Paper AA-14-2013

Information Value Statistic

Bruce Lund, Magnify Analytics Solutions, a Division of Marketing Associates, Detroit, MI
David Brotherton, Magnify Analytics Solutions, a Division of Marketing Associates, Detroit, MI

ABSTRACT

The Information Value (IV) statistic is a popular screener for selecting predictor variables for binary logistic regression. Familiar, but perhaps mysterious, guidelines for deciding if the IV of a predictor X is high enough to use in modeling are given in many textbooks on credit scoring. For example, these texts say that IV > 0.3 shows X to be a strong predictor. These guidelines must be considered in the context of binning. A common practice in preparing a predictor X is to bin the levels of X to remove outliers and reveal a trend. But IV decreases as the levels of X are collapsed. This paper has two goals: (1) Provide a method for collapsing the levels of X which maximizes IV at each iteration and (2) show how the guidelines (e.g. IV > 0.3) relate to other measures of predictive power. All data processing was performed using Base SAS®.

INTRODUCTION

Information Value Statistic Defined: The information value (IV) of a predictor X and the binary target Y can be given as a formula involving an X-Y frequency table as shown in Table 1.

Notation: "G" and "B" are taken from credit scoring where "G" is "good" (paid as agreed) and "B" is "bad" (default). G_k refers to the count of "goods" corresponding to $X = X_k$. In contrast, g_k refers to the percent of all goods corresponding to $X = X_k$. Likewise for "bads", B_k and b_k .

Table 1 – Information Value Example

X	Y = 0 "B"	Y = 1 "G"	b: Col % Y = 0	g: Col % Y = 1	Log(g/b) (base e)	g - b	(g - b) * Log(g/b)
1	2	1	0.400	0.333	-0.1823	-0.067	0.0122
2	1	1	0.200	0.333	0.5108	0.133	0.0681
3	2	1	0.400	0.333	-0.1823	-0.067	0.0122
SUM	5	3				**IV =**	**0.0924**

As a formula IV is written as:

$$IV = \sum_{k=1}^{K} (g_k - b_k) * \log(g_k / b_k)$$

where the count of levels of X is $K \geq 2$ and g_k and b_k are positive for all k = 1,…,K [1]

The IV statistic is appropriate for a predictor X with a modest number of levels, typically under 20, with no zero cells. Predictors with "continuous" value ranges (e.g. dollars, distances) must first undergo preliminary binning.[2]

Naively, we can say that $\log(g_k / b_k)$ measures the deviation between the distributions of g and b while $(g_k - b_k)$ measures the importance of the deviation. For example, considering the two equal odds of 0.02 / 0.01 and 0.2 / 0.1, the odds of 0.02 / 0.01 is less important in IV since it is weighted by (0.02 - 0.01) in the computation.

Popular credit scoring text books give guidelines for evaluation of the strength of a predictor X for a binary target Y in terms of its IV statistic. See Finlay (2010)[3], Mays and Lynas (2010)[4], and Siddiqi (2006).

---

[1] Since IV is defined by column percents of goods and bads, expected value of IV would be unchanged by stratified sampling of goods and bads (e.g. 100% of bads and 10% of goods). This is also true for c-stat and x-stat which are discussed later in the paper.
[2] See Finlay (2010) chapter 5
[3] page 139
[4] page 95

The following is taken from Siddiqi, page 81.

IV Rules of Thumb for evaluating the strength a predictor

- Less than 0.02: unpredictive
- 0.02 to 0.1: weak
- 0.1 to 0.3: medium
- 0.3 +: strong

These guidelines are familiar but perhaps mysterious. Although they are firmly grounded in good practice, can these guidelines be related to other metrics? This question is discussed in the second major section of this paper.

A Brief Discussion of the c-statistic

The c-statistic is a commonly used statistic to evaluate the strength of a numeric (or ordered) predictor X for potential usage in a logistic regression model with binary target Y.

A formula for the c-statistic is given below:

$$\textbf{c-stat} = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} (g_i * b_j) + (0.5) * (\sum_{i=1}^{K} g_i * b_i)$$

The c-statistic's range is 0 to 1. It is customary to require c-statistic $\geq$ 0.5 by taking max(c-stat, 1 – c-stat).

The "c" that occurs in the output of PROC LOGISTIC; MODEL Y = <predictors> is the c-statistic of P, the probability from the MODEL, and the target Y.[5]

Weight-of-Evidence of a Predictor: The log-odds factor "$\log(g_k / b_k)$" in IV is the familiar quantity from the weight-of-evidence (WOE) recoding of X.

This recoding is given by:

$$\text{IF } X = X_k \text{ THEN } X\_woe = \log(g_k / b_k) \text{ for } k = 1 \text{ to } K.$$

The x-statistic of X and Y

We will use "**x-statistic of X and Y"** to refer to "c" from logistic regression PROC LOGISTIC; MODEL Y = X_woe.[6]

Here are two important equivalent characteristics of the **x-statistic of X and Y**.

a) **x-statistic** equals the "c" from: PROC LOGISTIC; CLASS X; MODEL Y = X;
b) **x-statistic** = $0.5 * (1 + \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} \text{Abs} (g_i*b_j - g_j*b_i))$ where Abs = absolute value.

Of particular significance is that (b) gives a way to compute the x-statistic within a data step. This is used in the SAS code in the macro %BEST_COLLAPSE discussed later and appearing in the Appendix.

When X is numeric, the c-stat is defined and the x-stat is always equal to or greater than the c-stat. When x-stat equals c-stat, then X is monotonic versus Y. That is, $G_k / (G_k + B_k)$ is non-decreasing (or non-increasing) with respect to the ordering of X.

What is a good x-statistic value? The logistic model "c" (often called AUC for "area under ROC curve") is a common measure of the discriminatory power of a logistic regression model. Hosmer and Lemeshow (2000 p. 162) state that a logistic model with "c" of at least 0.7 provides acceptable discrimination. As noted, the x-statistic is the "c" for the single variable model: MODEL Y = X_woe.

Some of the individual WOE predictors which have entered into a model may have an x-statistic vs. Y which is much less than 0.7.[7] Applications and data, of course, vary across industries. Our experience in automotive direct

[5] From the PROC LOGISTIC output section: Association of Predicted Probabilities and Observed Responses.
[6] The term x-statistic is preferable because the x-statistic can be computed without reference to PROC LOGISTIC as shown by (b). See Raimi and Lund (2012) for more discussion. For Table 1, x-stat = 0.5667.

marketing models is that a predictor with an x-statistic of 0.55 is at the low end of being useful and that a predictor with an x-statistic of 0.60 is likely to be included in the model.

