Paper 089-2021

Screening, Binning, Transforming Predictors for a Generalized Logit Model

Bruce Lund, Statistical Consultant and Trainer, Novi MI

ABSTRACT

The generalized logit model is a logistic regression model where the target (or dependent variable) has 3 or more levels, and the levels are unordered. Predictors for the generalized logit model may be NOD (nominal, ordinal, discrete-numeric) where, generally, the number of levels is under 16. Alternatively, predictors may be continuous where the predictor is numeric and has many levels. This paper discusses methods that screen, bin, or transform the NOD and continuous predictors, as preparation for model fitting. These same methods also apply to the cumulative logit model (where the target is ordered). The binning methodology is applied to NOD predictors and generalizes the concept of information value. The method of transforming a continuous predictor is an extension of the function selection procedure (FSP) to the multinomial target. SAS® macros are presented which implement the methods for screening, binning, and transforming. Familiarity with PROC LOGISTIC is assumed.

INTRODUCTION

The generalized logistic model extends the binary logistic model to the case where the target has J > 2 levels, and the levels are unordered. Some papers and books use the term multinomial logit model. But in conformance with SAS terminology, the term, generalized logistic model, or generalized logit, is used. The conditional logit, which is a generalization of the generalized logit, is not discussed in this paper.

The cumulative logit (cum logit) extends the binary logistic model to the case where the target has J > 2 levels, and where the levels are ordered. The number of levels J should be relatively small with J=10 being large. If a target has J > 10, then an alternative modeling approach is probably better. The common (but restrictive) form of the cum logit is the proportional odds (PO) model. The cum logit partial proportional odds (PPO) model provides a generalization. If the target is count-data, then the Poisson, Negative Binomial, or ZIP regression model may be preferred. These are performed by PROC GENMOD or PROC COUNTREG (in SAS/ETS).

The purpose of this paper is to discuss methods to screen, bin, or transform predictors prior to the model fitting stage for the generalized logit but with reference to the cumulative logit. At each stage (screening, binning, transforming) SAS macros will be demonstrated.

“Screening” is the process of finding predictors with enough predictive power to be considered further.

For predictors having only a few levels, “binning” is a process to reduce the number of levels while maintaining predictive power.[1] Binning achieves parsimony and can reveal logical relationships between the predictor and the target.[2]

Finally, “transforming” of continuous numeric refers to how such a predictor is represented in a model in order to have good predictive power. For example, a transformation might replace continuous numeric (positive) X with Log(X) or by adding a second order term X^2.

THE GENERALIZED LOGIT

For the generalized logit the levels of the target can be character or numeric provided the ordering is not meaningful. It is most convenient to let the target levels be integers 1 to J. Let J ≥ 2 and denote the target by Y. Let $P_{i,j}$ be the probability that $P(Y_i=j)$ for the i^{th} subject or observation.

[1] “Few” is a subjective term but can be viewed as under 16 and perhaps no more than 10. Predictors with a few levels fall into three categories: nominal, ordinal, and discrete-numeric.
[2] Binning for the binary logit model is presented in Lund (2017) and for the cumulative logit model in Lund (2019).

When J = 2 the generalized logit is the familiar binary logistic. Let a binary target have levels 1 and 2 with probabilities $P_1 = P(Y=1)$ and $P_2 = P(Y=2)$. In binary logistic there is a linear combination of predictors and coefficients, denoted by xbeta = $\sum_{k=0}^{K} \beta_k x_{i,k}$, related to odds $P_{i,1} / P_{i,2}$ as shown:

$$P_{i,1} / P_{i,2} = \exp(\sum_{k=0}^{K} \beta_k x_{i,k}) = \exp(\text{xbeta}) \quad \ldots x_{i,0} = 1$$

where "i" indexes the observation and there are K predictors.

For the generalized logit the level "J" is traditionally taken as a reference level, or base level. With this convention, the probability of each level j = 1 to J-1 is compared to the probability for level J to form J-1 odds for j = 1 to J-1. Each odds equals an $\exp(\text{xbeta}_j)$ where the coefficients $\beta_{j,k}$ for predictor X_k depend on j.

$$P_{i,j} / P_{i,J} = \exp(\text{xbeta}_j) = \exp(\sum_{k=0}^{K} \beta_{j,k} x_{i,k}) \quad \ldots x_{i,0} = 1$$

These J-1 equations are called the response equations.

Using these J-1 equations and $\sum_{j=1}^{J} P_{i,j} = 1$, the formulas for $P_{i,j}$ are found:

$$P_{i,j} = \exp(\text{xbeta}_{i,j}) / (1 + \sum_{h=1}^{J-1} \exp(\text{xbeta}_{i,h})) \text{ for } j = 1 \text{ to } J\text{-}1$$

$$P_{i,J} = 1 / (1 + \sum_{h=1}^{J-1} \exp(\text{xbeta}_{i,h}))$$

Formally, $P_{i,J} / P_{i,J} = \exp(\text{xbeta}_J) = 1$ for all X. This implies that each $\beta_{J,k}$ is set to 0.

The coefficients (and probabilities) are fit simultaneously by maximizing the likelihood function:

$$L(\underline{b}) = \prod_{i=1}^{n} \prod_{j=1}^{J} P_{i,j}^{Y_{i,j}}$$

where $Y_{i,j} = 1$ if $Y_i = j$, else $Y_{i,j} = 0$ and sample size is n.

A new feature was added to PROC LOGISTIC in SAS/STAT 14.1 for the generalized logit. This new feature is the statement EQUALSLOPES. This statement allows the modeler to specify which predictors may have equal slopes across the response equations.

For example, if the predictors X1 and X2 are in a generalized logit model, then the use of EQUALSLOPES, below, forces X2 to have equal slopes across the response equations.

```
PROC LOGISTIC DATA= Test;
MODEL Y= X1 X2 / LINK= glogit EQUALSLOPES= (X2);
run;
```

UNEQUALSLOPES is the default for PROC LOGISTIC and it will be the main focus of this paper.

CUMULATIVE LOGIT PROPORTIONAL ODDS MODEL

Let J ≥ 2 and denote the target by Y with levels 1 to J. For the cumulative (cum) logit, the levels for the target can be character or numeric provided the ordering is meaningful. Let $P_{i,j}$ be the probability that $P(Y_i=j)$ for the i^{th} observation. The Cum Logit Proportional Odds (PO) model is defined by the J-1 response equations as follows:

$$\sum_{j=1}^{J_o} P_{i,j} / \sum_{j=J_o+1}^{J} P_{i,j} = \exp(\alpha_{J_o} + \sum_{k=1}^{K} \beta_k x_{i,k}) \ldots \text{ for } 1 \le J_o \le J\text{-}1$$

where $\alpha_j < \alpha_{j+1}$ for all j, the β_k are slope parameters with no restriction, and X_k are predictors.

Note:

- β_k does not depend on j.
- $\alpha_{j_1} + \sum_{k=1}^{K} \beta_k x_{i,k} < \alpha_{j_2} + \sum_{k=1}^{K} \beta_k x_{i,k}$ for $j_1 < j_2$ for all x.

