Paper 2019-0026

Logistic Regression, Basics and Beyond

Bruce Lund, Independent Consultant

ABSTRACT

This paper presents light theory, supported by simulations, as well as practical suggestions for developing binary logistic regression models. Topics include: Firth method versus maximum likelihood method for estimating parameters; screening, binning, transforming predictors; identification of multicollinearity; predictor selection methods using PROC LOGISTIC, PROC HPLOGISTIC, PROC HPGENSELECT; and measures of fit and predictive accuracy.

INTRODUCTION

The goal of the paper is to present insights into theory of binary logistic regression and to give practical advice. There is a collage of topics beginning with the basics of maximum likelihood estimation (MLE) and its large sample properties. Next, there is a comparison of the Firth method of parameter estimation to MLE. Finally, topics are taken from various steps in fitting a logistic model. These topics include:

- Screening predictors
- Binning and transforming predictors
- Comparing dummy variable coding with weight of evidence coding
- Detecting multicollinearity
- Comparing methods of selecting predictors to fit a model and model validation

Attention is given to appealing capabilities of PROC HPLOGISTIC for fitting logistic models. For example, HPLOGISTIC provides predictor selection and choice of final model using the Schwarz-Bayes criterion (SBC). Finally, MLE is compared to fitting models by LASSO,[1] provided by PROC HPGENSELECT.

Throughout the paper there are citations to these two references:

- LRuS: Allison, P. (2012a), *Logistic Regression Using SAS®: Theory and Application 2nd Ed*
- ALR: Hosmer D., Lemeshow S., Sturdivant R. (2013). *Applied Logistic Regression 3rd Ed*.

Several technical discussions have been placed in the Appendix.

MAXIMUM LIKELIHOOD ESTIMATION

CONCEPT OF A LOGISTIC MODEL POPULATION

The paper begins by presenting the concept of a logistic model population $\mathcal{P}$. First, the *design matrix* is defined. The *design matrix* [X] has N rows $\{(x_{i,0}, x_{i,1}, \ldots, x_{i,K})$ for i = 1 to $N\}$ with K+1 columns, indexed by j = 0 to K. Column 0 has value 1 in each row. Columns j = 1 to K are called *predictors* X_j. The jth predictor in the ith row has value $x_{i,j}$. The symbol N is a positive integer.

For example, here is a design matrix where there are two predictors X1 and X2 and N = 3:

$$\begin{bmatrix} 1 & x_{1,1} & x_{1,2} \\ 1 & x_{2,1} & x_{2,2} \\ 1 & x_{3,1} & x_{3,2} \end{bmatrix}$$

For the ith row in the design matrix there is an associated number Y_i, called the *target*, with levels 0 and 1. The levels of Y are given names: “1” is a “success” or an “event”, while “0” is a “failure” or “non-event”. The rows from the design along with the associated target value are called observations or cases or subjects.

[1] Least absolute shrinkage and selection operator

Parameters for the population are numbers $\underline{\beta} = (\beta_0, \ldots, \beta_K)$.[2] For the ith row, *xbeta*$_i$ is defined by

$$\text{xbeta}_i = \textstyle\sum_{j=0}^{K} \beta_j \, x_{i,j}.$$

If a row is randomly taken and Y_i is observed, the probability of "success" or an "event" is:

$$P(Y_i = 1 \mid \underline{X} = \underline{x}_i) = \exp(\text{xbeta}_i) \,/\, (1 + \exp(\text{xbeta}_i))$$

The probability of a "failure" or "non-event" is given by: $P(Y_i = 0 \mid \underline{X}_i = \underline{x}_i) = 1 - P(Y_i = 1 \mid \underline{X}_i = \underline{x}_i)$.

Let $[\underline{X}_S]$ be a subset of the rows from the population design matrix (with same columns). A sample $S = \{(\underline{x}_i, y_i)$ for i = 1 to n$\}$ with sample design $[\underline{X}_S]$ is a subset of the population $\mathcal{P}$. A sample is *random* if, given a row from $[\underline{X}_S]$, then each observation from the population with this row (these predictor values) has equal chance of selection for the sample. The sample size is "n". Samples will be assumed to be random.

THE LIKELIHOOD FUNCTION

Using a sample, the population parameters $\underline{\beta}$ may be estimated by various formulas. An estimate of the parameters is denoted by $\hat{\underline{\beta}} = (\hat{\beta}_0 \ \ldots, \hat{\beta}_K)$. Maximum likelihood estimation (MLE) is the common method to produce an estimate $\hat{\underline{\beta}}$. MLE is discussed next. Let $\underline{b} = (b_0, \ldots, b_K)$ be parameter values and write

$$p_i = \exp(\textstyle\sum_{j=0}^{K} b_j \, x_{i,j}) \,/\, (1 + \exp(\sum_{j=0}^{K} b_j \, x_{i,j}))$$

The likelihood function is defined as $L(\underline{b}) = \prod_i^n p_i^{\,y_i} \, (1 - p_i)^{1-y_i}$. It is a function of $\underline{b}$.

Assuming that $L(\underline{b})$ has a maximum value, then $\text{Log}(L(\underline{b}))$ also has a maximum. The maximum likelihood estimates (also abbreviated by MLE) of $\underline{\beta}$ are found by finding a point $\hat{\underline{\beta}}$ which maximizes $\text{Log}(L(\underline{b}))$. A point $\hat{\underline{\beta}}$ which does maximize $\text{Log}(L(\underline{b}))$, as was proved in the 1980's, must be unique.[3]

However, there are data sets and associated models for which the maximum of $\text{Log}(L(\underline{b}))$ does not exist. In this situation the model is said to have either complete separation or quasi complete separation. The topic of separation is discussion in LRuS, pp. 49-59. Separation can occur in very small samples as a natural consequence of the rarity of data or in larger samples due to inadequate attention to pre-modeling data preparation. There are remedial actions available to the modeler. See LRuS, pp. 52-56.

For medium to large samples, the usual and natural method in logistic modeling of finding $\hat{\underline{\beta}}$, is by MLE.[4] This assertion raises two questions:

(1) What constitutes "medium and large samples"? This guideline from Allison (2012b) will be used:

> Let n1 be the count of cases where Y=1 and n0 be the count of cases where Y=0 in a sample of size n (so that n = n1 + n0). Let n* = min(n1, n0). The description "medium to large" will be taken as n* ≥ 200.

(2) Why is MLE the usual and natural method of finding $\hat{\underline{\beta}}$ in cases of medium to large samples?

> For medium and, even more so, for large samples, the estimators $\hat{\beta}_j$ can be treated as unbiased (with mean equaling β_j) and as approximately normally distributed. The variance of $\hat{\beta}_j$ decreases as 1/n (formally, $\sigma^2_{\hat{\beta}_j}$ = O[1/n]).[5]

[2] An English or Greek letter with an underscore will indicate a *tuple*, an ordered sequence of numbers. E.g. $\underline{x}$ = (4, 3, 1)

[3] Albert and Anderson (1984), Santer and Duffy (1989)

[4] In the 1950's the linear probability model, with least squares, was used to fit predictors to a binary target. As late as the 1980's the two-group discriminant model was used as an alternative to logistic regression with MLE. Both the linear probability model and the discriminant model provided closed form solutions for estimating the model parameters. But both approaches have theoretical deficiencies (linear probability, Allison (2012, pp. 10-14); (discriminant, HLS (2013, p. 21)

[5] See the Appendix, Topic 1 for motivation to support the statement that the variance of $\hat{\beta}_j$ decreases as 1/n (formally, $\sigma^2_{\hat{\beta}_j}$ = O[1/n]). The mathematics required to understand the assertion that the MLE $\hat{\beta}_j$ converges to a normal distribution is beyond the scope of this paper. (But see Table 2 for a "proof" by simulation.)

MLE is the default method of fitting a logistic model for PROC LOGISTIC and PROC HPLOGISTIC. Unlike least squares estimation for linear regression, there is not a closed form solution for solving for $\underline{\hat{\beta}}$. But there are fast and accurate algorithms that converge to the MLE (unless there is separation).

Another method, called the Firth Method, is preferred for small samples, and will be discussed in a later section. The Firth method converges to estimates of $\underline{\beta}$ even in the presence of separation.

The statistical reporting by LOGISTIC and HPLOGISTIC is based on the normal approximations, regardless of sample size.[6] The suitability of the normal approximation for small vs. medium/large samples will be investigated by simulation studies.

