Paper 969-2017

SAS® Macros for Binning Predictors with a Binary Target

Bruce Lund, Magnify Analytic Solutions, Detroit MI, Wilmington DE, Charlotte NC

ABSTRACT

Binary logistic regression models are widely used in CRM (customer relationship management) or credit risk modeling. In these models it is common to use nominal, ordinal, or discrete (NOD) predictors. NOD predictors typically are binned (reducing the number of their levels) before usage in a logistic model. The primary purpose of binning is to obtain parsimony without greatly reducing the strength of association of the predictor X to the binary target Y. In this paper two SAS® macros are discussed. %NOD_BIN bins predictors with nominal values (as well as ordinal and discrete) by collapsing levels so as to maximize information value (IV). %ORDINAL_BIN is applied to predictors which are ordered and where collapsing can occur only for levels that are adjacent in the ordering of X. %ORDINAL_BIN finds all possible binning solutions by complete enumeration. These solutions are ranked by IV and monotonic solutions are identified. The purpose of this paper is to present these two macros.

INTRODUCTION

The goal of binning a predictor X with respect to a binary target Y is to simplify X by reducing the number of levels (distinct values) while maintaining the power of X to predict Y.

Let X have L levels (distinct values) and let k satisfy $2 \le k \le L$. A **k-bin solution** is an *eligible assignment* (see below) of the L levels into k bins (each bin has at least one level). If the k-binned X has almost as much power in predicting Y (as measured by information value or log likelihood) as the unbinned X, then the binning was successful in simplifying the model.[1] A modeler is guided by various statistics in selecting a k-bin solution, but the final decision ultimately relies on the expert judgment of the modeler.

An eligible assignment of levels to bins is defined in terms of the ordering of the levels of X. There are two alternatives:

(i) Solutions with **ordered bins**: the levels within a bin are adjacent (no gaps) in the ordering of X
(ii) Solutions with **unrestricted bins**: the levels within a bin are unrestricted relative to the ordering of X

For example: If X has 4 ordered levels A, B, C, D, then the 2-bin solutions with ordered bins are {A} {B C D}, {A B} {C, D}, and {A B C} {D}. In contrast, a 2-bin solution which is not ordered is {A C} {B D}.

Two SAS macros are presented for binning of a nominal, ordinal, or discrete[2] (NOD) predictor X for a binary target Y.[3]

1. %NOD_BIN: User specifies whether solutions are ordered (J) or unrestricted (A). If (A), then any pair of levels can be collapsed. If (J), then only a pair composed of adjacent levels in the ordering of X can be collapsed. Starting with L levels of X, eligible pairs are selected for collapsing so as to maximize IV at each step. Alternatively, log likelihood (LL) may be maximized. The collapsing step is repeated until there are 2 bins. If (J) is selected, then each k-bin solution for $k \le L$ has ordered bins.

2. %ORDINAL_BIN: This macro is applied to ordered X and finds ALL k-bin solutions with ordered bins where $2 \le k \le L$. The algorithm to produce the binning solutions is simply a complete enumeration.

   To illustrate, consider the simple case of a predictor with three levels: 1, 2, 3. There are 2 ordered bin solutions with 2 bins: {1 2} {3} and {1} {2 3} and one solution with 3 bins: {1} {2} {3}. %ORDINAL_BIN finds them all, finds which are monotonic with respect to odds (equivalently, event rate), and computes IV and -2*Log L for each.

[1] Two approaches of how to use a k-binned X in a model are: (1) as a group variable (via a CLASS variable) or (2) as a weight of evidence coded variable. The pros and cons of (1) and (2) are not discussed in this paper.
[2] A discrete predictor is a numeric predictor with only a “few” values. Often these values are counts. The designation of “few” is subjective. It is used here to distinguish discrete from continuous (interval) predictors with “many levels”.
[3] Another method of binning is based on decision trees. A decision tree is used by the Interactive Grouping Node in the Credit Scoring Application in SAS Enterprise Miner. Decision tree binning is not discussed in this paper.

BINNING AND NUMBER OF LEVELS OF X

For X with L levels the number of ordered bin solutions is $2^{(L-1)} - 1$. For %ORDINAL_BIN, the run-time doubles with each added level. %ORDINAL_BIN was run on a data set with 20 levels for X, and the run-time was acceptable.[4] %NOD_BIN is tolerant of much larger numbers of X levels. %NOD_BIN was run on a data set with 30 levels for X, and the run-time was very fast.

If the number of levels L of X is too large, then preliminary binning is needed to reduce L to a manageable number. For numeric X, preliminary binning can be performed by PROC HPBIN or PROC RANK. If X is character, SAS code to reduce the number of levels of X is discussed in **Appendix 1**.

What about complete enumeration of solutions for "unrestricted bins"? First, the programming challenge to produce the unrestricted bins is daunting, at least to me. Second, the number of computations for complete enumeration become much larger than for ordered bins as L increases.[5] These computations might be possible for L ≤ 10.

Therefore, %NOD_BIN (or some other algorithmic approach) is a practical approach when X is unordered. A down-side is that binning by %NOD_BIN can become sub-optimal with respect to IV or LL as the collapsing process goes beyond the first collapse. There is no easy way to detect if and when the binning process has become sub-optimal.[6]

MACRO %NOD_BIN

Features of %NOD_BIN

The target Y is binary with levels 0 and 1 with no missing values.

