauty example) and you want to do "all subsets" selection, you can accomplish that by coding the dummy variables by hand, and then using the all subsets features of PROC REG.