SIMULATIONS OF LOGISTIC MODEL POPULATIONS

Steps to create a simulated logistic model population are given below:

- A design matrix $[\underline{X}] = \{(x_{i,0}, \ldots, x_{i,K})$ for i = 1 to N $\}$ is specified.
- Observations from a logistic distribution are randomly generated.[7] These are denoted by ε_i.
- A variable Z is constructed as a sum of a linear combination of predictors $\underline{X} = (X_1, \ldots, X_K)$ and parameters $\underline{\alpha} = (\alpha_0, \ldots, \alpha_K)$, and a logistic term:

$$Z_i = \sum_{j=0}^{K} \alpha_j \, x_{i,j} + \varepsilon_i \quad \text{for } i = 1 \text{ to } N$$

- The dichotomization of Z by a value λ defines a variable Y with values 0 and 1 where:

$$\text{If } P(Z_i > \lambda \mid \underline{X_i} = \underline{x_i}), \text{ then } Y_i = 1, \text{ and otherwise } Y_i = 0$$

Because ε_i has a logistic distribution, the probability $P(Y_i = 1 \mid \underline{X_i} = \underline{x_i})$ is computed to be:

$$P(Y_i = 1 \mid \underline{X_i} = \underline{x_i}) = \exp\left(\sum_{j=0}^{K} \beta_j \, x_{i,j}\right) / \left(1 + \exp\left(\sum_{j=0}^{K} \beta_j \, x_{i,j}\right)\right) \ldots (*)$$

where $\beta_0 = -\lambda + \alpha_0$ and $\beta_j = \alpha_j$ for j = 1 to K.

The mathematical derivation of equation (*) is given in Appendix, Topic 2. As a result, $\{(\underline{x_i}, y_i)$ for i = 1 to N $\}$ gives a logistic model population with parameters $\underline{\beta}$. (Also see LRuS pp. 87-89.)

SAS® CODE TO CREATE A LOGISTIC MODEL POPULATION

The code below creates 100 samples, each of size 5,000, from a logistic model population with one predictor X and parameters $\beta_0 = 1$, $\beta_X = 5$. Each sample has the same design $[\underline{X_S}]$. The only difference between the samples is the random effect of "e" on the determination of the values of Y.

```
DATA Samples;
   do s = 1 to 100;
      call streaminit(s);
      do ID = 1 to 5000;
         c = rand("Uniform");
         e = 1*log(c / (1-c)); /* e is a logistic random variable */
         X = (mod(ID,5) - 2)/10;
         Z = 1 + 5*X + e;
         Y = (Z > 0);
         output;
         end;
      end;
PROC MEANS DATA= Samples NOPRINT; CLASS s; VAR Y;
   OUTPUT OUT= MEANOUT(where=(_TYPE_ = 1)) SUM(Y)= Y_SUM;
PROC MEANS DATA= Meanout mean stddev; VAR Y_SUM;
run;
```

[6] This comment excludes models where EXACT method of parameter estimation is used. See LRuS pp. 55-57.

[7] A logistic random variable has cumulative distribution function $F(\varepsilon) = \exp(\varepsilon) / (1 + \exp(\varepsilon))$

The second PROC MEANS shows that average number of events (i.e. observations where Y=1) in the 100 samples is 3,553 with a standard deviation of 31.2

DISTRIBUTION OF β_X FROM SIMULATION OF 100 SAMPLES WITH SAMPLE SIZES 5,000

PROC LOGISTIC is run on each of the 100 samples. The 100 estimates $\widehat{\beta_X}$ are processed by PROC UNIVARIATE.

```
ods exclude all;
/* parameter estimates of β0 and βX are output to EST */
ods output ParameterEstimates= EST;
PROC LOGISTIC DATA= Samples desc;
   BY s;
   MODEL Y= X;
run;
ods exclude none;
PROC UNIVARIATE DATA= EST(where= (variable="X"));
   VAR estimate;
   HISTOGRAM estimate / normal;
run;
```

As shown in Table 1 the mean value of $\widehat{\beta_X}$ is 5.0711. This value is very near to $\beta_X = 5$, the population value. The standard deviation of 0.2475 is also near to the theoretical standard deviation of 0.2460 for the distribution of $\widehat{\beta_X}$ (for 5,000 observations with design [$\underline{X}_S$], and $\beta_0 = 1$, $\beta_X = 5$).[8]

Moments	
N	100
Mean (of $\widehat{\beta_X}$)	5.0177
Std Deviation	0.2475

Table 1.

In Table 2 the quantiles of $\widehat{\beta_X}$ are compared to a normal with mean 5.0177 and standard deviation 0.2475 (estimated from the 100 samples). Upon visual inspection the quantiles show close agreement.

Quantiles for Normal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	4.492	4.442
5.0	4.631	4.611
10.0	4.671	4.701
25.0	4.847	4.851
50.0	5.034	5.018
75.0	5.145	5.185
90.0	5.368	5.335
95.0	5.464	5.425
99.0	5.609	5.594

Table 2

[8] See Appendix, Topic 3 for the calculation that gives the theoretical standard deviation of 0.2460

DISTRIBUTION OF β_X FROM SIMULATION OF 100 SAMPLES WITH SAMPLE SIZE 100

It is not appropriate to use a normal approximation for small samples. Consider again the logistic model with a single predictor X, parameter β_X, and MLE $\widehat{\beta_X}$. If $n^* \le 200$, then it is often observed that:

$$| \widehat{\beta_X} | > | \beta_X |$$

This is not a theorem but rather it is an empirical tendency. As samples are drawn and $\widehat{\beta_X}$ is estimated, the distribution of $| \widehat{\beta_X} |$ is skewed to the right of $| \beta_X |$. This skewness pulls the mean of $| \widehat{\beta_X} |$ to the right of $| \beta_X |$. Qualitative, graphical explanations of this tendency are found in Firth (1993 p. 29) and Rainey and McCaskey (2017, pp. 1-7).

To supplement these references, I'll present a simplified situation that illustrates the over-estimation property of MLE for small samples.

Suppose an idealized large population has the following relationship between X and Y:

X↓ / Y→	0	1
-1	100	50
1	50	100

This implies that the population logistic model parameters are: $\beta_0 = 0$ and $\beta_X = 0.6931$

If there are two equal-size samples “equally distant” from the population distribution but on different “sides”, then the over-estimate of β_X on “one side” is amplified and the underestimate of β_X on the “other side” is constricted. Consider these two samples:

Sample 1	0	1	Sample 2	0	1
-1	90	60	-1	110	40
1	60	90	1	40	110

For sample 1, $\widehat{\beta_X} = 0.4055$ and for sample 2 $\widehat{\beta_X} = 1.0116$.

The under-estimate from sample 1 is 0.6931 - 0.4055 = 0.2876 while the over-estimate from sample 2 is 1.0116 - 0.6931 = 0.3185. The over-estimate is greater than the under-estimate, supporting the tendency that, on average, $| \widehat{\beta_X} | > | \beta_X |$.

Of course, this tendency continues for large samples but is not noticeable. This is due to large-sample convergence in distribution of $\widehat{\beta_X}$ to a normal with mean β_X and a variance $\sigma^2_{\widehat{\beta_X}}$, which decreases as 1/n.

Perhaps a more convincing demonstration that $| \widehat{\beta_X} | > | \beta_X |$, on average, is given by simulations. The simulation of 100 samples with size 5,000, described earlier, is now changed to have 100 samples with sample size 100. The coding for the simulated logistic population is the same as generated earlier except now the SAS code is revised by replacing

“`do ID = 1 to 5000;`“ with “`do ID = 1 to 100;`“

Since the simulated coefficient of X is 5.0, well above zero, we can dispense with absolute values in the discussion which follows. The estimator $\widehat{\beta_X}$ over-estimates β_X, on average, (5.1723 vs. 5). The average of the estimates of standard deviation is also biased upward (1.9290 vs. 1.7391).[9]

Moments	
N	100
Mean (of $\widehat{\beta_X}$)	5.1723
Std Deviation	1.9290

Table 3.

[9] See Appendix, Topic 4 for the calculation that gives the theoretical standard deviation of 1.7391.

In Table 4 the quantiles of $\widehat{\beta_X}$ are compared to a normal with mean 5.1723 and standard deviation 1.9290. Upon visual inspection the right and left tail quantiles do not show close agreement to the estimated normal.

Quantiles for Normal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	1.9262	0.6846
5.0	2.4948	1.9993
10.0	2.8170	2.7001
25.0	3.5290	3.8711
50.0	5.0275	5.1723
75.0	6.3765	6.4734
90.0	7.9787	7.6444
95.0	8.7224	8.3453
99.0	10.0065	9.6599

Table 4.

Although $\widehat{\beta_X}$ over-estimates β_X, on average, there are many samples among the 100 where $\widehat{\beta_X}$ is less than $\beta_X = 5$, as seen by examining the Observed Quantiles in Table 4.

BIAS CORRECTION FOR SMALL SAMPLES, THE FIRTH METHOD

An alternative method to estimate parameters β_j is given by the Firth Method, introduced by David Firth (1993). In the Firth Method the estimates of β_j, denoted $\hat{\beta}_j^{\text{firth}}$, are obtained by finding $\underline{b}$ which maximize:

$$\text{FLog}(\text{L}(\underline{b})) = \text{Log}(\text{L}(\underline{b})) + \text{Log}(\text{A}(\underline{b}))$$

where $\text{A}(\underline{b})$ is an adjustment factor which reduces the bias of MLE for small samples.[10]

Very roughly, $\text{FLog}(\text{L}(\underline{b}))$ tends toward a maximum when $\text{P}(\text{Y}=1 \mid \underline{x}, \underline{b})$ is closer to 0.5 for all $\underline{x}$. Therefore, each b_j, in absolute value, is closer to zero versus the absolute values of the MLE's.

