

If There Is A Will, There Is A Way: Getting Around Defaults of PROC LOGISTIC

Chao-Ying Joanne Peng, Indiana University, Bloomington IN
Tak-Shing Harry So, Indiana University, Bloomington IN

ABSTRACT

Logistic regression modeling can be a powerful tool for analyzing data collected for social science research. Specifically, logistic regression is used to predict the likelihood of an odds ratio for a dichotomously scaled outcome measure from a series of explanatory variables. SAS/STAT® software has a versatile procedure (LOGISTIC) for performing logistic regression, both for fitting a specific logistic regression model and for selecting the best model to fit the data. Despite the wide range of very useful options/features already implemented into PROC LOGISTIC, this procedure has certain defaults and syntax conventions which can pose difficulties for social science researchers to deal with. In this paper, we identify six areas of PROC LOGISTIC for which alternative solutions may be needed in order to get around its limiting conventions. These areas include:

- Handling data structured in tabulated form or covariate pattern matrix, instead of the raw data;
- Treating the "success or event" category which is numerically coded at a higher level than the "failure or non-event" category;
- Converting categorical covariates into design variables for a logistic model;
- Incorporating interaction terms into a logistic model;
- Calculating the Somers' D statistics; and
- Selecting a hierarchical logistic model of the best covariates to fit the data.

All solution discussed in this paper are obtained within the SAS® System by using tactful SAS statements. Our paper also demonstrates SAS codes for generating three diagnostic plots (delta Pearson chi-square versus estimated probabilities, delta deviance versus estimated probabilities, and delta parameter estimate versus estimated probabilities) and the Receiver Operating Characteristic (ROC) curve using SAS powerful graphical procedure—GPLOT.

INTRODUCTION

Binary logistic regression is a regression technique that can be used to predict the likelihood of the odds ratio for a dichotomous outcome¹ variable. The logistic regression has the form

$$\ln \left(\frac{\hat{Y}}{1 - \hat{Y}} \right) = \alpha + \beta'x,$$

where $\hat{Y}$ is the outcome probability to be modeled, α is the intercept parameter, β is a vector of slope parameters, and x is a vector of predictors.

Logistic regression has been widely employed in epidemiological research, where often the outcome variable is presence or absence of some disease state (e.g., Yarandi and Simpson, 1991). Meanwhile, the use of logistic regression continues to grow in social sciences (e.g., Chuang, 1997; Janik & Kravitz, 1994; Ott, 1988; Tolman and Welsz, 1995) and researchers routinely rely on commercially available programs for estimation of parameters (i.e., α and β above).

The LOGISTIC procedure in SAS/STAT software is a versatile procedure for performing logistic regression, both for fitting a specific logistic regression model and for selecting the best model to fit the data. Despite the wide range of very useful options/features already implemented into PROC LOGISTIC, certain statistics, graphic output, and modeling techniques discussed in statistical literatures are not available. These shortcomings can be overcome within the SAS System by tactful SAS statements. These tactful statements are illustrated in this paper with a real-world data set adopted from Mroz (1987).

The data, used throughout this paper, described 752 married women's decision to (or not to) enter the labor forces in 1975, along with other relevant information. Table 1 presents descriptive information of eight variables from the data set.

Table 1. Descriptive statistics for the married women labor force participation data

Variable Name	Mean	S.D.	Min	Max	Description
lfp	0.567	0.496	0	1	1 if wife is in the paid labor force; else 0
k5	0.238	0.524	0	3	Number of children ages 5 or younger
k618	1.352	1.321	0	8	Number of children ages 6 to 18
age	42.547	8.073	30	60	Wife's age in years
hc	0.392	0.489	0	1	1 if husband attended college; else 0
wc	0.281	0.450	0	1	1 if wife attended college; else 0
inc	20.156	11.619	1.12	96	Family income excluding wife's wages (in \$1,000)
hwg	1.096	0.567	-2.054	3.219	Logarithm of wife's average hourly earnings

Note: Sample size = 752.

The dependent variable (i.e., lfp) is coded either 1 or 0, representing respectively whether married women were (1) or were not (0) participating in the paid labor force in 1975.

THREE INPUT DATA FORMATS

Three data formats are accepted by PROC LOGISTIC as input. These formats are illustrated in Table 2, using profiles from ten cases extracted from the "married women labor force participation data." Only the first six variables in Table 1 are used for illustrations.

Table 2. Three input data formats

(a) Data listed in raw data format

case	lfp	k5	k618	age	hc	wc
1	1	0	0	30	0	0
2	1	1	2	36	0	0
3	0	1	0	43	1	1
4	1	0	0	30	0	0
5	1	1	2	36	0	0
6	1	1	0	43	1	1
7	0	1	2	36	0	0
8	1	1	2	36	0	0
9	0	2	6	39	0	0
10	0	1	2	36	0	0

(b) Data listed in frequency data format

lfp	k5	k618	age	hc	wc	count
1	0	0	30	0	0	2
0	1	2	36	0	0	2
1	1	2	36	0	0	3
0	1	0	43	1	1	1
1	1	0	43	1	1	1
0	2	6	39	0	0	1

(c) Data listed in covariate pattern (or event/trial) format

k5	k618	age	hc	wc	event	trial
0	0	30	0	0	2	2
1	2	36	0	0	3	5
1	0	43	1	1	1	2
2	6	39	0	0	0	1

Raw data format [(a) above] is the style in which one row represents a case with the outcome and predictors² stored as variables across columns. Raw data format is the most common way of entering data for logistic regression. Data sets stored in a raw data format contain the richest information. McCullagh and Nelder (1989) suggest using this format for inputting data if the serial order of cases is relevant. An example of PROC LOGISTIC syntax for analyzing data stored in this format is as follows:

```
PROC LOGISTIC;
  MODEL lfp=k5 k618 age hc wc;
```

Sometimes, data are stored in a frequency data format—one row represents a set of cases which have the same pattern of outcome and predictors with frequency information stored in a variable (see Table 2b). For example, combining cases 1 and 5 in Table 2a forms the first row in Table 2b. The value (i.e., 2) under the variable *count* represents the number of occurrences in the raw data. The PROC LOGISTIC syntax below can be used to analyze data structured in frequency format (Table 2b):

```
PROC LOGISTIC;
  MODEL lfp=k5 k618 age hc wc;
  FREQ count;
```