RELATED WORK:

Alec Zhixiao Lin (2013) contributed a paper to the SAS Global Forum called "Variable Reduction in SAS by Using Weight of Evidence and Information Value". This paper includes a SAS macro which comparatively ranks predictors for a binary response model in terms of their predictive power as measured by Information Value.

SECTION ONE:
COLLAPSING LEVELS OF X WHILE MAXIMIZING IV

A common practice in preparing a predictor X for use in a logistic model is to bin the levels of X to remove outliers and reveal a trend. But IV decreases when two levels of X are collapsed with equality occurring only when the odds-ratios from the two levels are equal.[8] In some cases the modeler employs business knowledge when forming the bins. Alternatively, the modeler may wish to rely on an algorithm to perform the collapsing into bins. In this section a Best Collapse algorithm is described for collapsing the levels of X which maximizes IV at each iteration.

Recalling the formula for IV:

$$IV = \sum_{k=1}^{K} (g_k - b_k) * \log(g_k / b_k).$$

The algorithm finds the two levels (call these levels i and j) when combined together decreases IV the least. This is equivalent to finding i and j so that D is minimized where:

$$D = (g_i - b_i) * \log(g_i / b_i) + (g_j - b_j) * \log(g_j / b_j) - (g_i + g_j - b_i - b_j) * \log((g_i + g_j)/ (b_i + b_j))$$

The expression: $(g_i + g_j - b_i - b_j) * \log((g_i + g_j) / (b_i + b_j))$ is the contribution to IV from the combined levels i and j.

The algorithm, as coded in SAS, at each iteration checks each pair (i, j) to find the minimum D. This pair is then collapsed.

Alternatively, if the predictive variable X is ordered and the modeler want to maintain the ordering of X during the collapsing, the algorithm has an option to collapse only adjacent levels of X.

The algorithm is coded in a macro which we call **%BEST_COLLAPSE**.

%BEST_COLLAPSE also provides the option to the modeler of using maximum log likelihood of X as a predictor of Y (as in logistic regression) as the criterion for selecting the levels of X to collapse. This maximum log likelihood algorithm also has Modes A and J.[9] The major focus of this paper will be on the Information Value statistic.

MACRO %BEST_COLLAPSE

This section discusses %BEST_COLLAPSE macro and gives several examples. SAS code is given in the Appendix.

Macro Call:

```
%MACRO BEST_COLLAPSE(DATASET,X,Y,W,METHOD,MODE,VERBOSE,LL_STAT);
```

---

[7] The statement applies also to predictors X which are entered as a CLASS variable and to numeric predictors X which are monotonic versus Y (that is, monotonic versus $P(Y=1 \mid X = X_k)$ since, for these, x-statistic = c-statistic..

[8] See the Appendix for a mathematical proof.

[9] The maximum log likelihood criterion for collapsing was discussed and included in the macro by Lund and Raimi (2012) called %COLLAPSE_LEVELS. The more complex %COLLAPSE_LEVELS includes full data input checking and also collapsing for multinomial targets but it does not include the option to collapse by maximizing IV at each iteration. The %COLLAPSE_LEVELS is built around PROC FREQ and ODS outputs. The algorithms in %BEST_COLLAPSE are implemented in a data step.

Parameter Definitions:

DATASET: A dataset name - either one or two levels
X: Character or numeric variable which can have MISSING values. Missing values are ignored in all calculations.
Y: Binary Target which is numeric and must have values 0 and 1 without MISSING values.
W: Numeric frequency variable which has values which are positive integer values.
METHOD: IV or LL
For METHOD = IV the criterion for selecting two eligible levels to collapse is to maximize the IV. The levels that are eligible for collapse is determined by the MODE parameter.
For METHOD = LL the criterion for selecting two eligible levels to collapse is to maximize the Log Likelihood. The levels that are eligible for collapse is determined by the MODE parameter.
MODE: A or J
For MODE = A all pairs of levels are compared when collapsing
For MODE = J only adjacent pairs of levels are compared when collapsing
VERBOSE: If YES, then the entire history of collapsing is displayed in the SUMMARY REPORT. Otherwise this history is not displayed in the SUMMARY REPORT.
LL_STAT: If YES, then Log Likelihood for Model and Likelihood Ratio Chi Square Probability are displayed.

LL_STAT is optional since the log likelihood and chi-square probability are not especially useful in practical situations. Specifically, due to large samples, the chi-square probability is often essentially equal to one.

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".

Predictor variables X with values having more than 2 characters or having a large number of levels may cause the lines from the PROC PRINT reports to wrap around.

If the modeler wants to model missing values, then the missing values must be pre-coded to a non-missing value in a preliminary data step.

Example 1 - Data:

```
data IV_test_data;
length x $1;
input x $ w y @@;
datalines;
1 2 0   1 1 1   2 1 0   2 1 1   3 2 0   3 3 1
;
run;
proc freq data = IV_test_data;
tables x * y / norow nocol nopercent;
weight w;
run;
```

x y

Freq	0	1	Total
1	2	1	3
2	1	1	2
3	2	3	5
Total	5	5	10

Example 1 - Macro Call:

```
%BEST_COLLAPSE(IV_test_data, X, Y, W, IV, J, YES, NO);
```

Note: MODE = J, so only adjacent pairs are considered for collapsing (these pairs are: 1+2 and 2+3).

Example 1 - Reports:

The levels of X before collapsing:

```
Dataset = IV_Example1_data, Predictor = X, Target = Y, Method = IV, Mode = J
Collapse Step: Levels = 3

Obs    x_char    _TYPE_    G    B

 1                  0      5    5
 2       1          1      1    2
 3       2          1      1    1
 4       3          1      3    2
```

The 2 and 3 levels of X were collapsed in this iteration.

```
Dataset = IV_Example1_data, Predictor = X, Target = Y, Method = IV, Mode = J
Collapse Step: Levels = 2

Obs    x_char    _TYPE_    G    B

 1                  0      5    5
 2       1          1      1    2
 3       2+3        1      4    3
```

The VERBOSE = YES parameter caused the three columns L1 L2 L3 to be printed. Note that the SUMMARY includes the c-stat of Y and X. The c-statistic is meaningful only if the ordering of X is meaningful.

```
Dataset = IV_Example1_data, Predictor = X, Target = Y, Method = IV, Mode = J
Summary Report

k       IV       X_STAT    C_STAT    L1     L2      L3

3    0.21972    0.62000   0.62000    1      2       3
2    0.19617    0.60000   0.60000    1      2+3
```

The "Binary Splits" report is produced only when MODE = J. It gives the IV (or LL) values for the binary splits of the values of X. For Example 1 there are only 2 binary splits which are: 1 and 2+3 (Split1) and 1+2 and 3 (Split2).

