

DISCRETE DATA CODING USING PROC TRANSREG

Michael Lerner

Ontario Ministry of Education

June 3, 2011

Purpose

- To discuss how to code discrete data in different ways in PROC TRANSREG for analysis in other procedures.

What Are Discrete Data?

- Discrete data are indicators that use categories or levels such as gender or different types of post-secondary degrees.
- Typically appear in class statements in many procedures such as GLM, LOGISTIC or GENMOD.

Why Do This?

- Sometimes it makes sense to analyze discrete data using deviations from the average effect.
 - Makes results more immediately interpretable.
 - Eliminates time and possible error in constructing contrast or estimate statements as a work around or creating dummy variables in a data step.
 - Effect of omitted level = -1 (sum of levels with estimates)
- For instance, GLM, MIXED, PROBIT estimate the difference in the effects of each level compared to the last level.
 - Uses over-parameterized (not full rank) method.
 - Effect of omitted level = Intercept.

What Does TRANSREG Do?

- Fits linear models, optionally with smooth, spline, Box-Cox, and other nonlinear transformations of the variables.
- Codes experimental designs and classification variables prior to their use in other analyses.

Syntax

- Syntax is somewhat different from other procedures.
 - Class variables are invoked in the MODEL statement
 - Can output design matrix without performing analysis.

Sample Syntax from SAS 9.22 Manual

(Example 91.76)

```
proc transreg data=a design;  
model class(a b / deviations);  
id y w;  
output out=coded;  
run;
```

- 'Design' in PROC statement specifies design matrix to be output with no analysis
- Model statement
 - No dependent variable in model statement
 - Class(a b/deviations) specifies the classification variables and that deviations from the average effect is requested.
 - Last level is omitted in results by default
 - Could include interaction if required
 - Id statement includes the dependent variable, 'y' and another variable 'w' used to create the simulate data.
 - Output creates macro variable, &_TrgInd, has created independent variables to be used in subsequent analysis.

Example

- Uses fake data with interval outcome (Y), two categorical predictors (C1 and C2), and freq as the number of cases.
- Shows the data, deviation from mean effect results in GENMOD, what the default result is in GLM and that the 2nd GLM output looks like the GENMOD output because the discrete data have been re-parameterized appropriately.

The Data (Completely Fake):

Obs	Y	C1	C2	freq
1	12	a	w	20
2	9	b	w	25
3	14	c	w	17
4	8	d	w	11
5	6	a	x	33
6	-1	b	x	12
7	7	c	x	22
8	15	d	x	17
9	14	a	y	19
10	8	b	y	18
11	11	c	y	16
12	9	d	y	22

CODE GENMOD DEVIATION FROM MEAN EFFECT:

```
options pageno=1;  
title 'PROC GENMOD EFFECT  
PARAMETERIZATION';  
proc genmod data=x;  
class c1 c2/ param=effect;  
model y=c1 c2;  
freq freq;  
run;
```

GENMOD DEVIATION FROM MEAN RESULTS

Parameter		DF	Estimate	Standard Error
Intercept		1	9.4484	0.1837
C1	a	1	0.8487	0.2976
C1	b	1	-3.4999	0.3275
C1	c	1	1.1509	0.3221
C2	w	1	1.8592	0.2654
C2	x	1	-2.8035	0.2568

DEFAULT GLM ANALYSIS

```
options pageno=1;  
title 'PROC GLM DEFAULT  
      OVERPARAMETERIZED REFERENCE  
      PARAMETERIZATION';  
proc GLM data=x;  
class c1 c2;  
model y=c1 c2/SOLUTION;  
freq freq;  
run;  
QUIT;
```

DEFAULT GLM RESULTS

Intercept	11.89299	B	0.4553831
C1 a	-0.6515875	B	0.5212999
C1 b	-5.0002364	B	0.5562861
C1 c	-0.3493663	B	0.551457
C1 d	0	B	.
C2 w	0.9149122	B	0.4665676
C2 x	-3.7478561	B	0.4515621
C2 y	0	B	.

TRANSREG WRITES THE DESIGN MATRIX

```
options pageno=1;  
title 'SIMPLE TRANSREG OUTPUT';  
PROC TRANSREG DATA=X DESIGN;  
MODEL class(C1 C2/ deviations);  
id y freq;  
output out=coded;  
run;
```

GLM PRODUCES DEVIATION FROM AVERAGE EFFECT

```
OPTIONS PAGENO=1;
```

```
TITLE 'PROC GLM NOW DOES  
DEVIATION FROM AVERAGE EFFECT!!';
```

```
PROC GLM;
```

```
MODEL Y=&_trgind;
```

```
freq freq;
```

```
RUN;QUIT;
```

GLM DEVIATION FROM MEAN RESULTS AFTER TRANSREG

Parameter	Estimate	Standard Error
Intercept	9.4483782	0.18610998
C1a	0.84871	0.30150512
C1b	-3.4999389	0.33178669
C1c	1.1509313	0.32631929
C2w	1.8592268	0.26888441
C2x	-2.8035414	0.26020471

FINAL COMMENTS

- Have just scratched the surface of TRANSREG; it can also centre continuous variables as they are taken to higher powers.
- There is a general need in SAS to extend the param portion of the class statement to other procedures, especially GLMMOD, which is explicitly meant to write out design matrices.
 - Possible suggestion for SASware Ballot.