SAS Statistics Research and Applications
Paper #2022-01

Multinomial Regression Diagnostics in Logistic Regression*

Robert Derr, SAS Institute Inc., Cary, NC

Regression diagnostics are an important tool for model development. These diagnostic statistics enable you to find observations that are not explained well by your model, to find observations that have so much impact on the fit that the resulting model does not fit the other observations, and to identify patterns that can indicate a deficient model. SAS has provided diagnostics for binary logistic regression for decades, and now diagnostics are provided for multinomial response models. This paper describes the diagnostics now available for polytomous response models, provides the syntax for producing them, and demonstrates their use in some examples.

Introduction

Diagnostics for linear models are based on residuals and leverage, and they are described in many linear regression texts. Typically, after fitting a model, you perform basic goodness-of-fit testing to determine how well your model fits, but such tests do not provide a full picture. Diagnostics are used to locate observations that are not well fit by the model or that have overly influenced the resulting model, and subsequently they can suggest improvements to that model.

Diagnostics were first developed for binary logistic regression by Pregibon (1981). These diagnostics were made available in the LOGISTIC procedure almost 30 years ago, and most of them were added to the LOGSELECT procedure when it was released in 2016. Lesaffre and Albert (1989) extend these binary logistic diagnostics to a multinomial-response logistic regression setting, but they were never implemented in PROC LOGISTIC. Regression diagnostics for polytomous-response logistic models are now available in PROC LOGSELECT.

The following sections define the polytomous response logistic regression model and the regression diagnostics available in PROC LOGSELECT. The syntax that PROC LOGSELECT requires in order to generate these statistics is provided, and examples display these statistics in action.

*Weibin Mo, a summer intern from the University of North Carolina at Chapel Hill in 2019 and now an assistant professor at Purdue University, performed a literature review and developed SAS® programs to produce these diagnostics. His work is the basis of their current implementation in PROC LOGSELECT.

The Polytomous-Response Logistic Regression Model

PROC LOGSELECT supports two polytomous response logistic regression models: the generalized logit model for nominal responses and the proportional odds model for ordered responses.

Let an observation i consist of a set of $n_i \geq 1$ subjects that have the same p predictors $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})'$, where each subject has one of a possible J responses. Denote the response vector $\mathbf{y}_i = (y_{i1}, \ldots, y_{iJ})$ by letting each y_{ij} be the number of subjects in observation i that have the jth response. So observation i represents $n_i = \sum_{j=1}^{J} y_{ij}$ subjects. Let your data consist of N such observations, for a total of $n = \sum_{i=1}^{N} n_i$ subjects. Let the probability that a subject in observation i has a response j be p_{ij}, and denote $\mathbf{p}_i = (p_{i1}, \ldots, p_{iJ})'$.

Lesaffre and Albert (1989) give the observationwise covariance matrix as

$$\mathbf{V}_i = n_i[\text{diag}(\mathbf{p}_i - \mathbf{p}_i\mathbf{p}_i']$$

with the generalized inverse

$$\mathbf{V}_i^- = \text{diag}\left(\frac{1}{n_i p_{ij}}\right)$$

where $1 \leq j \leq J$. The overall covariance matrix is $\mathbf{V} = \text{diag}\{\mathbf{V}_1, \ldots, \mathbf{V}_N\}$.

Also let w_i be the total weight of the ith observation; that is, it is the product of the weight and the frequency of the observation.

The generalized logit model has a response function for each of the first $J - 1$ response levels that consists of a complete set of slopes and intercepts for each function, and it has a logit link for each function. In particular, if the Jth response level is the reference, the model is written

$$\log\left(\frac{p_{ij}}{p_{iJ}}\right) = \alpha_j + \mathbf{x}_i'\beta_j$$

for $i = 1, \ldots, N, j = 1, \ldots, J-1$, where the intercepts are denoted as $\alpha_1, \ldots, \alpha_{J-1}$ and the slopes are $\beta_j = (\beta_{j1}, \ldots, \beta_{jp})$.

The proportional odds model also has $J - 1$ response functions, with an intercept for each function, but it has a common set of slope parameters across all the functions and a cumulative logit link for each function. The model is written as

$$\log\left(\frac{p_{i1} + \cdots + p_{ij}}{p_{i,j+1} + \cdots + p_{iJ}}\right) = \alpha_j + \mathbf{x}_i'\beta$$

for $i = 1, \ldots, N, j = 1, \ldots, J - 1$, with intercepts $\alpha_1, \ldots, \alpha_{J-1}$ and common slopes $\beta = (\beta_1, \ldots, \beta_p)$.

Finally, for the generalized logit model, add the intercepts and expand the predictors for the ith observation by the number of response functions as $\mathbf{Z}_i = (1\mathbf{x}_i') \otimes I_{J-1}$ so that the first $p + 1$ columns are for the first response function, the second $p + 1$ columns are for the second response function, and so on. Also expand the predictors for the proportional odds model as $\mathbf{Z}_i = I_{J-1}||1_{J-1}\mathbf{x}_i'$, where 1_{J-1} is a vector of $J - 1$ ones. Denote the full design matrix as $\mathbf{X} = (\mathbf{Z}_1', \ldots, \mathbf{Z}_N')'$.

Diagnostics

Pregibon (1981) defines the basic building blocks for the identification of outlying and influential observations as the residuals and the leverage. We attack these building blocks first, then extend some of the other statistics to multinomial-response models. Note that in the binary logistic regression model, we dealt with a single response function for which all the resulting diagnostics are scalar values. However, a polytomous response model has several response functions, and the resulting diagnostics need to be combined across these functions.

Leverage

Leverage tells you how much influence an observation has on the fit of your model—large elements of the hat matrix can indicate extreme points in the design space. The hat matrix is $\mathbf{H} = \mathbf{V}^{1/2}\mathbf{X}(\mathbf{X}'\mathbf{V}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{1/2}$, and let $\mathbf{M} = \mathbf{I}_{(J-1)N} - \mathbf{H}$ (Williams 1987). For a multinomial-response model, each observation corresponds to a $(J-1) \times (J-1)$ block matrix $\mathbf{H}_i = \mathbf{V}_i^{1/2}\mathbf{Z}_i(\mathbf{X}'\mathbf{V}\mathbf{X})^{-1}\mathbf{Z}_i'\mathbf{V}_i^{1/2}$, instead of a single diagonal element that you would use for a single-response model. Let $\mathbf{M}_i = \mathbf{I}_{J-1} - \mathbf{H}_i$, where $\mathbf{I}_{J-1}$ is a $(J-1) \times (J-1)$ identity matrix. There are three definitions of leverage that summarize the hat matrix block matrices (Gupta, Nguyen, and Pardo 2008; Martín 2015), shown in Table 1.