When J = 2 the cumulative logit is the usual binary logistic model.

In order to simplify the equation above, suppose the target Y has levels A, B, C and there are two predictors X_1 and X_2. Here are the two response equations, this time after taking logarithms:

$$\text{Log}\ (P_{i,A} / (P_{i,B} + P_{i,C})) = \alpha_A + \beta_{X1}*X_{i,1} + \beta_{X2}*X_{i,2}$$
$$\text{Log}\ ((P_{i,A} + P_{i,B}) / P_{i,C}) = \alpha_B + \beta_{X1}*X_{i,1} + \beta_{X2}*X_{i,2}$$

CUMULATIVE LOGIT PARTIAL PROPORTIONAL ODDS (PPO) MODEL

In this generalization, one or more of the coefficients of the predictors can be depend on j. In the example below the target has three levels A, B, C and there are two predictors X_1 and X_2. The coefficients of X_2 will be allowed to be different in the two response equations:

$$\text{Log}(P_{i,A} / (P_{i,B} + P_{i,C})) = \alpha_A + \beta_{X1}*X_{i,1} + \beta_{X2,A}*X_{i,2}$$
$$\text{Log}(P_{i,A} + P_{i,B}) / P_{i,C})) = \alpha_B + \beta_{X1}*X_{i,1} + \beta_{X2,B}*X_{i,2}$$

In PROC LOGISTIC the cum logit PPO model is implemented by use of the UNEQUALSLOPES statement which allows the modeler to specify which predictors may have unequal slopes across the response equations. For example, if the predictors are X1 and X2 for a cum logit model, then the use of UNEQUALSLOPES, below, allows X2 to have unequal slopes across the response equations.[3]

```
PROC LOGISTIC DATA= Test;
MODEL Y= X1 X2 / UNEQUALSLOPES= (X2);
run;
```

EQUALSLOPES is the default. The cumulative logit model is fit by maximum likelihood estimation.

NOD PREDICTORS

A Nominal, Ordinal, or Discrete-numeric predictor X with number of levels more than 2 but generally less than 16 (and typically ≤ 10) will be called a NOD predictor. Normally, such predictors enter a logistic model as dummy variables via a CLASS statement. Alternatives to using the CLASS statement could be (i) weight of evidence coding or (ii) conversion of X to become a numeric predictor if X has numeric or ordinal scaling (not discussed in this paper). The meaning and usage of weight of evidence coding of a NOD X for the cumulative logit and generalized logit is defined and discussed in a later section.

THE SATURATED MODEL

For a NOD predictor X with L levels the *saturated* model, for cum logit or generalized logit, are given by:

Generalized Logit:

```
PROC LOGISTIC DATA= Test;
CLASS X;
MODEL Y= X / UNEQUALSLOPES= (X) /* = default */ LINK= glogit;
run;
```

Cumulative Logit:

```
PROC LOGISTIC DATA= Test;
CLASS X;
MODEL Y= X / UNEQUALSLOPES= (X);
run;
```

In the saturated model there are L*(J-1) parameters. The probabilities P(Y = j | X = i) for j =1 to J-1 and i=1 to L are the row percentages of the Table X*Y. In particular, the log-likelihood and likelihood ratio chi-square (LRCS) are equal for the two models above.[4] For large samples this LRCS has a chi-square distribution with (L-1)*(J-1) degrees of freedom.

---

[3] See NOTE at bottom of SAS webpage of Example 72.18 for discussion of the possibility of negative probabilities. http://support.sas.com/documentation/cdl/en/statug/68162/HTML/default/viewer.htm#statug_logistic_examples22.htm

[4] See Appendix 1 for formulas and an example calculation of the LRCS

The LRCS is a measure of predictive power of the saturated version of X to predict Y. If X and Y are independent, [i.e. P(Y=j | X=i) = P(Y=j)], then LRCS=0. At the other extreme, if for each j there is some X=i so that P(Y=j | X=i) = 1, then LRCS= -2*[Log-Likelihood of intercepts-only model] … using the convention that 0*Log(0) = 0. The larger the LRCS, the more the divergence from independence.

The "c" or Model c (or c-statistic) is computed for the cumulative logit model.[5] For the saturated model the Model c is simply the "c" from PROC LOGISTIC; CLASS X; MODEL Y=X / UNEQUALSLOPES=(X);.

I have not found a paper that gives guidelines for what would constitute a good "c" in general, nor for "c" in the special situation of the saturated model. I've run some simulations which suggest that a "c" above 0.60 indicates a strong predictor in the case of a saturated model.

SCREENING NOD PREDICTORS: %MULTI_LOGIT_SCREEN_1

The macro %MULTI_LOGIT_SCREEN_1 (DATASET, TARGET, INPUT, SORT) computes:
(i) likelihood ratio chi-square (LRCS) for saturated model, (ii) LRCS significance level, and (iii) model c (meaningful only if cum logit is being considered). This is done for each predictor listed in the INPUT parameter. Target and predictors can be numeric or character.

The major processing by the macro is performed by just one PROC SUMMARY.

The right-most macro parameter is called SORT. It designates the sort sequence of the output Table. Choices are "LRCS" (significance of LRCS), "Model_c", or "space". If space, then INPUT order is used.

EXAMPLE

The macro %MULTI_LOGIT_SCREEN_1 will be applied to the data set TEST01.[6] It is assumed for this example that the target Y is not ordered.

```
%LET SLOPE1 = 0.01; %LET SLOPE2 = 0.05; %LET SLOPE3 = 0.10;
%LET SLOPE4 = 0.20; %LET SLOPE5 = 0.99; %LET P_Seed = 5;
DATA TEST01;
Do i = 1 to 8000;
   X1 = floor(12*ranuni(2)) - 1.5;
   X2 = floor(2*ranuni(2)) - .5;
   X3 = floor(2*ranuni(2)) - .5;
   X4 = floor(2*ranuni(2)) - .5;
   X5 = floor(2*ranuni(2)) - .5;
   C1 = put(floor(4*ranuni(2)),z2.);
   C1_all = &SLOPE1*(C1='00') + &SLOPE2*(C1='01') + &SLOPE3*(C1='02');
   T = exp(0 + C1_all + &SLOPE1*X1 + &SLOPE2*X2 + &SLOPE3*X3 + &SLOPE4*X4 +
      &SLOPE5*X5);
   U = exp(1 + C1_all + &SLOPE1*X1 + &SLOPE2*X2 + &SLOPE3*X3 + &SLOPE4*X4 +
      &SLOPE1*X5);
   PA = 1 - 1/(1 + T);
   PB = 1/(1 + T) - 1/(1 + U);
   PC = 1 - (PA + PB);
   R = ranuni(&P_Seed);
   if R < PA then Y = "A"; /* Assign Y to match model probabilities */
   else if R < (PA + PB) then Y = "B";
   else Y = "C";
   Output;
   End;
run;
```

Here, the macro call and report from %MULTI_LOGIT_SCREEN_1 are shown:

[5] For the computation see Lund (2019 p. 5 footnote)
[6] This data set simulates data from a cum logit PO model, although this is not relevant for the example.

```
%Multi_Logit_Screen_1(TEST01, Y, X1 X2 X3 X4 X5 C1, LRCS);
```

Obs	Var_Name	Levels	LRCS	df	Pr > ChiSq	MODEL_C
1	X5	2	708.03	2	<.0001	0.578
2	X4	2	27.70	2	<.0001	0.525
3	C1	4	11.27	6	0.0804	0.517
4	X3	2	1.25	2	0.5365	0.505
5	X2	2	1.19	2	0.5512	0.504
6	X1	12	20.41	22	0.5572	0.518

NOTE: Model c is not meaningful for the generalized logit

Table 1

One use of right-tail probability, Pr > ChiSq, is to rank the predictors for the given data and target variable, and then investigate the least significant for elimination. Here, X1, X2, X3 are candidates for elimination.