The Binary Split report is used to check if the IV (or LL) collapsing became sub-optimal at some point during the iterations. This sub-optimality would be shown if the maximum IV for the binary splits was greater than the IV in the Summary Report for k = 2. In Example 1 the maximum binary split occurs for 1 vs. 2+3. This agrees with the IV value for k=2 from the Summary Report.[10]

```
Dataset = IV_Example1_data, Predictor = X, Target = Y, Method = IV, Mode = J
Final Step Binary Splits for MODE = J

Obs      Split1      Split2

 1      0.19617     0.16219
```

Example 2 (below) will provide an example where the IV collapsing process does become suboptimal.

[10] Even if the maximum IV from the binary split equals the IV from k=2 it is not ruled out that at some earlier iteration IV departed from optimal but then later returned to optimal.

Example 2 - Data

The Table 2 has coded income levels called **income_c** versus a binary response **Y**. The income_c will be regarded as ordered and %BEST_COLLAPSE will be run with METHOD = IV and MODE=J.

Table 2 - IV_Test_Income Dataset

	income_c												
Y	01	02	03	04	05	06	07	08	09	10	11	12	**Total**
0	1393	6009	5083	4519	8319	4841	2689	2090	729	292	253	294	36511
1	218	890	932	1035	2284	1593	1053	872	311	136	120	142	9586
Total	1611	6899	6015	5554	10603	6434	3742	2962	1040	428	373	436	46097

Example 2 - Macro Call:

The Macro call is: %BEST_COLLAPSE(IV_Test_Income, Income_C, Y, W, IV, J, YES, NO);

Example 2 - Reports:

Table 3 shows a partial listing of the Summary Report.

Table 3

Dataset = IV_Test_Income, Predictor = Income_C, Target = Y, Method = IV, Mode = J							
Summary Report							
k	IV	X_STAT	C_STAT	L1	L2	L3	L4 to L12 OMITTED
12	0.12145	0.59795	0.59775	01	02	03	
11	0.12145	0.59795	0.59775	01	02	03	
10	0.12144	0.59795	0.59775	01	02	03	
9	0.12143	0.59793	0.59773	01	02	03	
8	0.12136	**0.59783**	**0.59783**	01+02	03	04	
7	0.12113	0.59753	0.59753	01+02	03	04	
6	0.12046	0.59707	0.59707	01+02	03	04	
5	0.11792	0.59463	0.59463	01+02	03	**04+05**	
4	0.11513	0.59282	0.59282	01+02+03	04+05	06	
3	0.11029	0.58905	0.58905	01+02+03	04+05	06+07+08+09+10+11+12	
2	**0.08439**	0.56457	0.56457	01+02+03	04+05+06+07+08+09+10+11+12		

When the collapsing process reached k = 8 the x-stat equaled the c-stat. Therefore, the collapsed X has a monotonic relationship to Y starting with k = 8.

The final collapse to k=2 levels gave a binary split of the values of X into [01 to 03] and [04 to 12]. The Binary Split report (Table 4) shows that this IV collapsing process became sub-optimal. Specifically, the split [01 to 04] and [05 to 12] gave the highest binary split with IV = **0.08883** which is greater than the final IV in Table 3 of **0.08439**. A "wrong path" occurred when the point "04" was joined to "05" instead of to "01+02+03" at k = 5.

As a practical matter in this example the modeler would certainly stop the collapsing process before k=4 due to the large drop-offs in both IV and x-stat at k=6 and further down.

Table 4

Dataset = IV_Test_Income, Predictor = Income_C, Target = Y, Method = IV, Mode = J										
Final Step Binary Splits for MODE = J										
IVsplit1	IVsplit2	IVsplit3	IVsplit4	IVsplit5	IVsplit6	IVsplit7	IVsplit8	IVsplit9	IVsplit10	IVsplit11
0.00822	0.05801	**0.08439**	**0.08883**	0.07797	0.05937	0.03710	0.01788	0.01132	0.00758	0.00417

Log Likelihood and Information Value do not Always Collapse in the Same Way

Using the Income data set (Table 2) and collapsing by MODE = J, the maximum log likelihood and the IV algorithms collapse X differently.

- LL: For k = 5 this algorithm collapsed "03" with "01+02"
- IV: For k = 5 this algorithm collapsed "04" with "05".

An Algorithm For Collapsing That Appeared Promising But Failed

The idea for an algorithm that collapses the levels of X comes from noting that collapsing two levels i and j where $g_i/b_i = g_j/b_j$ gives the same IV as before the collapse. So, collapsing two levels i and j where g_i/b_i and g_j/b_j are closest together should seemingly maximize IV among the other choices. Such an algorithm would be efficient, needing a sort by g/b and an inspection of differences in the g/b across the successive observations to find the minimum. But the algorithm fails.

An example is given in Table 5. Levels 3 and 4 have the closest g/b. But this approach does not pick the levels to collapse which would maximize IV. Collapsing levels 3 and 4 gives IV = 0.012497. However, collapsing levels 2 and 3 gives a higher IV = 0.012524.[11]

Table 5 – Example showing the "minimum difference of g/b" algorithm fails to maximize IV.

X	Y = 0 "B"	Y = 1 "G"	b: Col % Y = 0	g: Col % Y = 1	g/b	row to row change in g/b	
1	272	325	0.2747	0.3250	0.84538		
2	100	100	0.1010	0.1000	1.01010	0.16472	
3	99	95	0.1000	0.0950	1.05263	0.04253	
4	519	480	0.5242	0.4800	1.09217	**0.03954**	← Minimum g/b change

STOPPING GUIDELINES

Subjective judgment by the modeler will inevitably play a large role in deciding when to stop collapsing levels when applying %BEST_COLLAPSE. This is sound and practical since the modeler will be familiar with the predictor variable. This judgment can be assisted by the statistics produced by %BEST_COLLAPSE:

IV, x-stat, and c-stat

The modeler can inspect the changes in IV and x-stat to determine when too much predictive power is lost by a collapse. In the case of numeric predictors, the equality of x-stat and c-stat signals monotonicity.