Table 1: Leverage Definitions

Leverage	Equation
Determinant	$h_{di} = \det(\mathbf{M}_i)$
Potential	$h_{pi} = \frac{1}{J-1}\text{trace}(\mathbf{M}_i^{-1}\mathbf{H}_i)$
Trace	$h_{ti} = \frac{1}{J-1}\text{trace}(\mathbf{H}_i)$

Residuals

Residuals describe how far an observation lies from your model's regression surface; observations that are far from the surface are not well approximated by your model. For each observation i and response j, the raw residual is

$$r_{ij} = y_{ij}/n_i - \hat{p}_{ij}$$

where $\hat{p}_{ij}$ is the model-predicted probability for a subject in observation i to have response j. These are accumulated across the response levels in the following fashion:

$$r_i = \sqrt{\sum_{j=1}^{J} y_{ij} r_{ij}^2}$$

Pearson's chi-square statistic for a multinomial response is

$$\chi^2 = \sum_{i=1}^{N}\sum_{j=1}^{J} w_i \frac{n_i^2 r_{ij}^2}{n_i \hat{p}_{ij}}$$

and the Pearson residual for the ith observation is the square root of its contribution to this statistic,

$$r_{Pi} = \sqrt{\sum_{j=1}^{J} r_{Pij}^2}$$

where r_{Pij} is the contribution from each response level j,

$$r_{Pij} = \sqrt{w_i}\frac{n_i r_{ij}}{\sqrt{n_i \hat{p}_{ij}}}$$

Similarly, the deviance residual for the ith observation and jth response level is the square root of its contribution to the deviance

$$r_{Dij} = \text{sign}(r_{ij})\sqrt{2w_i y_{ij}\left|\log\left(\frac{y_{ij}}{n_i \hat{p}_{ij}}\right)\right|}$$

and accumulating this for the total contribution from the ith observation gives

$$r_{Di} = \sqrt{2w_i \sum_{j=1}^{J} y_{ij}\left|\log\left(\frac{y_{ij}}{n_i \hat{p}_{ij}}\right)\right|}$$

Standardized residuals also describe how far an observation lies from your model's regression surface.

Lesaffre and Albert (1989) standardize the residual as

$$\mathbf{r}_{Si} = \mathbf{M}_i^{-1/2}\mathbf{r}_i$$

which corresponds to the single-response standardized residual of the form $\mathbf{r}_i/\sqrt{1-h_i}$. Similarly there is the standardized Pearson residual

$$r_{SPi} = \sqrt{\mathbf{r}'_{Pi-J}\mathbf{M}_i^{-1}\mathbf{r}_{Pi-J}}$$

where $\mathbf{r}_{Pi-J} = (r_{Pi1}, \ldots, r_{pi,J-1})'$, and the standardized deviance residual

$$r_{SDi} = \text{sign}(\mathbf{r}'_{Di-J}\mathbf{M}_i^{-1}\mathbf{r}_{Di-J})\sqrt{\left|\mathbf{r}'_{Di-J}\mathbf{M}_i^{-1}\mathbf{r}_{Di-J}\right|}$$

where $\mathbf{r}_{Di-J} = (r_{Di1}, \ldots, r_{Di,J-1})'$.

The likelihood residuals are a weighted combination of the standardized Pearson and deviance residuals:

$$r_{Li} = \sqrt{\mathbf{r}'_{Pi-J}\mathbf{H}_i\mathbf{M}_i^{-1}\mathbf{r}_{Pi-J} + \mathbf{r}'_{Di-J}\mathbf{r}_{Di-J}}$$

Cook's Distance (CBAR)

The $\overline{C}_i$ statistic, which is based on Cook's distance, measures the influence of an individual observation on the parameter estimates. Lesaffre and Albert (1989) derive this statistic as

$$\overline{C}_i = \mathbf{r}'_{Pi-J}\mathbf{M}_i^{-1}\mathbf{H}_i\mathbf{r}_{Pi-J}$$

DIFCHISQ and DIFDEV

DIFCHISQ and DIFDEV estimate the effect of deleting an observation on the value of the Pearson chi-square statistic and the deviance statistic. Large values indicate observations that make a large contribution to any disagreement between the data and the model-predicted values. (Lesaffre and Albert 1989) define these statistics as

$$\text{DIFDEV}_i = \mathbf{r}'_{Di-J}\mathbf{r}_{Di-J} + \overline{C}_i$$
$$\text{DIFCHISQ}_i = \mathbf{r}'_{Pi-J}\mathbf{r}_{Pi-J} + \overline{C}_i$$

DFBETAS

DFBETA is the difference in the parameter estimates when you leave out an observation, and DFBETAS is its standardized value. These statistics can detect observations that have an excessive effect on a particular parameter. This statistic is not currently implemented in PROC LOGSELECT, because it produces one column for each predictor parameter, and also because you might prefer deleting just one trial from that observation instead of the entire observation. In any case, Lesaffre and Albert (1989) provide the one-step estimate for deleting the entire observation:

$$\text{DFBETA}_{(i)} = (\mathbf{X}'\mathbf{V}\mathbf{X})^{-1}\mathbf{Z}'_i\mathbf{V}_i^{1/2}\mathbf{M}_i^{-1}\mathbf{V}_i^{1/2}r_{Pi}$$
$$\text{DFBETAS}_{(i)k} = \text{DFBETA}_{(i)k}/\sqrt{\mathbf{X}'\mathbf{V}\mathbf{X})^{-1}_{kk}}$$

where (i) indicates that the ith observation is deleted and k indicates the kth parameter. Alternatively, and more precisely, you can write a program to store the model, delete the observation, take one or more steps from the current model, and then compare the new parameter estimates to the full-model estimates, as shown in the examples.

Practical Guidelines (Rules of Thumb)

To use these diagnostic statistics, various papers derive practical guidelines, or rules of thumb, based on large-value approximations to indicate when the diagnostics are too extreme. These guidelines are not implemented in PROC LOGSELECT. Although you can use these guidelines to decide whether to look at the diagnostics, it is also informative to simply note which observations have the most extreme values by listing the observations with the largest values. Creating plots of the statistics against an identification variable enables you to see which observations stand out from the rest. Also note that although derivation of some of these rules of thumb requires more than one subject per observation, you can still use the diagnostics to find outliers even when you have only one subject per observation.

If you have J response levels, N observations, and s parameters in your model, then the practical guidelines for the statistics are provided in Table 2 (Lesaffre and Albert 1989; Gupta, Nguyen, and Pardo 2008; Martín and Pardo 2009; Martín 2015).

Table 2: Practical Guidelines for the Diagnostics

Statistic	Equation
Hat trace	$\geq 2s/((J-1)N)$
Hat potential	$\geq 2\sum(h_{pi})/N$
Hat determinant	$\leq 1-2s/N$
Pearson's chi-square residual	$\geq (1-s/((J-1)N))\chi^2_{J-1}$
Standardized Pearson residual	$\geq \chi^2_{J-1}$
Deviance residual	$\geq (1-s/((J-1)N))\chi^2_{J-1}$
Standardized deviance residual	$\geq \chi^2_{J-1}$
Residual likelihood	$\geq \chi^2_{J-1}$
Pearson difference	$\geq \chi^2_{J-1}$
Deviance difference	$\geq \chi^2_{J-1}$
C bar	$\geq s\chi^2_{J-1}/((J-1)N)$

Syntax

To obtain multivariate regression diagnostics from PROC LOGSELECT, you must use the *multinomial-trial* syntax in your MODEL statement. For this syntax, if you have J response levels, then you must have J variables that contain the number of subjects in that observation with that response value. For example, suppose you perform an experiment with three possible response levels, Y= $\{1, 2, 3\}$. Define three response variables, Y1, Y2, and Y3, to contain the number of Y = 1, 2, and 3 responses for each observation. Suppose that for one multinomial trial, or observation, of 20 subjects you obtain six 1s, nine 2s, and five 3s; for that observation you set the value of Y1 to 6, Y2 to 9, and Y3 to 5. In PROC LOGSELECT you specify the MODEL statement as follows:

```
model Y1 Y2 Y3 = <predictors> </options>;
```

In the `logistic` action you specify the model parameter as follows:

```
model={depVars={'Y1', 'Y2', 'Y3'}, ... }
```

Note that you can have Y1 + Y2 + Y3 = 1 for every observation, especially when your data include continuous-valued covariates.