Covariate pattern data format (also called the event/trial format) utilizes one row for each pattern of covariates with a variable containing the number of case(s) sharing the same covariate pattern, while another variable containing the number of case(s) having the "event" with that covariate pattern (see Table 2c). The five cases (i.e., cases 2, 5, 7, 8, and 10), sharing the same covariate pattern in Table 2a,

form the second row in Table 2c. A value of 5 is therefore entered under the variable *trial* to reflect the number of cases having the same covariate pattern. A value of 3 for the variable *event* indicates that three out of the five cases were working women (i.e., lfp=1). Below is PROC LOGISTIC syntax which can be used to analyze data in Table 2c:

```
PROC LOGISTIC;
  MODEL event/trial=k5 k618 age hc wc;
```

The use of different data formats as input does not affect parameter estimates. However, if input data are formatted as covariate patterns, the calculation of regression diagnostic statistics will be different than if the other two data formats are employed. Hosmer and Lemeshow (1989) suggest that an aggregated data set based on the predictor(s) in the model be created in order to calculate the diagnostic statistics correctly. A SAS program to transform data stored in the raw data format to the covariate pattern format is shown in Table 3.

Table 3. A SAS program to transform data stored in raw data format to covariate pattern format

```
DATA mwlfip;
  INFILE 'a:\mwlfip.dat';
  INPUT lfp k5 k618 age hc wc inc lwg;

PROC FREQ;
  TABLE lfp*k5*k618*age*hc*wc / NOPRINT OUT=out1;

DATA mwlfip1 mwlfip2;
  SET out1;
  IF lfp=0 THEN OUTPUT mwlfip1;
  IF lfp=1 THEN OUTPUT mwlfip2;

DATA a1;
  SET mwlfip1;
  nonevent=count;
  DROP lfp count percent;

PROC SORT;
  BY k5 k618 age hc wc;

DATA a2;
  SET mwlfip2;
  event=count;
  DROP lfp count percent;

PROC SORT;
  BY k5 k618 age hc wc;

DATA final;
  MERGE a1 a2;
  BY k5 k618 age hc wc;
  IF nonevent=. THEN nonevent=0;
  IF event=. THEN event=0;
  trial=nonevent+event;

RUN;
```

CODINGS OF THE DEPENDENT VARIABLE

PROC LOGISTIC accepts either a numeric or character variable as the dependent variable. In binary logistic regression modeling, it is a common practice to use the value 0 for the "non-event" outcome and 1 for the "event" outcome. Unlike most of the logistic regression programs in other statistical packages which, by default, consider the higher value (i.e., 1 in this case) as the "event" outcome, PROC LOGISTIC models the probability of the first ordered level (i.e., 0 in this case). Consequently, logistic regression

coefficients obtained from PROC LOGISTIC have opposite signs as those modeled otherwise. There are two solutions to get around this default setting.

The first solution is to use the DESCENDING option, available in SAS/STAT 6.07 or later versions, on the PROC LOGISTIC statement. The example below shows how this option is used:

```
PROC LOGISTIC DESCENDING;
MODEL lfp=k5 k618 age hc wc;
```

The DESCENDING option reverses the ordering of the two outcome categories. Consequently, the category coded 1 becomes the first ordered category, also the "event" category.

The second solution is to create a SAS format and use it to label the values of the dependent variables. The following example demonstrates this solution:

```
PROC FORMAT;
VALUE depend 1='event' 0='nonevent';

PROC LOGISTIC;
MODEL lfp=k5 k618 age hc wc;
FORMAT lfp depend.;
```

The example above specifies a SAS format, named *depend*, which labels the two outcome categories with "event" for the group coded as 1, and "nonevent" for the group coded as 0. The category labeled "event", which is now alphabetically before the category labeled "nonevent"; hence, it correctly becomes the first ordered category.

USE OF CATEGORICAL PREDICTORS

Similar to the REG procedure, PROC LOGISTIC cannot directly employ categorical predictors in analysis. A categorical predictor must be transformed to a set of design variables. Although PROC LOGISTIC does not provide any mechanism for generating design variables, this transformation can be achieved through a data modification step. Three commonly used coding schemes of generating the design variables—dummy coding, effect coding, and orthogonal polynomials coding—are discussed in this section. As the number of categorical predictors employed grows and the number of categories of each predictor increases, the data manipulation process can be cumbersome. To help with this process, three SAS macro programs were written (see APPENDIX). These macros generate design variables based on three coding schemes. For a detailed discussion of these coding schemes, you are referred to Hosmer and Lemeshow (1989, pp.47-58).

Dummy Coding

Dummy coding, also known as reference cell coding, is the most frequently used scheme in generating design variables for regression analysis. A set of $(k - 1)$ dummy variables is used to render the information of a categorical predictor with k levels. Each dummy variable is coded 1 for one level of a predictor and 0 for the other levels. The referent level has 0's in all dummy variables. This coding scheme is demonstrated in Table 4 for variable *newage* which has six categories. The macro program DUMMY is used to achieve the dummy coding.

Table 4. Demonstration on Dummy Coding

```
%INCLUDE 'a:\category.mac';

DATA example;
INPUT newage @@;
%DUMMY(newage, newage, 6, 6);
DATALINES;
1 2 3 4 5 6
;

PROC PRINT;
VAR newage newage1-newage5;
RUN;
```

[Output]

The SAS System							1
OBS	NEWAGE	NEWAGE1	NEWAGE2	NEWAGE3	NEWAGE4	NEWAGE5	
1	1	1	0	0	0	0	
2	2	0	1	0	0	0	
3	3	0	0	1	0	0	
4	4	0	0	0	1	0	
5	5	0	0	0	0	1	
6	6	0	0	0	0	0	

Effect coding

Effect coding, also known as deviation from means coding, is more frequently used in analysis of variance and linear regression than in logistic regression. This coding scheme is similar to dummy coding except that the referent level is coded -1 in all design variables. Table 5 demonstrates such a coding scheme for variable *newage* which has six categories. The macro program EFFECT is invoked to complete the effect coding process.