Log Odds Ratio of the Levels to be Collapsed

If levels i and j are selected to be collapsed, then their log-odds ratio = LO = $\log((G_i / B_i) / (G_j / B_j))$. The approximate standard deviation of the LO is LO_SD = SQRT $(1/G_i + 1/B_i + 1/G_j + 1/B_j)$. Assuming cell counts in rows i and j are large, then LO is normally distributed and an approximate 95% confidence interval (CI) is:

LO +/- 2 * LO_SD (approximate 95% confidence interval for true LO).

If LO = 0, then $g_i / b_i = g_j / b_j$ and the collapsing of i and j is a good decision. Roughly[12], the more that LO deviates from 0, the greater will be the decrease in IV from collapsing. A potential guideline is to consider stopping the collapsing process if LO +/- 2* LO_SD does not include 0.

[11] This example was found by a trial and error process and after we failed to prove that the algorithm would maximize IV.
[12] Recall the discussion surrounding Table 5.

In Table 6 the 95% CI for the log-odds at level 6 omits zero. This suggests stopping at k = 7. This conclusion is reinforced by examining the change in the IV and x-stat when going from 7 to 6 levels. For each statistic there is a noticeable drop between k = 7 and k = 6. (For example, IV drops from 0.12113 at k = 7 to 0.12046 at k = 6.)

Table 6

Dataset = IV_test_Income, Predictor = Income_C, Target = Y, Method = IV, Mode = J							
Log-odds with 95% CI							
k	IV	x-stat	Collapsing to	LO	LO_SD	LOminus2SD	LOplus2SD
12	0.12145	0.59795	11	-0.01820	0.15187	-0.32193	0.28553
11	0.12145	0.59795	10	-0.02786	0.12722	-0.28229	0.22658
10	0.12144	0.59795	9	-0.02225	0.07882	-0.17989	0.13539
9	0.12143	0.59793	8	0.05507	0.08121	-0.10735	0.21749
8	0.12136	0.59783	7	-0.06920	0.05022	-0.16963	0.03123
7	**0.12113**	**0.59753**	6	-0.15575	0.06583	**-0.28741**	**-0.02410**
6	**0.12046**	**0.59707**	5	-0.18128	0.04178	-0.26483	-0.09772
5	0.11792	0.59463	4	-0.20287	0.04803	-0.29894	-0.10680
4	0.11513	0.59282	3	-0.23202	0.03703	-0.30609	-0.15796
3	0.11029	0.58905	2	-0.37940	0.02655	-0.43251	-0.32629
2	0.08439	0.56457					

SECTION TWO:
INFORMATION VALUE STATISTIC GUIDELINES - COMPARISON OF IV AND X-STATISTIC

This section will focus on comparing IV to the x-statistic in order to better understand the **IV Rules of Thumb** for evaluating the strength a predictor.

In addition to the x-statistic it is possible to compute chi-square statistics between X and Y and to look for significant values of association. But the chi-square may be highly significant simply due to large sample sizes. In contrast, the x-statistic and the IV statistic are not dependent on sample size.

How to actually perform the comparison of IV and x-statistic?

A program can be written to produce all the frequency tables with specified "N" total observations and "K" rows and where the cell counts G_k and B_k are non-zero (so that IV can be computed.) Then IV and x-stat are computed for each table of the form of Table 7.

Table 7 – Generic Table in the IV population with parameters K and N
G_k and B_k required to be non-zero

	Y		
X	Y = 0	Y = 1	TOTAL
X_1	B_1	G_1	N_1.
...	...	...	…
X_K	B_K	G_K	N_K.
			N

A Small Complete Enumeration Example

The IV and x-statistic values for all tables where N = 8 and K = 3 are shown below. There are 21 tables but only 4 unique combinations of IV and x-stat values, as shown in Table 8. See the Appendix for a complete list of the 21 tables.

Table 8 – All unique IV and x-stat pairs for the population of tables with N = 8 and K = 3

IV	x-stat_mean
0.00000	0.50000
0.09242	0.56667
0.29296	0.63333
0.34657	0.65625

Complete Enumeration for N and K having a Large Number of Tables

For fixed N and K there are examples where two tables have the same IV but have different x-stat values. So there is not the concept of a list of IV values with their associated x-stat.[13] But, as a practical matter, even for small N and K, there are far too many unique IV values to list. Instead, the IV values are grouped into narrow ranges and x-stat distributional values are computed for the mean, 10th percentile, 90th percentile, 1st percentile and 99th percentile.

In the Tables below four exemplary values of IV were selected and ranges of width +/- 0.005 were formed around each of them. IV and x-stat distributional statistics are shown for these value-ranges of IV. See Table 9A, 9B, 9C.

For **N = 50 and K = 3** there are 1,906,884 tables but only 155,351 unique pairs of IV and x-stat values.[14]

Table 9A: N = 50 and K = 3 based on enumeration

IV range	x_stat count	x_stat_mean	x_stat_P10	x_stat_P90	x_stat_P01	x_stat_P99
0.02 +/- .005	1,303	0.53289	0.52564	0.53906	0.51843	0.54160
0.1 +/- .005	1,278	0.57498	0.56160	0.58333	0.54800	0.58571
0.2 +/- .005	1,209	0.60488	0.58732	0.61680	0.56981	0.62000
0.3 +/- .005	1,126	0.62825	0.60480	0.64260	0.58847	0.64571

For **N = 50 and K = 4** there are 85,900,584 tables but only 1,709,364 unique pairs of IV and x-stat values.

Table 9B: N = 50 and K = 4 based on enumeration

IV range	x_stat count	x_stat_mean	x_stat_P10	x_stat_P90	x_stat_P01	x_stat_P99
0.02 +/- .005	3,412	0.53424	0.52778	0.54000	0.52083	0.54221
0.1 +/- .005	7,680	0.57732	0.56571	0.58508	0.55263	0.58766
0.2 +/- .005	9,756	0.60967	0.59524	0.61969	0.57738	0.62240
0.3 +/- .005	10,775	0.63361	0.61630	0.64569	0.59621	0.64881

For **N = 100 and K = 3** there are 71,523,144 tables but only 5,876,866 unique pairs of IV and x-stat values.

