Fitting a Lognormal Regression Model

Recall from the diagnostic plots we reviewed in the previous demonstration that the assumption of normality for the **cars** data set looked fine. But we did observe heterogeneous variance. In particular, the variability of price was higher for the more expensive cars than for the lower-priced cars. This was clear from the fanning to the right in the plot.

In order to correct this heterogeneity of variance, we can use PROC GLIMMIX to apply a lognormal distribution to this data and to fit a lognormal regression model. Let's review the program that is in the editor.

```
title 'Lognormal model for CARS data set';

ods output ParameterEstimates=params;
proc glimmix data=mydata.cars;
   effect q_hwympg = polynomial(hwympg / degree=2
          standardize(method=moments)=center);
   model price = q_hwympg horsepower / dist=lognormal solution;
   output out=out pred=pred resid=resid;
   id model price;
run;
```

In the PROC GLIMMIX statement, we specify the **cars** data set.

The syntax for the EFFECT statement is identical to the syntax that we used with PROC GLMSELECT. We're creating the same polynomial effect, which has linear and quadratic terms for **Hwympg**. We'll center these variables in order to avoid the multicollinearity that often comes with polynomial terms.

The MODEL statement again specifies **Price**, predicted using the polynomial terms for **Hwympg**, as well as **Horsepower**.

This time, we add the DIST= option to apply the lognormal distribution. We use the SOLUTION option in order to specify the inclusion of a table of parameter estimates in the output. This table is not included in PROC GLIMMIX output by default.

We want to output the predicted values and the residuals so that after the model is fit, we can assess whether we have corrected the heterogeneity of variance. One way to do that is by plotting the predicted values versus the residuals. In the OUTPUT statement, we use the OUT= option to specify that the output data set will be named **out**.

In the ID statement, we specify that SAS should include the values of **Model** and **Price** in the output data set.

Because we've specified the lognormal distribution in our code, the parameter estimates from PROC GLIMMIX will be on the log scale. We want to get predicted values that are on the original scale of the data. In order to do that, we will need to apply an adjustment factor that involves an estimate of sigma squared.

In order to calculate the adjustment factor for backtransforming **Logprice** to **Price**, we need the scale parameter value of 0.05317 from the Parameter Estimates table. Therefore, we need to output the contents of this table to an output data set. To do this, we use the ODS OUTPUT statement with the PARAMETERESTIMATES=option, specifying the name of the data set as **params**.

Let's run the code. There's a lot of output from this procedure, much of which we don't need for our current task. At the top of the output, SAS provides information about the model – the data set name, the response variable, the type of distribution, information about the matrices, and so on. PROC GLIMMIX uses restricted maximum likelihood as the default estimation method for normal or lognormal data.

The Fit Statistics table provides several information criteria statistics, including the AIC, AICC, and BIC. Note that PROC GLIMMIX computes these statistics differently than PROC GLMSELECT does, so you should not make comparisons on the basis of these criteria between models, which were fit using different procedures. These statistics are useful if you run multiple models using PROC GLIMMIX.

The results in the Parameter Estimates table indicate that all three predictors in the model are highly significant. If you compare the significance tests for each of the predictors to the results of the model fitted using the normal distribution, the *p* values are very similar. The conclusions have not changed after applying the lognormal distribution. While correcting for heterogeneity of variance is important to consider and could be significant, in this particular case, it did

not change the results.

Based on the parameter estimates, you can write the following regression equation:
log(**Price**) = 2.1022 – 0.04193*__Hwympg__ + 0.001599*__Hwympg__^{2} + 0.004907*__Horsepower__

The **Scale** parameter shown in the table is the estimate of the residual variance for the lognormal model. We'll use that in our adjustment factor computation.

Let's look at the output data set called **params** that was created by the ODS OUTPUT statement. You can see that it includes the values from the Parameter Estimates table, including **Scale**.