But simply shrinking $\mid \underline{b} \mid$ toward zero is not enough.

Instead, this shrinking also accomplishes the desired goal of providing estimates with less bias than the ML estimates. The relationships of Firth to MLE and to the population parameter are summarized below:

$$|\hat{\beta}_j^{\text{FIRTH}}| < |\hat{\beta}_j^{\text{ML}}| \text{ and } \hat{\beta}_j^{\text{FIRTH}} \sim \beta_j \text{ … on average}$$

The same simulation with 100 samples, each with sample size 100, is used in fitting PROC LOGISTIC with the Firth method to estimate the coefficient of X.

To run Firth, simply modify the MODEL statement as shown:

```
MODEL Y= X / FIRTH;
```

The average Firth estimate $\hat{\beta}_X^{\text{FIRTH}}$, at 4.9815, is closer to the population value of 5 than is the average MLE. See Table 5. The Firth estimate, on average, was shrunk below the average MLE.

Since the Firth estimates, in absolute value, are closer to zero, the standard deviation of these estimates tends to be lower than for the MLE's. See Table 5.

[10] $\text{A}(\underline{b}) = \{\det \mathbf{I}(\underline{b})\}^{0.5}$ where $\mathbf{I}(\underline{b})$ is the observed information matrix evaluated at $\underline{b}$

Moments	FIRTH	MLE
N (samples)	100	100
Mean (of $\widehat{\beta_X}$)	4.9815	5.1723
Std Deviation	1.8401	1.9290

Table 5.

In Table 6 the quantiles of $\hat{\beta}_X^{\mathrm{firth}}$ are compared to a normal with mean 4.9815 and standard deviation 1.8401. Upon visual inspection the right and left tail quantiles do not show close agreement to the estimated normal, but agreement appears to be better than for the quantiles from MLE.

Quantiles for Normal Distribution		
	Quantile	
Percent	Observed	Estimated
1.0	1.8658	0.7009
5.0	2.4172	1.9549
10.0	2.7222	2.6234
25.0	3.4125	3.7404
50.0	4.8586	4.9815
75.0	6.1169	6.2226
90.0	7.6476	7.3396
95.0	8.3770	8.0081
99.0	9.5520	9.2621

Table 6.

The Firth method does not provide a panacea. The predictions from a model fit by the Firth method are biased in the following way:

For MLE, the probabilities p_i, where i = 1 to n, have the following property (where y is coded 0 and 1):

$$\sum_{i=1}^{n} y_i = \sum_{i=1}^{n} p_i$$

Dividing by n gives: $\bar{y} = \bar{p}$. In words, this equation says that that success rate equals the average prediction rate. This is similar to least squares for linear regression where the sum of the residuals equals zero.

For Firth estimation:

$$\sum_{i=1}^{n} y_i \neq \sum_{i=1}^{n} p_i$$

Generally, p_i^{FIRTH} is closer to 0.5 than is p_i^{MLE}.

The mean Firth prediction for the 100 samples was averaged and compared to the mean success rate for the 100 samples. The results are 0.7089 vs 0.7126. See Table 7.

```
ods exclude all;
PROC LOGISTIC DATA= Samples desc;
   BY s;
   MODEL Y= X / Firth;
   OUTPUT OUT= Scored P= Predict;
run;
ods exclude none;
PROC MEANS DATA= Scored NOPRINT; CLASS s;
   VAR Y Predict;
   OUTPUT OUT= MEANOUT(where=(_TYPE_ = 1))
   Mean(Y)= Y_mean
```

```
    Mean(Predict)= Predict_Mean;
 run;
 PROC MEANS DATA= Meanout mean stddev P50;
    VAR Y_mean Predict_Mean;
 run;
```

Variable	Mean	Std Dev	50th Pctl
Y_mean	0.7126	0.0430	0.7150
Predict_Mean (Firth)	0.7089	0.0423	0.7113

Table 7.

ALGORITHMIC METHODS OF SOLVING FOR $\hat{\beta}_j^{FIRTH}$ AND $\hat{\beta}_j^{ML}$

Algorithms for finding the ML estimates begin by taking the derivatives of Log(L($\underline{b}$)) with respect to each b_j and setting these equations to zero. This gives K+1 equations, non-linear in $\underline{b}$. These equations are solved iteratively. In PROC LOGISTIC there are two iterative methods: Fisher scoring and Newton-Raphson.[11] For a brief discussion, see LRuS, pp. 42-45.

The approach for finding the Firth estimates follows the same steps. The derivatives of FLog($\underline{b}$)) with respect to each b_j are computed and set to zero. This gives K+1 equations, non-linear in $\underline{b}$. The equations are made much more complicated by the term Log(A($\underline{b}$)). But despite the more complicated formula, the Fisher scoring or Newton-Raphson can be applied without significant increase in computational complexity. And as a bonus, the Firth estimates exist even in the case of complete separation, where MLE's do not exist. Additional discussion of the estimation of Firth coefficients is given in the Appendix, Topic 5.

FIRTH OR MLE

When n* ≤ 200 and if the primary interest of the modeler is coefficient estimation, the Firth method is preferred to MLE. Confidence intervals for parameters and odds-ratios as well as hypothesis tests for parameters will suffer somewhat from the moderate deviation of the distribution of the Firth estimates from normality (see Table 6) but, at the least, the bias of the parameter estimates is reduced vs. MLE.[12]

But if prediction is the primary goal, then MLE would be preferred since its predictions on a validation sample would not include a built-in bias.

In the 100 simulated samples, each with sample size of 100, the value of n* averages to about 29.[13] The question of doing predictive modeling is almost moot in this case. Fitting a predictive model when n* = 29 presents significant challenges for fitting and then validating the logistic model.[14]

For larger n the impact of the correction term Log(A($\underline{b}$)) on the Firth estimates tends to zero with n. In this case the MLE and Firth parameter estimates are similar.[15] The more conventional MLE should be used.

Since the Firth method utilizes the same algorithms as MLE to find coefficient estimates, there is no advantage to either method based on computational issues.

NOTE

For the remainder of the paper the assumption will be made that any sample is medium / large (n* ≥ 200) and that MLE is used in fitting a logistic model.

[11] http://support.sas.com/documentation/onlinedoc/stat/151/logistic.pdf p. 5835

[12] The profile likelihood method of computing confidence intervals does not depend on normality approximations. It is preferable to the (Wald) confidence interval approach. See LRuS pp. 40-41 and ALR, pp. 18-20.

[13] See Table 7: Average of occurrences of Y=0 over the 100 samples is 100 * (1 - 0.7126) ~ 29.

[14] PROC LOGISTIC provides "leave one out" cross-validation. For n* = 29, this is the only method that could be used for model validation.

[15] For support of this statement, see the final paragraph of the discussion in Appendix, Topic 5.

SCREENING PREDICTORS HAVING ONLY A FEW LEVELS

If X is numeric with many levels or is binary (2 levels), then X can go directly into the PROC LOGISTIC Model Statement: MODEL Y = X;

In contrast, consider predictors that could be Nominal, Ordinal, or Discrete and with few but more than 2 levels. Such predictors will be referred to as "NOD" predictors. A NOD predictor requires processing before it is used in a model. Alternatives for entering a NOD predictor into a logistic model include dummy variable coding, weight of evidence coding, or, for discrete predictors and ordinal predictors (after recoding), as a numeric variable.[16]

A predictor which is numeric and has many levels will be called "continuous". The distinction between "few" and "many" is subjective. Screening and transforming of continuous predictors is taken up in a later section.[17]

Weak NOD predictors should be removed from consideration for use in the logistic model at an early stage. A widely used measure of predictive power to screen NOD predictors is Information Value (IV). The calculation of IV is illustrated in Table 8. The predictor X can be any NOD predictor (character or numeric).

X	Frequencies		Col % Y=0 "b_k"	Col % Y=1 "g_k"	Log(g_k/b_k) = X_woe	g_k - b_k	IV Terms (g_k - b_k) * Log(g_k/b_k)
	Y = 0	Y = 1					
X1	2	1	25.0%	12.5%	-0.69315	-0.125	0.08664
X2	1	1	12.5%	12.5%	0.00000	0	0.00000
X3	5	6	62.5%	75.0%	0.18232	0.125	0.02279
SUM	8	8	100%	100%		**IV =**	0.10943

Table 8.

