Performing Diagnostics and Remedial Measures on an ANCOVA Model

This demonstration shows how to perform diagnostics and remedial measures on an ANCOVA model.

First, we use PROC GLMSELECT to output the design matrix for further analysis in PROC REG. In the PROC GLMSELECT statement, the OUTDESIGN= option saves the design matrix to a temporary data set named **design**. In PROC GLMSELECT, the CLASS and MODEL statements specify the same model that was shown earlier in PROC GLM: **BPChange** = **Treatment**, **BaselineBP**, and the interaction of **Treatment** and **BaselineBP**. At the end of the MODEL statement, SELECTION=NONE prevents SAS from performing the default process of selecting variables for the model. The design matrix is the only output that we want, so we use the statement ODS SELECT NONE above the PROC GLMSELECT step to suppress all printed output.

When PROC GLMSELECT runs, it creates an automatic macro variable named **_glsmod**, which stores all of the independent variables in the model. The %PUT statement tells SAS to write the value of the **_glsmod** macro variable (that is, the list of the variables) to the log.

Let's submit the PROC GLMSELECT code.

```
ods select none;
proc glmselect data=mydata.trials outdesign=design;
   class treatment;
   model bpchange = treatment|baselinebp / selection=none;
run;
%put macro variable _glsmod=&_glsmod;
```

First, we look at the design matrix. We navigate to the work library and open the **design** data set. This design matrix has a column for each of the following independent variables: the intercept, each of the three treatments, the continuous predictor **BaselineBP**, and the three interaction effects (that is, each treatment with the continuous predictor variable). At the end is a column for the dependent variable **BPChange**.

Now let's look at the log. As expected, the log displays the names of the variables in the model, which are stored in the **_glsmod** macro variable. These names correspond to the columns that we just saw in the design matrix, with the exception of the intercept.

Next, we want to analyze the design matrix using PROC REG. First, the ODS SELECT statement specifies the names of the plots that we want PROC REG to create: the parameter estimates (because those include the variance inflation factors), the diagnostics panel, the DFFITS plot and the DFBETAS panel. The PROC REG statement specifies **design** as the input data set. The PLOTS= option specifies the DFBETAS and DFFITS plots because these are not generated by default. In the MODEL statement, notice that the model is specified more concisely than earlier. Specifically, the **_glsmod** macro variable is used to reference the names of the independent variables in the model. At the end of the MODEL statement, options specify that we want to calculate the variance inflation factors and influence diagnostics.

Let's run PROC REG.

```
ods select ParameterEstimates DiagnosticsPanel
    DFFITSPlot DFBETASPanel;
proc reg data=design plots=(dfbetas dffits);
   model bpchange=&_glsmod / vif influence;
title 'Check Collinearity on ANCOVA Model';
run;
quit;
```

At the top of the PROC REG results, the Parameter Estimates table has the same parameter estimates that PROC GLM produced earlier. This is, in fact, the same model. We'll come back to these tables later. For now, we'll scroll down to the diagnostic plots so that we can check the assumptions for ANCOVA. First, do we have normally-distributed residuals? The histogram of residuals in the bottom left corner looks pretty good – this is a normal distribution. In the Q-Q plot immediately above, the data hugs the line well—again, indicating a normal distribution. Next, we want to verify that we have equal variances. In the top left corner of the panel, the plot of residuals versus predicted values shows no drastic changes from left to right – no funnel shape as we've seen in some cases before. So this plot supports the assumption of constant variance. In other words, the residuals appear to be a random scatter around a zero reference line and display no heteroscedasticity.

We can also use the diagnostic plots to evaluate the fit of the model. In the middle of the panel is the plot of the observed values versus the predicted values of **BPChange**. In this plot, most of the points fall along the 45-degree line, so the actual and predicted values match fairly well. This indicates a good model fit. Immediately below, the Fit-minus-Mean residual plot shows a bigger vertical spread on the Fit-minus-Mean side than on the Residual side. This indicates that the model accounts for a good deal of the variability in the change in diastolic blood pressure measurements, as does the adjusted R square of 0.9002 that is shown in the bottom right corner of the panel.

Finally, we can also examine these plots for outliers and influential points. In the residual plot in the top left corner, we can see the two main outliers. We might want to go back and look at those two data points later. To the right, the two RSTUDENT plots are helpful in identifying influential observations. The RSTUDENT plots are scaled, so we can expect that 95% of these RSTUDENT residuals occur within the bounds of −2 to 2. This provides a better indication of the degree of the influence. The second RSTUDENT plot displays the RSTUDENT values versus the leverage, which is a function of the distance between each point and the center of the distribution. The observation at the bottom has both a large value of the RSTUDENT score and a leverage value higher than the suggested cutoff that is indicated by the vertical line. This is another point that we probably want to investigate. It wouldn't be surprising if this same observation is the one that shows a high Cook's D score in the plot immediately below. A high Cook's D score indicates that the observation has a large influence on the collective parameter estimates.