For %NOD_BIN the predictor X may be character or numeric. If numeric, X must have integer values between 0 and 99. Missing values for X are supported and are processed according to the MISS parameter.

%NOD_BIN bins X according to the user's choice of METHOD and MODE. These parameters are explained below.

Required Parameters for %NOD_BIN

There are seven required parameters to run %NOD_BIN. The defaults can be accepted for the other eight parameters. But in practice, the user will want to specify some of these parameters.

DATASET is a dataset name - either one or two components

X (Predictor) is a numeric or character variable which can have MISSING values. If **X** is numeric, then **X** must have integer values between 0 and 99. Otherwise, the macro STOPs.

Y (Target) has values 0 and 1 without MISSING values.

W (Freq) has values which are non-negative integers. It represents a FREQUENCY variable. If there is no frequency variable in **DATASET**, then enter 1.

METHOD is IV or LL. For **METHOD** = IV the collapsing maximizes IV at each step. For **METHOD** = LL the collapsing maximizes log likelihood at each step.

MODE is A or J. For **MODE** = A all pairs of levels are compared when collapsing. For **MODE** = J only pairs of adjacent levels in the ordering of **X** are compared when collapsing. For **MODE** = J the Missing Level is not eligible to be collapsed with any other Level of **X**.

---

[4] %Ordinal_Bin finds the number of solutions implied by L and range of k (this range is controlled by user parameters) and STOP's if more than 600,000. If 20 levels and no restrictions, there are 524,287 ordered bin solutions.
[5] Counts for unrestricted bins are the Bell Numbers minus one. See http://mathworld.wolfram.com/BellNumber.html. There are 115,974 unrestricted bin solutions for L = 10. For L = 11 and 12 there are 678,569 and 4,213,596 respectively. For L = 20 there are 51,724,158,235,371.
[6] For an example of sub-optimal binning see Lund and Brotherton (2013) p. 6.

If the user wants to impose the restriction of collapsing only pairs of adjacent levels via **MODE** = J, a more effective binning process is provided by %ORDINAL_BIN, to be discussed later.

ORDER = D | A. If D, then the higher value of Y (which is Y=1) is modeled. That is, counts of "1" appear in the numerator of the weight-of-evidence expressions or in an odds calculation. If A, then the reverse is true. In reports, the modeled value of Y is called "good" and is denoted by G. Otherwise, B for "bad".

Often, the user will specify the optional parameter **VERBOSE.** This is selected for Example 1 below.

VERBOSE = YES | <other> is used to display the entire history of collapsing in the SUMMARY REPORT. Otherwise this history is not displayed in the SUMMARY REPORT.

Example 1: Using Required Parameters for %NOD_BIN and the additional parameter VERBOSE

```
data TEST_A;
input x $ y w @@;
datalines;
AAAAAAAAA 0 500   C 1 310
AAAAAAAAA 1 330   D 0 400
BA 0 300          D 1 210
BA 1 270          E 0 550
C 0 400           E 1 400
;
%NOD_BIN(
DATASET =  TEST_A,
X =  X,
Y =  Y,
W =  W,
METHOD = IV,    /* IV | LL */
MODE = A,       /* A | J */
ORDER = A,      /* D | A */
MISS =  ,       /* MISS | <all other is noMISS> */
MIN_PCT =   ,   /* space | integer 0 to 99 */ /* space = 0 */
MIN_NUM =   ,   /* space | integer >= 0 */ /* space = 0 */
VERBOSE = YES, /* YES | <all other is NO> */
VERBOSE2 =  ,   /* YES | <all other is NO> */
LL_STAT =   ,   /* YES | <all other is NO */
WOE =  ,        /* WOE | <all other is NO */
WOEADJ = ,      /* space | 0 | 0.5 */  /* space = 0 */
RUN_TITLE = Example1  /* Title for run, NO commas */
);
```

Two reports are generated: SUMMARY REPORT and LOG-ODDS RATIO REPORT.

TABLE 1 Summary Report for Example 1

Example 1
NOD_BIN Version v13c, RUN ON 03DEC16 19:20
Dataset= TEST_A, Predictor= X, Target= Y, Freq= W, Method= IV, Mode= A, Miss= , Order= A
Min_Pct = 0, Min_Num = 0, WOEADJ = N/A

SUMMARY REPORT

k	Collapse via Min Pct | Num	IV	-2*Log L	X_STAT model c	L1	L2	L3	L4	L5
5		0.02636	4955.65	0.54406	AAAAAAAA	BA	C	D	E
4	NO	0.02591	4956.06	0.54246	AAAAAAAA	BA	C+E	D	
3	NO	0.02359	4958.13	0.53611	AAAAAAAA+C+E	BA	D		
2	NO	0.01699	4964.04	0.52394	AAAAAAAA+C+E+BA	D			

Discussion of SUMMARY REPORT

The first column gives the number of bins at that step. The next column “Collapse via Min Pct | Num” indicates if the collapse of bins at this step was forced by either parameter **MIN_PCT** or **MIN_NUM**. The action of these parameters is more fully explained in the parameter documentation. Simply stated, these parameters require that small-size bins be binned with other bins as the first priority.

The next three columns, IV, -2*Log L, and X-STAT[7] give statistics for this bin solution relative to target **Y**.

The **VERBOSE** = YES causes columns L1 – L5 be printed. These columns give the history of binning. For example, for k = 4, C was collapsed with E as indicated by “C+E”.