You request the multinomial-response regression diagnostics in the usual fashion, by specifying an OUTPUT statement in PROC LOGSELECT (or an `output` parameter in the `logistic` action, or a `logisticScore` action) and requesting the CBAR, DIFCHISQ, DIFDEV, H, RESCHI, RESDEV, RESLIK, STDRESCHI, and/or STDRESDEV statistics, or by specifying the ALLSTATS option. For example, the syntax to specify a CBAR statistic for PROC LOGSELECT is

```
OUTPUT OUT=caslib.OUTDS CBAR<=cbarname>;
```

The syntax for the `logistic` action, using the CASL language, is

```
output OUT=caslib.OUTDS CBAR<=cbarname>;
```

The syntax for the `logisticScore` action is

```
logisticScore ... cbar='cbarname';
```

Most of these diagnostics require the PREDPROBS option to be in effect; this is the case by default when you use multinomial-trial syntax. Table 3 matches all available multinomial diagnostic keywords to the preceding equations and also shows what is produced when you specify the NOPREDPROBS option in the OUTPUT statement.

Table 3: Diagnostic Keywords

Option	Statistic	NOPREDPROBS
CBAR	$\overline{C}_i$	.
DIFCHISQ	DIFDEV$_i$	.
DIFDEV	DIFCHI$_i$	.
H	h_{di}, h_{ti}, and h_{pi}	.
RESCHI	r_{Pi}	r_{Pij}
RESDEV	r_{Di}	r_{Dij}
RESLIK	r_{Li}	.
STDRESCHI	r_{SPi}	.
STDRESDEV	r_{SDi}	.

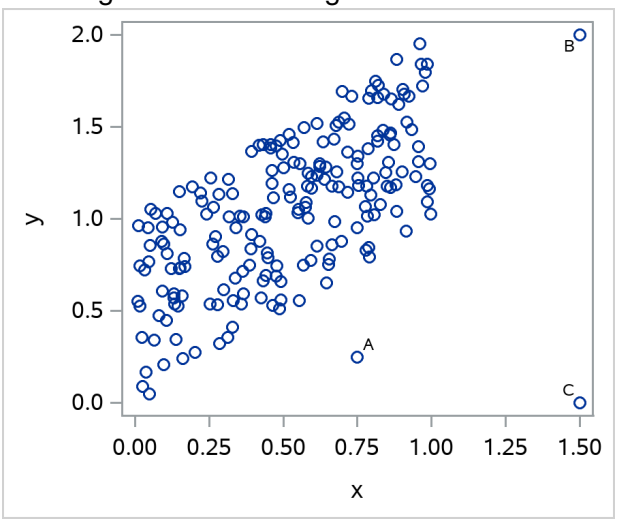
Examples

Figure 1 displays a standard plot of the response Y versus a predictor X that is used to discuss outliers and high-leverage points in linear regression models. In this plot, you can see that point A is an outlier because it does not follow the trend of the data, but because it lies well within the range of the X and Y data, it is not a high-leverage point. Point B is not an outlier because it follows the data trend, but it is a high-leverage point because its position highly influences the angle of the regression line. Point C is both an outlier and a high-leverage point.

```
data one;
   do i=1 to 200;
      x=ranuni(3939);
      y= x+ranuni(3939);
      label=' ';
      output;
   end;
   label='A'; x=0.75; y=0.25; output;
   label='B'; x=1.5;  y=2;    output;
   label='C'; x=1.5;  y=0;    output;
run;

ods graphics on;
proc sgplot data=one;
   scatter x=x y=y / datalabel=label;
run;
```

Figure 1: Linear Regression Outliers



You have more than one response function to consider for polytomous response models, so this simple plot does not describe the situation. In the following examples, the first example

investigates what type of data the regression diagnostics identify by using a simple generalized linear model, while the second example fits a proportional odds model to real data.

Example 1: An Investigation Using the Generalized Logit Model

For a model with a categorical response and more than one response function, you need to look at different plots to get an idea of what these regression diagnostics are identifying. Consider the following simple one variable generalized logit model:

$$\log\left(\frac{\Pr(Y=1)}{\Pr(Y=3)}\right) = 1 - 2x, \quad \log\left(\frac{\Pr(Y=2)}{\Pr(Y=3)}\right) = -4 + 3x$$

or for $\mathbf{x}'\beta_1 = 1 - 2x$ and $\mathbf{x}'\beta_2 = -4 + 3x$,

$$\Pr(Y=1) = \frac{\exp(\mathbf{x}'\beta_1)}{1 + \exp(\mathbf{x}'\beta_1) + \exp(\mathbf{x}'\beta_2)}, \quad \Pr(Y=2) = \frac{\exp(\mathbf{x}'\beta_2)}{1 + \exp(\mathbf{x}'\beta_1) + \exp(\mathbf{x}'\beta_2)}$$

and $\Pr(Y=3) = 1 - \Pr(Y=1) - \Pr(Y=2)$.

The following DATA step samples points from this model and creates the SASCAS1.ONE data table:

```
%macro makeProb;
   xb1= 1-2*x; exb1=exp(xb1);
   xb2= -4+3*x; exb2=exp(xb2);
   p3= 1/(1+exb1+exb2); p1= exb1*p3; p2= exb2*p3;
%mend;
data one;
   call streaminit(54011);
   do obsnum=1 to 100;
      x=2*rand("Uniform");
      %makeProb;
      y1=0;y2=0;y3=0;id='.';
      do j=1 to 6; /*1+5*rand("Uniform"); seed=51749 */
         rand= rand("Uniform");
         if (rand<p1) then y1=y1+1;
         else if (rand<p1+p2) then y2=y2+1;
         else y3=y3+1;
      end;
      output;
   end;
run;
data sascas1.one; set one; run;
```

The following program fits a generalized logit model to these multinomial-response data. The ALLSTATS option outputs all appropriate statistics to the SASCAS1.OUT data table. The COPYVARS= option is specified to additionally output some ID variables, the response counts, and the predictor variable.

```
proc logselect data=sascas1.one;
   model y1 y2 y3 = x / link=glogit;
   output out=sascas1.out allstats ipred=p copyvars=(obsnum id y1 y2 y3 x);
run;
```

You can also use the `logistic` action and the CASL language to fit the same model and output the same table, as in the following program:

```
proc cas;
   action regression.logistic /
      table='two,
      model={depVars={'y1', 'y2', 'y3'},
             effects='x', link='glogit'},
      output={casOut={name='out', replace=true},
         copyVars={'obsnum', 'id', 'y1', 'y2', 'y3', 'x'},
         allstats=true, ipred='p'};
run;
```

Index Plots

You can easily display index plots of the regression diagnostics versus their observation number by using PROC SGPLOT; these plots are shown in Figure 2 and Figure 3. You can use the %INDEXPLOTS macro provided in the appendix to create the individual plots. Although there do not seem to be any extreme values in these plots, you should not be surprised if some observations stand out even though all the data are drawn from the model.