Table 5. Demonstration of Effect Coding

```
%INCLUDE 'a:\category.mac';

DATA example;
INPUT newage @@;
%EFFECT(newage, newage, 6, 6);
DATALINES;
1 2 3 4 5 6
;

PROC PRINT;
VAR newage newage1-newage5;
RUN;
```

[Output]

The SAS System							1
OBS	NEWAGE	NEWAGE1	NEWAGE2	NEWAGE3	NEWAGE4	NEWAGE5	
1	1	1	0	0	0	0	
2	2	0	1	0	0	0	
3	3	0	0	1	0	0	
4	4	0	0	0	1	0	
5	5	0	0	0	0	1	
6	6	-1	-1	-1	-1	-1	

Orthogonal polynomials coding

Orthogonal polynomials coding is designed to assess the trend in the outcome variable over increasing levels of an ordinal categorical predictor. The number of trends allowed equals the number of categories minus one. The example in Table 6 shows that for an ordinal categorical variable such as *newage*, which has six levels, five trends (i.e., linear, quadratic, cubic, quartic, and quintic) can be assessed. The macro program POLY, invoked in Table 6, is capable of generating design variables corresponding to eight trends in a categorical variable with nine levels.

Table 6. Demonstration of Orthogonal Polynomials Coding

```
%INCLUDE 'a:\category.mac';

DATA example;
  INPUT newage @@;
  %POLY(newage, newage, 6);
DATALINES;
1 2 3 4 5 6
;

PROC PRINT;
  VAR newage newage1-newage5;
RUN;
```

[Output]

The SAS System							1
OBS	NEWAGE	NEWAGE1	NEWAGE2	NEWAGE3	NEWAGE4	NEWAGE5	
1	1	-0.5976	0.5455	-0.3727	0.1890	-0.0630	
2	2	-0.3586	-0.1091	0.5217	-0.5669	0.3150	
3	3	-0.1195	-0.4364	0.2981	0.3780	-0.6299	
4	4	0.1195	-0.4364	-0.2981	0.3780	0.6299	
5	5	0.3586	-0.1091	-0.5217	-0.5669	-0.3150	
6	6	0.5976	0.5455	0.3727	0.1890	0.0630	

INCORPORATING INTERACTION TERMS

Incorporating interaction terms into PROC LOGISTIC requires two steps. The first step is to generate interactions as cross-products of predictors in the DATA step. The second step is to include these interaction variables in the MODEL statement. An example is shown in Table 7.

Table 7. Example on generating and incorporating interaction terms in logistic regression modeling

```
%INCLUDE 'a:\category.mac';
DATA mwlfp;
  INFILE 'a:\mwlfp.dat';
  INPUT lfp k5 k618 age hc wc inc lwg;
  IF age <= 34 THEN newage=1;
  ELSE IF 34 < age <= 39 THEN newage=2;
  ELSE IF 39 < age <= 44 THEN newage=3;
  ELSE IF 44 < age <= 49 THEN newage=4;
  ELSE IF 49 < age <= 54 THEN newage=5;
  ELSE newage=6;

  %DUMMY(newage, newage, 6, 6);

  wc_age1=wc*newage1;
  wc_age2=wc*newage2;
  wc_age3=wc*newage3;
  wc_age4=wc*newage4;
  wc_age5=wc*newage5;

PROC LOGISTIC DESCENDING;
  MODEL lfp=k5 k618 newage1-newage5 hc wc
    wc_age1-wc_age5;
```

Even though interaction terms are created in the DATA step, the process can be tedious, especially when interaction terms involve categorical variables. When in this situation, you may turn to the GENMOD procedure which allows you to specify the interaction terms directly in the MODEL statement. PROC GENMOD can also be used as a preliminary tool to identify potentially significant interaction effects. The SAS program shown below generates the same parameter estimates as those presented in Table 7, yet it is written for the GENMOD procedure.

```
PROC GENMOD;
  CLASS newage;
  MODEL lfp=k5 k618 newage wc hc wc*newage /
    DIST=BIN LINK=LOGIT;
```

CALCULATING SOMERS' D STATISTICS

PROC LOGISTIC calculates three measures of association: Tau-a, Gamma, and Somers' D statistics. These statistics measure the association between the observed outcomes and the estimated probabilities. Tau-a statistic is Kendall's rank-order correlation coefficient without the adjustment for ties. Gamma statistic is calculated from a contingency table composed of (1) observed outcomes and (2) estimated probabilities categorized into intervals of length 0.002. In logistic regression, the Gamma statistic is more useful and appropriate than Tau-a statistic. Unfortunately, the Gamma statistic has a tendency to overstate the strength of relationship between predicted probabilities and the observed outcomes (Demaris, 1992).

The Somers' D statistic is considered an even better measure of association than the Gamma statistic. Somers' D statistic is an extension of the Gamma statistic for which one variable is designated as the dependent and the other as the predictor (Siegel and Castellan, 1988). The Somers' D (i.e., Dxy) statistic reported by PROC LOGISTIC is calculated using the categorized estimated probability (x) as the independent variable and the observed outcome (y) as the dependent variable. Demaris (1992) suggests that the Somers' Dxy statistic that utilizes the observed outcomes as independent variable and the categorized estimated probabilities as dependent variable is the ideal index for association in binary logistic regression.

The alternative Somers' D statistics can be obtained by following these steps (the implementation is shown in Table 8):

1. Fits the logistic regression model and generates the estimated probabilities.
2. Categorize the estimated probabilities into intervals of length 0.002 by dividing the estimated probabilities with 0.002 and keeping only the integers of these values.
3. Use the FREQ procedure to calculate the Somers' D statistics by employing the variable in step 2 as the column variable and the observed outcome variable as the row variable.

Table 8. Calculating Alternative Somers' D Statistics

```
PROC LOGISTIC DESCENDING NOPRINT;
  MODEL lfp=k5 k618 newage1-newage5 hc wc
    wc_age1-wc_age5;
  OUTPUT OUT=out1 PRED=predict;

DATA somers;
  SET out1;
  probcat=INT(predict/.002);

PROC FREQ;
  TABLE lfp*probcat / MEASURES NOPRINT;
  TITLE 'SAS FREQ';

RUN;
```

[Output]

SAS FREQ

1

STATISTICS FOR TABLE OF LFP BY PROBCAT		
Statistic	Value	ASE
Gamma	0.405	0.039
Kendall's Tau-b	0.281	0.027
Stuart's Tau-c	0.389	0.038
Somers' D C R	0.397	0.038
Somers' D R C	0.199	0.019
Pearson Correlation	0.342	0.032
Spearman Correlation	0.341	0.033
Lambda Asymmetric C R	0.001	0.014
Lambda Asymmetric R C	0.317	0.041
Lambda Symmetric	0.101	0.018
Uncertainty Coefficient C R	0.039	0.003
Uncertainty Coefficient R C	0.242	0.021
Uncertainty Coefficient Symmetric	0.067	0.006

Sample Size = 752

The Somers' D C|R (i.e., 0.397) reported in the output is identical to the one reported by PROC LOGISTIC which is the Somers' Dxy statistic. The Dyx statistic is reported on the output as Somers' D R|C (i.e., 0.199).