Table 9C: N = 100 and K = 3 based on enumeration

IV range	x_stat count	x_stat_mean	x_stat_P10	x_stat_P90	x_stat_P01	x_stat_P99
0.02 +/- .005	47,931	0.53256	0.52505	0.53881	0.51662	0.54140
0.1 +/- .005	44,521	0.57335	0.55838	0.58313	0.54071	0.58590
0.2 +/- .005	41,830	0.60392	0.58472	0.61682	0.56151	0.61980
0.3 +/- .005	37,742	0.62730	0.60471	0.64224	0.57791	0.64560

Observations:

- The mean x-stat increases as K increases from 3 to 4 for N = 50.
- The mean x-stat decreases slightly as N increases from 50 to 100 for K = 3

Table 9D: Summary

Complete Enumeration			
IV range	N = 50 and K = 3	N = 50 and K = 4	N = 100 and K = 3
	x_stat_mean		
0.02 +/- .005	0.53289	0.53424	0.53256
0.1 +/- .005	0.57498	0.57732	0.57335
0.2 +/- .005	0.60488	0.60967	0.60392
0.3 +/- .005	0.62825	0.63361	0.62730

[13] There are also examples of two tables with the same x-stat but different IV values. Contact the authors for examples.
[14] **proc sort** data = population out = unique nodupkey; by IV6 x_stat6; (To avoid spurious non-dupes due to calculation imprecision of IV and x-stat, the IV and x-stat were rounded to 6 decimal places to create IV6 and x_stat6.)

PROBLEM:

A complete enumeration for larger tables is not practically possible even for modest size N and K. For K levels there are 2*K cells in the table. The general formula[15] for the number of tables with N total frequency and 2*K cells (all being non-zero) is:

C(N-1, 2*K-1) where C(n, k) = n! / ((n-k)! * k!) is the combination symbol.

For K = 2 levels and N $\geq$ 4 the formula works out to be $(N^3 - 6*N^2 + 11*N - 6) / 6$. As shown by the formula the growth in table count is polynomial in N. Expressed in terms of DO LOOPs the formula is:

```
/* This formula is only valid for K = 2. For each one unit increase in K, two more DO
LOOPs must be added following the pattern shown below */
N = <>;  /* N >= 4 */
K = 2;
count = 0;
do i1 = 1 to (N -(2*K - 1));
   do i2 = 1 to (N - i1 - (2*K - 2));
      do i3 = 1 to (N - i1 - i2 - (2*K - 3));
         count = count + 1;  /* Gives the count of tables */
         end;
      end;
   end;
```

For N = 50 and K = 2 the table count is 18,424 (by the formula). From Table 9A, there are 1,906,884 tables for N = 50 and K = 3. From Table 9B the count climbs to 85,900,584 for N = 50 and K = 4.

SOLUTION:

A SAS program was written to sample from the population of all possible tables for given N and K and then to compute IV and x-stat for the sampled tables. Using this sample a function of the form

F(N, K, IV) = x-stat

can be developed by linear regression.

THE REGRESSION EQUATION: F(N, K, IV) = X-STAT

Values of N and K were selected that arise in the actual practice of building models. Samples[16] from the populations of tables determined by the N and K were obtained to form the data set for regression. IV and x-stat were computed for each table in the sample. For a given N and K only unique pairs of IV and x-stat were retained for fitting the model.[17]

The **Design** of the data set for regression followed these rules:

- Restricted the IV values to a range of practical interest. IV in range 0.0 to 0.5.
- Selected N for sizes commonly used for developing predictive models. N: 500, 1000, 2000, 3000, 4000
- Selected K for counts of levels often encountered. K: 4, 6, 8, 10, 12

Predictor variables for use in fitting F(N, K, IV) = x-stat were:

IV
IV squared
N_1K (=N/1000)
K

[15] This is a formula from the mathematical subject of Partition of Integers. See http://mathforum.org/library/drmath/view/52268.html where the formula is derived. This derivation uses an approach involving a "generating function". See the Appendix for an alternative elementary proof.
[16] SAS code is not included in this paper but is available from the authors.
[17] The use of unique pairs for given N and K will give equal weight in the regression to each value of IV occurring in the sample.

Results are given in Table 10.

Table 10: F(N, K, IV) = x-stat

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	11.95321	2.9883	21420.7	<.0001
Error	12206	1.70281	0.00013951		
Corrected Total	12210	13.65602			

Root MSE	0.01181	R-Square	0.8753
Dependent Mean	0.63598	Adj R-Sq	0.8753
Coeff Var	1.85718		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	0.53577	0.00056869	942.1	<.0001
IV	1	0.42194	0.00394	107.12	<.0001
IV_sq	1	-0.32069	0.0067	-47.88	<.0001
N_1K	1	-0.00002699	0.00008419	-0.32	0.7485
K	1	0.00071647	0.00004201	17.05	<.0001

Except for N_1K, all predictors are significant. Overall R-squared is strong at 88%.

The regression model's fitted-values were computed for the design points of IV, N, and K and then extrapolated to K=14.

Results are shown in Table 11.

Table 11: Fitted values of x-stat for design points of N and K and extrapolated to K = 14.

		X-STAT VALUES						
TABLE SIZE		LEVELS = K						
	IV	4	6	8	10	12	14
N=500	0.02	0.54693	0.54836	0.54980	0.55123	0.55266	0.55410
	0.1	0.57761	0.57904	0.58047	0.58191	0.58334	0.58477
	0.2	0.61018	0.61161	0.61305	0.61448	0.61591	0.61734
	0.3	0.63634	0.63777	0.63921	0.64064	0.64207	0.64350
N=1000	0.02	0.54692	0.54835	0.54978	0.55122	0.55265	0.55408
	0.1	0.57759	0.57903	0.58046	0.58189	0.58333	0.58476
	0.2	0.61017	0.61160	0.61303	0.61447	0.61590	0.61733
	0.3	0.63633	0.63776	0.63919	0.64062	0.64206	0.64349
N=2000	0.02	0.54689	0.54832	0.54976	0.55119	0.55262	0.55406
	0.1	0.57757	0.57900	0.58043	0.58187	0.58330	0.58473
	0.2	0.61014	0.61157	0.61301	0.61444	0.61587	0.61730
	0.3	0.63630	0.63773	0.63916	0.64060	0.64203	0.64346
N=3000	0.02	0.54686	0.54830	0.54973	0.55116	0.55260	0.55403
	0.1	0.57754	0.57897	0.58041	0.58184	0.58327	0.58470
	0.2	0.61011	0.61155	0.61298	0.61441	0.61584	0.61728
	0.3	0.63627	0.63770	0.63914	0.64057	0.64200	0.64344
N=4000	0.02	0.54684	0.54827	0.54970	0.55114	0.55257	0.55400
	0.1	0.57751	0.57895	0.58038	0.58181	0.58324	0.58468
	0.2	0.61009	0.61152	0.61295	0.61438	0.61582	0.61725
	0.3	0.63624	0.63768	0.63911	0.64054	0.64198	0.64341

As shown in Table 12, the extrapolated regression values for N = 50 and K = 4 are close to the complete enumeration averages except for the case of IV = 0.02 +/- .005 where the values are different by 0.0127.