TABLE 2 Log-Odds Ratio Report for Example 1

LOG-ODDS RATIO WITH 95% CI

Consider stopping at k if +/- 2SD interval after collapse omits zero

k	collapsing_to	LO Ratio after collapse	LO Ratio Std Dev	LOminus2SD	LOplus2SD
5	4	-0.0636	0.1002	-0.2640	0.1369
4	3	0.1243	0.0866	-0.0488	0.2974
3	2	0.2270	0.0932	0.0406	0.4134

Discussion of LOG-ODDS RATIO REPORT

To discuss the first row of the Log-Odds Report, expanded details of the bins at step 5 need to be viewed:

TABLE 3 Details of 5-bin Solution for Example 1

Obs	X	_TYPE_	G	B	G/B	Row_Total
1		0	2150	1520	1.4145	3670
2	BA	1	300	270	1.1111	570
3	C	1	400	310	1.2903	710
4	E	1	550	400	1.3750	950
5	AAAAAAAA	1	500	330	1.5152	830
6	D	1	400	210	1.9048	610

The LO Ratio, when collapsing at k = 5 to k = 4, is the log of the ratio of odds for C and odds for E. The odds for C are 1.2903 and for E they are 1.3750. Log Odds Ratio = log (1.2903 / 1.3750) = -0.0636. An approximate standard deviation is sqrt (1/400 + 1/310 + 1/550 + 1/400) = 0.1002.

The +/- 2 std. dev. around the log odds-ratio gives a test for “stopping”. This test says that a collapse can be made if the interval formed from +/- two standard deviations around the log-odds ratio contains zero.

The logic is heuristic. If the odds from C and E are about the same (in which case the log odds-ratio is near 0), then collapsing C and E would not lose much information. If the interval does not contain zero, then collapsing would cause a loss of too much information.

In this first example for k = 5 going to k = 4, zero is well within the +/- 2 std. dev. interval of -0.2640 to 0.1369 and so a decision to collapse to k = 4 is supported.

ADDING THE OPTIONAL PARAMETER: LL_STAT = YES

LL_STAT = YES | <other> is used to display entropy (base e), Nested_ChiSq, and Prob > ChiSq. The Prob > ChiSq gives a statistic for an additional stopping criterion.

The chi-square (Pr > ChiSq) column tests whether the dummy variable coefficients, corresponding to the two levels that are collapsed, are equal.

[7] X_STAT is the same as the model "c" statistic for PROC LOGISTIC; CLASS <binned X>; MODEL Y = <binned X>;. The name X_STAT is introduced because it can be computed in a DATA Step (does not require PROC LOGISTIC).

In the case of the first collapse between C and E the report in TABLE 4 gives Pr > ChiSq as 0.5260. This statistic is also found by testing the equality of the dummy variable coefficients for C and E. This test is conducted by the SAS code shown below and the result is given in TABLE 5.

TABLE 4 Output Added by LL-STAT in Example 1

Consider stopping at k+1 if (Pr > ChiSq) < 0.05, or other alpha, at k

k	Collapse via Min Pct | Num	IV	-2*Log L	X_STAT model c	Entropy (base e)	Nested ChiSq	Pr > ChiSq	L1-L5 Same
5		0.02636	4955.65	0.54406	0.6752	N/M	N/M	
4	NO	0.02591	4956.06	0.54246	0.6752	0.4021	0.5260	Note: C + E
3	NO	0.02359	4958.13	0.53611	0.6755	2.0692	0.1503	
2	NO	0.01699	4964.04	0.52394	0.6763	5.9103	0.0151	

```
data dummies; set TEST_A;
   d_AAAAAAAAA = (X = "AAAAAAAAA");
   d_BA = (X = "BA");
   d_C = (X = "C");
   d_D = (X = "D");
   d_E = (X = "E");
run;
proc logistic data = dummies;
model Y = /*d_AAAAAAAAA*/ d_BA d_C d_D d_E;
test d_C = d_E;
freq W;
run;
```

TABLE 5 Unequal Coefficients Test for Stopping in Example 1

Linear Hypotheses Testing Results			
Label	Wald Chi-Square	DF	Pr > ChiSq
Test 1	0.4022	1	0.5260

The collapsing of C and E for k = 4 is supported.

MODE = J

In Example 1 the choice of MODE was MODE = A. The alternative is MODE = J. For this choice there are only 4 eligible pairs of levels for the first collapse. (AAAAAAAA with BA, BA with C, C with D, D with E). The first collapse when MODE = J is BA with C.

There are the following changes to reporting when running MODE = J

- LL_STAT is disabled and the Log Odds-Ratio Report is not displayed. Since binning is based on maximizing IV (or LL) subject to adjacency, these stopping tests are compromised.
- The C-Stat between the Bins of **X** and **Y** is produced.
- A new column called MONO is added. When the value is YES, then the odds for the bins are monotonic (increasing or decreasing) vs. the ordering of X for the k bin solution which was found at this step. (Once a monotone solution is found, all solutions from the following steps remain monotonic.)

