Performing Model Diagnostics – Part 1

Let's assume that we selected a final model for the car company analysis that has three variables, **Hwympg** [linear], **Hwympg^2** [quadratic], and **Horsepower**. Now we use SAS to evaluate this model by checking for violations of the assumptions, model fit, and collinearity. We'll also identify any observations that appear to be influential.

The first thing we're going to do is run our model in PROC GLMSELECT to create the model terms that we need. Then we'll use PROC REG to provide diagnostic plots. In PROC GLMSELECT, we specify the **cars** data set. Then we use the OUTDESIGN= option to tell SAS to send variables for all of the predictors to the data set **d_carfinal**. As you learned in Lesson 1, we use the EFFECT statement to create polynomial terms for **Hwympg**. Specifying degree=2 tells SAS to produce linear and quadratic terms. The STANDARDIZE option specifies that the polynomial terms should be centered. Recall that this is a way to mitigate the effects of multicollinearity. In the MODEL statement we specify **Price** as the outcome. Then we include the polynomial effects for **Hwympg**, as well as **Horsepower**. We specify the option SELECTION=NONE, because we are not doing any model selection.

```
ods graphics / imagemap=on;

proc glmselect data=mydata.cars outdesign(addinputvars)=d_carfinal;
   effect q_hwympg = polynomial(hwympg / degree=2
      standardize(method=moments)=center);
   model price = q_hwympg horsepower / selection=none;
run;
```

Let's run the code. We're not interested in the output. In this case, we ran PROC GLMSELECT purely to create the variables that we will next analyze using PROC REG. Here's our data set. Notice that we have centered variables for **Hwympg** and **Hwympg^2**. Then we have the other variables that were included in our original data set.

Now let's review the PROC REG code. We specify the **d_carfinal** data set that we created in the previous program. We use the PLOTS=ALL option to tell SAS to produce all of the plots available in PROC REG (which is quite a few). The UNPACK option unpanels the display and produces a series of individual plots. The LABEL option tells SAS to label extreme observations.

```
proc reg data=d_carfinal plots(unpack label)=all;
   model price = &_GLSMOD / vif collin collinoint influence spec partial;
   id model;
   output out=check r=residual p=pred rstudent=rstudent h=leverage;
run;
quit;
```

Again in the MODEL statement our outcome is **Price**. We use the macro variable &_GLSMOD to give us the names of the predictors from the final model resulting from PROC GLMSELECT.

Then we specify options that turn on the diagnostic plots and statistics needed to diagnose multicollinearity and other issues. In this case we're requesting the following:

- VIF – variance inflation factors
- Collin and Collinnoint – to produce collinearity analyses
- influence, which requests a table displaying the values of all the statistics for influential observations, as well as turning on the plots
- spec – a formal test for homogeneity of variance, and
- partial, which produces the partial leverage plot

The ID Statement lists the variable to use to identify observations, in this case, the variable indicating the model of the car. The OUTPUT statement specifies that the output will be sent to a data set named **check**. Each of the single letters in this statement indicates the statistic to be output, along with the name of that variable in the output data set. Let's go ahead, run the program and review some of the output. Note that you've already seen the top part of this output in Lesson 1.

The F-test shown in the Analysis of Variance table indicates that at least one of the predictors in this model is a highly significant predictor of our outcome. The R-square value is 0.72, indicating that a large proportion of the variability in our outcome is being explained by our model. In the Parameter Estimates table, we'll look at the last column, the Variance Inflation Factor. Remember, our recommendation is that if the VIF is greater than 10, that's indicative of

multicollinearity. Also, a condition index value greater than 30 indicates moderate multicollinearity. Here the VIF values are around 2 and 4, indicating that multicollinearity is not a problem in our data. Remember that this is what we expect, because we've already centered our polynomial terms. Therefore, we don't need to inspect the collinearity diagnostics.

Next we'll review the output from the SPEC option, giving us a formal test, one part of which diagnoses heterogeneity of variance.

The SPEC option performs a test so that the first and second moments of the model are correctly specified. The null hypothesis for this test includes the following:

- The errors are homoscedastic (assuming that we have homogeneity of variance)
- The errors are independent of the predictor variables.
- Several technical assumptions about the model specification are valid. For example, the correct model is specified. For details, see theorem 2 and assumptions 1 through 7 of White (1980).

If this test is significant, you don't know which piece of the null hypothesis is being violated. We can see from the *p*-value of 0.0359 that this test is significant. But we also have a note in our log – remember that it's always good to look at your log, even if your code produces the output that you expect. The note here means that the results for this test might not be valid. We have some evidence that we might not have homogeneity of variance, so we will need to use our plots in conjunction with this test to confirm if this piece of the null hypothesis is not valid. This warning message tends to appear when you have dummy variables or higher-ordered terms in the model. You might consider an alternative to evaluate the constant variance assumption, such as computing the Spearman Rank Correlation Coefficient between the absolute values of the residuals and the predicted values.

Note that the SPEC test is a simultaneous test of multiple null hypotheses. If you reject the test, then you cannot identify, without further investigation, which particular null hypothesis caused the rejection.

Next let's look briefly at the table of output statistics. This table displays statistics useful for detecting influential observations for every observation in the data set. The table is very long and hard to eyeball. The power of this information comes in the plots.

Now we'll look at other plots that assist in evaluating the assumptions. The Distribution of Residuals for Price plot assesses the assumption of normality. The red line indicates observed values; the blue line references the normal distribution. As you can see, this data looks normal. The normal Quantile plot, or Q-Q plot, also looks pretty good. The observations are closely clustered around the reference line. The plot of Residuals vs. Predicted Values, on the other hand, does exhibit a pattern. Notice that residuals are larger on the right side of the plot than on the left, so there's some fanning to the right. This pattern indicates a lack of homogeneity of variance and gives support to the finding from the SPEC test. Note that observations with higher predicted values have greater variability than observations with lower predicted values.

To help assess homogeneity of variance, we're going to compute the Spearman Rank Correlation between the absolute values of the residuals and the predicted values.

To compute the absolute value of the residuals, we'll use a DATA step that takes the output data set from PROC REG, check, and then creates the variable named **Abserror**, which is equal to the absolute value of the residuals.

```
data check;
   set check;
   abserror=abs(residual);
run;
```

Let's run this code and briefly inspect the data set, **check**. You can see the absolute value of the residuals, as well as the residuals themselves.

Now let's use PROC CORR to compute the Spearman Rank Correlation by correlating the absolute values of the residuals to the predicted values. The predicted values came from PROC REG.

```
proc corr data=check spearman nosimple;
   var abserror pred;
run;
```

Let's run the PROC CORR program.

We see that the Spearman rank correlation coefficient between the absolute values of the residuals and the predicted values is 0.6. The highly significant *p*-value (<.0001) indicates a strong correlation between the absolute values of the

residuals and the predicted values. The positive correlation coefficient indicates that the residuals increase as the predicted values increase. This matches what we saw in the plot of Residuals vs. Predicted Values.

Close