Table 12: For N=50 and K=4 Comparison of Enumeration and Regression

	x-stat	
IV range	Regression Fitted	Enumeration
0.02 +/- .005	0.54694	0.53424
0.1 +/- .005	0.57762	0.57732
0.2 +/- .005	0.61019	0.60967
0.3 +/- .005	0.63635	0.63361

Tables 11 and 12 support the conclusion that IV can be related to the x-stat across wide ranges of N and K according to the simplified Table 13 below:

Table 13: Simplified IV to x-Stat Relationship

IV	X-STAT
0.02	0.55
0.1	0.58
0.2	0.61
0.3	0.64

How to receive the programs in the IV and x-stat study:

Contact the authors for the completion enumeration program, sampling program, and regression program.

REFERENCES

Finlay, S. (2010). *Credit Scoring, Response Modelling and Insurance Rating*, New York: Palgrave MacMillan.

Laurendi, J. (2005), Partitions of Integers, http://www.artofproblemsolving.com/Resources/Papers/LaurendiPartitions.pdf

Lin, A. Z. (2013). "Variable Reduction in SAS by Using Weight of Evidence and Information Value", *SAS Global Forum 2013 Proceedings,* Paper 095-213.

Lund, B. and Raimi, S. (2012). "Collapsing Levels of Predictor Variables for Logistic Regression and Weight of Evidence Coding, *MWSUG 2012, Proceedings,* Midwest SAS Users Group, Inc, Paper SA-03.

Mays, E. and Lynas, N. *Credit Scoring for Risk Managers,* CreateSpace Independent Publishing Platform.

Raimi, S. and Lund, B. (2012). "Efficiently Screening Predictor Variables for Logistic Models", *NESUG 2012, Proceedings*, Northeast SAS Users Group, Inc, Paper SA9.

Siddiqi, N. (2006). *Credit Risk Scorecards,* Hoboken, NJ: John Wiley & Sons, Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Bruce Lund
Marketing Associates, LLC
777 Woodward Ave, Suite 500
Detroit, MI, 48226
blund@marketingassociates.com

David Brotherton
Marketing Associates, LLC
777 Woodward Ave, Suite 500
Detroit, MI, 48226
dbrotherton@marketingassociates.com