TABLE 6 Summary Report When MODE = J in Example 1

k	Collapse via Min Pct | Num	IV	-2*Log L	X_STAT model c	C-STAT	MONO	L1	L2	L3	L4	L5
5		0.026	4955.7	0.544	0.507		AAAAAAAA	BA	C	D	E
4	NO	0.024	4957.4	0.542	0.504		AAAAAAAA	BA+C	D	E	
3	NO	0.017	4963.8	0.527	0.513		AAAAAAAA+BA+C	D	E		
2	NO	0.007	4973.0	0.520	0.520	YES	AAAAAAAA+BA+C	D+E			

If predictor X has a meaningful and important ordering, then often the modeler wants to find a solution where the odds are monotonic and, only then, to find the best IV solution. But %NOD_BIN does not look for monotonic solutions, and is likely to bypass good monotonic solutions while seeking to maximize IV (or LL). In these cases, a much better approach is %ORDINAL_BIN, to be discussed later.

SMALL BINS CAN BE FORCED TO COLLAPSE FIRST

In Example 1, if MIN_NUM = 600, then level BA (having only 570 observations), will be collapsed first according to the best IV (or LL) collapse. This parameter is useful since it removes meaninglessly small bins that might survive the IV (or LL) selection process until a final solution. An alternative is MIN_PCT which allows the user to specify the minimum percent of a bin which will force immediate binning.

DOCUMENTATION OF THE PARAMETERS FOR %NOD_BIN

The complete parameter list for %NOD_BIN is documented in **Appendix 2**.

MACRO %ORDINAL_BIN

Features of %ORDINAL_BIN

The target Y is binary with levels 0 and 1 with no missing values.

For %ORDINAL_BIN it is assumed that the predictor X is ordered. The predictor X may be character or numeric. If numeric, X must have integer values between 0 and 99. If X is character, there is no restriction but the user needs to be satisfied with the implied ordering of the values of X. For example, the character values: 1.4, 3.0, 11.2 will be ordered as follows: 1.4, 11.2, 3.0 and this is probably not desired.[8]

Missing values for X are supported and are processed according to the MISS parameter.

%ORDINAL BIN finds ALL k-bin solutions with ordered bins for k in the range 2 to L where L is the number of levels of X. If X has four ordered levels A, B, C, D, then there is a total of seven solutions where X remains ordered:

There is a single ordered 4-bin solution is {A} {B} {C} {D}

There are three 3-bin solutions: {A B} {C} {D}, {A} {B C} {D}, {A} {B} {C D}

There are three 2-bin solutions: {A B C} {D}, {A B} {C D}, {A} {B C D}

If X has L levels there are $2^{(L-1)}$ - 1 total solutions. For L = 20, there are $2^{(L-1)}$ - 1 = 524,287 total solutions.

For fixed k the solutions are ranked in %ORDINAL_BIN output reports by IV or -2*Log L according to the parameter METHOD.

Since %ORDINAL_BIN assumes that X is ordered and finds only ordered solutions, the modeler may want to find “monotonic binning solutions”. The definition of a monotonic binning solution is given below:

Let i = 1 to L be an index for the level of X. Define Gi and Bi as shown (“G” is for “good” and “B” for bad):

Gi = count of Y = 1 when X = Xi
Bi = count of Y = 0 when X = Xi

The odds at X = Xi is the ratio Gi / Bi

A binning solution is monotonic if Gi / Bi is either is always increasing or always decreasing across the ordered levels of X. In the second step of binning and beyond, the number of levels of the binned X goes below L (but is at least 2), and the definition of a monotonic binning solution continues to apply to the current binned ordering of X.

For each solution %ORDINAL_BIN determines if the solution is monotonic. In addition, the number of “turns” are computed where a “turn” occurs when the difference in odds changes signs as shown below:

If (odds(i1) – odds(i2)) * (odds(i2) – odds(i3)) < 0, then a turn occurs at i = i2

[8] The user could first apply the Z4.1 format to these values to obtain character values with the desired order.

Required Parameters for %NOD_ORDINAL

There are 6 required parameters to run %NOD_ORDINAL. The defaults can be accepted for the other parameters. But in practice, the user will want to specify some of these parameters.

DATASET is a SAS dataset name - either one or two components.

X (Predictor) is a numeric or character variable which can have MISSING values. If **X** is numeric, then **X** must have integer values between 0 and 99. Otherwise, the macro STOPs.

Y (Target) is a numeric variable that has values 0 and 1 without MISSING values. For the current release %ORDINAL_BIN does not accept a target with values other than 0 and 1.

W (Freq) is a numeric variable that has values which are positive integers. It represents a FREQUENCY variable. If there is no frequency variable in **DATASET**, then enter 1. Do not leave as spaces.

METHOD = IV | LL. Solutions are ranked by descending IV or ascending -2*LL. These solutions are highlighted by "*" in Best_IV_LL column in PROC PRINT. **METHOD** is displayed in the PROC PRINT title.

ORDER = D | A. If D, then the higher value of Y (which is Y=1) is modeled. That is, counts of "1" appear in the numerator of the weight-of-evidence expressions or in an odds calculation. If A, then the reverse is true. In reports, the modeled value of Y is called "good" and is denoted by G. Otherwise, B for "bad".