There is no cut-off value across applications for Pr > ChiSq since the tail probability is influenced by sample size. In this regard, LRCS might be computed for bootstrap samples of size 1000 (assuming a full sample of N ≥ 1000) to obtain average Pr > ChiSq's and a measure of variability. The bootstrap sample size is set to 1000 so that the average right-tail probability and its variability can be compared to a standard alpha, such as 0.15.

In this example just 10 bootstrap samples were used. PROC SURVEYSELECT provides bootstrap sampling.[7] Averages of Pr > ChiSq from 10 bootstrap samples for X5, X4, X1, X2, X3, C1 are shown below. Only X5 had right-tail probability consistently below 0.15. Predictors X2, X3, and C1 are candidates for elimination.

Var_Name	Average of Pr > ChiSq for 10 bootstrap samples	Percent where (Pr > ChiSq) < 0.15
X5	<.0001	100%
X4	0.189	80%
X1	0.219	40%
X2	0.624	10%
X3	0.520	10%
C1	0.402	30%

Table 1b

BINNING NOD PREDICTORS

Binning is widely used for simplifying NOD predictors for binary logistic models. Binning is the process of reducing the number of levels of a NOD predictor to achieve parsimony while preserving, as much as possible, the power of the predictor. This practice is especially prevalent in credit risk modeling.

Suppose X has levels 1, 2, 3. A step in binning might be to combine 1 and 3. Now the binned X has 2 levels {1, 3}, {2}. Here, "*non-adjacent*" binning was allowed (1 and 3 are not adjacent in the ordering of X). If only *adjacent* levels may be combined, then {1, 2}, {3} is an allowable step. The other allowable step is {1}, {2, 3}. A modeler chooses *non-adjacent* or *adjacent* binning depending on meaning of X and its purpose in the model.

[7]
```
PROC SURVEYSELECT DATA=TEST01 out=BootSamples noprint seed=123
 n=1000 method=urs  reps=10 /* 10 resamples */;
```

THE FINAL BINNING SOLUTION

A k-bin solution is defined by the membership of levels of X within k bins. The *final binning solution* is given by specifying the number k of bins and the membership of the levels of X in these k bins.

When binning predictors for binary logistic regression, the binning process is designed to attempt to maximize the information formation (IV) of the final binning solution. As an alternative to maximizing IV, the -2*log(likelihood) for the saturated model of Y for the binned predictor X can be minimized.[8]

EXTENSION OF IV TO GENERALIZED LOGIT AND CUM LOGIT

How can the ideas of information value and -2*log(likelihood) be extended to the generalized logit and cumulative logit in order to guide the binning process? The extension of -2*log(likelihood) is straightforward, simply compute -2*log(likelihood) for the saturated model of Y and predictor X.

But an extension of IV is needed in order to apply the concept of information value, for the purpose of binning, to the generalized logit and cumulative logit.

This extension depends on forming “binary splits” of target Y. But the definition of “binary split” varies with whether cumulative or generalized logit is being considered.

For the cumulative logit the observations are divided (low vs. high) at a split point of the ordered target levels. There are J-1 splits. For example, if target has levels A, B, C there are 2 splits: {A} v. {B, C} and {A, B} v. {C}. For the cumulative logit, all observations are utilized for each binary split.

For the generalized logit the binary splits are between a target level “j” and the target base “J”. The target base takes a special role. Only the observations with target levels “j” or “J” are used in the “j” split. For a target with levels A, B, C, with C as the base, there are 2 splits: {A} v. {C} and {B} v. {C}.

Consider the X*Y Data Sample (Table 2) and the two different forms of binary splits. For the generalized logit the {A} v. {C} split uses only 13 observations in total. The ordered split {A} v. {B, C} for the cum logit uses all 17 observations in the sample.

	Data Sample				Generalized Logit					Cum Logit			
	Y				Split: Level v. Base					Ordered Binary Split			
X	A	B	C	X	A	C	B	C	X	A	{B, C}	{A, B}	C
1	4	1	1	1	4	1	1	1	1	4	2	5	1
2	3	1	3	2	3	3	1	3	2	3	4	4	3
3	1	2	1	3	1	1	2	1	3	1	3	3	1

Table 2

Now the usual binary information value is computed for each split (see Table 3).

Generalized Logit Split: Level v. Base	IV (Info Value)	Cumulative Logit Ordered Split	IV (Info Value)
A v. C	0.4158	A v. {B, C}	0.4413
B v. C	0.5924	{A, B} v. C	0.3269

Table 3

%MULTI_LOGIT_BIN

An algorithm, in the form of a SAS macro %MULTI_LOGIT_BIN, performs binning for either cumulative logit or generalized logit.[9] The modeler must first select adjacent or non-adjacent binning in order to identify which pairs of levels are eligible for collapse. Then the modeler picks a criterion (same criterion at every k^{th} step) to optimize in the search for the eligible pair to collapse.[10]

[8] Equivalent to maximizing LRCS for saturated models.
[9] Binning for binary logit model was presented in Lund (2017) and for cumulative logit model in Lund (2019).
[10] In %MULTI_LOGIT_BIN parameter MODE: non-adjacent binning is coded “A” and adjacent binning is coded “J”.

Here are four criteria that can guide the binning process at each k^{th} step. These criteria apply to both the cumulative and generalized logit, but with the distinctive computations of the IV's.

1. Maximize IV: IV is the average of the IV's for the binary splits
2. Maximize MIN_IV: MIN_IV is the smallest IV across the binary splits
3. Maximize MAX_IV: MAX_IV is the largest IV across the binary splits
4. Minimize -2*LL: for saturated model of Y and X. This criterion is equivalent to maximizing LRCS.

The X*Y sample from Table 2 is first regarded as a sample from a cumulative logit population. If non-adjacent binning is used, then there are 3 possible 2-bin solutions. The best 2-bin solution according to IV, MAX_IV, and MIN_IV is {1} {2, 3}. But {1, 2} {3} is best for criterion -2*LL. See Table 4.