Figure 2: Linear Regression Outliers

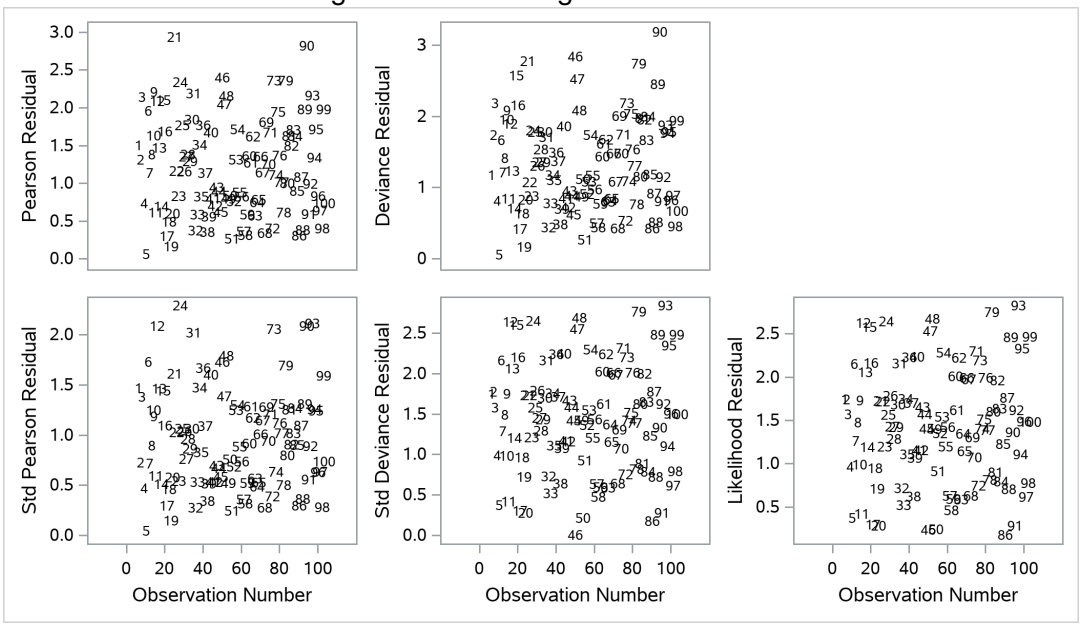
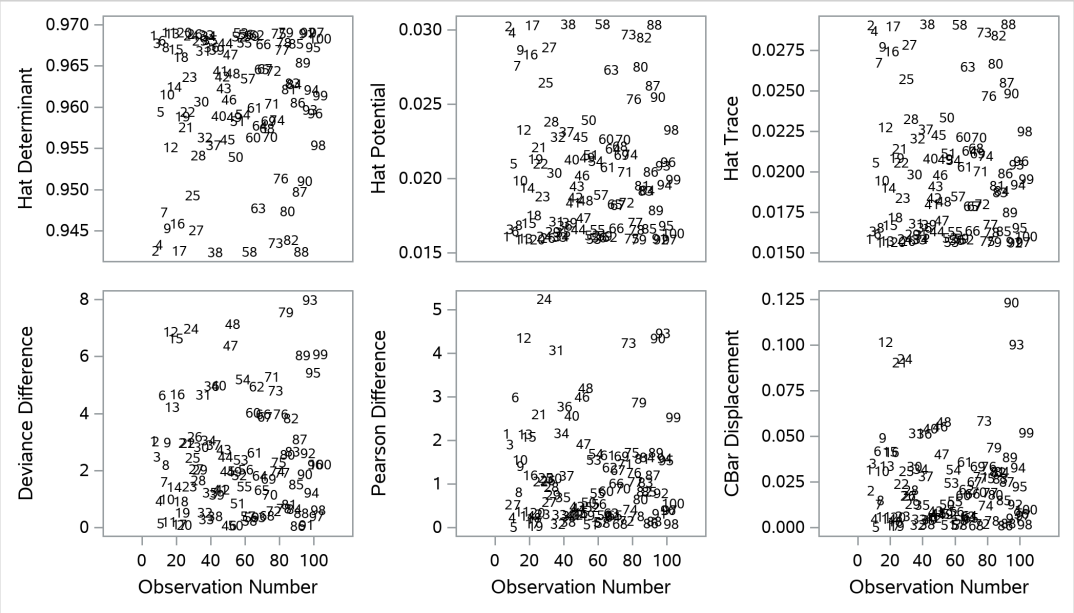


Figure 3: Other Diagnostics



Using the Practical Guidelines

PROC LOGSELECT does not currently implement the practical guidelines when identifying outliers, but you might find them useful especially when you are programmatically deciding whether you have any outlying observations. The %THUMB macro, defined in the appendix and invoked in the following one-line program, lists the top five observations for each statistic that also satisfy the conditions shown in Table 2. Running the full-model SASCAS1.ONE data table through this macro produces the table in Figure 4.

```
%thumb;
```

Figure 4: Top 5 Observations That Satisfy a Practical Guideline

Statistic	rank	obsnum	value	thumb
Cbar	1	90	0.12315	0.11983
deviance difference	1	93	7.96120	5.99146
deviance difference	2	79	7.53480	5.99146
deviance difference	3	48	7.12185	5.99146
deviance difference	4	24	6.96195	5.99146
deviance difference	5	12	6.85375	5.99146
hat potential	1	38	0.03034	0.02000
hat potential	2	58	0.03033	0.02000
hat potential	3	88	0.03031	0.02000
hat potential	4	17	0.03026	0.02000
hat potential	5	2	0.03021	0.02000

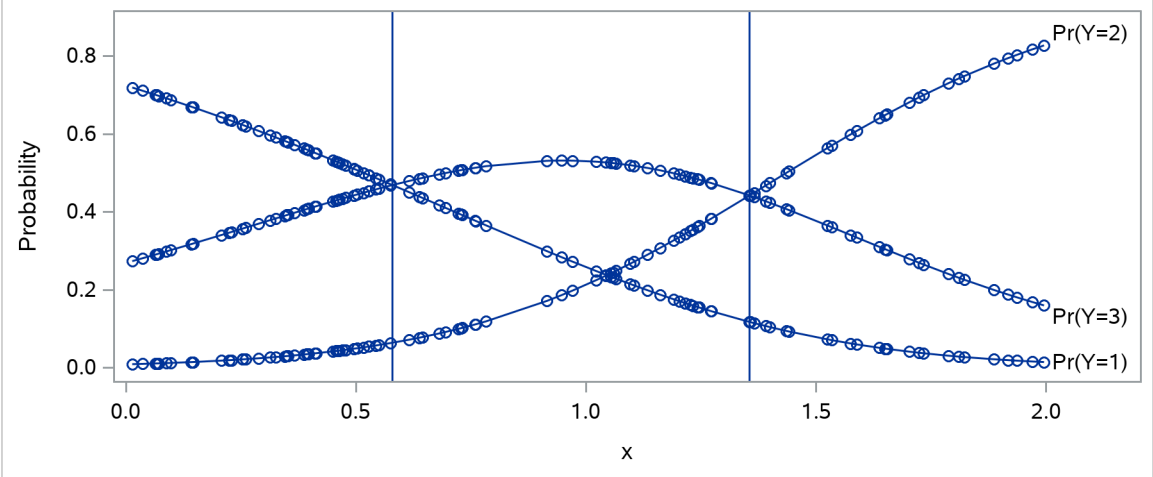
When you compare these observations to the plots in Figure 2 and Figure 3, nothing stands out. Programmatically, using these guidelines can reduce the number of observations that you would investigate. However, sometimes no observations exceed the practical guideline, but when you look at the plots some observations stand apart.