The example in Table 8 is for data stored in raw data format. If the data are stored in frequency or covariate pattern format, additional SAS statements are needed in order to calculate the Somers' D statistics. Table 9 demonstrates another SAS program for computing Somers' D based on data structured in frequency or covariate pattern format.

Table 9. Example for calculating Somers' D statistics for data set structured in frequency data format and covariate pattern format

```

/* Data set in frequency format */
PROC LOGISTIC DESCENDING;
  MODEL lfp=k5 k618 newage1-newage5 wc hc
        wc_age1-wc_age5;
  OUTPUT OUT=out1 PRED=predict;
  FREQ count;

DATA somers;
  SET out1;
  probcat=INT(predict/0.002);

PROC FREQ;
  TABLE lfp*probcat / MEASURES NOPRINT;
  WEIGHT count;

/* Data set in covariate pattern format */
PROC LOGISTIC DESCENDING;
  MODEL event/trial=k5 k618 newage1-newage5 wc hc
        wc_age1-wc_age5;
  OUTPUT OUT=out1 PRED=predict;

DATA somers;
  SET out1;
  probcat=INT(predict/0.002);
  lfp=0; count=(trial-event); OUTPUT;
  lfp=1; count=event; OUTPUT;

PROC FREQ;
  TABLE lfp*probcat / MEASURES NOPRINT;
  WEIGHT count;

```

SELECTING THE BEST HIERARCHICAL MODEL

The term "hierarchical" in logistic regression modeling has two meanings. The first meaning refers to a series of models which are nested within each other. When a model (A) contains a subset of predictors of another model (B), model A is said to be nested within model B. Hence, Models A and B are hierarchical. Table 10 demonstrates the use of SELECTION=SCORE option for generating possible nested models with number of predictors ranging from 1 to 16.

Table 10. Using the SELECTION=SCORE method to generate the best n-predictor model

```

%INCLUDE 'a:\category.mac';
DATA mwlfp;
  INFILE 'a:\mwlfp.dat';
  INPUT lfp k5 k618 age hc wc inc lwg;
  IF age <= 34 THEN newage=1;
  ELSE IF 34 < age <= 39 THEN newage=2;
  ELSE IF 39 < age <= 44 THEN newage=3;
  ELSE IF 44 < age <= 49 THEN newage=4;
  ELSE IF 49 < age <= 54 THEN newage=5;
  ELSE newage=6;

%DUMMY(newage, newage, 6, 6);

wc_age1=wc*newage1;
wc_age2=wc*newage2;
wc_age3=wc*newage3;
wc_age4=wc*newage4;
wc_age5=wc*newage5;

PROC LOGISTIC DESCENDING;
  MODEL lfp=k5 k618 newage1-newage5 hc wc
        inc lwg wc_age1-wc_age5
        / SELECTION=SCORE;
  TITLE 'SAS Logistic - Best';

```

[Output]

SAS Logistic - Best 1
The LOGISTIC Procedure

Data Set: WORK.TEMP
Response Variable: LFP In paid labor force.
Response Levels: 2
Number of Observations: 752
Link Function: Logit

Response Profile

Ordered Value	LFP	Count
1	1.00	427
2	0.00	325

SAS Logistic - Best 2
The LOGISTIC Procedure

Regression Models Selected by Score Criterion

In	Score Value	Variables Included in Model
1	34.233	K5

[Part of output is omitted]

2	57.785	K5 LWG
2	54.315	K5 WC

[Part of output is omitted]

3	73.609	K5 WC INC
3	70.807	K5 INC LWG

[Part of output is omitted]

```

-----
4  88.240 K5 WC INC LWG
[Part of output is omitted]
-----
5  96.341 K5 NAGE1 WC INC LWG
[Part of output is omitted]
-----
6  104.803 K5 NAGE1 NAGE2 WC INC LWG
[Part of output is omitted]
-----
7  108.619 K5 NAGE1 NAGE2 NAGE4 WC INC LWG
[Part of output is omitted]
-----
8  112.455 K5 NAGE1 NAGE2 NAGE3 NAGE4 WC INC LWG
[Part of output is omitted]
-----
9  113.716 K5 NAGE1 NAGE2 NAGE3 NAGE4 NAGE5 WC INC LWG
[Part of output is omitted]
-----
10 114.440 K5 NAGE1 NAGE2 NAGE3 NAGE4 NAGE5 WC WC_NAGE4 INC
    LWG
[Part of output is omitted]
-----
11 115.032 K5 NAGE1 NAGE2 NAGE3 NAGE4 NAGE5 WC WC_NAGE2
    WC_NAGE3 INC LWG
[Part of output is omitted]
-----
12 115.840 K5 NAGE1 NAGE2 NAGE3 NAGE4 NAGE5 WC WC_NAGE1
    WC_NAGE2 WC_NAGE3 INC LWG
[Part of output is omitted]
-----
13 116.414 K5 K618 NAGE1 NAGE2 NAGE3 NAGE4 NAGE5 WC
    WC_NAGE1 WC_NAGE2 WC_NAGE3 INC LWG
[Part of output is omitted]
-----
14 116.555 K5 K618 NAGE1 NAGE2 NAGE3 NAGE4 NAGE5 WC HC
    WC_NAGE1 WC_NAGE2 WC_NAGE3 INC LWG
[Part of output is omitted]
-----
15 116.680 K5 K618 NAGE1 NAGE2 NAGE3 NAGE4 NAGE5 WC HC
    WC_NAGE1 WC_NAGE2 WC_NAGE3 WC_NAGE5 INC LWG
[Part of output is omitted]
-----
16 116.780 K5 K618 NAGE1 NAGE2 NAGE3 NAGE4 NAGE5 WC HC
    WC_NAGE1 WC_NAGE2 WC_NAGE3 WC_NAGE4 WC_NAGE5 INC
    LWG
-----

```

Based on the information presented in Table 10, a series of nested models can be examined. This series of nested models can be generated with multiple runs of PROC LOGISTIC. Luckily, with the help of FORWARD selection method and SEQUENTIAL plus DETAILS options, nested models can be generated and examined in only one run of the LOGISTIC procedure. The predictors in the MODEL statement below are specified in the order that mimics the increasing sequence in Table 10.

```

PROC LOGISTIC DESCENDING;
  MODEL lfp=k5 wc inc lwg newage1-newage5
    wc_age1-wc_age5 k618 hc /
  SELECTION=FORWARD SLENTRY=1
  SEQUENTIAL DETAILS;

```

The second meaning of the term "hierarchical" implies a type of modeling restriction that when interactions among predictors are included in a model, all main effects and lower order interactions of those predictors are also included in the model (Tabachnick and Fidell, 1996). Unfortunately, there is no shortcut in the SAS System to implement the second meaning of hierarchical logistic models. The implementation of this modeling rule requires you to adjust the model manually with multiple runs of PROC LOGISTIC.