Cum Logit (non-adjacent binning)					Ordered Binary Splits	
X BINs	- 2_LL	AVG_IV	MIN_IV	MAX_IV	IV: A v {B C}	IV: {A B} v C
{1} {2 3}	34.392	0.288	0.227	0.348	0.348	0.227
{1 3} {2}	34.653	0.157	0.020	0.293	0.020	0.293
{1 2} {3}	33.901	0.138	0.014	0.261	0.261	0.014

Table 4: Non-adjacent Bins

Choice of which of the 3 variants of IV to use is subjective. MAX_IV may insure that a single binary split determines the final binning solution. This approach could lead to a solution where X strongly differentiates between, say, {A, B, C, D} v. {E}, but is weak otherwise. MIN_IV looks for a final solution for X where even the weakest split is strong. Average IV is a compromise between MAX_IV and MIN_IV.

The X*Y sample from Table 2 is used again, now to show non-adjacent binning for the generalized logit model. When considering the splits for the generalized logit, the best 2-bin solution is {1, 3} {2} according to IV, MAX_IV, and MIN_IV. But {1, 2} {3} is best for the criterion -2*LL. See Table 5. Choice of base (here it is "3") can significantly affect the IV rankings. But -2*LL is not affected by choice of base.

Generalized Logit (non-adjacent binning)					Split: Level v. Base	
X BINs	- 2_LL	AVG_IV	MIN_IV	MAX_IV	IV: A v B	IV: B v C
{1,3} {2}	34.653	0.366	0.206	0.526	0.206	0.526
{1,2} {3}	33.901	0.229	0.042	0.416	0.042	0.416
{1} {2,3}	34.392	0.215	0.014	0.416	0.416	0.014

Table 5 Non-adjacent Bins

If predictor X has L levels, the binning takes L-2 steps to reach a 2-bin solution. Stopping rules for binning are, frankly, subjective. If there is a sharp drop-off in the binning criterion from k bins to k-1 bins, then the "rule" is stop at k bins. Other factors that affect when to stop are: (i) There should be no bins with a very small count of observations. (ii) Relationship between the target and the bins should be logical.

When a final binning solution is selected, it may not be optimal according to whatever criterion was selected. Even though at each step the optimal eligible pair is selected for collapse, it is possible that a non-optimal choice could eventually have led to a better final binning solution. I think the potential impact on the quality of the final binning solution, if reaching a non-optimal solution, is minor.

SAS statements for Weight of Evidence Coding and for the Bin Coding for each binary split are generated by %MULTI_LOGIT_BIN. For the binning solution {1} v. {2 3} for the cumulative logit model, the WOE and BIN codings are below. The WOE coding values are different for the generalized logit model.

CUM LOGIT: Weight of Evidence Coding		CUM LOGIT: Bin Coding
if X in (1) then X_woe1 = 0.8109302162 ;		if X in (1) then X_bin = 1;
if X in (1) then X_woe2 = 0.7339691751 ;		if X in (2, 3) then X_bin = 2;
if X in (2,3) then X_woe1 = -0.441832752 ;		
if X in (2,3) then X_woe2 = -0.315852949 ;		

Table 6

The parameter list for %MULTI_LOGIT_BIN and an example of a macro call is given in the Appendix 2.

REMARKS CONCERNING GENERALIZED LOGIT

There may be concern about using the three IV binning methods for the generalized logit because of the preferential role played by the selected base level.[11] For the generalized logit I believe that if the base has a large observation count, is distinctive in meaning, and is important to the interpretation of the model, then these IV-based binning approaches are useful in modeling.[12] Each level is compared to the "important" base. E.g. Consider the base of "no crime" versus "property crime" or "assault crime" for a sample of teenage victims. Here, it seems natural to compare the "crimes" against a base of "no crime".

In contrast, the method of -2*LL does not depend on the base and is fully general. Since the probabilities for the saturated model for cumulative logit and generalized logit are the row percentages from the X*Y table, these two -2*LL are equal for a given binning and lead to the same final binning solution.

DUMMIES AND WOE'S IN CUMULATIVE AND GENERALIZED LOGIT MODELS

Dummy variable coding of X_bin can be used in either model. If X has L levels, then there are L-1 dummies. The modeler must specify equal or unequal slopes.[13] For J target levels there are J-1 WOE transforms. These provide an alternative to using dummies. Again, choice of equal slopes or unequal slopes is needed. The macro computes the correlations between the J-1 WOE transforms. If high correlations, then some WOE transforms might be omitted from the model. For k=2 bins, correlation among any of the J-1 WOE's is 1. See example in Appendix 2 of using WOE transforms in a model.

But there is ambiguity regarding the degrees of freedom (d.f.) to assign to a WOE variable. For discussion of the d.f. problem for WOE's, see Lund (2021). This makes predictor selection methods that depend on d.f. (e.g. P-values, AIC, BIC) dubious when applied to WOE predictors.[14]

TRANSFORMING CONTINUOUS NUMERIC PREDICTORS

A *continuous* predictor X is *numeric with many levels*. Predictor X could measure miles, minutes, dollars, etc. The Function Selection Procedure (FSP) recommends a transformation of X, such as Log(X) or X**2. The FSP is discussed in detail in Royston and Sauerbrei (2008) (hereafter R-S) in their book *Multivariate Model-building*. In R-S the FSP is applied to find transformations of predictors for the binary logistic model as well as for ordinary regression and Cox regression.

The FSP approach can be applied to the generalized logit model and the cumulative logit model (PO and PPO). Most of the mechanics of FSP extend without change from the binary case.

FSP HAS TWO PRELIMINARY STEPS

First, the predictor X must be positive and, if needed, a translation of X is applied. In fact, for numerical stability, it is better to translate X so that the minimum of X is 1 (if minimum is not already 1 or more).

Next, a class of transformations of X, called fractional polynomials (FP), is defined. These fractional polynomials are given by:

X^p where p is taken from S = {-2, -1, -0.5, 0, 0.5, 1, 2, 3} and where "X^0" denotes log(X)

FP1 refers to the collection of 8 linear functions formed by the selection of one X^p. That is,

$$g(X,p) = \beta_0 + \beta_1 X^p$$

In this definition it is the "p" that matters. For the generalized logit and cum logit PPO the coefficients would vary across the response equations. This variation is not relevant to the definition of FP1.

---

[11] The total number of binary splits is $\binom{J}{2}$. Would computing IV's for all these (i) improve, (ii) not affect, or (iii) make worst the binning solutions based on total IV? I have not investigated this reasonable question.

[12] Compare to the approach by Begg and Gray (1984) of fitting J-1 binary logistics of non-base levels to a base.

[13] A model comparison test of Unequalslopes v. Equalslopes for CLASS X; MODEL Y=X / … could be conducted.

[14] Recommendation: Use CLASS X (i.e. dummy variables) when employing a predictor selection method to fit a model. Once predictors have been selected, then replace CLASS variables by WOE transforms and refit the existing model. Fit statistics, on a validation sample, of the CLASS model and WOE model are compared to determine if all (or some) of the WOE predictors can be used.

FP2 refers to the collection of 36 linear functions formed by selection of two X^p as shown below:

$$g(X,p_1,p_2) = \beta_0 + \beta_1 X^{p1} + \beta_2 X^{p2} \qquad p_1 \neq p_2 \ldots 28 \text{ pairs}$$
$$g(X,p_1,p_1) = \beta_0 + \beta_1 X^{p1} + \beta_2 X^{p1} \log(X) \qquad p_1 = p_2 \ldots 8 \text{ pairs}$$

For each response equation, FP1 produces only monotonic curves (since [translated] X ≥ 1). FP2 produces curves with a variety of non-monotonic shapes.