Discovering Outliers and High-Leverage Observations

Consider a plot of the model-predicted probabilities of each response level versus the X variable, shown in Figure 5, which is produced by the following program. This plot includes reference lines at the intersections of these probability curves, which divide the plot into three bins. This plot essentially replaces the standard outlier plot shown in Figure 1.

```
proc sgplot data=sascas1.out noautolegend;
   series  x=x y=py1 / curvelabel='Pr(Y=1)';
   series  x=x y=py2 / curvelabel='Pr(Y=2)';
   series  x=x y=py3 / curvelabel='Pr(Y=3)';
   scatter x=x y=py1;
   scatter x=x y=py2;
   scatter x=x y=py3;
   lineparm slope=. x=0.5796121 y=0;
   lineparm slope=. x=1.3559387 y=0;
   yaxis label='Probability';
run;
```

Figure 5: Simple GLOGIT Model



From the curves in Figure 5, you can see that the Y = 1 response is most likely when $X < 0.58$, the Y = 3 response is most likely when $0.58 < X < 1.36$, and the Y = 2 response is most likely when $X > 1.36$. To investigate the regression diagnostics, you can add points at different values of X with different numbers of trials and different sets of response values. For example, in the first (leftmost) bin, where $X < 0.58$, any observation with a relatively large number of Y = 2 responses should be rare and might be tagged as either an outlier or an influential observation.

The mean values of the response variables and the covariate for each of these bins are shown in Table 4.

Table 4: Bin Means

Bin	X	Y1	Y2	Y3
1	0.33	3.5	0.2	2.4
2	1.00	1.7	1.5	2.8
3	1.66	0.3	3.8	2.0

For this example, every observation has exactly six trials. One obvious way for an observation to be influential is for it to have an extremely large number of trials. To investigate this, the %ADDONE macro, provided in the appendix, adds one observation with the provided response counts for each response level to the SASCAS1.ONE data table and refits the model. It then determines whether this observation has a regression diagnostic that is the most extreme, indicating that it is either very influential or very poorly fit.

Now see what happens when you add an observation with a large number of trials to each bin by submitting the following program. In this case, each X value is set to its mean in the bin, and each response value is approximately double its mean.

```
%addone(x=0.33, y1=7, y2=0, y3=5);
%addone(x=1.00, y1=3, y2=3, y3=6);
%addone(x=1.66, y1=1, y2=8, y3=4);
```

The results from the first macro call are displayed in Figure 6. RANK=1 means that it has the largest value of that regression diagnostic.

Figure 6: Results from the %ADDONE Macro

Statistic	rank	value
hat determinant	1	0.93583
hat potential	1	0.03376
hat trace	1	0.03258

In all three bins, the three leverage values (hat determinant, hat potential, and hat trace) are identified as the most extreme among all these observations. These three diagnostics capture observations that have an abnormally large number of trials, and this larger frequency gives them a larger influence on the model fit.

For the other diagnostics, location is more important. The %ADDONETOALL macro invokes the %ADDONE macro for every response distribution of six trials for the given X value, and it refits the model for each distribution. This time the observation is flagged as an outlier only if it is the most extreme diagnostic and it is more than 1.5 times the second-most extreme value. The following program calls this macro for five different X values. Table 5 shows which combinations of response levels have at least one extreme diagnostic and then summarizes the results.

```
%addonetoall(x=-1);
%addonetoall(x=0.33);
%addonetoall(x=1);
%addonetoall(x=1.66);
%addonetoall(x=3);
```

Table 5: Results of the %ADDONETOALL Macro Calls

Y1	0	0	0	0	0	0	0	1	1	1	1	1	1	2	2	2	2	2	3	3	3	3	4	4	4	5	5	6
Y2	6	5	4	3	2	1	0	5	4	3	2	1	0	4	3	2	1	0	3	2	1	0	2	1	0	1	0	0
Y3	0	1	2	3	4	5	6	0	1	2	3	4	5	0	1	2	3	4	0	1	2	3	0	1	2	0	1	0
X = -1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x		x		
X = 0.33	x	x	x	x	x	x		x	x	x	x			x	x	x			x	x			x					
X = 1	x	x						x																		x	x	x
X = 1.66													x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
X = 3			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

X	Summary
-1	Outliers indicated for all responses except when Y2 = 0 and Y1 is large
0.33	No outliers indicated when Y2 is small
1.0	Outliers indicated when Y1 is much different from Y2
1.66	No outliers indicated when Y1 is small
3	Almost all are indicated as outliers

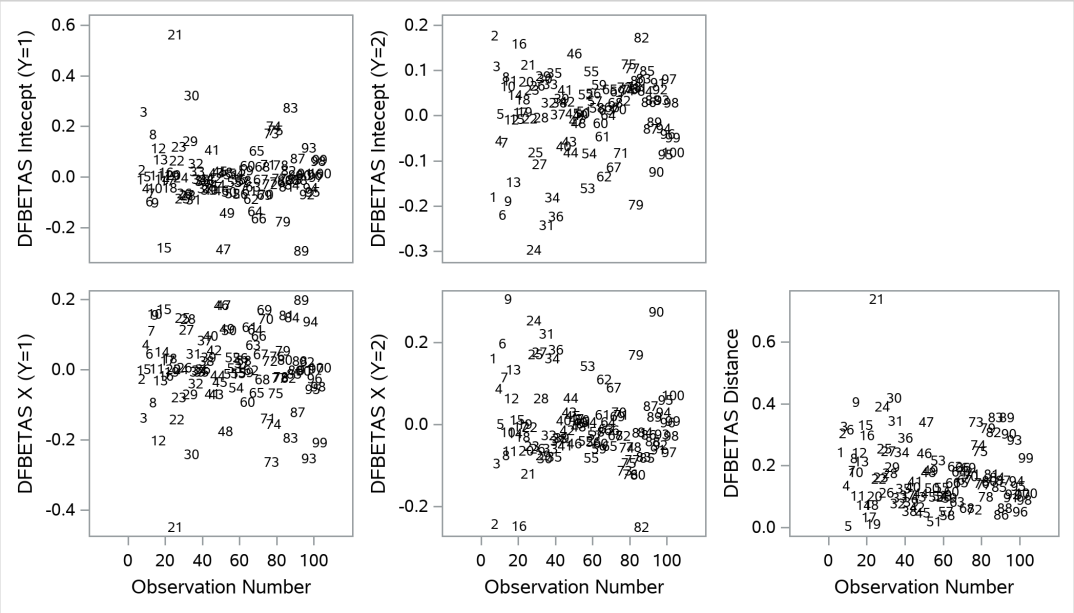
In summary, the regression diagnostics identify observations that have a larger number of trials and/or a different distribution of response values than their neighboring observations. These observations might be overly influencing fit and predictions, and they might not be well described by your model.

Programming Your Own DFBETAS Statistics

The generalized logit regression example has four parameters, so there are four DFBETAS statistics to compute. The square root of the sums of squares of these DFBETAS statistics is also computed and labeled as “DFBETAS Distance”. The %DFBETAS macro, provided in the appendix and invoked in the following one-line program, creates the RESULTS data set, which you can then use with PROC SGPLOT to create the plots displayed in Figure 7.

```
%dfbetas;
```

Figure 7: DFBETAS



The 21st observation stands out as excessively influencing the parameter estimates.