Below the diagnostic plot are the plots produced by the INFLUENCE option: DFFITS and DFBETAS. These plots indicate that several observations might be exerting influence on the model and should be investigated. In the DFFITS plot, several lines extend beyond the horizontal bounds above and below the zero line. These represent data points that exert a large influence on the predicted values—that is, on the predictions. Next is the panel of DFBETAS plots. Whereas Cook's D indicates whether a point has an influence on all the parameter estimates collectively, the DFBETAS indicate a point's influence on each individual parameter estimate. Observations that are outside the bounds are influential on the effect in each plot. Remember that this is an overparameterized model, so there is no parameter effect for the *Placebo* treatment. However, the Intercept plot tells us about the *Placebo* treatment. For **BaselineBP**, likewise, there is no parameter effect for its interaction with the *Placebo* treatment, but the **BaselineBP** plot tells us about its interaction with the *Placebo* treatment.

Let's go back to the tables at the top of the PROC GLMSELECT results to look for multicollinearity. In the Parameter Estimates table, let's focus on the variance inflation factors in the last column. As a cutoff, we used a VIF value of 10; anything above this value indicates strong multicollinearity. Many of the VIF values in this table are greater than 1000, which indicates a serious multicollinearity problem. We will need to fix this.

Multicollinearity is frequently present for ANCOVA. Remember that including polynomial effects in a model necessarily introduces multicollinearity. Of course, there is a correlation between powers of variables, such as X, X^2, and X^3. There is also multicollinearity between interaction effects, which are another type of higher-order term. To reduce multicollinearity in a model that has any type of higher-order terms, we can center the data. We will use PROC STDIZE to center the continuous predictor (that is, the covariate) by subtracting the mean. Then we will reanalyze the data to see if the VIF values have been reduced.

Let's look at the code. In the PROC STDIZE statement, we specify the original data set. The METHOD= option specifies MEAN as the type of standardization. PROC STDIZE will subtract the mean from each value of the covariate **BaselineBP**, which is specified in the VAR statement below. In the PROC STDIZE statement, the OUT= option saves the standardized data to a temporary output data set named **trials2c**. (In that name, the c stands for "centered.") The RENAME= option renames the **BaselineBP** variable in the output data set to **BaselineBPC**.

After the data is centered, we need to run PROC GLMSELECT to rebuild the model based on the standardized data. As before, ODS SELECT NONE suppresses printed output. In the PROC GLMSELECT statement, we specify the standardized data set as the input. The OUTDESIGN= option specifies **design2c** as the name of the temporary data set that will contain the design matrix. The model specification is the same as earlier.

Again, we use PROC REG to run diagnostics, using the design matrix for the model that is based on the centered variables.

Let's run this code.

```
proc stdize data=mydata.trials method=mean
            out=trials2c (rename=(baselinebp=baselinebpc));
   var baselinebp;
run;

ods select none;
proc glmselect data=trials2c outdesign=design2c;
   class treatment;
```

```
   model bpchange = treatment|baselinebpc / selection=none;
title 'Check Collinearity on Centered ANCOVA Model';
run;

ods select ParameterEstimates DiagnosticsPanel
    DFFITSPlot DFBETASPanel;
proc reg data=design2c plots=(dfbetas dffits);
   model bpchange=&_glsmod / vif influence;
run;
quit;
```

At the top of the results, we look at the Parameter Estimates table. These parameter estimates can be used to write the regression equation for each treatment. To see the three equations and additional information related to the parameter estimates, click the Information button.

The values in the Variance Inflation column are much smaller than in the last model. In fact, the VIF values are well below the cutoff of 10. We have successfully dealt with the mullicollinearity!

Now let's look at the Parameter Estimate column. Compared with the model we created at the beginning of this demonstration, only the estimated intercepts (the first three values in this column) have changed. The slope parameters below (that is, the first three variable names that start with **BaselineBP**) are unchanged. To see why this is so, let's compare plots of the original and centered data with the regression lines overlaid.

In the code, the first PROC SGPLOT step will create a plot based on the original data set, **trials**. The Y variable is **BPChange** and the X variable is **BaselineBP**. The GROUP= option specifies **Treatment** as the grouping variable. The plot will display one line for each treatment group. A reference line is specified at 95 on the x axis.

The second PROC SGPLOT step will create a similar plot based on the standardized data set, **trials2c**. The only difference is that the reference line is now drawn at zero on the x axis.

Let's run this code and compare the two plots.

```
proc sgplot data=mydata.trials;
   reg y=bpchange x=baselinebp / group=treatment;
   refline 95 / axis=x;
title 'Original Data';
run;

proc sgplot data=trials2c;
  reg y=bpchange x=baselinebpc / group=treatment;
  refline 0 / axis=x;
title 'Centered Data';
run;
```

You can see that the only effect of centering **BaselineBP** is to shift the cloud of data points so that it is centered around 0 instead of (approximately) 95. The intercepts for the three equations change, but the slopes do not.

Close