FSP HAS TWO MAIN STEPS

I. Searching for Best Transformations: There is an exhaustive search of FP1 to find the function with maximum likelihood in a logistic regression.[15] This is called the FP1 solution. Then, a second exhaustive search of FP2 is conducted to find the function with maximum likelihood within this collection. This is called the FP2 solution.

II. Performing Significance Testing: FSP significance testing of X has three steps called A, B, and C. The test-statistics for the three steps are the differences of -2 Log Likelihood's, as shown below. The test statistic is assumed to be chi-square with degrees of freedom give in Table 8. The degrees of freedom for the test statistic depends on the Step and the type of logistic regression.

$$\text{Test-Statistic} = (-2 \text{ Log Likelihood}_{\text{ Restricted Model}}) - (-2 \text{ Log Likelihood}_{\text{ FP2 Solution}})$$

where the "restrictive model" is, respectively, Null, Linear, FP1 models.

THE THREE STEPS (with references to Table 8):

A. Perform a "A" d.f. test at the α level of the -2 log likelihood of the FP2 solution against the -2 log likelihood null model (no predictor). If the test is not significant, drop X from consideration and stop; otherwise continue.

B. Perform a "B" d.f. test at the α level of the FP2 solution against X. If the test is not significant, stop, the recommended transform is linear X; otherwise continue.

C. Perform a "C" d.f. test at the α level of the FP2 solution against the FP1 solution. If the test is significant, the recommended transform is the FP2 solution, otherwise the FP1 solution is the recommended transform.[16]

	A: FP2 solution against Null (Intercept)	B: FP2 solution against X (Linear)	C: FP2 solution against the FP1 solution
Binary Logistic	4	3	2
Cum Logit PO	4	3	2
Cum Logit PPO	2*(J - 1) + 2	(J - 1) + 2	(J - 1) + 1
GL (unequalslopes)	2*(J - 1) + 2	(J - 1) + 2	(J - 1) + 1

Table 8

The significance testing and associated degrees of freedom for GL (and Cum Logit PPO) are based on:

- Assumption of large sample chi-square distribution for the test-statistic
- Intuitive, but not rigorously established, formulas for degrees of freedom shown in Table 8.

For example, the degrees of freedom for GL of "FP2 versus Null" is 2*(J-1) + 2. The first term, 2*(J-1), counts the coefficients for the two fractional polynomials across the J - 1 response equations. The additional term of "2" counts the choice of fractional polynomial exponent (from the list of 8).

For binary logistic the rationale for degrees of freedom (4, 3, 2) in the 3-step tests of FSP is given in R-S (p. 79). A similar rationale applies to cumulative logit PO model. See Lund (2018) for discussion. See Appendix 3 for some simulations have been run to support the d.f. formulas for the generalized logit.

[15] "Logistic regression" applies to model of interest: binary logistic, cum logit PO, cum logit PPO, or generalized logit.
[16] If FP2 solution is selected, then the two fractional polynomial transforms are both entered into the model.

IMPLEMENTATION OF FSP BY MACROS [17]

Presently, there are 3 macros for implementing FSP for (i) Cum Logit PO, (ii) Cum Logit PPO, (iii) Generalized Logit. All three run the binary logistic model but the Cum Logit PO version is recommended in this case. The macro names are: FSP_8LR, FSP_8LR_PPO, FSP_8LR_GLOGIT. The macro parameters are the same for all three macros. Here is the macro call for FSP_8LR_GLOGIT

%FSP_8LR_GLOGIT (DATASET, TARGET, INPUT, VERBOSE, ORDER, WEIGHT);

Parameter definitions:

DATASET: The data set containing the target and predictors
TARGET: Target variable (character or numeric). At least two non-missing levels
INPUT: Numeric predictors (at least one). Predictors are delimited by a space.
VERBOSE: YES … "YES" produces more output.
ORDER: A | D … Order for modeling the TARGET (A=ascending, D=descending). This is a dummy parameter (has no effect) except for %FSP_8LR
WEIGHT: A variable for the WEIGHT statement in PROC LOGISTIC | space (no weight variable)

Data set GL_01 simulates a generalized logit dataset with target Y with 3 levels and predictors X1 and X2. Predictor X2 enters the model as a linear effect, but X1 enters as the transform 1/X1.

```
Data GL_01;
  do i = 1 to 8000;
     X1 = floor(15*ranuni(3)) + 1;
     X2 = rannor(4);
     xbeta1 = (2*(1/X1) - 2*X2);
     xbeta2 = (1*(1/X1) + 2*X2);
     P1 = exp(xbeta1) / (1 + exp(xbeta1) + exp(xbeta2));
     P2 = exp(xbeta2) / (1 + exp(xbeta1) + exp(xbeta2));
     P3 = 1 - P1 - P2;
     R = ranuni(6);
     if R < P1 then Y = 1;
     else if P1 <= R < P1 + P2 then Y = 2;
     else Y = 3;
     output;
     end;
run;
%FSP_8LR_GLogit(GL_01, Y, X1 X2, NO, DUMMY,  );
```

Results are given in Table 9. The best transform for predictor X1 is (X1)**(-2) (i.e. "p=-2"). [But p-value for the FP2 solution of (X1)**(-1) and X1**2 is borderline insignificant at 10.5%.] The best transform for predictor X2 is "linear" (simply X2). The p-value for moving to FP1 is insignificant at 67.2%. In order to apply the eight fractional polynomial transforms, X2 had been translated by 4.665 units. Now this translation is unneeded.

Pred	Off-set	Test	Deviance	Test Stat	df	p-value	transform1	transform2
X1	0	Null v. FP2	16177.2	86.47	6	0		
X1	0	Linear v.	16143.5	52.79	4	0	Linear	
X1	0	FP1 v. FP2	16096.9	6.14	3	0.105	p=-2	
X1	0		16090.8				p=-1	p=2
X2	4.665	Null v. FP2	16177.2	6260.73	6	0		
X2	4.665	Linear v.	9918.8	2.35	4	0.672	Linear	
X2	4.665	FP1 v. FP2	9918.8	2.35	3	0.503	Linear	
X2	4.665		9916.5				log	p=3

Table 9

[17] Versions of this macro for cumulative logit model are presented in Lund (2018, 2019).

CONCLUSIONS

This paper presented three macros that can perform for the generalized logit model, the screening, binning, or transforming of predictors. These steps are performed prior to beginning the stage of model fitting.

One screening of NOD predictors uses the right-tail probability of the LRCS. This probability is sensitive to sample size with larger samples producing smaller, but perhaps less meaningful, right-tail probabilities. Instead, the macro MULTI_LOGIT_SCREEN_1 can be applied to numerous bootstrap samples of fixed sample size (e.g. 1000) to provide LRCS's. Then average right-tail probabilities, with a measure of variability, can be compared to a benchmark cutoff alpha value.