If a column percentage, "b_k" or "g_k", is zero (a zero cell), then IV is undefined. Sometimes the modeler adds a small number, such as 0.1, to a zero cell to allow the calculation of IV. But this IV adjustment can drastically inflate the IV if the other b_k or g_k is comparatively large. The use of IV is not suitable for a predictor with more than one or two zero cells. If IV calculation is mandated, then preliminary binning of levels of X is needed to eliminate the zero cells.

The well-known book by Siddiqi (2017, p. 179) gives an interpretation of a predictor in terms of its IV. See Table 9.

IV Range	Interpretation
IV < 0.02	"Not Predictive"
IV in [0.02 to 0.1)	"Weak"
IV in [0.1 to 0.3)	"Medium"
IV ≥ 0.3	"Strong"

Table 9: Interpretation of IV values in terms of predictive power. Siddiqi (2017, p. 179)

An IV over 0.5 is regarded as suspicious by Siddiqi. The modeler should investigate whether the values of such a predictor are determined, at least partially, as a result of the coding of the target.

As a by-product of the IV calculation, the weight of evidence (WOE) transformation X_woe of X is obtained. (See the column in Table 8 with heading "Log(g_k/b_k) = X_woe".) Predictors that pass the IV screening might be entered into the logistic model using WOE coding in preference to dummy variable coding. These two alternatives are discussed further in a later section.

[16] The question might be asked, why not simply put numeric X in the logistic model. But the few levels of X often have unique meanings. Consider credit modeling and let X be number of bankruptcies. Then X=0 probably requires an indicator variable. Perhaps, X=1 has unique importance versus X=2, etc. Soon, the modeler may decide simply to use dummy variable coding.

[17] Nominal or Ordinal predictors with many levels (e.g. ZIP codes) require preprocessing, using Clustering, Decision Tree, or some other method. This topic is not discussed.

Another statistic that characterizes an ordinal or discrete NOD predictor X is the c-statistic. Let G_k = count of (Y=1) when X=x_k, and let B_k = count of (Y=0) when X= x_k. Unlike IV, the c-statistic is not a screener for X but is an indicator of the strength of a monotonic tendency between X=x_k and G_k / (G_k + B_k).

The calculation of the c-statistic follows these rules:

- IP ["Informative Pairs"] are pairs of observations (r, s) where Targets $Y_r \neq Y_s$
- If $Y_r > Y_s$ and $X_r > X_s$, then IP is "concordant"
- If $Y_r > Y_s$ and $X_r < X_s$, then IP is "discordant"
- Else the IP is a "tie"

Let C be the number of concordant pairs, T the number of ties, and IP the number of informative pairs. The c-statistic formula is below. The "max" transform makes 0.5 ≤ c-statistic ≤ 1.0.

c-statistic = max(1 - (C + 0.5*T) / IP, (C + 0.5*T) / IP).

Calculation of a c-statistic is illustrated in Table 10 with 4 rows. Predictor X has levels A and B with A < B.

ID	X	Y
#1	A	0
#2	B	0
#3	A	1
#4	B	1

IP's	c, d, or t
(#1, #3)	tie
(#1, #4)	concordant
(#2, #3)	discordant
(#2, #4)	tie

Table 10: The c-statistic for Table 10 equals 0.5

If X is discrete and there is a strong monotonic tendency, then X might be considered for direct entry into the logistic model (or, possibly, as a transformation of X). In the case of ordinal X with a strong monotonic tendency, a further investigation is needed to decide how to recode the ordinal levels of X to become numeric levels. Guidelines for converting an ordinal X to numeric are given in Lund (2019a).

A predictor X with c-statistic ≥ 0.65 merits consideration for use (after recoding, if ordinal) as a numeric predictor in the logistic model, perhaps after a transformation. The benefit of using X (or recoded X) directly in the logistic model versus dummy variable or WOE coding is reduction in degrees of freedom. (The d.f. associated with a WOE coded predictor is not well-defined but, arguably, exceeds 1. The determination of d.f. for X_woe is discussed in the following major section.)

An alternative to IV which does not have the problem of zero-cells is the "x-statistic".[18] Here are the steps to compute the x-statistic of X for target Y:

1. Let $P_k = G_k / (G_k + B_k)$ for k = 1 to J, but now re-label "k's" so that the P_k are non-decreasing.
2. In the algorithm for computing the c-statistic of X vs. Y, use the newly re-labelled P_k in place of X_k.
3. Now use the same formula to compute the x-statistic as for the c-statistic

Example:

In the table on the left for X vs. Y there are 7 concordance pairs and 6 ties.
The c-statistic = max(1 - (C + 0.5*T) / IP, (C + 0.5*T) / IP) = (7 + 0.5*6)/18 = 10/18

On the right, after re-labelling and using P_k vs. Y, there are 8 concordant pairs and 6 ties.
The x-statistic = max(1 - (C + 0.5*T) / IP, (C + 0.5*T) / IP) = (8 + 0.5*6)/18 = 11/18

c-statistic					x-statistic		
X	Counts				P_k (ascending)	Counts	
(ascending)	Y=0	Y=1	Memo: P_k		Re-labelled k	Y=0	Y=1
X1	2	1	1/3		1/4	3	1
X2	3	1	1/4		1/3	2	1
X3	1	1	1/2		1/2	1	1

[18] x-statistic is a name the author assigned to this statistic for the purpose of this paper. It does not appear elsewhere (except in some earlier papers by the author).

The “x-statistic” is exactly the “c” (or model c) from PROC LOGISTIC; CLASS X; MODEL Y=X;

The x-statistic serves as an alternative to IV (and has the benefit of being computable in the presence of zero-cells).

A relationship between the x-statistic and IV for the IV’s from Siddiqi’s Table 9 is given in Table 11. This relationship between 0.1 and 0.3 is essentially linear. There is, however, variation around the x-statistic values of approximately +/- 0.02. There are also exceptional values well outside this range. An x-statistic ≥ 0.64 indicates a very strong relationship between X and Y (perhaps non-monotonic).

IV	x-stat	approx. 95% CI
0.02	~ 0.55	+/- 0.02
0.1	~ 0.58	+/- 0.02
0.2	~ 0.61	+/- 0.02
0.3	~ 0.64	+/- 0.02

Table 11. Lund and Brotherton (2013)

The x-statistic and c-statistic are related as follows: (See Appendix, Topic 6 for a proof.)

x-statistic ≥ c-statistic

G_k / (G_k + B_k) is monotonic [19] in the ordering of X=x_k if and only if x-statistic = c-statistic

It was mentioned earlier that a discrete or ordinal predictor X with c-statistic ≥ 0.65 would support the use of X (after recoding, if ordinal)) as a numeric predictor in the logistic model. Now it is seen that the relationship of the c-statistic to the x-statistic is also relevant. If c-statistic ≥ 0.65 and x-statistic = c-statistic, then X is truly monotonic and the monotonicity is “steep” (although, not necessarily, strictly monotonic).

The IV, c-statistic, and x-statistic can be computed for multiple X’s in (essentially) one DATA Step. These statistics are computed by a macro %CUM_LOGIT_SCREEN_2.

MACRO CUM_LOGIT_SCREEN_2

The macro parameters are shown:

Dataset: Data set containing the Target and Predictors

Target: At least 2 levels (missing are ignored)

N_Input: Numeric Predictors (space delimited)

C_Input: Character Predictors (space delimited)

IV_Adj: YES, adds 0.1 to a zero cell to allow IV calculation

Miss: If YES, statistics for each INPUT variable are computed for its non-missing values. If not YES, then only "complete cases" are used (all variables are non-missing for observation to be used). Missing values for an INPUT must be re-coded to be used as true values in IV or x-stat calculations.

The macro name “cum_logit_screen_2” arises from its use for the cumulative logit model. Binary logistic is a special case. See Lund (2019b) for details. The macro is illustrated for data set SCREEN given below.

```
DATA SCREEN;
INPUT C1 $ C2 $ X1 X2 X3 Y @@;
DATALINES;
A AA 1 2 1 0 D CC 1 1 1 1 A CC 3 1 1 0 D XX 3 4 5 1 A XX 3 4 5 0 D XX 1 5 4 0
B AA 1 7 3 1 E CC 0 2 1 0 B AA 1 2 1 1 E AA 1 1 3 0 B CC 3 0 1 1 E XX 3 4 5 0
C XX 3 4 5 0 F XX 1 5 4 1 C AA 1 7 4 0 F CC 0 2 3 1 C XX 3 1 5 0 F XX 3 1 5 0
;
run;
%CUM_LOGIT_SCREEN_2(SCREEN, Y, X1 X2 X3, C1 C2, NO,  );
```

[19] Monotonic means that empirical G_k / (G_k + B_k) is non-decreasing vs. the levels of X or, alternatively, non-increasing.