GENERATING GRAPHICAL OUTPUT

Graphic output results are useful visual tools for assessing the adequacy of the fitted model. These graphs can be classified into two categories based on their functions. One category is concerned with regression diagnostics. The other is related to the accuracy of prediction.

Diagnostics-related graphic output

Hosmer and Lemeshow (1989) suggest that three plots—delta Pearson chi-square versus estimated probabilities, delta deviance versus estimated probabilities, and delta parameter estimate versus estimated probabilities—be considered in any logistic regression diagnostic.

To generate these three diagnostic plots, the first step is to form an aggregated data set based upon the predictors in the model in order to calculate the diagnostic statistics correctly. The next step is to apply the logistic regression model to the aggregated data set, and use the OUTPUT statement to save the diagnostic statistics in a separate SAS data set. A slightly modified version of Table 3 program demonstrates how these two steps are accomplished (see Table 11).

Table 11. Generating the diagnostic statistics based on covariate patterns

```

DATA temp;
  INFILE 'a:\mwlfip.dat';
  INPUT lfp k5 k618 age hc wc inc lwg;
  IF age <= 34 THEN newage=1;
  ELSE IF 34 < age <= 39 THEN newage=2;
  ELSE IF 39 < age <= 44 THEN newage=3;
  ELSE IF 44 < age <= 49 THEN newage=4;
  ELSE IF 49 < age <= 54 THEN newage=5;
  ELSE newage=6;

PROC FREQ;
  TABLE lfp*k5*k618*newage*hc*wc / NOPRINT OUT=out1;

DATA mwlfip1 mwlfip2;
  SET out1;
  IF lfp=0 THEN OUTPUT mwlfip1;
  IF lfp=1 THEN OUTPUT mwlfip2;

DATA a1;
  SET mwlfip1;
  nonevent=count;
  DROP lfp count percent;

PROC SORT; BY k5 k618 newage hc wc;

DATA a2;
  SET mwlfip2;
  event=count;
  DROP lfp count percent;

PROC SORT; BY k5 k618 newage hc wc;

DATA final;
  MERGE a1 a2;
  BY k5 k618 newage hc wc;
  IF nonevent=. THEN nonevent=0;
  IF event=. THEN event=0;
  trial=nonevent+event;

  %DUMMY(newage, newage, 6, 6);

  wc_age1=wc*newage1;
  wc_age2=wc*newage2;
  wc_age3=wc*newage3;
  wc_age4=wc*newage4;
  wc_age5=wc*newage5;

```

```

PROC LOGISTIC DESCENDING;
  MODEL event/trial=k5 k618 newage1-newage5 hc wc
    wc_age1-wc_age5;
  OUTPUT OUT=out1 DFBETAS=dbeta0 dbk5 dbk618 dbage1
    dbage2 dbage3 dbage4 dbage5
    dbwc dbhc dbwcage1 dbwcage2
    dbwcage3 dbwcage4 dbwcage5
    DIFCHISQ=dx2 DIFDEV=ddev
    PRED=predict;

```

Once the diagnostic statistics are generated, they can be plotted by the GPLOT procedure of SAS/GRAPH® software. The three diagnostic plots mentioned earlier and the PROC GPLOT statements are shown in Table 12. A discussion on how to interpret these diagnostic plots is found in Hosmer and Lemeshow (1989, pp.149-170).

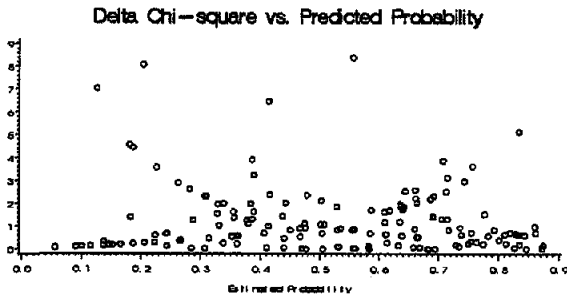
Table 12. PROC GPLOT statements to generate the three diagnostic plots

```

PROC GPLOT DATA=out1;
  SYMBOL1 VALUE=circle INTERPOL=none;
  PLOT dx2*predict=1;
  TITLE 'Delta Chi-square vs. Predicted Probability';

```

[output]

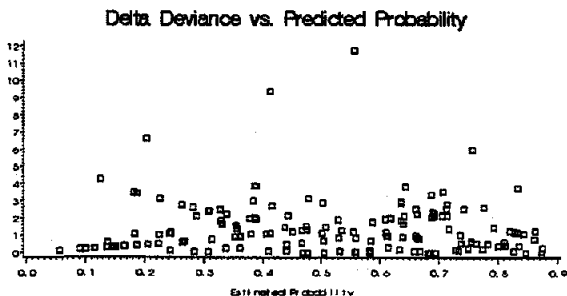


```

PROC GPLOT DATA=out1;
  SYMBOL1 VALUE=square INTERPOL=none;
  PLOT ddev*predict=1;
  TITLE 'Delta Deviance vs. Predicted Probability';

```

[output]

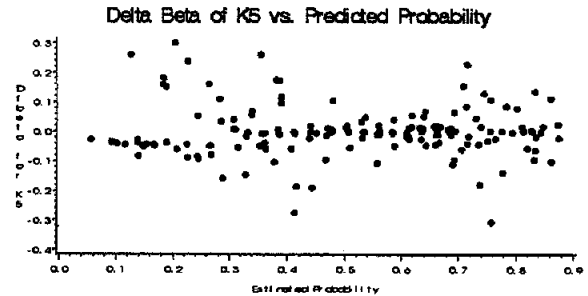


```

PROC GPLOT DATA=out1;
  SYMBOL1 VALUE=dot INTERPOL=none;
  PLOT dbk5*predict=1;
  TITLE 'Delta Beta of K5 vs. Predicted Probability';

```

[output]



Prediction-related graphic output

One of the purposes of logistic regression is to generate an equation which can reliably classify the cases into one of the two outcome categories. Affifi and Clark (1990) suggest two graphs that can help researchers to assess the accuracy of prediction. The first graph is called a Receiver Operating Characteristic (ROC) curve. The ROC curve is based on the proportion of observations correctly classified as the "event" (i.e., sensitivity) versus the proportion of those misclassified as the "event" (i.e., one minus specificity) under various probability cutoff points. An example of generating ROC curve by the SAS System is shown in Table 13.