APPENDIX

%BEST_COLLAPSE

```
%MACRO BEST_COLLAPSE(DATASET, X, Y, W, METHOD, MODE, VERBOSE, LL_STAT);
* Best Collapse Version 6a;

options ls=230 nocenter;

/* !!! WARNING: There is No Input Data Checking in this Program !!! */

/* DATASET is a dataset name - either one or two levels */
/* X (Predictor) is a numeric or character variable which can have MISSING values */
   /* Missing values of X are ignored in all calculations */
   /*  "___x_Char" is RESERVED.  Do not use ___x_char as name of predictor */
/* Y (Target) has values 0 and 1 without MISSING values */
/* W (Freq) has values which are positive integers.  It represents a FREQUENCY variable */
/* METHOD is IV or LL */
   /* For METHOD = IV the collapsing maximizes IV
      For METHOD = LL the collapsing maximizing Log likelihood */
/* MODE is A or J */
   /*    For MODE = A all pairs of levels are compared when collapsing IV
      For MODE = J only adjacent pairs of levels are compared when collapsing IV */
/* VERBOSE = YES is used to display the entire history of collapsing in the SUMMARY REPORT */
   /* Otherwise this history is not displayed in the SUMMARY REPORT */
/* LL_STAT = YES is used to display Log Likelihood for Model and Likelihood Ratio Chi Square
Probability */

/* 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" */

%global num_levels;
%global STOP;
%global LL_inter;

%IF &METHOD NE LL
%THEN
%DO;
   %IF &METHOD NE IV
   %THEN
   %DO;
      %PUT INVALID SUBSTITUTION METHOD = &METHOD;
      %PUT ENDING EXECUTION;
      %GOTO EXIT;
      %END;
   %END;

proc means data = &DATASET noprint; class &X; var &Y; freq &W;
types () &X;
output out = mean_out_0
sum = y;
run;

%let STOP = NO;

data mean in; set mean out_0 nobs = num_levels;
   length ___x_char $75;
   LABEL ___x_char = "&X";
   keep ___x_char G B;
   ___x_char = trim(&X);
   B = _freq_ - y;
   G = y;
   if _n_ = 1 then call symput('num_levels',num_levels - 1); /*Subtracts 1 for _TYPE_=0*/
   if _n_ = 1 then call symput('num_levels_minus1',num_levels - 2);
```

```
   if _n_ = 1
   then
   do;
      LL_Inter = B*log(B/_freq_) + G*log(G/_freq_);
      call symput('LL_inter',LL_inter);
      end;
   if G = 0 or B = 0 then call symput("STOP","YES");
   if _type_ = 1 then output;
run;

   %IF &STOP = YES %THEN
   %DO;
      %PUT ZERO CELL DETECTED;
      %PUT ENDING EXECUTION;
      %GOTO EXIT;
      %END;

%MACRO BEST_COLLAPSE_LEVELS(NUM_LEVELS_R);

proc means data = mean_in noprint; class ___x_char; var G B;
output out = mean_out(keep = ___x_char G B _type_)
sum = G B;
run;

proc print data = mean_out label;
title1
"Dataset= &DATASET, Predictor= &X, Target= &Y, Method= &METHOD, Mode= &MODE, RUN ON &SYSDATE
&SYSTIME";
title2 " ";
title3 "Collapse Step: Levels = &num_levels_r";
run;

data
   denorm&num_levels_r
   mean_in(keep = ___x_char G B)
   %IF ("%UPCASE(&MODE)" = "J" AND &num_levels_r = &num_levels)
   %THEN %DO;
      Split(keep = split1 - split%cmpres(&num_levels_minus1))
      %END;
      ;
   set mean_out end = eof;

   length L1 - L&num_levels_r $75;
   length ___x_char $75;
   array Gx{*} G1 - G&num_levels_r;
   array Bx{*} B1 - B&num_levels_r;
   array LEVELx{*} $ L1 - L&num_levels_r;

   array Splitx{*} Split1 - Split%cmpres(&num_levels_minus1);
   retain G_total B_Total k collapsing_to IV LL_Model LRCS LR_Chi_Sq_Prob;
   retain G1 - G&num_levels_r B1 - B&num_levels_r L1 - L&num_levels_r;
   if _type_ = 0
   then
   do;
      G_total = G;
      B_total = B;
      k = 0;
      IV = 0;
      LL_Model = 0;
      end;
   if _type_ = 1
   then
   do;
      k + 1;
      collapsing_to = k - 1;
      Gx{k} = G;
      Bx{k} = B;
      LEVELx{k} = trim(left(___x_char));
      IV = IV + (G/G_total - B/B_total)*log((G/G_total) / (B/B_total));
      LL_Model = LL_Model + G * log(G/(G+B)) + B * log(B/(G+B));
      end;
```

```
if eof
then
do;
   Minus2_LL = -2*LL_Model;
   LRCS = -2 * (&LL inter - LL Model);
   LR_Chi_Sq_Prob = 1 - PROBCHI(LRCS,k-1);
   LABEL Minus2_LL = "-2*Log L";
   LABEL LR_Chi_Sq_Prob = "Prob(x > LR_Chi_Sq)";
   LABEL LRCS = "Lik-Ratio Chi_Sq";
   LABEL LL_Model = "LL for Model";

   %IF "%UPCASE(&MODE)" = "J" AND "%UPCASE(&METHOD") = "IV" AND &num_levels_r = &num_levels
   %THEN
   %DO;
      do r = 1 to &num_levels_r - 1;
         SUM_G_left = 0; SUM_B_left = 0;
         do s = 1 to r;
            SUM_G_left = SUM_G_left + Gx{s}/G_total;
            SUM_B_left = SUM_B_left + Bx{s}/B_total;
            end;
         SUM_G_right = 0; SUM_B_right = 0;
         do s = r+1 to &num_levels_r;
            SUM_G_right = SUM_G_right + Gx{s}/G_total;
            SUM_B_right = SUM_B_right + Bx{s}/B_total;
            end;
         Splitx{r} = (SUM_G_left - SUM_B_left)     * log(SUM_G_left / SUM_B_left) +
                     (SUM_G_right - SUM_B_right)   * log(SUM_G_right / SUM_B_right);
         end;
      OUTPUT Split;
      %END;
   %IF "%UPCASE(&MODE)" = "J" AND "%UPCASE(&METHOD") = "LL" AND &num_levels_r = &num_levels
   %THEN
   %DO;
      do r = 1 to &num_levels_r - 1;
         SUM_G_left = 0; SUM_B_left = 0;
         do s = 1 to r;
            SUM_G_left = SUM_G_left + Gx{s};
            SUM_B_left = SUM_B_left + Bx{s};
            end;
         SUM G right = 0; SUM B right = 0;
         do s = r+1 to &num_levels_r;
            SUM_G_right = SUM_G_right + Gx{s};
            SUM_B_right = SUM_B_right + Bx{s};
            end;
         Splitx{r} = SUM_G_left*log(SUM_G_left/(SUM_G_left+SUM_B_left)) +
                     SUM_B_left*log(SUM_B_left/(SUM_G_left+SUM_B_left)) +
                     SUM_G_right*log(SUM_G_right/(SUM_G_right+SUM_B_right)) +
                     SUM_B_right*log(SUM_B_right/(SUM_G_right+SUM_B_right));
         end;
      OUTPUT Split;
      %END;

   min C = 99999999;
   X_STAT = 0;
   C_STAT = 0;
   do i = 1 to &num_levels_r - 1;
      %IF "%UPCASE(&MODE)" = "A" %THEN %DO; do j = i+1 to &num_levels_r; %END;
      %IF "%UPCASE(&MODE)" = "J" %THEN %DO; do j = i+1 to i+1; %END;
         %IF &METHOD = LL
         %THEN
         %DO;
            L_i = Gx{i}*log(Gx{i}/(Gx{i}+Bx{i})) + Bx{i}*log(Bx{i}/(Gx{i}+Bx{i}));
            L_j = Gx{j}*log(Gx{j}/(Gx{j}+Bx{j})) + Bx{j}*log(Bx{j}/(Gx{j}+Bx{j}));
            C_ij = L_i + L_j -
               ( (Gx{i}+Gx{j})*log((Gx{i}+Gx{j})/(Gx{i}+Gx{j}+Bx{i}+Bx{j})) +
               (Bx{i}+Bx{j})*log((Bx{i}+Bx{j})/(Gx{i}+Gx{j}+Bx{i}+Bx{j})) );
            if C_ij <= min_C
            then
            do;
               i_index = i;
```

```
               j_index = j;
               min_C = C_ij;
            %END;
         %ELSE %IF &METHOD = IV
         %THEN
         %DO;
            L_i = ( Gx{i}/G_total - Bx{i}/B_total ) *
                  log( (Gx{i}/G_total) / (Bx{i}/B_total) );
            L_j = ( Gx{j}/G_total - Bx{j}/B_total ) *
                  log( (Gx{j}/G_total) / (Bx{j}/B_total) );
            C_ij = L_i + L_j -
                  ( (Gx{i} + Gx{j})/G_total - (Bx{i} + Bx{j})/B_total ) *
                  log( ((Gx{i} + Gx{j})/G_total) / ((Bx{i} + Bx{j})/B_total) );
            if C_ij <= min_C
            then
            do;
               i_index = i;
               j_index = j;
            min_C = C_ij;
            %END;

            if &num_levels_r >= 3
            then
            do;
               LO = log((Gx{i}*Bx{j})/(Gx{j}*Bx{i}));
               LO_SD = sqrt(1/Gx{i} + 1/Gx{j} + 1/Bx{i} + 1/Bx{j});
               LOplus2SD = LO + 2*LO_SD;
               LOminus2SD = LO - 2*LO_SD;
               end;

            end;
         END; /* END OF J loop */

      do j = i+1 to &num_levels_r;
         C_STAT = C_STAT  + Bx{I}*Gx{J};
         X_STAT = X_STAT  + ABS(Bx{i}*Gx{j} - Gx{i}*Bx{j});

         end; /* END OF: J loop */
      end; /* END OF: I loop */

   do i = 1 to &num levels r;
      C_STAT  = C_STAT + .5*Bx{I}*Gx{I};
      END;

   C_PAIR  = B_TOTAL * G_TOTAL;
   C_STAT  = MAX( C_STAT  / C_PAIR,  1 - C_STAT  / C_PAIR );
   X_STAT  = .5 * (X_STAT  / C_PAIR  + 1);

   OUTPUT denorm&num_levels_r;

   do i = 1 to &num_levels_r;
      if i = i_index or i = j_index
         then ___x_char = compress(LEVELx{i_index}||"+"||LEVELx{j_index});
       else ___x_char = LEVELx{i};
      G = Gx{i};
      B = Bx{i};
      OUTPUT mean_in;
      end;

   end; /* END OF: if eof then do */
run;
proc append base = denorm data = denorm&num_levels_r force nowarn;
run;
%MEND;

%MACRO INTER;
proc delete data = denorm;
run;
%do k = &num_levels %to 2 %by - 1;
   %BEST_COLLAPSE_LEVELS(&k);
   %end;
```

```
proc print data = denorm noobs label;
var K
%IF &LL_STAT = YES %THEN LL_Model Minus2_LL LRCS LR_Chi_Sq_Prob;
IV X_STAT C_STAT
%IF &VERBOSE = YES %THEN L1 - L%cmpres(&num_levels); ;
format IV X_STAT C_STAT 8.5;
title2 " ";
title3 "Summary Report";
run;
%IF ("&MODE" = "J")
%THEN %DO;
   proc print data = Split;
   title2 " ";
   title3 "Final Step Binary Splits for MODE = J";
   %END;
run;

proc print data = denorm noobs;
var K  IV X_STAT collapsing_to LO LO_SD LOminus2SD LOplus2SD;
format IV X_STAT LO LO_SD LOminus2SD LOplus2SD 8.5;
title2 " ";
title3 "Log-odds with 95% CI";
run;
%MEND;
   %INTER;
%EXIT: %MEND;
```