Obs	Var Name	Levels	Character	Monotonic	C-Stat	X-Stat	IV (Info Value)
1	C1	6	YES		0.5584	0.9480	n/a
2	C2	3	YES		0.5974	0.6623	0.3801
3	X1	3	NO	YES	0.6298	0.6298	0.2853
4	X2	6	NO		0.5129	0.7077	n/a
5	X3	4	NO		0.6753	0.7142	0.6636

Table 12 (some columns omitted)

Comments: For this toy example, predictors C2, X3 have very strong IV while X1 is only of medium strength. Predictor X3, with c-stat of 0.6753 may be a good candidate for direct entry into the logistic model as a simple linear effect. Predictors C1 and X2 have one or more zero-cells and IV cannot be computed. No predictors would be "screened out" based on weak IV below 0.1

PROC HPBIN can compute IV for numeric predictors.[20] The c-statistic and x-statistic are not provided.

WEIGHT OF EVIDENCE, DUMMY VARIABLES, AND DEGREES OF FREEDOM

In Table 8 the weight of evidence transformation of predictor X was illustrated. It is given here as a formula. If X = x_k then X_woe = log(g_k / b_k) where g_k is the column percentage of Y = 1 in the k^{th} row, down the levels of X, and b_k is the column percentage of Y = 0 in the k^{th} row.

If either g_k or b_k is zero, then X_woe cannot be defined. (But a small number such as 0.1 might be assigned in the case of a zero.)

There is not a "tidy" formula for assigning values to X_woe during processing of the observations in a DATA step. Both the count of (Y=1) and count of (Y=0) must first be computed. Then the count of all occurrences of (Y=1) and (Y=0) must be made for each X = x_k.

Suppose X is ordered. If the relationship between X and X_woe is monotonic, then the relationship between X and G_k / (G_k + B_k) is also monotonic, and conversely.[21] Likewise, if there is a curvilinear relationship, such as quadratic, between X and X_woe, then this also holds for X and G_k / (G_k + B_k).

Weight of evidence (WOE) is an alternative to the use of dummy variable coding for entering a NOD predictor in a logistic model. WOE is widely used in credit risk modeling.[22]

Some advantages of WOE include:

- Modeler has control over the effect of X_woe in the logistic model. Provided X_woe is not part of an interaction with another predictor in the logistic model, then:

$$\text{Log}(\ p\ /\ (1-\ p)\ |\ X = x_k\ ) = \text{xbeta} = \widehat{\beta_{X\_woe}} * \text{Log}(g_k/b_k) + Z$$

  where:

  Z gives the terms in xbeta for the other predictors,
  The model probability, p is computed for X = x_k while other predictors are fixed,
  Log(g_k/b_k) is the weight of evidence for X = x_k.

[20] http://support.sas.com/documentation/cdl/en/prochp/67530/HTML/default/viewer.htm#prochp_hpbin_overview.htm. HPBIN is principally designed to perform "fine classing". Fine classing is preliminary binning of levels of numeric predictors, especially, continuous predictors. Fine classing would be followed by final binning which utilize an algorithm designed to maintain predictive power as the number of bins is decreased. Binning is discussed in a later section.

[21] $\text{Log}\ (g_k/b_k) \leq \log(g_{k+1}/b_{k+1}) \Rightarrow \text{Log}\ (G_k/B_k) - \text{Log}(G/B) \leq \log(G_{k+1}/B_{k+1}) - \text{Log}(G/B)$
$\Rightarrow \text{Log}\ (G_k/B_k) \leq \log(G_{k+1}/B_{k+1})$
$\Rightarrow G_k/B_k \leq G_{k+1}/B_{k+1} \Rightarrow G_kB_{k+1} \leq G_{k+1}+B_k \Rightarrow G_kB_{k+1} + G_kG_{k+1} \leq G_{k+1}B_k + G_kG_{k+1}$
$\Rightarrow G_k/(G_k+B_k) \leq G_{k+1}/(G_{k+1}+B_{k+1})$

[22] For example, see: Siddiqi (2017), Finlay (2010)

The log-odds of p for X = x_k, with other predictors held fixed, is linearly related to the weight of evidence of X.

If X_woe is monotonic with respect to X, then there is a monotonic relationship between X and Log(p / (1-p)), where p is computed for X = x_k with other predictors held fixed.

In the case of dummy variable coding with J-1 fitted coefficients, the effect of X on Log(p / (1-p)) is given through $\sum_{j=i}^{J-1} \hat{c}_j * (X = xj)$ with other predictors held fixed. Here, $\hat{c}_j$ is the coefficient of X = x_j with reference level coding, across the first J-1 levels of X.

If the relationship in the model between X and X_woe is monotonic, the dummy variable relationship between X and $\sum_{j=i}^{J-1} \hat{c}_j * (X = xj)$ need not be monotonic.

- Fewer parameters are added to the logistic model. If X has J levels, then dummy coding adds J-1 parameters versus only 1 for WOE.
- X_woe is numeric and can be compared with other numeric predictors to assess collinearity.

A disadvantage of WOE is that the degrees of freedom for X_woe, when entered into a logistic model, are generally unknown. Here is an explanation.

Let X have J > 2 levels. Then the following two models are the same (i.e. produce the same probabilities):

(A) PROC LOGISTIC DESCENDING; CLASS X; MODEL Y=X;

(B) PROC LOGISTIC DESCENDING; MODEL Y=X_woe;

Model (A) uses J-1 degrees of freedom. Therefore, Model (B) must also use J-1 d.f.

Suppose numeric predictors W and Z are added to models (A) and (B) to form models (A2) and (B2):

(A2) PROC LOGISTIC DESCENDING; CLASS X; MODEL Y=X W Z;

(B2) PROC LOGISTIC DESCENDING; MODEL Y=X_woe W Z;

Then $\text{Log}(L)_{A2} \geq \text{Log}(L)_{B2}$. The degrees of freedom for Model A2 equals J-1 + 2. The degrees of freedom for Model B2 are undetermined but lie between 3 and J-1 + 2.

It is difficult to accept that the d.f. of Model B2 would only be 3 after noting that nested Model B has J-1 d.f. On the other hand, to fully load Model B2 with J-1 + 2 d.f. seems wrong where, as in the usual case, $\text{Log}(L)_{A2} > \text{Log}(L)_{B2}$. But if one is forced to choose, the choice of J-1 + 2 is the more conservative.

If X_woe is uncorrelated with W and Z, then, based on observing a few examples, $\text{Log}(L)_{A2} = \text{Log}(L)_{B2}$. Here, definitely, X_woe should be assigned J-1 d.f.

The d.f. assignment is important when considering confidence intervals and hypothesis tests of the parameter for X_woe as well as for use of p-values in predictor selection methods (e.g. stepwise p-value based) or in model comparisons and predictor selection methods based on SBC and AIC (as provided by PROC HPLOGISTIC).

None of the SAS procedures allow for d.f. adjustment for WOE predictors. Of course, more fundamentally, it is unclear how to make such an adjustment.

I have not seen a discussion of d.f. for WOE predictors and I assume that, in practice, X_woe is simply regarded as having 1 d.f. in conformance with its usage in PROC LOGISTIC and PROC HPLOGISTIC.

In situations with many predictors and with at least moderate collinearity, the 1 d.f. assignment may be a reasonable simplifying assumption.

In a later section there is a further discussion of the usage of WOE vs. dummies in the fitting of a logistic model.

BINNING

Binning is the process of reducing the number of levels of a NOD predictor by collapsing of levels. The goal is to achieve parsimony while preserving, as much as possible, the predictive power of the predictor. Binning should routinely be considered if the number of levels of the NOD predictor is 6 or more. The "binning solution" for a predictor X is simply the final configuration of bins (number of bins and membership of levels of X in the bins) decided upon by the modeler. This can be a somewhat subjective decision.

Adjacent vs. Non-Adjacent Binning:

Suppose X has levels A, B, C, D with alphabetical ordering. The first step in binning might be to combine A and C. Now the binned X has levels {A, C}, B, D. Here, "*non-adjacent*" binning was allowed (A and C are not adjacent)

If only *adjacent* levels may be combined, then {A, B}, C, D is an allowable first step. Then the second step would be either {A, B}, {C, D} or {A, B, C}, D.

Modeler chooses *non-adjacent* or *adjacent* depending on meaning of X.

Objectives:

A first objective is to maintain the IV of the final solution at a level near the starting IV.

The second objective: Binning should eliminate (through combining with other levels) a level with an overall low frequency count or any level having only a few events (or non-events).

A third objective, possibly at odds with the first objective, is to find a business or scientific relationship between a binning solution and X_woe. If X is ordered, then the relationship may be monotonicity.

Algorithms and Macros

See Lund (2017) for an in-depth discussion of binning as well as a description of two SAS macros. One macro, %NOD_BIN, performs non-adjacent binning (and adjacent binning) and the other, %ORDINAL_BIN, performs adjacent binning only. An example is given below of a summary report from %NOD_BIN.