Table 13. Receiver Operating Characteristic (ROC) curve

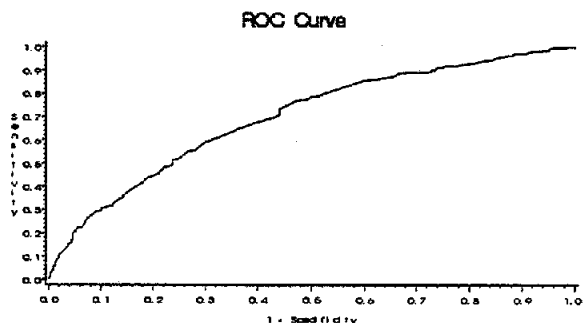
```

PROC LOGISTIC DESCENDING;
  MODEL lfp=k5 k618 newage1-newage5 wc hc
    wc_age1-wc_age5 / OUTROC=roc1;

PROC GPLOT DATA=roc1;
  SYMBOL1 INTERPOL=join VALUE=none;
  PLOT _sensit*_lmspec_=1 / VAXIS=0 TO 1 BY .1;
  TITLE 'ROC Curve';

```

[output]



The second graph is an overlay plot of sensitivity (and specificity) against various probability cutoff points. It is a valuable tool for determining an appropriate probability cutoff point for classification. An example of such a plot is shown in Table 14.

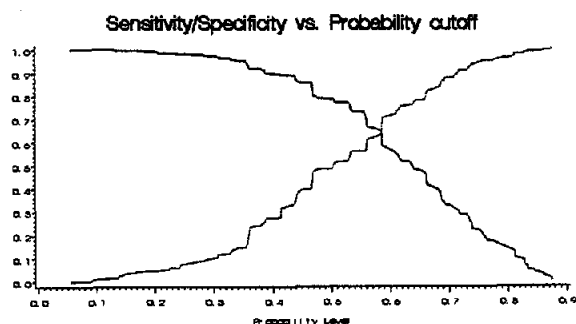
Table 14. Percentage of correct classification as a function of cutoff points

```
PROC LOGISTIC DESCENDING;
  MODEL lfp=k5 k618 newage1-newage5 wc hc
    wc_age1-wc_age5 / OUTROC=roc1;

DATA roc2;
  SET roc1;
  spec=1-_lmspec_;

PROC GPLOT DATA=roc2;
  SYMBOL1 I=JOIN;
  PLOT _pos*_prob_ spec*_prob_ / OVERLAY;
  TITLE 'Sensitivity/Specificity vs. Probability
  cutoff';
```

[Output]



In Table 14, the curve for sensitivity runs from top left corner to the bottom right corner. The other curve is for specificity. The point at which these two curves intersect provides a good choice for a probability cutoff point for future classification.

CONCLUSION

In this paper, six areas of difficulty a researcher may encounter when using PROC LOGISTIC are identified. Solutions written in tactful SAS statements and illustrated by a real-world data set are also provided. It is hoped that naïve and experienced users alike will find this paper to be an useful supplement to the *SAS/STAT User's Guide, Version 6, Fourth Edition*, and *SAS/STAT Software: Changes and Enhancements Through Release 6.12*.

REFERENCES

- Afifi, A.A. and Clark, V. (1990), *Computer-Aided Multivariate Analysis*, 2nd edition, New York, NY: Van Nostrand Reinhold.
- Chuang, H.L. (1997), "High School Youth's Dropout and Re-Enrollment Behavior," *Economics of Education Review*, 16(2), 171-186.
- Demaris, A. (1992), *Logit Modeling: Practical Applications*, Newbury Park, CA: SAGE Publications, Inc.

Hosmer, D.W., Jr. and Lemeshow, S. (1989), *Applied Logistic Regression*, New York, NY: John Wiley & Sons, Inc.

Janik, J. and Kravitz, H.M. (1994). "Linking work and domestic problem with police suicide," *Suicide and Life Threatening Behavior*, 24(3), 267-274.

Long, J.S. (1997), *Regression Models for categorical and limited dependent variables*, Thousand Oaks, CA: SAGE Publications, Inc.

McCullagh, P. and Nelder, J.A. (1989), *Generalized Linear Models*, 2nd edition, London: Chapman & Hall.

Mroz, T. A. (1987). "The sensitivity of an empirical model of married women's hours of work to economic and statistical assumptions," *Econometrica*, 55, 765-799.

Ott, M.D. (1988), "An Analysis of Predictors of Early Academic Dismissal," *Research in Higher Education*, 28(1), 34-48.

SAS Institute Inc. (1997), *SAS/STAT Software: Changes and Enhancements Through Release 6.12*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1990), *SAS/STAT User's Guide, Version 6*, 4th Edition, Cary, NC: SAS Institute Inc.

Siegel, S. and Castellan, N.J., Jr. (1988), *Nonparametric Statistics for the Behavioral Sciences*, 2nd edition, New York, NY: McGraw-Hill, Inc.

Tabachnick, B.G. and Fidell, L.S. (1996), *Using Multivariate Statistics*, 3rd edition, New York, NY: HarperCollins College Publishers.

Tolman, R.M. and Weisz, A. (1995), "Coordinated Community Intervention for Domestic Violence: The Effects of Arrest and Prosecution on Recidivism of Woman Abuse Perpetrators," *Crime and Delinquency*, 41(4), 481-495.

Yarandi, H.N. and Simpson, S.H. (1991). "The Logistic Regression Model and the odds of testing HIV positive," *Nursing Research*, 40(6), 372-373.