Example 2: Using Required Parameters for %ORDINAL_BIN

```
data level_4;
do X = 1 to 4;
   Y = 1;
   W = floor(200*ranuni(1)) + 1;
   output;
   Y = 0;
   W = floor(250*ranuni(1)) + 1;
   output;
   end;
run;

%Ordinal_Bin(
DATASET = Level_4,
X = X,
Y = Y,
W = W,
METHOD = IV, /* IV | LL */
ORDER = D, /* D | A */
MISS = ,  /* MISS | <other> */
N_BEST = ,  /* space = 1 or positive integer */
N_MONO = , /* space = 1 or positive integer */
MIN_PCT =  , /* space = 0 | integer from 0 to 99 */
MIN_NUM =  , /* space = 0 | negative integer */
MIN_BIN = ,  /* space or integer >= 2 (see rules) */
MAX_BIN = ,  /* space or integer >= 2 (see rules) */
NOPRINT_WOE = , /* YES | other */
PRINT1_WOE = ,  /* space or integer >= 2 (see rules) */
PRINT2_WOE = ,  /* space or integer >= 2 (see rules) */
RUN_TITLE = Example2,  /* Title for run, NO commas */
DELETE_PRIOR =  /* DELETE_YES or <other> */
)
;
```

TABLE 7 Default Value Settings for ORDINAL_BIN in Example 2

Example2
ORDINAL_BIN Version v13b, RUN ON 07DEC16 15:35
Dataset= Level_4, Predictor= X, Target= Y, Freq= W, Miss= , N_Best= 1, N_Mono= 1
Reset values of BIN min / max and/or PRINT_WOE according to DEFAULT RULES

Obs	message
1	BIN min / max and/or PRINT_WOE were reset
2	MIN_BIN=2, MAX_BIN=4, PRINT1_WOE=2, PRINT2_WOE=4

TOTAL SOLUTIONS TO BE INSPECTED

Obs	SOLUTIONS
1	7

TABLE 8 gives results for k-bins as specified by default values of MIN_BIN = 2 and MAX_BIN = 4. The number of solutions for each k to be printed is determined by default values of N_BEST= 1 and N_MONO = 1. There will be one solution for the best_rank and possibly one additional solution for the best_mono (if there is a monotone solution and, if so, if it not already found by N_BEST = 1.

There is no monotonic solution for k = 4 bins. The best IV solution (note: Method=IV) has 2 turns. There is a monotonic solution for k = 3 which is different from the best IV solution. For k = 2 the best IV solution and the best monotonic solution are the same solution. The default values of N_BEST and N_MONO limited the printed output. The number of solutions for k = 4, 3, 2 are, respectively, 1, 3, 3 for a total of 7, but only 4 solutions are reported below.

The “missing” column in TABLE 8 is “N” if X has no missing values or parameter MISS is not “MISS”. “Solution_num” gives the rank of the solution in terms of METHOD (here it is IV).

TABLE 8 Summary Report for Example 2

Example2
ORDINAL_BIN Version v13b, RUN ON 07DEC16 15:35
Dataset= Level_4, Predictor= X, Target= Y, Miss= , N_Best= 1, N_Mono= 1, Method= IV, Order= D
Min_Pct= 0, Min_Num= 0, Min_Bin= 2, Max_Bin= 4, Print1_WOE= 2, Print2_WOE= 4

Obs	BINS	missing	best_rank	best_mono	solution_num	turns	IV	minus2LL	L1	L2	L3	L4
1	4	N	*		1	2	0.4800	1336.65	1	2	3	4

Obs	BINS	missing	best_rank	best_mono	solution_num	turns	IV	minus2LL	L1	L2	L3
1	3	N	*		1	1	0.4792	1336.86	1	2	3+4
2	3	N		*	3	0	0.1204	1418.73	1+2	3	4

Obs	BINS	missing	best_rank	best_mono	solution_num	turns	IV	minus2LL	L1	L2
1	2	N	*	*	1	0	0.4555	1342.96	1	2+3+4

The SAS coding of the weight-of-evidence (WOE) transformation is given for both of the 3-bin solutions found above, as specified by the default values of PRINT1_WOE = 2 and PRINT2_WOE = 4. There are similar reports (not shown) for k = 4 and k = 2.

TABLE 9 Illustrations of Weight of Evidence Coding for Example 2

Obs	BINS	Solution_num	all_code	X_WOE	G_Count	B_Count
1	3	1	if X in (1) then X_woe = -1.372778828 ;	-1.3728	37	243
2	3	1	if X in (2) then X_woe = 0.7170040679 ;	0.7170	80	65
3	3	1	if X in (3,4) then X_woe = 0.2633553271 ;	0.2634	294	376
4	3	3	if X in (1,2) then X_woe = -0.458561145 ;	-0.4586	117	308
5	3	3	if X in (3) then X_woe = 0.2366590849 ;	0.2367	185	243
6	3	3	if X in (4) then X_woe = 0.3103634571 ;	0.3104	109	133

ZERO CELLS ARE HANDLED AUTOMATICALLY BY %ORDINAL_BIN

In data set ZERO_CELL there is a zero count for Level C (there is no dataline for C=0). %ORDINAL_BIN detects this, prints a message, and simply moves to the next lower number of bins. Here, this is k = 2.

```
data ZERO_CELL;
input x $ y w;
datalines;
C 1 310
D 0 400
D 1 210
E 0 550
E 1 400
;
```

TABLE 9 ORDINAL_BIN Handling of Data Sets with Zero Cells

Test Zero Cell Handling

Obs	message
1	There are no solutions for BINS = 3
2	No Solutions meet the Min_Pct or Min_Cnt requirements, OR
3	Every solution has a bin with zero counts for Y=0 or Y=1 ... ZERO CELL DETECTED

Obs	BINS	missing	best_rank	best_mono	solution_num	turns	IV	minus2LL	L1	L2
1	2	N	*	*	1	0	0.08373	2552.89	C+D	E

USER OPTION: SOLUTIONS WITH BINS WITH SMALL COUNTS ARE NOT REPORTED

The parameters **MIN_PCT** and **MIN_NUM** allow the user to suppress the reporting of binning solutions where a bin falls below the percent of **MIN_PCT** or the bin count of **MIN_NUM**. These small count solutions are, however, output to a data set. The documentation provides details.