Example of %NOD_BIN Summary Report

%NOD_BIN is applied to data set LEVEL_8 with Target Y and Predictor X, with 8 levels 1, …, 8.

```
DATA LEVEL_8;
do X = 1 to 8;
   Y = 1;
   W = floor(100*ranuni(1)) + 1;
   output;
   Y = 0;
   W = W + floor(200*ranuni(1));
   output;
   end;
run;
```

In the example of %NOD_BIN, non-adjacent binning is performed (despite the fact that X is numeric). Table 13 gives the summary report.

At step k=7, levels {2} and {6} were binned. The loss in IV is not noticeable when going from k=8 and k=7. The binning process continues but should stop no later than k=4 for the following reasons:

(1) IV dropped noticeably from k=4 to k=3 (0.3199 vs. 0.3076)

(2) The chi-square (Pr > ChiSq) column tests whether the dummy variable coefficients, corresponding to the two levels that are collapsed, are equal. For k=3 the bins {2, 6} and {3, 4, 5} are collapsed. But in a logistic model the dummy variables corresponding to these two bins are statistically unequal (Pr > ChiSq of 0.0422). Collapsing should only be performed when two bins are statistically equal.

But, an alternative argument can be made for stopping at k=5 based on IV and the chi-square test.

k	IV	X_ stat	Nested ChiSq	Pr > ChiSq	L1	L2	L3	L4	L5	L6	L7	L8
8	0.3287	0.642	N/M	N/M	1	2	3	4	5	6	7	8
7	0.3287	0.642	0.000	0.9898	1	2+6	3	4	5	7	8	
6	0.3285	0.641	0.063	0.8021	1	2+6	3+4	5	7	8		
5	0.3282	0.641	0.120	0.7289	1	2+6	3+4+5	7	8			
4	0.3199	0.637	2.969	0.0849	1	2+6	3+4+5	7+8				
3	0.3076	0.625	4.128	0.0422	1	2+6+3+4+5	7+8					
2	0.2232	0.568	29.839	0.0000	1	2+6+3+4+5+7+8						

Table 13.

At each step in the binning process the algorithm for %NOD_BIN finds the two bins to collapse which maximizes IV after the collapse versus all the other possible pairs to collapse. It is possible that this algorithm leads to a non-optimal final k-bin solution. This is because a sub-optimal collapse at an earlier step could lead to a better k-bin solution. %NOD_BIN does not detect such as outcome. As a practical matter, I conjecture that suboptimal solutions are uncommon in the running of %NOD_BIN or that loss in IV of a suboptimal solution would not be material.[23]

In contrast, %ORDINAL_BIN performs only adjacent binning. It is guaranteed to find the best solution for each k. That is, if k=8, then %ORDINAL_BIN finds the 7-bin solution with maximum IV, and likewise for k = 6, …, 2. %ORDINAL_BIN also finds the best k-bin monotonic solution, if such solutions exists for a given k. The algorithm for %ORDINAL_BIN is the complete enumeration of all solutions.

Although not designed to perform binning, PROC HPSLPIT might be used as an alternative for both adjacent and non-adjacent binning.[24]

SCREENING AND TRANSFORMING CONTINUOUS PREDICTORS

A continuous predictor X is numeric with many levels. Examples of X include measurements of distance in miles, money in dollars, time in minutes. While X may be entered into the logistic model, a better usage of X may be: Log(X), 1/X, or two variable combinations of X such as X and X^2, or Log(X) and $X^{-0.5}$.

The Function Selection Procedure (FSP) provides a method to (i) select a very good transformation of X to use in the logistic model or (ii) recommend that X be dropped from consideration.

FSP was developed in the 1990's for bio-medical applications. P. Royston and W. Sauerbrei were among the principal developers. They discuss FSP in great detail in their book *Multivariate Model-building* (2008).

A macro %FSP_8LR for applying FSP to process multiple predictors in a single macro call was given by Lund (2016, 2018). In Lund (2018) the FSP is applied to the cumulative logit model, with binary logistic as a special case. See Royston and Sauerbrei (2008), Lund (2016, 2018) for the FSP methodology.

[23] For this data set %NOD_BIN becomes IV-suboptimal for non-adjacent binning at k=2. The 2-bin solution from %NOD_BIN is {01, 02, 03} and {04, 05, 06, 07, 08, 09, 10, 11, 12}. The optimal IV solution is {01, 02, 03, 04} and {05, 06, 07, 08, 09, 10, 11, 12}. %NOD_BIN found the optimal 3-bin solution {01, 02, 03} {04, 05} {06, 07, 08, 09, 10, 11, 12} but is unable to split up {04, 05} to reach the optimal 2-bin solution.

	X												
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

[24] HPSPLIT is a decision tree procedure. It performs splitting of the sample in a manner that is the reverse of binning. The terminal nodes (the leaf's) provide binning solutions. I have not been able to work with the data set that is output by HPSPLIT to create a usable binning report. Aside, from this difficulty, the decision tree approach can also accomplish binning.

Below, an example is given of applying %FSP_8LR to a dataset with target Y and two predictors, X and Rannorx. First, the logistic model population is generated by simulation.

```
%LET SLOPE1 = 0.2;
%LET SLOPE2 = -0.5;
DATA FSP_Example;
   do i = 1 to 8000;
      call streaminit(1);
      c = rand("Uniform");
      Rannorx = rand("Normal");
      e = 1*log(c / (1-c));
      X = mod(i,16) + 1;
      Z = 0 + &SLOPE1*LOG(X) + &SLOPE2*(1/X) + e;
      Y = (Z > 0);
      output;
      end;
run;
```

The SAS code created a predictor X which is transformed by T(x) = 0.2*Log(x) - 0.5*(1/x). There is also a logistic random error that is added to T(x). This gives T(x) + error which is used in producing Y. Rannorx is a standard normal that is unrelated to Y. The macro call is:

```
%FSP_8LR(FSP_Example, Y, X Rannorx, NO, A,  );
```

%FSP_8LR parameters are: DATASET, TARGET, List of predictors with separation by space, VERBOSE = YES | NO for more reports, A or D for modeling ascending or descending TARGET, and finally a WEIGHT parameter for use in the WEIGHT statement in PROC LOGISTIC. In this example, there is a space to indicate that the WEIGHT parameter is not being used.

There are three steps to reach a final decision regarding the screening and transforming of X and of Rannorx. Each step involves a decision These decisions are shown in the column with heading "Decision" in the FSP_8LR summary report. See Table 14.

If p-value exceeds 0.05 (or modeler's favorite α), then STOP and accept the "decision". First consider X.

- In the first row, the p-value = 0. The process moves to the next row.
- In the second row, the p-value = 0. Use of X (linear) is rejected. The process moves to the next row.
- In the third row, the p-value is 0.052. A borderline situation. The process might STOP with the decision to use $X^{-0.5}$ as the transform. Otherwise, it moves to the next row.
- Here, both X^{-2} and X^{-2} * log(X) (= transform1 * log) are selected for the model.

FSP recommends that Rannorx be dropped (p-value = 0.15 in the first row).

Summary Report (some columns are omitted)								
Predictor	Off-set (i)	Decision	-2log(L)	Test Stat (ii)	df	p Value (iii)	trans1 (iv)	trans2 (v)
X	0	Drop	10962.5	105.05	4	0		
X	0	Linear	10894.6	37.11	3	0	Linear	
X	0	FP1	10863.4	5.93	2	0.052	p=-0.5	
X	0	FP2	10857.5				p=-2	transform1*log
Rannorx	4.6583	Drop	10962.5	6.74	4	0.150		
Rannorx	4.6583	Linear	10957.1	1.30	3	0.729	Linear	
Rannorx	4.6583	FP1	10956.6	0.83	2	0.659	p=-0.5	
Rannorx	4.6583	FP2	10955.8				p=3	transform1*log

Table 14

Table 14 notes: See reference to column headings:

(i) Off-Set is 4.6583 for Rannorx. This is the amount added to predictor Rannorx so that min(Rannorx + 4.6583) = 1. The FSP transformations of Rannorx are applied to the translated Rannorx. (But this is moot since Rannorx was dropped.)

(ii) Test Stat: Difference of -2*Log(L) for this row and -2*Log(L) for FSP2.

Test Stat is approximately a chi-square with d.f. as shown.

(iii) P Value: Right tail p-value for Test Stat.

(iv) Trans1: "p" is the power (exponent) for transformed X. e.g. p=-0.5 denotes $X^{-0.5}$

(v) Trans2: "transform1*log" is the transform of X which is X^{-2} * Log(X). Note that "trans1" is X^{-2}.

In Figure 1 the fitted xbetas for X, FSP1, and FSP2 are plotted versus the average Log Odds of Y at the 16 levels of X. (The coefficients of the transforms were determined during the running of %FSP_8LR but were omitted from Table 14.)

```
LINEAR = -0.08789 + 0.04046*X;
FP1 =  0.73628 - 1.15458*X**(-0.5);
FP2 = 0.45942 - 0.80225*X**(-2) - 2.81401*X**(-2)*Log(X);
```

Both FP1 and FP2 do a good job of tracking Log-Odds, but FP2 is better on the left end of the chart. Clearly, the use of an un-transformed X would fail to track Log-Odds of Y.