APPENDIX

```
*****/
/* FILENAME: CATEGORY.MAC */
/* */
/* This file contains three SAS macro programs */
/* to generate three different codings for */
/* categorical variable. */
/* */
/* CVAR = Name of the categorical variable */
/* PREFIX = Name prefix for the set of design */
/* variables */
/* NCAT = Number of categories */
/* RCAT = The referent categories */
*****/

/* MACRO DUMMY */
%MACRO DUMMY(CVAR, PREFIX, NCAT, RCAT);
  %DO I=1 %TO &NCAT;
    &PREFIX&I={&CVAR=&I};
  %END;
  DROP &PREFIX&RCAT;
%MEND DUMMY;

/* MACRO EFFECT */
%MACRO EFFECT(CVAR, PREFIX, NCAT, RCAT);
  %DO I=1 %TO &NCAT;
```

```

&PREFIX&I=(&CVAR=&I);
IF &CVAR=&RCAT THEN &PREFIX&I=-1;
%END;
DROP &PREFIX&RCAT;
%MEND EFFECT;

/* MACRO POLY */
%MACRO POLY(CVAR, PREFIX, NCAT);
  %IF &NCAT=3 %THEN %DO;
    SELECT (&CVAR);
    WHEN (1) DO;
      &PREFIX&1=-.7071; &PREFIX&2=.4082; END;
    WHEN (2) DO;
      &PREFIX&1=0; &PREFIX&2=-.8165; END;
    WHEN (3) DO;
      &PREFIX&1=.7071; &PREFIX&2=.4082; END;
    OTHERWISE;
  END;
%END;
%ELSE %IF &NCAT=4 %THEN %DO;
  SELECT (&CVAR);
  WHEN (1) DO;
    &PREFIX&1=-.6708; &PREFIX&2=.5;
    &PREFIX&3=-.2236; END;
  WHEN (2) DO;
    &PREFIX&1=-.2236; &PREFIX&2=-.5;
    &PREFIX&3=.6708; END;
  WHEN (3) DO;
    &PREFIX&1=.2236; &PREFIX&2=-.5;
    &PREFIX&3=-.6708; END;
  WHEN (4) DO;
    &PREFIX&1=.6708; &PREFIX&2=.5;
    &PREFIX&3=.2236; END;
  OTHERWISE;
  END;
%END;
%ELSE %IF &NCAT=5 %THEN %DO;
  SELECT (&CVAR);
  WHEN (1) DO;
    &PREFIX&1=-.6325; &PREFIX&2=.5345;
    &PREFIX&3=-.3162; &PREFIX&4=.1195; END;
  WHEN (2) DO;
    &PREFIX&1=-.3162; &PREFIX&2=-.2673;
    &PREFIX&3=.6325; &PREFIX&4=-.4781; END;
  WHEN (3) DO;
    &PREFIX&1=0; &PREFIX&2=-.5345;
    &PREFIX&3=0; &PREFIX&4=.7171; END;
  WHEN (4) DO;
    &PREFIX&1=.3162; &PREFIX&2=-.2673;
    &PREFIX&3=-.6325; &PREFIX&4=-.4781; END;
  WHEN (5) DO;
    &PREFIX&1=.6325; &PREFIX&2=.5345;
    &PREFIX&3=.3162; &PREFIX&4=.1195; END;
  OTHERWISE;
  END;
%END;
%ELSE %IF &NCAT=6 %THEN %DO;
  SELECT (&CVAR);
  WHEN (1) DO;
    &PREFIX&1=-.5976; &PREFIX&2=.5455;
    &PREFIX&3=-.3727; &PREFIX&4=.189;
    &PREFIX&5=-.063; END;
  WHEN (2) DO;
    &PREFIX&1=-.3586; &PREFIX&2=-.1091;
    &PREFIX&3=.5217; &PREFIX&4=-.5669;
    &PREFIX&5=.315; END;
  WHEN (3) DO;
    &PREFIX&1=-.1195; &PREFIX&2=-.4364;
    &PREFIX&3=.2981; &PREFIX&4=.378;
    &PREFIX&5=-.6299; END;
  WHEN (4) DO;
    &PREFIX&1=.1195; &PREFIX&2=-.4364;
    &PREFIX&3=-.2981; &PREFIX&4=.378;
    &PREFIX&5=.6299; END;
  WHEN (5) DO;
    &PREFIX&1=.3586; &PREFIX&2=-.1091;
    &PREFIX&3=-.5217; &PREFIX&4=-.5669;
    &PREFIX&5=-.315; END;
  WHEN (6) DO;
    &PREFIX&1=-.5976; &PREFIX&2=.5455;
    &PREFIX&3=.3727; &PREFIX&4=.189;
    &PREFIX&5=.063; END;
  OTHERWISE;
  END;
%END;
%ELSE %IF &NCAT=7 %THEN %DO;
  SELECT (&CVAR);
  WHEN (1) DO;
    &PREFIX&1=-.5669; &PREFIX&2=.5455;
    &PREFIX&3=-.4082; &PREFIX&4=.2417;
    &PREFIX&5=-.1091; &PREFIX&6=.033; END;
  WHEN (2) DO;
    &PREFIX&1=-.378; &PREFIX&2=0;
    &PREFIX&3=.4082; &PREFIX&4=-.5641;
    &PREFIX&5=.4364; &PREFIX&6=-.1974; END;
  WHEN (3) DO;
    &PREFIX&1=-.189; &PREFIX&2=-.3273;
    &PREFIX&3=.4082; &PREFIX&4=.0806;
    &PREFIX&5=-.5455; &PREFIX&6=.4935; END;
  WHEN (4) DO;
    &PREFIX&1=0; &PREFIX&2=-.4364;
    &PREFIX&3=0; &PREFIX&4=.4835;
    &PREFIX&5=0; &PREFIX&6=-.658; END;
  WHEN (5) DO;
    &PREFIX&1=.189; &PREFIX&2=-.3273;
    &PREFIX&3=-.4082; &PREFIX&4=.0806;
    &PREFIX&5=.5455; &PREFIX&6=.4935; END;
  WHEN (6) DO;
    &PREFIX&1=.378; &PREFIX&2=0;
    &PREFIX&3=-.4082; &PREFIX&4=-.5641;
    &PREFIX&5=-.4364; &PREFIX&6=-.1974; END;
  WHEN (7) DO;
    &PREFIX&1=.5669; &PREFIX&2=.5455;
    &PREFIX&3=.4082; &PREFIX&4=.2417;
    &PREFIX&5=.1091; &PREFIX&6=.033; END;
  OTHERWISE;
  END;
%END;
%ELSE %IF &NCAT=8 %THEN %DO;
  SELECT (&CVAR);
  WHEN (1) DO;
    &PREFIX&1=-.5401; &PREFIX&2=.5401;
    &PREFIX&3=-.4308; &PREFIX&4=.282;
    &PREFIX&5=-.1498; &PREFIX&6=.0615;
    &PREFIX&7=-.0171; END;
  WHEN (2) DO;
    &PREFIX&1=-.3858; &PREFIX&2=.0772;
    &PREFIX&3=.3077; &PREFIX&4=-.5238;
    &PREFIX&5=.4922; &PREFIX&6=-.3077;
    &PREFIX&7=.1195; END;
  WHEN (3) DO;
    &PREFIX&1=-.2315; &PREFIX&2=-.2315;
    &PREFIX&3=.4308; &PREFIX&4=-.1209;
    &PREFIX&5=-.3638; &PREFIX&6=.5539;
    &PREFIX&7=-.3585; END;
  WHEN (4) DO;
    &PREFIX&1=-.0772; &PREFIX&2=-.3858;
    &PREFIX&3=.1846; &PREFIX&4=.3626;
    &PREFIX&5=-.321; &PREFIX&6=-.3077;
    &PREFIX&7=.5974; END;
  WHEN (5) DO;
    &PREFIX&1=.0772; &PREFIX&2=-.3858;
    &PREFIX&3=-.1846; &PREFIX&4=.3626;
    &PREFIX&5=.321; &PREFIX&6=-.3077;
    &PREFIX&7=-.5974; END;
  WHEN (6) DO;
    &PREFIX&1=.2315; &PREFIX&2=-.2315;
    &PREFIX&3=-.4308; &PREFIX&4=-.1209;
    &PREFIX&5=.3638; &PREFIX&6=.5539;
    &PREFIX&7=.3585; END;
  WHEN (7) DO;
    &PREFIX&1=.3858; &PREFIX&2=.0772;
    &PREFIX&3=-.3077; &PREFIX&4=-.5238;
    &PREFIX&5=-.4922; &PREFIX&6=-.3077;
    &PREFIX&7=-.1195; END;
  WHEN (8) DO;
    &PREFIX&1=-.5401; &PREFIX&2=.5401;
    &PREFIX&3=.4308; &PREFIX&4=.282;
    &PREFIX&5=-.1498; &PREFIX&6=.0615;
    &PREFIX&7=.0171; END;
  OTHERWISE;
  END;
%END;