SOLUTION NUMBER LIMIT

Based on L (number of levels of X) and parameters **MIN_BIN** and **MAX_BIN** the number of solutions to be inspected is computed. If the number is over 600,000, the macro STOPs. Note that L = 20 with no restrictions on max and min bins requires only 524,287. But L = 21 without max / min bin restrictions requires more than 600,000.

COMMENTS APPLYING TO BOTH %NOD_BIN and %ORDINAL_BIN

- The restriction that numeric X have integer values from 0 to 99: To generalize, numeric X can be formatted to character by PUT(X, Zw.d). Care is needed to assure that resulting character X has the desired sort order, if ordering is to be used in finding binning solutions.
- Length of the levels of X: If X has levels “AAA” and “BBBB” and these levels are collapsed to “AAA+BBBB” then the internal variable ___x_char which stores the collapsed levels of X must have length at least eight. At some point the length of ___x_char could be exceeded as collapsing continues. In the macros ___x_char is set to have length 300. The macros STOP if 300 is exceeded. To determine whether the collapsing process could exceed this limit, the user should add 1 (for “+”) to the length each level of X and then sum the lengths (this limit would never be achieved since the final collapse ends with k = 2).

SAS MACROS DISCUSSED IN THIS PAPER

Contact the author for copies of %NOD_BIN or %ORDINAL_BIN. See the SAS Global Forum site for SAS code for NOMINAL_ODDS_BIN_VIA_RANK.

REFERENCES

Lund B. and Brotherton D. (2013). Information Value Statistic, *MWSUG 2013, Proceedings*, Midwest SAS Users Group, Inc., paper AA-14.

ACKNOWLEDGMENTS

Dave Brotherton and Michael Davidson of Magnify Analytic Solutions of Detroit provided helpful insights and suggestions.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:
blund@magnifyas.com, blund_data@mi.rr.com, or blund.data@gmail.com

DISCLAIMER

All SAS code in this paper is provided by Marketing Associates, LLC "as is" without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Recipients acknowledge and agree that Marketing Associates, LLC shall not be liable for any damages whatsoever arising out of their use of this material. In addition, Marketing Associates, LLC will provide no support for the materials contained herein.

TRADEMARK NOTICE

APPENDIX 1: PRELIMINARY BINNING WHEN X IS CHARACTER WITH MANY LEVELS

NOMINAL_ODDS_BIN_VIA_RANK reads a data set of character predictors and a binary target (assumes values of 1 and 0). For each level of a character variable the "odds" are formed where the odds are the count of 1's divided by the count of 0's. A reasonable rule for preliminary binning is to combine levels of a character predictor having similar odds. This combining process is implemented by PROC RANK where the variable containing the "odds" is ranked.

Finally, SAS statements are created which map the levels of a predictor that are combined (i.e. given the same rank) by PROC RANK to the ranked value. This mapping can be copied into a DATA STEP to implement the binning. The ranks become the new binned levels of the character predictor. Then either %ORDINAL_BIN or %NOD_BIN with MODE = J is applied to these ranks.

See the SAS Global Forum site for SAS code for NOMINAL_ODDS_BIN_VIA_RANK and an example data set.

APPENDIX 2: PARAMETERS FOR %NOD_BIN (v13c)

DATASET is a dataset name - either one or two components

X (Predictor) is a numeric or character variable which can have MISSING values
If **X** is character, then ! is not an allowed value. The value ! is reserved for %NOD_BIN use.
See parameter **MISS** for how missing values of **X** are processed.
"___X_Char" is RESERVED. Do not use ___X_Char as the name of a predictor.
If **X** is numeric, then **X** must have integer values between 0 and 99. Otherwise, the macro STOPs.

Y (Target) has values 0 and 1 without MISSING values. The current version %NOD_BIN does not accept a target with values other than 0 and 1.

W (Freq) has values which are non-negative integers. It represents a FREQUENCY variable. If there is no frequency variable in **DATASET**, then enter 1.

METHOD is IV or LL. For **METHOD** = IV the collapsing maximizes IV at each step. For **METHOD** = LL the collapsing maximizes log likelihood at each step.

MODE is A or J. For **MODE** = A all pairs of levels are compared when collapsing. For **MODE** = J only pairs of adjacent levels in the ordering of **X** are compared when collapsing. For **MODE** = J the Missing Level is not eligible to be collapsed with any other Level of **X**.

If the user wants to impose the restriction of collapsing only pairs of adjacent levels via **MODE** = J, a more comprehensive binning solution is provided by %ORDINAL_BIN.

ORDER = D | A. If D, then the lower value of Y is set to B and the greater value of Y is set to G. The G value is modeled. That is, G appears in the numerator of the weight-of-evidence expression. If A, then the reverse is true.