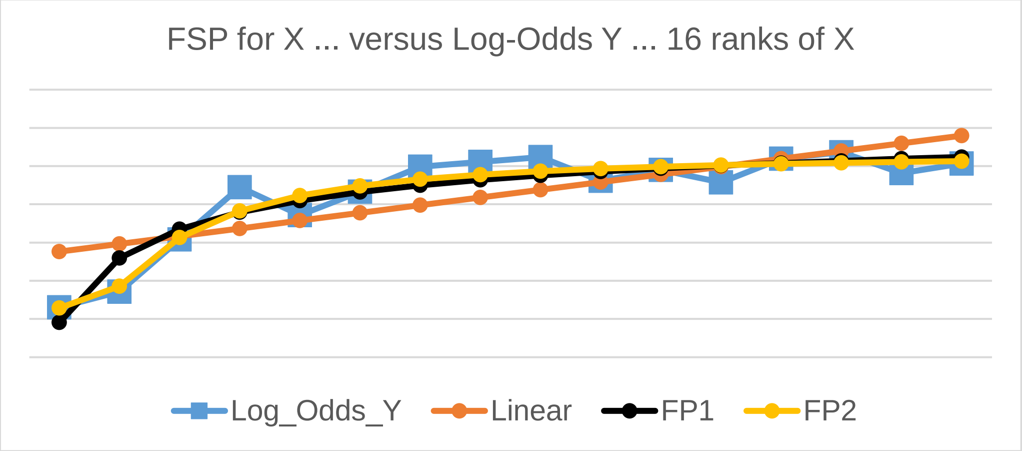

Figure 1 (created in Excel)

MULTICOLLINEARITY AND PREDICTOR REDUCTION

There are explicit formulas which show the effect of multicollinearity on standard errors of predictors in least squares estimation for linear regression (e.g. see O'Brien, (2007)). High collinearity clearly inflates the standard errors.

For logistic regression the $\hat{\beta}_j$ are estimated by an iterative algorithm. There is no possibility for a closed-form formula to show the effect of multicollinearity. Instead, to illustrate the effect of multicollinearity in logistic models, several examples are given in ALR, p. 149. Additionally, simulations are given in Appendix, Topic 7 to show the effect of multicollinearity on the standard errors of predictor estimates.

Examples from ALR and the discussion in the Appendix support the following points:

- Multicollinearity inflates the standard errors of predictor estimates. Since standard errors enter the denominator of test statistics and as a multiplier in confidence interval end-points, a significant predictor may appear as insignificant and the length of a confidence interval may be inflated.
- Predictor selection methods using p-values (discussed in a later section) are distorted, resulting in suboptimal models.

- Large sample size overcomes the impact of multicollinearity. This is simply a restatement of the effect of sample size on p-values. Insignificant predictors, whether involved in collinearity or not, become significant with increasing sample size.
- Predictor coefficient estimates are not biased. The model can function as a predictive model.

Predictor variable selection methods that do not depend on p-values (discussed in later sections) can eliminate collinearity by not selecting or by removing collinear predictors during predictor selection.

Except for designed experiments, some correlation is inevitable. Only blatant correlation needs to be removed before beginning the model fitting. For more discussion, see O'Brien (2016).

Correlations of numeric predictors are calculated by PROC CORR. A high correlation, perhaps greater than |0.90| between two predictors, should be resolved by removing one of the predictors.

But it could happen that there are K predictors with no two having correlation above |0.90| but that one predictor is highly correlated to a linear combination of the other K-1. To detect this occurrence the variance inflation factor (VIF) of each predictor can be computed. The definition of VIF for predictor X_1 versus X_2 - X_K is given by:

$\mathrm{VIF}(X_1) = 1 / (1 - R^2)$ where R^2 is the r-square of the least squares linear regression of X_1 against X_2 -X_K

There are at least three issues with using VIF:

- The handling of classification variables: If classification variable X has three levels X1, X2 and X3 with 54% for X1, 44% for X2, and only 2% for X3 (any n), then X1 and X2, after conversion to dummy variables, have correlation of -0.96. But this is not true correlation. This example and other considerations show that the usage of dummy variables in VIF introduces confusion and may be lead to bad decisions. An alternative to dummies is weight of evidence coding.
- The VIFs might be used to rank the variables from high (bad) to low, but there is no guidance from the target. Perhaps the x-statistic could be added in some manner to the VIF rankings to decide on elimination of predictors. This thought, however, requires a more complete process definition
- The VIF process should be iterative. Once a variable is removed, the VIF's of the remaining variables should be recomputed. Such an iterative process is unwieldy and time consuming when there are many predictors.

A second approach is the usage of PROC VARCLUS. There are issues with the VARCLUS approach.

- As with VIF, this method does not apply to classification variables. The conversion to dummy variables creates confusion and may be lead to bad decisions. Again, WOE coding is an alternative.
- VARCLUS determines the number of clusters (subsets) of predictors, unless the modeler has insights that would override the VARCLUS process. The appropriateness of the cluster definitions are critical because the modeler, using subject matter expertise, must select one or several predictors from each cluster to "represent" that cluster in the model.
- The selection of predictors from the clusters is not guided by the target.

An alternative approach to reduction of collinear (numeric) predictors is outlined below and the SAS code for implementation is given in Appendix, Topic 8. The data set Correlation_Data with predictors X1-X5 appears in Appendix, Topic 8 and will used in the discussion below.

Step 1: Classification variables are converted to WOE. But none of X1-X5 is a classification variable.[25]

Step 2: The x-statistic (or alternatively the IV) is computed for each predictor. This may be done by the macro %CUM_LOGIT_SCREEN_2. These x-statistics rank the predictors.

Step 3: PROC CORR computes correlations. For a pair of predictors with high absolute correlation, the predictor with lower x-statistic (or IV) is eliminated. Once a predictor is eliminated, it is no longer used in correlation comparisons with lower ranked predictors.

[25] The author has a method and rough SAS code that detects dependence among classification predictors. No conversion to numeric predictors is required when using this method.

The x-statistic, most likely, will be required for this process because of excessive zero cells when computing IV for predictors with many levels.

Table 15 shows Step 2 after applying %CUM_LOGIT_SCREEN_2 to X1-X5.

Obs	Var Name	Levels	Character	Monotonic	C-Stat	X-Stat	IV (Info Value)
1	X2	13	NO		0.6500	0.6625	0.3519
2	X1	10	NO		0.6500	0.6593	0.3767
3	X3	13	NO		0.6334	0.6381	n/a
4	X4	20	NO		0.5116	0.6013	0.1622
5	X5	5	NO		0.5258	0.5402	0.0227

Table 15

The correlation cutoff was selected as |0.80| for the purpose of this illustration. A higher cutoff of |0.90| might normally be considered.

Here is the report from Step 3. In the first row of the correlation report, predictor X1 is eliminated since it has correlation 0.818 with higher ranked predictor X2. The second line of the correlation report giving correlations with X1 is deleted. Predictor X3 will be kept since it is not compared with X1. Finally, predictors X4 and X5 are kept.

Pearson Correlation Coefficients, N = 1000				
	X1	X3	X4	X5
X2	0.818	0.706	-0.004	0.012
~~X1~~		~~0.857~~	~~0.002~~	~~0.003~~
X3			0.032	-0.001
X4				-0.034
X5				

Variables to be removed due to correlation
Correlation max = .80

Obs	removed	step	correlation
1	X1	1	0.818

Table 16.

Once blatant collinearity has been removed, then predictor selection methods during model fitting can avoid adding strongly collinear predictors to the model.

In the section "predictor selection methods" a method is discussed where predictors are added (or removed) that so that Schwarz-Bayes criterion (SBC) is minimized at each step. Then, as will be explained below, the use of SBC chooses the best model (according to SBC theory), and also tends to avoid the problem of collinear predictors.

PREDICTOR SELECTION METHODS

"*Purposeful selection*" of effects (predictors) for a model occurs when modeler considers the theoretical relationship between X and Y (as in bio-statistics). The modeler uses the theory to help construct the model. In contrast, the "*automated selection"* of predictors for fitting a model uses a predictor selection technique (for example, stepwise using p-value thresholds for entry and removal). The modeler only chooses the method of automated selection.

In this section the focus is on automated selection. Earlier in the paper, the samples sizes were restricted to medium / large. Now we should regard the sample as large or very large, as encountered in data mining.

Three SAS procedures that do automated selection for logistic models will be discussed

- PROC LOGISTIC
- PROC HPLOGISTIC
- PROC HPGENSELECT

PROC LOGISTIC METHODS FOR PREDICTION SELECTION

PROC LOGISTIC has the following predictor selection methods: FORWARD, BACKWARD, BACKWARD FAST, and STEPWISE with each requiring the modeler to specify a p-value for entry, removal, or both. Selection of predictors using p-values have been criticized on several grounds. Harrell (2001, p.56-60) summarized these criticisms in the case of least squares linear regression.