```

```

END;
%END;
%ELSE %IF &NCAT=9 %THEN %DO;
  SELECT (&CVAR);
  WHEN (1) DO;
    &PREFIX&1=-.5164; &PREFIX&2=.5318;
    &PREFIX&3=-.4449; &PREFIX&4=.3129;
    &PREFIX&5=-.1849; &PREFIX&6=.0899;
    &PREFIX&7=-.0341; &PREFIX&8=.0088; END;
  WHEN (2) DO;
    &PREFIX&1=-.3873; &PREFIX&2=.133;
    &PREFIX&3=.2225; &PREFIX&4=-.4693;
    &PREFIX&5=.5085; &PREFIX&6=-.382;
    &PREFIX&7=.2048; &PREFIX&8=-.0705; END;
  WHEN (3) DO;
    &PREFIX&1=-.2582; &PREFIX&2=-.1519;
    &PREFIX&3=.4132; &PREFIX&4=-.2458;
    &PREFIX&5=-.1849; &PREFIX&6=.4944;
    &PREFIX&7=-.478; &PREFIX&8=.2468; END;
  WHEN (4) DO;
    &PREFIX&1=-.1291; &PREFIX&2=-.3229;
    &PREFIX&3=.286; &PREFIX&4=.2011;
    &PREFIX&5=-.416; &PREFIX&6=.0225;
    &PREFIX&7=.478; &PREFIX&8=-.4936; END;
  WHEN (5) DO;
    &PREFIX&1=0; &PREFIX&2=-.3799;
    &PREFIX&3=0; &PREFIX&4=.4023;
    &PREFIX&5=0; &PREFIX&6=-.4495;
    &PREFIX&7=0; &PREFIX&8=.617; END;
  WHEN (6) DO;
    &PREFIX&1=.1291; &PREFIX&2=-.3229;
    &PREFIX&3=-.286; &PREFIX&4=.2011;
    &PREFIX&5=.416; &PREFIX&6=.0225;
    &PREFIX&7=-.478; &PREFIX&8=-.4936; END;
  WHEN (7) DO;
    &PREFIX&1=.2582; &PREFIX&2=-.1519;
    &PREFIX&3=-.4132; &PREFIX&4=-.2458;
    &PREFIX&5=.1849; &PREFIX&6=.4944;
    &PREFIX&7=.478; &PREFIX&8=.2468; END;
  WHEN (8) DO;
    &PREFIX&1=.3873; &PREFIX&2=.133;
    &PREFIX&3=-.2225; &PREFIX&4=-.4693;
    &PREFIX&5=-.5085; &PREFIX&6=-.382;
    &PREFIX&7=-.2048; &PREFIX&8=-.0705; END;
  WHEN (9) DO;
    &PREFIX&1=.5164; &PREFIX&2=.5318;
    &PREFIX&3=.4449; &PREFIX&4=.313;
    &PREFIX&5=.1849; &PREFIX&6=.0899;
    &PREFIX&7=.0341; &PREFIX&8=.0088; END;
  OTHERWISE;
END;
%END;
%MEND POLY;

```

ACKNOWLEDGEMENTS

We are grateful to Dr. John Samuel and Mr. Choong-Geun Chung of Center of Statistical and Mathematical Computing and Professor Scott Long of Department of Sociology, all at Indiana University, for their valuable comments on earlier drafts of this paper.

SAS, SAS/STAT, and SAS/GRAPH are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.

FOOTNOTES

¹The terms "outcome variable" and "dependent variable" are interchangeable when referring to logistic regression modeling.

²The terms "predictor," "covariate," and "independent variable" are interchangeable when referring to logistic regression modeling.

AUTHORS

Chao-Ying Joanne Peng
Room 4050, School of Education
Indiana University
Bloomington, IN 47405-1006
E-mail: peng@indiana.edu
Phone: (812) 856-8337, FAX: (812) 856-8333

Tak-shing Harry So
Room 4008, School of Education
Indiana University
Bloomington, IN 47405-1006
E-mail: tso@indiana.edu
Phone: (812) 855-4724, FAX: (812) 855-8802