The introduction of generalized IV criteria provides an alternative approach to screening. The usage of generalized IV is specialized to cases where there is a distinctive reference level for the target. Values of the generalized IV's are dependent of the choice of base level.

Binning of NOD predictors can be guided by maximizing LRCS at each step or by maximizing one of the versions of the generalized IV's.

The binning macro MULTI_LOGIT_BIN also provides SAS code for weight of evidence transformations of NOD predictors. These WOE variables may be a useful replacement to dummy variables produced by a CLASS statement in PROC LOGISTIC.

The degrees of freedom formulas for the steps of FSP for the generalized logit have not been validated by simulation studies. A limited simulation study is given Appendix 3. A definitive simulation study would be a large undertaking.

In a strict sense, the macros %FSP_LR8 … do not precisely find the FP2 solution having maximum likelihood. To achieve a significant improvement in computational efficiency, the score chi-square is involved in finding the FP2 solution. See Lund (2018 p. 7) for discussion.

REFERENCES

Begg, C. and Gray, R. (1984). Calculation of polychotomous logistic regression parameters using individualized regressions, *Biometrika* Vol. 71, No. 1

Chatfield, C. (1995). *Problem Solving: A Statistician's Guide, 2nd Ed.*, Chapman & Hall/CRC.

Lund, B. (2017). SAS® Macros for Binning Predictors with a Binary Target, *Proceedings of the SAS Global Forum 20178 Conference,* Cary, NC, SAS Institute Inc., paper 969.

Lund, B. (2018). The Function Selection Procedure, *Proceedings of the SAS Global Forum 2018 Conference,* Cary, NC, SAS Institute Inc., paper 2390.

Lund, B. (2019). Screening, Transforming, and Fitting Predictors for Cumulative Logit Model, *Proceedings of the SAS Global Forum 2019*, paper 3067.

Lund, B. (2021). Weight of Evidence, Dummy Variables, and Degrees of Freedom, *Proceedings of the SAS Global Forum 2021* https://communities.sas.com/t5/SAS-Global-Forum-Proceedings/Weight-of-Evidence-Dummy-Variables-and-Degrees-of-Freedom/ta-p/726292

Royston P. and Sauerbrei W. (2008). *Multivariate Model-building,* John Wiley & Sons.

CONTACT INFORMATION

Your comments, questions, and requests for macro code are valued and encouraged. Contact the author Bruce Lund at: blund_data@mi.rr.com or blund.data@gmail.com

APPENDIX 1: LIKELIHOOD RATIO CHI-SQUARE FOR THE SATURATED MODEL

The data set is used to illustrate the saturated models for the cum logit PPO and for the generalized logit for predictor X and target Y. PROC LOGISTIC fits the saturated models.

```
DATA Test;
input X Y @@;
datalines;
1 1 1 2 1 3 1 1 2 1 2 3 2 3
2 3 2 2 3 3 3 1 3 1 3 3 3 2
;
PROC LOGISTIC DATA= Test;
CLASS X1;
MODEL Y= X / unequalslopes = (X);
run;
```

or

```
PROC LOGISTIC DATA= Test;
CLASS X1;
MODEL Y= X / link = GLOGIT;
run;
```

But to fit the saturated model, PROC LOGISTIC need not be run. The probabilities $p_{i,j}$ of target level j for row level i are equal to the row percentages in the X * Y cross-tabs, as in Table 10.

Here is the formula, with reference to Table 10:

$p_{i,j} = n_{i,j} / n_{i\cdot}$ … where $n_{i,j}$ gives cell counts, $n_{i\cdot}$ gives row total.

	Y			
X	1	2	3	Tot
1	$n_{1,1}$ = 2 .50	1 .25	1 .25	$n_{1\cdot}$ = 4
2	1 .20	1 .20	3 .60	$n_{2\cdot}$ =5
3	2 .40	1 .20	2 .40	$n_{3\cdot}$ =5
	$n_{\cdot 1}$= 5	$n_{\cdot 2}$ = 3	$n_{\cdot 3}$ = 6	n = 14

Table 10. Row percentages are the probabilities for the Saturated Model

The log-likelihoods and likelihood ratio chi-square can be computed from these probabilities by the formulas below:

LL(full) = Log-Likelihood (intercept and covariates) = $\sum_{i=1}^{3} \sum_{j=1}^{3} n_{i,j} * \log(p_{i,j})$ = -14.185
LL(restricted) = Log-Likelihood (intercept Only) = $\sum_{j=1}^{3} n_{\cdot j} * \log(n_{\cdot j} / n)$ = -14.853
Likelihood Ratio Chi-Sq = -2 * { LL(restricted) - LL(full) } = 1.337 (4 d.f.)

If independence, then P(Y=j | X=i) = P(Y=j) or $n_{i,j} / n_{i\cdot} = n_{\cdot j} / n$. Substituting $n_{i,j} / n_{i\cdot} = n_{\cdot j} / n$ into $\sum_{i=1}^{3} \sum_{j=1}^{3} n_{i,j} * \log(p_{i,j})$, after some manipulation, gives $\sum_{j=1}^{3} n_{\cdot j} * \log(n_{\cdot j} / n)$. Then, LRCS = 0.

APPENDIX 2: PARAMETERS AND EXAMPLE FOR %MULTI_LOGIT_BIN

MODEL: GLOGIT | CUMLOGIT, (space defaults to CUMLOGIT)

DATASET: Data set to be processed

TARGET: Target variable (numeric or character) with at least 2 levels. Missing is ignored.

X: Predictor variable (numeric or character).
If numeric, then **X** has integer values from 0 to 99. If character, then “embedded space”, !, +, _, # may

not occur in the value. If character values represent an ordered predictor, then care is needed to assign values that give the intended ordering. **X** must have at least 2 levels. Missing is ignored.

W: A frequency variable if present in **DATASET**. Otherwise enter 1. Space is not permitted as an entry.

METHOD: LL or IV or MIN_IV or MAX_IV

MODE: A or J. A = Any pairs of levels can be combined. J = Only pairs with adjacent levels can be combined.

ONE_ITER: YES | <other>. Statistics *only* for no-binning solution. **ONE_ITER** has priority over **MIN_BIN**

MIN_BIN: INTEGER > 1 | space.
Integer value restricts the processing to bin solutions where the number of BINs is greater or equal to the INTEGER. If <space>, then all bin solutions are processed.

VERBOSE: If not YES, then only Summary Report is displayed. **VERBOSE**=YES can be run to obtain SAS code statements for WOE and BINs for all k^{th} steps.

ZERO_ADJ: YES | any other. If YES, then adds 1 to a cell with zero count. (Zero count causes a STOP.)

RUN_TITLE: Title1 for all Reports. No commas in the title.