MISS = MISS | <other> is used if missing values for **X** (Predictor) are treated as a Level. If **MODE** = J, then missing is not allowed to collapse with any other Level. The missing Level will be one of the 2 Levels appearing in the final collapse k = 2. If **X** is character, then the missing Level appears as "!" in the Reports. If **X** is numeric, then the missing Level appears as "." in the Reports. WOE SAS coding: The SAS code statements do not need modification when **MISS** = MISS is specified. In the WOE SAS code statement, if **X** is character, then "space" is used in the SAS code for missing. In the WOE SAS code statement, if **X** is numeric, then "." is used in the SAS code for missing.

MIN_PCT = space or an integer number from 0 to 99. If space, then **MIN_PCT** defaults to 0. **MIN_PCT** collapses a level of **X** where the sample size of the level is below the **MIN_PCT** ("minimum percent") of the total.

Description of Process: As the algorithm finds pairs of levels to collapse to maximize IV (or LL), it also identifies pairs where one (or both) or the levels has a count below **MIN_PCT**% of total sample. If one or more such pairs exist, the algorithm will collapse the pair which maximizes IV (or LL).

Example: Suppose **X** has 5 levels: A, B, C, D, E and A has 2%, B has 4%, C has 2%, D has 40% and E has 52% and suppose **MIN_PCT** = 5 (=5%) and **MODE** = A. There are 10 possible of pairs to collapse in the first iteration but of these there are 9 that involve at least one level under 5%. Among these 9, suppose that collapsing A and B gives the maximize IV (or LL). Now there are four bins {A, B}, {C}, {D}, {E} with {A, B} having 6%. The process described above now repeats. Only {C} falls below 5% so this current iteration will remove all bins under 5% by collapsing {C} with some other bin.

MIN_NUM = space or a number from 0 or higher. If space, then **MIN_NUM** defaults to 0. **MIN_NUM** replaces the percentage given by **MIN_PCT** with a count. **MIN_NUM** has the same effect on collapsing as does **MIN_PCT**. Both **MIN_PCT** and **MIN_NUM** can be specified at the same time and each will affect the collapsing algorithm as described above.

VERBOSE = YES | <other> is used to display the entire history of collapsing in the SUMMARY REPORT. Otherwise this history is not displayed in the SUMMARY REPORT.

VERBOSE2 = YES | <other>. **VERBOSE2** creates a report for each step in the collapsing process that is similar to TABLE 3.

WOE = WOE | <other> is used to print the WOE coded transform of **X** for each iteration of collapsing.

WOEADJ = space, 0 or 0.5. If space, then space is converted to 0. If **WOEADJ** = 0.5 then 0.5 replaces 0 in a "zero cell". A "zero cell" is a level of **X** where the count of **Y**=1 is zero or the count of **Y**=0 is zero. By replacing "0" with "0.5" the macro will not STOP with ZERO CELL DETECTED error. Binning will continue and will include (as an initial binning) all levels of **X**. The user should consider setting the **MIN_NUM** parameter to a small integer value so that the **WOEADJ** cells are collapsed quickly if the total count of that cell is small. If there are no zero cells, then the title in Reports will show **WOEADJ** = N/A regardless of the input parameter value of **WOEADJ**.

RUN_TITLE gives the contents of TITLE1 in SAS reports prepared by %NOD_BIN. Do not use commas. The default is “No Title”.

COMMENT AND RESTRICTIONS

1. SAS code for WOE transformation of **X** is saved in ___X_woe&num_levels_r for each collapsed level (value of &num_levels_r). Only the first 22 characters of X are used when naming these data sets and only first 28 characters of **X** are used when creating WOE-coded variable X_woe.
2. If **X** is numeric, then **X** must have only integer values between 0 and 99
3. The binning process does not stop until there are only 2 bins remaining. There are no stopping rules given by macro parameters.
4. It is required that ALL cell counts in the X-Y Frequency Table are positive. The Program ENDS if there is a zero cell and prints "ZERO CELL DETECTED". But see parameter **WOEADJ** as a means to avoid STOPPING in the case of a zero cell.

APPENDIX 3: PARAMETERS FOR %ORDINAL_BIN (v13c)

DATASET is a SAS dataset name - either one or two components.

X (Predictor) is a numeric or character variable which can have MISSING values. To avoid excessively long run times, **X** should have no more than L = 20 levels. The computational algorithms in this program are of the order of $2^{(L-1)}$. If **X** is numeric, then **X** must have integer values between 0 and 99. Otherwise, the macro STOPs. See the parameter **MISS** for how missing values are used. "___X_Char" is RESERVED. Do not use ___X_CHAR as the name of the predictor.

Y (Target) is a numeric variable that has values 0 and 1 without MISSING values. For the current release %ORDINAL_BIN does not accept a target with values other than 0 and 1.

W (Freq) is a numeric variable that has values which are positive integers. It represents a FREQUENCY variable. If there is no frequency variable in **DATASET**, then enter 1.

METHOD = IV | LL. Solutions are ranked by descending IV or ascending -2*LL. These solutions are highlighted by Best_IV_LL column in PROC PRINT. **METHOD** is displayed in the PROC PRINT title.

ORDER = D | A. If D, then the lower value of Y is set to B and the greater value of Y is set to G. The G value is modeled. That is, G appears in the numerator of the weight-of-evidence expression.