Now we want to determine if this model fit with PROC GLIMMIX is an improvement over the model fit with PROC GLMSELECT. We know that it's a significant model, but can we trust it? To do that, let's check for homogeneity of variances using the Spearman correlation coefficient on the log-scale data. First, we use a DATA step to obtain the absolute value of the residuals. Then we use PROC CORR to look at the Spearman correlation coefficient of the absolute values of the residuals and the predicted values.

```
data check3;
   set out;
   abserror=abs(resid);
run;

proc corr data=check3 spearman nosimple;
   var abserror pred;
run;
```

Let's run the program. The Spearman correlation coefficient between the absolute value of the residuals and the predicted values on the log scale is 0.19492. The *p*-value of 0.0812 indicates that there is not enough evidence to reject the null hypothesis of constant variance. From these values, we can definitely say that this model is a big improvement over the earlier one.

Now we want to create a plot of the residuals versus the predicted values. We want those predicted values to be on their original scale. In order to obtain predicted values on the original data scale, you must back-transform the predicted values produced by PROC GLIMMIX. This is done using a DATA step to compute the estimated means on the scale of the original data.

```
data _null_;
   set params;
   if Effect='Scale' then call symput('var',Estimate);
run;

data back;
   set check3;
   Estimate = exp(pred + &var/2);
   Difference = Price-Estimate;
run;

proc sgplot data=back;
   scatter x=Estimate y=Difference / datalabel=model;
   xaxis min=0 max=60;
   yaxis min=-30 max=30;
   refline 0;
run;
```

The first DATA step reads in the data set **params**, taking the value of the row called scale, which is 0.053 (the estimate of residual variation from the lognormal model), and storing it as a macro variable called VAR (for variance). The _NULL_ option allows you to use the DATA step to process an input data set without creating an output data set. In the second DATA step, we back-transform the predicted values on the log scale to get our predicted values on the original data scale. In the ESTIMATE line we calculate the back transformation by exponentiating the predicted values from log transformed model added to an adjustment factor, the estimate of sigma squared. This gives us low-biased estimates of these means. In the DIFFERENCE line, we calculate the residuals, which is the difference between our observed values minus predicted values. Then we plot the data using PROC SGPLOT, placing predicted values on the x-axis and residuals on the y-axis. The DATALABEL= option specifies that SAS labels data points with model names.

Let's run the program. Here's our new plot, which exhibits much less spread than the plot in the previous

demonstration. This confirms the indications we saw in the correlation analysis. Overall, the lognormal model seems to provide a better fit to the data than the original model. There is still more variability in the more expensive cars. The residuals for cars with predicted values above $30,000 are all negative, and the residual for the Dodge Stealth is still large. Remember, the predictor variables in this model were selected using ordinary least squares regression, assuming that the errors were normally distributed. When switching to a lognormal model, you should return to the modeling cycle and repeat the variable selection process.

Let's look at another way of dealing with non constant variance, which is to use heteroscedasticity-consistent standard errors. We'll use PROC REG to analyze the data, using the same response variable, **Price**, knowing that this data violates the assumption of constant error variance and that this throws off the standard errors. In the MODEL statement, we specify the HCC option, which requests heteroscedasticity-consistent standard errors of the parameter estimates. Then we specify the HCCMETHOD= option to specify the method that is used to compute the heteroscedasticity-consistent covariance matrix. We'll use method 3, which is the method recommended when the sample size is less than 250. For more information about the HCCMETHOD= option, click the Information button.

```
proc reg data=d_carfinal;
   model price = &_GLSMOD / hcc hccmethod=3;
run;
quit;
```

Let's run this code and review the ANOVA and Parameter Estimates tables. The ANOVA table is the same as what we saw earlier. What's new is that the Parameter Estimates table now includes heteroscedasticity-consistent standard errors and the results of the tests based on them. From these results, all three of the independent variables, **Hwympg**, **Hwympg^2**, and **Horsepower**, are significant. In this particular instance, the results of the tests do not differ substantially from what we saw earlier. However, in general, in the presence of heteroscedasticity, you should select candidate models using the consistent variance estimates.

Close