EXAMPLE DATA SET AND %MULTI_LOGIT_BIN MACRO CALL [18]

```
Data Summary;
input obs Severity Age _freq_ @@;
datalines;
1  1  16 1   17 1  33 2  32 2  26 10  47 3  24 1
2  1  18 4   18 1  34 2  33 2  27 3   48 3  25 2
3  1  19 5   19 1  35 1  34 2  28 4   49 3  26 2
4  1  20 3   20 1  37 1  35 2  29 3   50 3  27 1
5  1  21 12  21 1  39 1  36 2  30 4   51 3  28 1
6  1  22 5   22 1  42 4  37 2  31 2   52 3  29 1
7  1  23 7   23 2  17 1  38 2  32 3   53 3  30 2
8  1  24 12  24 2  18 3  39 2  35 1   54 3  31 1
9  1  25 7   25 2  19 1  40 2  36 1   55 3  32 2
10 1  26 8   26 2  20 3  41 2  37 1   56 3  33 1
11 1  27 4   27 2  21 3  42 3  15 1   57 3  34 1
12 1  28 4   28 2  22 6  43 3  19 1   58 3  35 2
13 1  29 3   29 2  23 3  44 3  20 1   59 3  36 1
14 1  30 3   30 2  24 5  45 3  21 1   60 3  38 1
15 1  31 1   31 2  25 3  46 3  23 2   61 3  39 2
16 1  32 3
;
DATA Backache; length Age_group $8; Set Summary;
  If SEVERITY=1 then TARGET="C";
  Else If SEVERITY=2 then TARGET="B";
  Else If SEVERITY=3 then TARGET="A";
   if age <= 19 then Age_group="15to19";
   else if age <= 22 then Age_group="20to22";
   else if age <= 24 then Age_group="23to24";
   else if age <= 26 then Age_group="25to26";
   else if age <= 28 then Age_group="27to28";
   else if age <= 30 then Age_group="29to30";
   else if age <= 32 then Age_group="31to32";
   else if age <= 35 then Age_group="33to35";
   else                   Age_group="36andUP";
run;
```

[18] Chatfield (1995, Exercise D.2).

REMARKS: SEVERITY=1 indicates no symptoms. This category is distinctive from SEVERITY 2 and 3. For this reason TARGET = “C” (SEVERITY=1) is made the base. (All three levels have about the same counts.)

Here is the macro call for the generalized logit binning of TARGET versus AGE_GROUP.[19]

```
%MULTI_LOGIT_BIN( MODEL=GLOGIT, DATASET=BACKACHE, TARGET=TARGET,
X=Age_group, W=_FREQ_, MODE=A, METHOD=IV, ONE_ITER=, MIN_BIN=, VERBOSE=,
ZERO_ADJ=, RUN_TITLE=);
```

Table 11 gives the summary report of collapsing (“+” = just collapsed, “_” = previously collapsed) and the binning statistics for k = 9 to 2.

There is a large decrease in Avg_IV after Step 5. Stopping at Step 5 is suggested.

k	Levels collapsed (see “+”)	LRCS	AVG _IV	MIN _IV	MAX _IV	IV_1	IV_2	corr_woe _1_2
9		17.602	0.331	0.185	0.478	0.478	0.185	0.063
8	25to26+27to28	17.596	0.331	0.185	0.477	0.477	0.185	0.063
7	15to19+23to24	17.499	0.329	0.183	0.474	0.474	0.183	0.055
6	29to30+31to32	17.335	0.325	0.183	0.467	0.467	0.183	0.054
5	33to35+36andUP	17.086	0.322	0.178	0.466	0.466	0.178	0.077
4	15to19_23to24+20to22	16.151	0.303	0.168	0.439	0.439	0.168	0.126
3	25to26_27to28+29to30_31to32	14.668	0.265	0.157	0.373	0.373	0.157	0.054
2	25to26_27to28_29to30_31to32+33to35_36andUP	6.667	0.179	0.049	0.310	0.310	0.049	1.000

Table 11 Binning Summary Report

The best k-bin solutions for k = 8 to 2 are given in Table 12.

k	__BIN_1	__BIN_2	__BIN_3	__BIN_4	__BIN_5	__BIN_6	__BIN_7	__BIN_8
8	15to19	20to22	23to24	25to26+27to28	29to30	31to32	33to35	36andUP
7	15to19+23to24	20to22	25to26_27to28	29to30	31to32	33to35	36andUP	
6	15to19_23to24	20to22	25to26_27to28	29to30+31to32	33to35	36andUP		
5	15to19_23to24	20to22	25to26_27to28	29to30_31to32	33to35+ 36andUP			
4	15to19_23to24+ 20to22	25to26_27to28	29to30_31to32	33to35_ 36andUP				
3	15to19_23to24_ 20to22	25to26_27to28+ 29to30_31to32	33to35_ 36andUP					
2	15to19_23to24_ 20to22	25to26_27to28_ 29to30_31to32+ 33to35_36andUP						

Table 12

The macro produces SAS code for the two WOE transforms for the 5-bin solution. The correlation between the two WOE transforms is very low at 0.0772. The two predictors, Age_groups_woe1 and Age_groups_woe2, could be used in the model to replace:

CLASS Age_group_bin;

The SAS code below is generated by the macro. It would be inserted into a DATA Step before running PROC LOGISTIC.

[19] See Lund (2019) where this same example is binned as a cumulative logit

```
if Age_group in ( "15to19","23to24" ) then Age_group_woe1= -0.52109529 ;
if Age_group in ( "15to19","23to24" ) then Age_group_woe2= -0.364091542 ;
if Age_group in ( "20to22" ) then Age_group_woe1= -1.065822466 ;
if Age_group in ( "20to22" ) then Age_group_woe2= -0.072570693 ;
if Age_group in ( "25to26","27to28" ) then Age_group_woe1= -0.10697212 ;
if Age_group in ( "25to26","27to28" ) then Age_group_woe2= 0.2984929886 ;
if Age_group in ( "29to30","31to32" ) then Age_group_woe1= 0.7259370034 ;
if Age_group in ( "29to30","31to32" ) then Age_group_woe2= 0.6205764877 ;
if Age_group in ( "33to35","36andUP" ) then Age_group_woe1= 0.918308896 ;
if Age_group in ( "33to35","36andUP" ) then Age_group_woe2= -0.861028053 ;
```

A skeleton of the PROC LOGISTIC is:

```
PROC LOGISTIC DATA= <>;
CLASS <>;
MODEL TARGET= Age_group_woe1 Age_group_woe2 <others> / LINK=GLOGIT;
```

APPENDIX 3: FSP SIMULATIONS AND DISCUSSIONS

There are too many dimensions to control in order to conduct a definitive simulation study. These dimensions are: (1) the transformations of X that appears in the response equations of the model as well as other X's in the model, (2) the number of levels of the Target, (3) the values of the coefficients of X.

The data set GL_01, given earlier, is re-used in the simulations. Random seeds for X1, X2, and R are changed for each simulation run. Each simulated data set is run through %FSP_8LR_GLogit.