IV is non-increasing when collapsing levels of X

To show that IV is non-increasing when two levels of X are collapsed it must be shown that:

$$(g1 - b1) * \log(g1/b1) + (g2 - b2) * \log(g2/b2) \geq (g1 - b1 + g2 - b2) * \log((g1 + g2)/(b1 + b2))$$

where 1 and 2 refer to two arbitrary levels of X, g is the "good" percent and b is the "bad" percent.

This inequality is derived from the **Log-Sum Inequality** which says:

$$\sum_{i=1}^{n} A_i * \log(A_i/B_i) \leq (\sum_{i=1}^{n} A_i) * \log(\sum_{i=1}^{n} A_i / \sum_{i=1}^{n} B_i) \text{ for } A_i \text{ and } B_i \text{ positive and } i = 1 \text{ to } n$$

Equality occurs in the Log-Sum Inequality if only if A_i/B^i are equal for all i

Proof:

$(g1 - b1) * \log(g1/b1) + (g2 - b2) * \log(g2/b2) = g1 * \log(g1/b1) + b1 * \log(b1/g1) + g2 * \log(g2/b2) + b2 * \log(b2/g2)$

$= \{ g1 * \log(g1/b1) + g2 * \log(g2/b2) \} + \{ b1 * \log(b1/g1) + b2 * \log(b2/g2) \}$

Using 2 applications of the Log-Sum Inequality gives:

$\geq (g1 + g2) * \log((g1+g2)/(b1+b2)) + (b1 + b2) * \log((b1+b2)/(g1+g2))$

$= (g1 + g2) * \log((g1+g2)/(b1+b2)) - (b1 + b2) * \log((g1+g2)/(b1+b2))$

$= (g1 - b1 + g2 - b2) * \log((g1+g2)/(b1+b2))$

If g1/b1 = g2/b2 then the Log-Sum Inequality is an equality. The inequality in the calculation above becomes an equality.

Complete Enumeration of Tables for N = 8 and K = 3 (with nonzero cells)

Obs	IV	x_stat	G1	G2	G3	B1	B2	B3
1	0.29296	0.63333	1	1	1	1	1	3
2	0.09242	0.56667	1	1	1	1	2	2
3	0.29296	0.63333	1	1	1	1	3	1
4	0.09242	0.56667	1	1	1	2	1	2
5	0.09242	0.56667	1	1	1	2	2	1
6	0.29296	0.63333	1	1	1	3	1	1
7	0	0.5	1	1	2	1	1	2
8	0.34657	0.65625	1	1	2	1	2	1
9	0.34657	0.65625	1	1	2	2	1	1
10	0.29296	0.63333	1	1	3	1	1	1
11	0.34657	0.65625	1	2	1	1	1	2
12	0	0.5	1	2	1	1	2	1
13	0.34657	0.65625	1	2	1	2	1	1
14	0.09242	0.56667	1	2	2	1	1	1
15	0.29296	0.63333	1	3	1	1	1	1
16	0.34657	0.65625	2	1	1	1	1	2
17	0.34657	0.65625	2	1	1	1	2	1
18	0	0.5	2	1	1	2	1	1
19	0.09242	0.56667	2	1	2	1	1	1
20	0.09242	0.56667	2	2	1	1	1	1
21	0.29296	0.63333	3	1	1	1	1	1

Number of Positive Integer Solutions to $x_1 + \ldots + x_k = n$ is C(n-1, k-1)

If k and m are positive integers, Laurendi (2005) in his Example 5 gives a simple demonstration that the number of non-negative integer solutions ($x_1 \ldots x_k$) (each $x_i \geq 0$) to

$$x_1 + \ldots + x_k = m \quad \ldots (A)$$

is C(m+k-1, k-1)

We will show that there is a 1 to 1 correspondence between non-negative integer solutions to (A) and the positive integer solutions (each $x_i > 0$) to:

$$x_1 + \ldots + x_k = m + k \ldots (B)$$

This will prove that the number of positive solutions to (B) is C(m+k-1, k-1).

Since **m** is an arbitrary positive integer, we can define **n** = **m** + **k** and re-state the conclusion to say: If **n** $\geq$ 2 and **k** $\geq$ 1, then the number of positive solutions to

$$x_1 + \ldots + x_k = n \quad \ldots (C)$$

is C(n-1, k-1).

Now for the 1-1 correspondence:

Let $(a_1,\ldots, a_k)$ solve (A). Then $(a_1 +1, \ldots, a_k +1)$ solves (C).

Conversely, if $(b_1,\ldots, b_k)$ solves (C), then $(b_1 -1,\ldots, b_k -1)$ solves (A)