Example 2: The Proportional Odds Model

The following data are from Example 11 of the GLIMMIX procedure documentation. The data are taken from Gilmour, Anderson, and Rae (1987) and concern the foot shape of 2,513 lambs from 34 sires. The foot shape of the animals was scored in three ordered categories. Each observation corresponds to a sire and contains the outcomes for the three response categories in the variables k1, k2, and k3. For example, for the first sire, foot shape category 1 was observed for 52 of its offspring, foot shape category 2 was observed for 25 lambs, and none of its offspring were rated in foot shape category 3. The variables yr, b1, b2, and b3 represent contrasts of fixed effects.

```
data sascas1.foot;
   input yr b1 b2 b3 k1 k2 k3 @@;
   sire = _n_; id = sire; obsnum = sire;
   datalines;
 1  1  0  0  52 25  0     1  1  0  0  49 17  1     1  1  0  0  50 13  1
 1  1  0  0  42  9  0     1  1  0  0  74 15  0     1  1  0  0  54  8  0
 1  1  0  0  96 12  0     1 -1  1  0  57 52  9     1 -1  1  0  55 27  5
 1 -1  1  0  70 36  4     1 -1  1  0  70 37  3     1 -1  1  0  82 21  1
 1 -1  1  0  75 19  0     1 -1 -1  0  17 12 10     1 -1 -1  0  13 23  3
 1 -1 -1  0  21 17  3    -1  0  0  1  37 41 23    -1  0  0  1  47 24 12
-1  0  0  1  46 25  9    -1  0  0  1  79 32 11    -1  0  0  1  50 23  5
-1  0  0  1  63 18  8    -1  0  0 -1  30 20  9    -1  0  0 -1  31 33  3
-1  0  0 -1  28 18  4    -1  0  0 -1  42 27  4    -1  0  0 -1  35 22  2
-1  0  0 -1  33 18  3    -1  0  0 -1  35 17  4    -1  0  0 -1  26 13  2
-1  0  0 -1  37 15  2    -1  0  0 -1  36 14  1    -1  0  0 -1  63 20  3
-1  0  0 -1  41  8  1
;
```

All the available information is fit to a proportional odds model in the following program. The results are not displayed here, but the index plots and DFBETAS distance are displayed in Figure 8 and Figure 9.

```
proc logselect data=sascas1.foot;
   model k1 k2 k3 = b1 b2 b3 yr;
   output out=sascas1.out allstats predprobs copyvars=(obsnum id);
run;
```

Figure 8: Residuals

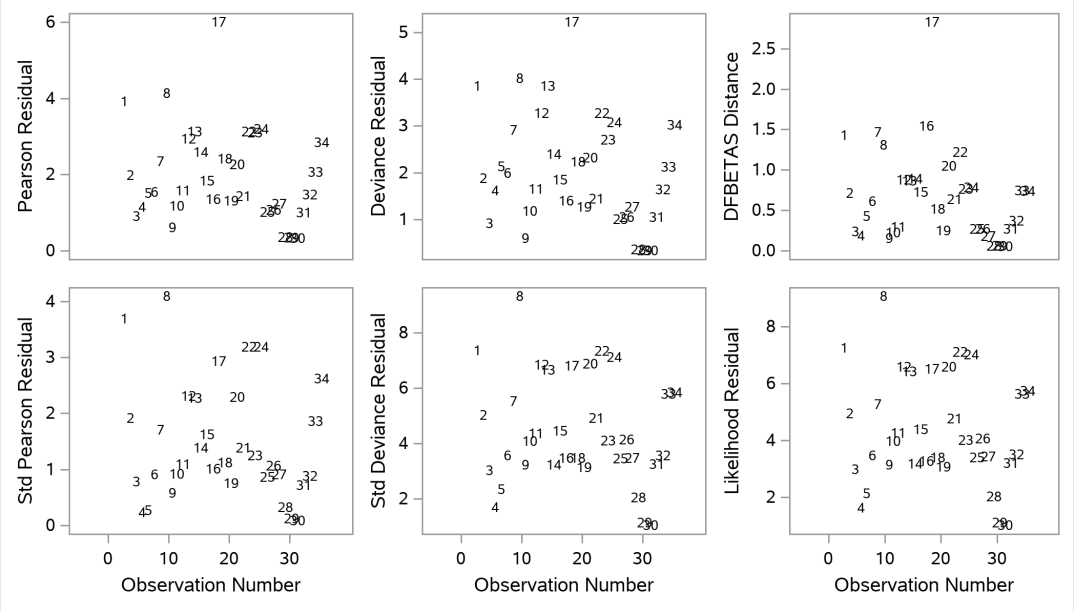
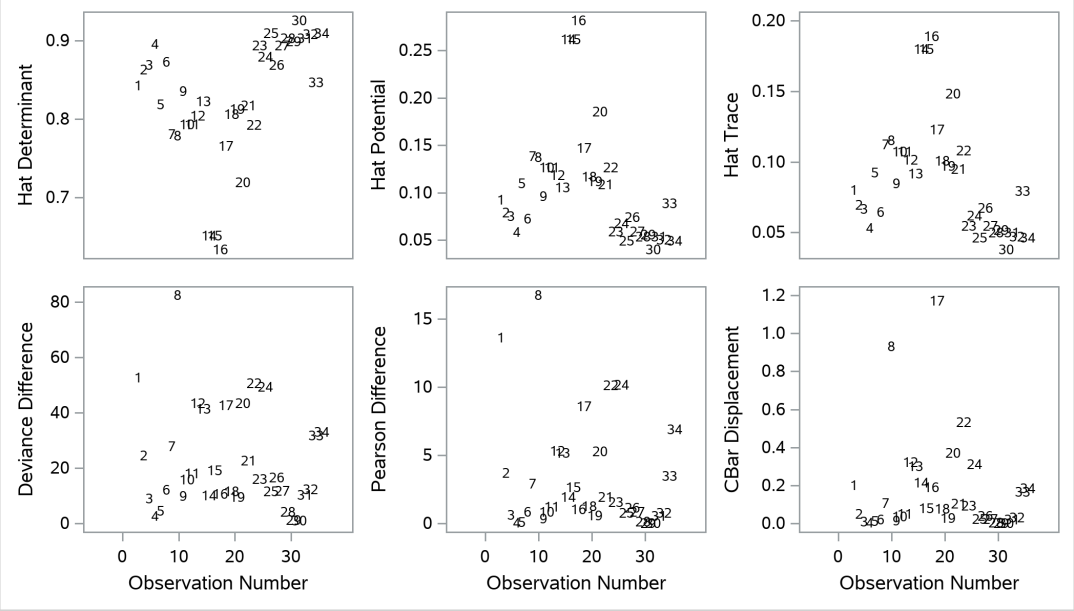


Figure 9: Other Diagnostics



The residual plots in Figure 8 indicate that sire 8 stands out from the herd; the DFBETAS distance plot identifies sire 17 as having a large influence on the predictors. The plots in Figure 9 find sires 1 and 14–17 to be influential. Sires 14–16 are the only sires with b2 = –1 and actually have the fewest lambs in the data set. Sires 8 and 17 have the largest values of k2 and k3 among sires with the same covariate values, and sire 1 has the largest k2 value compared to the first seven observations. In all cases, these sires have the largest proportion of k2 and k3 responses among sires with the same covariates.

Conclusion

The new multinomial-response syntax together with the new multinomial-response regression diagnostics provides you with more tools for evaluating the "badness" of the fit of your model to your data. In particular, the regression diagnostics identify observations that have a relatively large number of trials and/or an atypical response distribution that either drive the fit of your model or are not well fit by your model. As always, when you use diagnostics to identify outlying and influential observations, you should not simply remove those observations from your data; instead, you should make sure that your data are correct, and you should consider modifications to improve your model.