FSP STEP1: TESTING THE NULL CASE VS. FP2

TEST 1

The code that generates datasets SIM_1 to SIM_100 is shown below. The target Y has J=3 levels. The coefficient of 1/X1 is set to zero. Here is the macro call and the code to create the data sets:

```
%FSP_8LR_Glogit(SIM_&Seed, Y, X1, NO, );

%DO Seed = 1 %TO &Num;
DATA SIM_&Seed;
do i = 1 to 8000;
  X1 = floor(15*ranuni(&Seed)) + 1;
  X2 = rannor(&Seed);
  xbeta1 = (0*(1/X1) - 2*X2);
  xbeta2 = (0*(1/X1) + 2*X2);
  P1 = exp(xbeta1) / (1 + exp(xbeta1) + exp(xbeta2));
  P2 = exp(xbeta2) / (1 + exp(xbeta1) + exp(xbeta2));
  P3 = 1 - P1 - P2;
  R = ranuni(&Seed);
  if R < P1 then Y = 1;
   else if P1 <= R < P1 + P2 then Y = 2;
   else Y = 3;
  output;
  end;
/* more code follows */
```

The formula in Table 8 for degrees of freedom for testing NULL vs. FP2 gives 6 = (J-1)*2 + 2. Since the NULL is true (X1 does not enter the model), the expected number of rejections of the NULL with 6 d.f. at α = 5% out of 100 data sets should be about 5. Likewise, 10 if α = 10% and 15 if α = 15%.

The results are given in Table 13. The values of test statistic T are denoted by t. The percentage of rejections [P(T > t) < p-value] of the NULL model are shown for p-value cut-offs of 5%, 10% and 15%.

The simulations show that usage of 6 d.f., in this particular test, is good in that the number of rejections of the NULL hypothesis matches quite closely to the expected results. (See 5%, 10%, 17%).

p-value =	5%			10%			15%		
T (test-stat) d.f. =	5	6	7	5	6	7	5	6	7
P(T > t) < p-value	6% **(*)**	5%	1%	17%	10%	5%	26%	17%	10%

Table 13. FSP Step 1 – Rejection Rates of H_0: NULL model v. H_1: FP2 Solution … 100 cases
(*) Read: If the test statistic T is given 5 d.f., and if its test value is t, then there are 6% of cases where P(T > t) < 0.05

TEST 2

In TEST 2 a single change is made to SIM_1 to SIM_100. The coefficient of (1/X1) in the second response equation is changed to 1. Therefore, H_0: NULL model is false.

```
xbeta1 = (0*(1/X1) - 2*X2);
xbeta2 = (1*(1/X1) + 2*X2);
```

In the simulation, at 6 d.f. only 1 Type 2 error was committed at 5%.

p-value =	5%			10%			15%		
T (test-stat) d.f. =	5	6	7	5	6	7	5	6	7
P(T > t) < p-value	100%	99%	99%	100%	100%	99%	100%	100%	100%

Table 14. FSP Step 1 – Rejection Rates of H_0: NULL model v. H_1: FP2 Solution … 100 cases

FSP STEP2: TESTING LINEAR MODEL VS. FP2

TEST 3

The same code is used to generate datasets SIM_101 to SIM_200 with the exception of changes to the xbeta1 and xbeta2 statements. Now the focus is on X2. The predictor X2 is linear in the generalized logit model. Therefore, H_0: LINEAR model is true.

Here is the macro call and the code to create the data sets:

```
%FSP_8LR_Glogit(SIM_&Seed, Y, X2, NO, );

  xbeta1 = (2*(1/X1) - 2*X2);
  xbeta2 = (1*(1/X1) + 2*X2);
```

The d.f. formula of Table 8 gives 4 = (J-1) + 2. The simulations show that usage of 4 d.f., in this particular test, is overly conservative (too infrequently rejects NULL hypothesis). A test-statistic using 3 d.f. (3%, 12%, 16%) matches more closely to the expected results of 5, 10, 15.

p-value =	5%			10%			15%		
T (test-stat) d.f. =	3	4	5	3	4	5	3	4	5
P(T > t) < p-value	3%	2%	2%	12%	3%	2%	16%	8%	2%

Table 15. FSP Step 2 – Rejection Rates of H_0: LINEAR model v. H_1: FP2 Solution … 100 cases

For all 100 cases, with 6 d.f. the NULL model was rejected at <.0001 in the test:
H_0: NULL model v. H_1: FP2 Solution

FSP STEP3: TESTING FP1 VS. FP2

The selection of a specific FP1 transformation for a simulation does not really reflect the proper null hypothesis. The formal null hypothesis compares the FP1 model to the FP2 model where each model is the best fit to the given data. But in the simulation a specific FP1 solution must be selected in order to generate a dataset for running through FSP.

Further complicating the choice of an FP1 transformation for the simulation is the possibility that the LINEAR model will not be rejected. Without this rejection there is no test of FP1 vs. FP2. But many FP1 transformations may appear linear over the range of X.

TEST 4

The same code is used to generate datasets SIM_201 to SIM_300 with no changes to the xbeta1 and xbeta2 statements. But now the focus is on X1. The predictor X1 enters the generalized logit model as 1/X1. Therefore, H_0: FP1 model is true (with FP1 = 1/X1)

Here is the macro call and the code to create the data sets:

```
%FSP_8LR_Glogit(SIM_&Seed, Y, X1, NO, );

  xbeta1 = (2*(1/X1) - 2*X2);
  xbeta2 = (1*(1/X1) + 2*X2);
```

The d.f. formula of Table 8 gives 3 = (J-1) + 1. The simulations show that usage of 3 d.f., in this particular test, is good in that the number of rejections of the NULL hypothesis matches quite closely to the expected results (See 3%, 7%, 13%).

p-value =	5%			10%			15%		
T (test-stat) d.f. =	2	3	4	2	3	4	2	3	4
P(T > t) < p-value	10%	3%	1%	19%	7%	3%	26%	13%	4%

Table 16. FSP Step 3 – Rejection Rates of H_0: FP1 model v. H_1: FP2 Solution … 100 cases

For all 100 cases, at 6 d.f. the NULL model was rejected at <.0001 in the test:
H_0: NULL model v. H_1: FP2 Solution

For all 100 cases, at 4 d.f. the LINEAR model was rejected at < 0.05 in the test:
H_0: LINEAR model v. H_1: FP2 Solution

The 100 selected FP1 solutions are shown in Table 17. These transforms are very similar in their relationship to the target.

FP1_transform	Frequency
X1**(-0.5)	14
X1**(-1)	74
X1**(-2)	12

Table 16. FP1 Transforms selection in Step 3 … 100 cases

DISCUSSION OF DEGRESS OF FREEDOM

Degrees of freedom formulas for FSP in Table 8 as applied the generalized logit are not disqualified by these simulations. Furthermore, the d.f. formulas in Table 8 have appeal due to the logic of their formulation.

Many simulations would be required to provide definitive findings to determine appropriate degrees of freedom formulas. Notably, only the situation where Target Y had J=3 levels was simulated in this appendix and only two predictor variables were utilized in generating the simulated datasets.

DISCUSSION OF EQUAL SLOPES OPTION

PROC LOGISTIC offers the option of “equalslopes” for predictors in the generalized logit model. At this time there is no FSP macro for the generalized logit with the option for equalslopes. I recommend finding the FSP transformation for unequalslopes and then performing a model comparison test of this FSP solution with equalslopes vs. unequalslopes. If not significant, then the FSP solution could be implemented in the model with equalslopes.