MISS = MISS is used if missing values for **X** (Predictor) are to be treated as a Level. The missing level will not be combined with any non-missing level but the missing level will contribute to the calculation of IV (information value), -2*LL (-2 * log likelihood), and also WOE is computed for the missing level.

In the WOE SAS code statements that are produced by %ORDINAL_BIN, if **X** is character, then "space" is used in the SAS code for missing. In the WOE SAS code statements, if **X** is numeric, then "." is used in the SAS code for missing.

N_BEST is a positive number or space. **N_BEST** gives the number of best IV or -2*LL (See **METHOD**) solutions to be saved for each "bin-number" (solutions with the same number of bins). For example, if **X** has 10 levels and **N_BEST** = 2, then the 2 best solutions are saved for each "bin-numbers" 9 through 2. There is only one solution for bin-number 10. If "space" then **N_BEST** defaults to 1. See **METHOD** for designation of the use of IV or -2*LL.

N_MONO is a positive number or space. **N_MONO** gives the number of best IV (or -2*LL) solutions to be saved for each bin-number where the solution is monotonic. Some, all, or none of these solutions may have already been found via the specification of **N_BEST**. Solutions due to **N_BEST** and **N_MONO** are not duplicated in the Reports. If "space" then **N_MONO** defaults to 1.

MIN_PCT gives the minimum percent (0 to 99) of the total sample that each bin must satisfy in order for the solution not to be discarded. Discarded solutions are written to datasets with names &short_X._small_&num_bins where <X> is the predictor name and <k> gives the number of bins in the solution. If "space", then **MIN_PCT** defaults to 0.

MIN_NUM is similar to **MIN_PCT** except that **MIN_NUM** gives the minimum count requirement for a bin count. Discarded solutions are written to datasets with names &short_X._small_&num_bins where <X> is the predictor name and <k> gives the number of bins in the solution. If "space", then **MIN_NUM** defaults to 0.

MAX_BIN is an integer >= 2 or space. **MAX_BIN** gives the maximum number of bins to be used in a solution. See RULES below.

MIN_BIN is an integer >= 2 or space. **MIN_BIN** gives the minimum number of bins to be used in a solution. See RULES below.

RULES:

If any of these conditions is true, this is an error and the run STOPs:
- (**MAX_BIN** eq space & **MIN_BIN** ne space)
- (**MAX_BIN** ne space & **MIN_BIN** eq space)
- (**MIN_BIN** > **MAX_BIN**)
- (**MIN_BIN** < 2 and **MIN_BIN** ne space)
- (**MAX_BIN** < 2 and **MAX_BIN** ne space)

Otherwise:

If (**MAX_BIN** = space & **MIN_BIN** = space), then **MAX_BIN** is set to count of levels of **X** and **MIN_BIN**= 2.

Otherwise:

If **MAX_BIN** > number of levels of **X,** then **MAX_BIN** is reset to number of levels of **X**.
If **MIN_BIN** > number of levels of **X**, then **MIN_BIN** is reset to (the possibly reset) **MAX_BIN**.

NOPRINT_WOE = YES | <other>. If YES, then the printing of WOE SAS code is suppressed. NOPRINT_WOE over-rides **PRINT1_WOE** and **PRINT2_WOE**.

PRINT1_WOE is an integer >= 2 or space. SAS code for the WOE transformation of **X** is printed for solutions starting with **PRINT1_WOE** bins (subject to **N_BEST** and **N_MONO**).

PRINT2_WOE is an integer >= 2 or space. SAS code for the WOE transformation of **X** is printed for solutions ending with **PRINT2_WOE** bins (subject to **N_BEST** and **N_MONO**).

RULES for **PRINT1_WOE** and **PRINT2_WOE** are similar to **MIN_BIN** and **MAX_BIN**.

If **PRINT2_WOE** > **MAX_BIN**, then **PRINT2_WOE** is reset so that **PRINT2_WOE** = **MAX_BIN**

RUN_TITLE gives the contents of TITLE1 in SAS reports prepared by %ORDINAL_BIN. Do not use commas. The default is “No Title”.

DELETE_PRIOR: If given value = DELETE_YES, then all datasets in WORK of the form: ___<X>_WOE_<K> and ___<X>_BEST_<K> and ___<X>_SMALL_<K> and ___DENORM_<K>_SORT where <X> refers to the predictor variable and <K> refers to the number of bins in a solution are deleted before the current run of %ORDINAL_BIN.

COMMENT AND RESTRICTIONS

1. SAS code for WOE transformation of **X** is saved in ___X_woe&num_levels_r for each collapsed level (value of &num_levels_r). Only the first 22 characters of X are used when naming these data sets and only first 28 characters of **X** are used when creating WOE-coded variable X_woe.
2. In all references to the number of bins or levels, the missing level of X, if **MISS** = MISS, is not considered as a level. But the IV, LL, and WOE codes are computed for missing when **MISS** = MISS. If **MISS** = MISS is specified but there are no missing, this parameter is ignored. If **MISS** = No (or anything except MISS), then missing values, if any, are bypassed on all calculations.
3. Maximum number of solutions allowed is 600,000. The macro STOPs if there will be more than 600,000. The number of solutions equals $\sum_{k=A}^{B-1}\binom{L-1}{k-1}$ where X has L levels and MIN_BIN = A, MAX_BIN = B. Note: if L = 20, MAX_BIN = 20, and MIN_BIN = 2, then there are 524,287 solutions.