Appendix

%ADDONE and %ADDONETOALL Macros

The %ADDONE macro adds one observation to the ONE data set, creates the SASCAS1.TWO data table, and refits the model to this new CAS data table. You input the value of the X variable and the values of the Y1, Y2, and Y3 response variables. Specifying the option DISPLAY = 1 prints the values of the response variables. Specifying a CUT= option greater than 0 passes a cutpoint factor to the %TOP macro so that the most extreme statistic is not flagged unless it is factor-times the value of the second-most extreme statistic. The output displays the statistic, its rank, and its value.

```
%macro addone(x=,y1=,y2=,y3=,display=0,cut=0);
options nonotes nodate;
ods listing close;
data two;
    x=&x; y1=&y1; y2=&y2; y3=&y3; id='A';
run;
data two; set two one; i=_n_;
data sascas1.two; set two; run;
proc logselect data=sascas1.two;
   model y1 y2 y3 = x / link=glogit;
   display / excludeall;
   output out=sascas1.out2 allstats copyvars=(obsnum id y1 y2 y3 x);
run;
```

```
%top(num=1,dsname=out2,cutfactor=&cut);
data _null_;
%let allid=%sysfunc(OPEN(all,IN));
%let NOBS=%sysfunc(ATTRN(&allid,NOBS));
%let rc=%sysfunc(CLOSE(&allid));
run;
%if %eval(&NOBS > 0) %then %do;
   ods listing;
   %if %eval(&display=1) %then %put y1=&y1 y2=&y2 y3=&y3;
   %relabel;
   proc print data=all noobs label; var label rank value; run;
%end;
%mend;
```

The %ADDONETOALL macro scans through all combinations of Y1, Y2, and Y3 that have six trials, and it calls the %ADDONE macro to see which combinations are identified as outliers.

```
%macro addonetoall(x=);
%do y1=0 %to 6;
  %do y2=0 %to %eval(6-&y1);
     %let y3=%eval(6-&y2-&y1);
     %addone(x=&x,y1=&y1,y2=&y2,y3=&y3,display=1,cut=1.5);
  %end;
%end;
%mend;
```

%DFBETAS Macro

The %DFBETAS macro computes the full-model parameter estimates and stores them in the SASCAS1.PE data table. Then it sequentially drops each observation and refits the model but with only one iteration step, and then it stores the reduced-data parameter estimates in the SASCAS1.PE2 data table. Note that PROC LOGSELECT treats nonconvergence as an error unless you also specify the USELASTITER option. The DFBETAS for each parameter is the difference between these parameter estimates divided by the standard error of the full-model parameter estimate. Finally, the square root of the sums of squares of these DFBETAS is also computed and named dfbetas_dist. Results are stored in the RESULTS data set.

```
%macro dfbetas;
options nonotes nodate; ods listing close;
data one; set one; call symput('n',_n_); run;
proc logselect data=sascas1.one;
   model y1 y2 y3 = x / link=glogit;
   displayout parameterestimates=pe;
   display / excludeall;
run;
data pe; set sascas1.pe; Estimate1= Estimate;
   keep ParmName Estimate1 StdErr; proc sort; by ParmName; run;
%do i=1 %to &n;
```

```
data two; set one; if (_n_ ^= &i); run;
data sascas1.two; set two;
proc logselect data=sascas1.two inparmest=sascas1.pe maxiter=1 uselastiter;
   model y1 y2 y3 = x / link=glogit;
   displayout parameterestimates=pe2;
   display / excludeall;
run;
data pe2; set sascas1.pe2; keep ParmName Estimate; proc sort; by ParmName; run;
data tmp; merge pe pe2; by ParmName;
dfbetas=(Estimate-Estimate1)/StdErr; keep ParmName dfbetas; run;

proc transpose data=tmp out=tmp prefix=dfbeta_; id ParmName; run;
data tmp; set tmp; obsnum=&i;
   dfbeta_dist=sqrt(dfbeta_Intercept_y1**2+dfbeta_Intercept_y2**2
                  +dfbeta_x_y1**2+dfbeta_x_y2**2);
data results;
   %if %eval(&i=1) %then set tmp; %else set results tmp;;
run;
%end;
ods listing;
%mend;
```

%PLOTONE and %INDEXPLOTS Macros

The %PLOTONE macro displays an index plots of a diagnostic statistic, specified in the Y= option, by using PROC SGPLOT. The %INDEXPLOTS macro calls the %PLOTONE macro for all of the diagnostic statistics.

```
%macro plotone(y=);
   proc sgplot data=sascas1.out;
      scatter x=obsnum y=&y;
   run;
%mend;
%macro indexplots(y=);
   %plotone(y=_h_t);
   %plotone(y=_h_p);
   %plotone(y=_h_d);
   %plotone(y=_reschi_);
   %plotone(y=_stdreschi_);
   %plotone(y=_resdev_);
   %plotone(y=_stdresdev_);
   %plotone(y=_reslik_);
   %plotone(y=_difdeviance_);
   %plotone(y=_difchisquare_);
   %plotone(y=_cbar_);
%mend;
```

%RELABEL Macro

The %RELABEL macro simply changes the labels of diagnostics in the ALL data set from the output data table names to something more readable.

```
%macro relabel;
data all; length label $30; set all;
   if (label="_h_d")           then label="hat determinant";
   if (label="_h_p")           then label="hat potential";
   if (label="_h_t")           then label="hat trace";
   if (label="_reschi_")       then label="Pearson residual";
   if (label="_resdev_")       then label="deviance residual";
   if (label="_stdreschi_")    then label="standardized pearson residual";
   if (label="_stdresdev_")    then label="standardized deviance residual";
   if (label="_reslik_")       then label="likelihood residual";
   if (label="_difdeviance_")  then label="deviance difference";
   if (label="_difchisquare_") then label="Pearson difference";
   if (label="_cbar_")         then label="Cbar";
run;
%mend;
```

%THERULES and %THUMB Macros

The %THERULES and %THUMB macros are similar to the %TOP and %TOPONE macros, except that they also filter by the practical guidelines for each statistic (Table 2). For the %THERULES macro, the DSNAME= option specifies the output data set; the STAT= option specifies the diagnostic that you want to check; the NUM= option specifies the number of largest values to display, if any; the S= option specifies the number of parameters in the model; the J= option specifies the number of response levels; and the N= option specifies the number of observations in the data set. The %THUMB macro calls the %THERULES macro for all of the diagnostic statistics.

```
%macro therules(num=,dsname=out,s=,J=,N=,stat=);
proc sort data=&dsname; by descending &stat;
%if (&stat="_H_P") %then %do;
data _null_; retain sum 0; set &dsname; sum=sum+_H_P;
   call symput('sumtmp',sum); run;
%end; %else %let sumtmp=1;
data tmp; length label $ 14/*$*/; set &dsname; label="&stat"; value= &stat;
      keep obsnum id label value thumb;
   if ("&stat"="_h_t") then do; thumb=2*&s/((&J-1)*&N);
      if (_h_t > thumb) then output; end;
   else if ("&stat"="_h_p") then do; thumb=2*&sumtmp/&n;
      if (_h_p > thumb) then output;end;
   else if ("&stat"="_h_d") then do; thumb=1-2*&s/&N;
      if (_h_d < thumb) then output; end;
   else if ("&stat"="_reschi_") then do;
```

```
      thumb=(1-&s/((&J-1)*&N))*QUANTILE('CHISQUARE',.95,&J-1);
      if (_reschi_ > thumb) then output; end;
   else if ("&stat"="_stdreschi_") then do;
      thumb=QUANTILE('CHISQUARE',.95,&J-1);
      if (_stdreschi_ > thumb) then output; end;
   else if ("&stat"="_resdev_") then do;
      thumb=(1-&s/((&J-1)*&N))*QUANTILE('CHISQUARE',.95,&J-1);
      if (_resdev_ > thumb) then output; end;
   else if ("&stat"="_stdresdev_") then do;
      thumb=QUANTILE('CHISQUARE',.95,&J-1);
      if (_stdresdev_ > thumb) then output; end;
   else if ("&stat"="_reslik_") then do;
      thumb=QUANTILE('CHISQUARE',.95,&J-1);
      if (_reslik_ > thumb) then output; end;
   else if ("&stat"="_difdeviance_") then do;
      thumb=QUANTILE('CHISQUARE',.95,&J-1);
      if (_difdeviance_ > thumb) then output; end;
   else if ("&stat"="_difchisquare_") then do;
      thumb=QUANTILE('CHISQUARE',.95,&J-1);
      if (_difchisquare_ > thumb) then output; end;
   else if ("&stat"="_cbar_") then do;
      thumb=&s*QUANTILE('CHISQUARE',.95,&J-1)/((&J-1)*&N);
      if (_cbar_ > thumb) then output;
      end;
data tmp; set tmp; if _n_<=&num; rank=_n_; run; data all; set all tmp; run;
%mend;
%macro thumb(dsname=out,J=3,N=100,s=4,num=5,filter=0);
   data all; run;
   data out; set sascas1.&dsname; run;
   %therules(num=&num,dsname=&dsname,s=&s,J=&J,N=&N,stat=_h_t);
   %therules(num=&num,dsname=&dsname,s=&s,J=&J,N=&N,stat=_h_p);
   %therules(num=&num,dsname=&dsname,s=&s,J=&J,N=&N,stat=_h_d);
   %therules(num=&num,dsname=&dsname,s=&s,J=&J,N=&N,stat=_reschi_);
   %therules(num=&num,dsname=&dsname,s=&s,J=&J,N=&N,stat=_stdreschi_);
   %therules(num=&num,dsname=&dsname,s=&s,J=&J,N=&N,stat=_resdev_);
   %therules(num=&num,dsname=&dsname,s=&s,J=&J,N=&N,stat=_stdresdev_);
   %therules(num=&num,dsname=&dsname,s=&s,J=&J,N=&N,stat=_reslik_);
   %therules(num=&num,dsname=&dsname,s=&s,J=&J,N=&N,stat=_difdeviance_);
   %therules(num=&num,dsname=&dsname,s=&s,J=&J,N=&N,stat=_difchisquare_);
   %therules(num=&num,dsname=&dsname,s=&s,J=&J,N=&N,stat=_cbar_);
   data all; set all; if _n_>1; label label='Statistic';
      %if %eval(&filter=1) %then %do; if id ^= '.'; %end; run;
   proc sort data=all; by label rank obsnum; run;
   %relabel;
   proc print data=all noobs label; var label rank obsnum value thumb; run;
%mend;
```

%TOPONE and %TOP Macros

The %TOPONE and %TOP macros display the observations that have the largest diagnostic statistic values, as long as they have a nonmissing ID variable. In the %TOPONE macro, the STAT= option specifies the output statistic name, the NUM= option specifies the number of the largest values to display, the DSNAME= option specifies the input data set name, and the CUTFACTOR= option specifies that the identified extreme values must be larger than CUTFACTOR times the NUM+1th observation's value. In the %TOP macro, you specify the NUM= and DSNAME= options, and the FILTER=1 option suppresses results from the original data set. The %TOP macro calls the %TOPONE macro for all of the diagnostic statistics.

```
%macro topone(num=,dsname=,stat=,cutfactor=0);
   %if %eval(&stat=_h_d) %then %do;
      proc sort data=out; by &stat;
   %end;
   %else %do;
      proc sort data=out; by descending &stat;
   %end;
   data _null_; set out; if _n_=%eval(&num+1);
      %if %eval(&stat=_h_d) %then %do;
         cut=1-(1-&stat)*&cutfactor;
      %end;
      %else %do;
         cut=&cutfactor*&stat;
      %end;
      call symput('cutvalue',cut); run;
   data tmp; length label $ 14;/*$*/ set out; label="&stat";
   cut= &cutvalue;
   if _n_<=&num; rank=_n_; value=&stat; keep obsnum id label rank value cut;
   data all; set all tmp; run;
%mend;
%macro top(num=5,dsname=out,filter=1,cutfactor=);
   data all; run;
   data out; set sascas1.&dsname; run;
   %topone(num=&num,dsname=&dsname,cutfactor=&cutfactor,stat=_h_t);
   %topone(num=&num,dsname=&dsname,cutfactor=&cutfactor,stat=_h_p);
   %topone(num=&num,dsname=&dsname,cutfactor=&cutfactor,stat=_h_d);
   %topone(num=&num,dsname=&dsname,cutfactor=&cutfactor,stat=_reschi_);
   %topone(num=&num,dsname=&dsname,cutfactor=&cutfactor,stat=_stdreschi_);
   %topone(num=&num,dsname=&dsname,cutfactor=&cutfactor,stat=_resdev_);
   %topone(num=&num,dsname=&dsname,cutfactor=&cutfactor,stat=_stdresdev_);
   %topone(num=&num,dsname=&dsname,cutfactor=&cutfactor,stat=_reslik_);
   %topone(num=&num,dsname=&dsname,cutfactor=&cutfactor,stat=_difdeviance_);
   %topone(num=&num,dsname=&dsname,cutfactor=&cutfactor,stat=_difchisquare_);
   %topone(num=&num,dsname=&dsname,cutfactor=&cutfactor,stat=_cbar_);
   data all; set all; label label='Statistic';
      keep label rank value id cut; if _n_>1;
      %if %eval(&filter=1) %then %do; if id ^= '.'; %end;
```

```
      %if %eval(&cutfactor>0) %then %do;
         if (label='_h_d') then do; if value < cut; end;
         else do; if value > cut; end;
      %end; run;
   proc sort data=all; by label; run;
%mend;
```

References

Gilmour, A. R., Anderson, R. D., and Rae, A. L. (1987). "Variance Components on an Underlying Scale for Ordered Multiple Threshold Categorical Data Using a Generalized Linear Mixed Model." *Journal of Animal Breeding and Genetics* 104:149–155.

Gupta, A. K., Nguyen, T., and Pardo, L. (2008). "Residuals for Polytomous Logistic Regression Models Based on φ-Divergences Test Statistics." *Statistics* 42:495–514.

Lesaffre, E., and Albert, A. (1989). "Multiple-Group Logistic Regression Diagnostics." *Journal of the Royal Statistical Society, Series C* 38:425–440.

Martín, N. (2015). "Using Cook's Distance in Polytomous Logistic Regression." *British Journal of Mathematical and Statistical Psychology* 68:84–115.

Martín, N., and Pardo, L. (2009). "On the Asymptotic Distribution of Cook's Distance in Logistic Regression Models." *Journal of Applied Statistics* 36:1119–1146.

Pregibon, D. (1981). "Logistic Regression Diagnostics." *Annals of Statistics* 9:705–724.

Williams, D. A. (1987). "Generalized Linear Model Diagnostics Using the Deviance and Single Case Deletions." *Journal of the Royal Statistical Society, Series C* 36:181–191.

Acknowledgments

Thanks to Dr. Fang Chen for asking if I had a project for a summer intern to work on, Dr. Weibin Mo for the literature search, Dr. Weijie Cai for suggesting I write this paper, and Dr. Yiu-Fai Yung for a careful review and for ushering the paper through the publication process.