PROC LOGISTIC also provides selection method SCORE (best subsets). Comments related to SCORE are given in a later section.

PROC HPLOGISTIC METHODS FOR PREDICTION SELECTION

PROC HPLOGISTIC provides predictor selection methods that do not use p-values. One such method minimizes the Schwarz-Bayes criterion (SBC) at each step as a predictor enters or exits the model. Here is the formula for SBC (also called BIC): [26]

$$\text{SBC} = -2*\text{Log}(L) + \log(n)*(K+1)$$

where n=sample size, K=number of parameters (or d.f.) in the model excluding the intercept.[27]

There is complex theory, both conceptually and mathematically, that says that a model with smaller SBC is preferred to one with larger SBC. See Cavanaugh (2012a) for a PowerPoint presentation of this topic. Here is an implementation of predictor selection using SBC. The relevant statement is “SELECT=SBC”.

```
PROC HPLOGISTIC DATA = <dataset>;
CLASS <some of the predictors>;
MODEL Y (descending) = <predictors>;
SELECTION METHOD=FORWARD (SELECT=SBC CHOOSE=SBC STOP=SBC);
```

FORWARD is used to illustrate SELECT=SBC but, just as well, BACKWARD or STEPWISE could be used. With SELECT=SBC at each forward step, the predictor X is entered that gives best (smallest) SBC for the model at that step.

CHOOSE=SBC finds the model, along FORWARD path, giving smallest SBC.

It may be that after a few steps, the decrease in -2*Log(L) is lessened as dominant predictors have already been entered into the model. Meanwhile log(n)*(K+1) grows by log(n) at each step (assuming the predictor has 1 d.f.). For this reason, for large “n”, the CHOOSE=SBC tends to choose a model with “fewer” rather than “many” X’s.[28]

Coefficients and model statistics from HPLOGISTIC are reported for this chosen model (CHOOSE=SBC). No p-value guesses are required.

AKAIKE INFORMATION CRITERION (AIC)

An alternative to SBC is AIC where AIC = -2*Log(L) + 2*(K+1).[29] The multiplier of K+1 in AIC is 2 instead of log(n) as in SBC. There is also a complex theory, both conceptually and mathematically, that says that a model with smaller AIC is preferred to one with larger AIC. See Cavanaugh (2012b) for a PowerPoint presentation of this topic.

PROC HPLOGISTIC offers SELECT=AIC and CHOOSE=AIC. Additionally, SBC and AIC could be mixed, as in SELECT=AIC and CHOOSE=SBC.

[26] Schwarz, G. (1978) Estimating the Dimension of a Model, *Annals of Statistics*, **6**, (2): 461-464.

[27] Note that SBC is not well defined when the predictors include one or more with WOE coding.

[28] It is, however, possible that CHOOSE=SBC becomes too stringent in terms of allowing predictors into the model. A statistician at the University of Michigan once commented that “SBC was the Grinch that stole the model”.

[29] Akaike, H. (1974), A new look at the statistical model identification, *IEEE Transactions on Automatic Control*, **19** (6): 716–23

For large samples, SBC is much less likely to make a type I error by adding an insignificant predictor, under FORWARD, than is AIC. Of course, the price for low type I error is the risk that a significant predictor will be overlooked.

Here is an example of the logic of adding a predictor to a model where n=10,000, when employing, first SBC, and then AIC. It is assumed that X has 1 d.f.

- X is added if: $\text{SBC}_{(\text{after X})}$ < $\text{SBC}_{(\text{before X})}$
- Equivalently, add X if: $-2*\text{Log}(L)_{K+1} + \log(n)*(K+1)$ **<** $-2*\text{Log}(L)_{K} + \log(n)*K$
- Equivalently, add X if: $-2*\text{Log}(L)_{K} - (-2*\text{Log}(L)_{K+1})$ **>** log(n)
- Let $D = -2*\text{Log}(L)_{K} - (-2*\text{Log}(L)_{K+1})$
- If n= 10,000, then log(10,000) = 9.2
- Therefore …. add X if D > 9.2
- D is a Chi-Sq with 1 d.f. under H0: $\beta_X = 0$
- $\text{Prob}(D > 9.2 \mid \beta_X = 0) = 0.24\% = 0.0024$
- 0.24% is type I error probability … with n=10,000 it is very unlikely that insignificant X is added.

The same sequence of calculations lead to a type I error probability under AIC of 15.7%. (Replace log(10,000) in the above calculations with 2.)

PARTITION STATEMENT AND CHOOSE=VALIDATE IN PROC HPLOGISTIC

The PARTITION statement together with CHOOSE=VALIDATE provides another attractive feature for fitting and validating logistic models.

If "MyData" is the modeler's data set for logistic modeling, it can be partitioned by HPLOGISTIC into Training, Validation, Test by either of two methods:

Method 1: ROLEVAR. The modeler has a variable in Mydata called "Part" with levels 1, 2, 3 (the variable name and three levels are immaterial). Using Part the observations from Mydata are assigned to TRAIN, VALIDATE, and TEST. The values of Part are enclosed in quotes whether or not Part is numeric.

```
PROC HPLOGISTIC DATA= Mydata;
PARTITION ROLEVAR= PART (TRAIN="1" VALIDATE="2" TEST="3");
MODEL Y (descending) = <vars>;
SELECTION METHOD=FORWARD (SELECT=SBC CHOOSE=VALIDATE STOP=NONE);
```

The logistic model predictors are selected using FORWARD and SELECT=SBC on TRAIN. Then at each FORWARD step the VALIDATION sample is scored. The model step with the smallest average squared error on VALIDATION is the chosen model. Average squared error (ASE) = $\sum_{i=1}^{n}(p_i - y_i)^2$ / n.

At the chosen step TEST is also scored. Fit statistics are reported on each of TRAIN, VALIDATE, TEST.

But VALIDATE not available as a SELECT option. The computational burden would be significant. For example, consider FORWARD and 100 predictors. To select the predictor for the first step, the validation dataset would have to be scored 100 times, once for each of the 100 variables, in order to select the first predictor.

Method 2: The ROLEVAR statement is removed and is replaced by:

```
PARTITION FRACTION(VALIDATE=.2 TEST=.1 SEED=1);
```

This statement causes HPLOGISTIC to randomly assign 20% to VALIDATE, 10% to TEST, and 70% to TRAIN according to SEED=1.

Fit Statistics are computed for Training, Validation, Test.

A PROPOSAL FOR USING SELECT=SBC AND ENDING WITH WOE PREDICTORS

In this section the quandary is considered of how to include WOE predictors in predictor selection when fitting a logistic model. The problem was the inability to assign d.f. to WOE predictors.

Here are the steps of a proposal:

- Utilize CLASS predictors (instead of WOE) with SELECT=SBC or SELECT=AIC (here, there is proper d.f. counting)
- Obtain the best model using CHOOSE=VALIDATE
- Refit the chosen model with WOE variables in place of CLASS
- Compare CLASS and WOE MODEL fits by using statistics such as: -2*Log(L), Model c, ASE.
- If the loss in fit is unnoticeable or immaterial, then the use of the WOE model is supported.

An example of these steps is given next. First, is the coding of the example data set.

```
DATA Dummy_v_WOE;
   do I = 1 to 3000;
      random1 = ranuni(1);
      random2 = ranuni(1);
      random3 = ranuni(1);
      random4 = ranuni(1);
      random5 = ranuni(1);
      X1 = floor(2*random1); /* 2 levels */
      X2 = floor(3*(random1 + random2)/2);
      X3 = floor(3*(0.2*random2 + random3 + 1.8*random5)/3);
      X4 = floor(4*(0.1*random1 + 1.9*random4 + 2.0*random5)/4);
      xbeta = -.20 + .1*X1 + .1*X2 + .15*X3 + .15*X4 + rannor(1);
      Y = ( xbeta > 0);
      output;
      end;
run;
```

The logistic model is fit with the classification variable versions of X2-X4 (X1 is binary).

```
PROC HPLOGISTIC DATA= Dummy_v_WOE;
PARTITION FRACTION(VALIDATE=.3 TEST=.3 SEED=1);
CLASS X2 X3 X4;
MODEL Y (descending)= X1 X2 X3 X4;
SELECTION METHOD=FORWARD (SELECT=SBC CHOOSE=VALIDATE STOP=NONE);
run;
```

The smallest ASE on the validation data set occurred at step 4. The chosen logistic model includes all four predictors X1-X4.

Selection Summary (class)				
Step	Effect	Number	SBC	Validation
0	Intercept	1	1371.84	0.2361
1	X4	2	1370.95*	0.2296
2	X1	3	1374.35	0.2279
3	X3	4	1386.21	0.2267
4	X2	5	1399.78	**0.2264***

Table 17

Now the selected predictors X2-X4 are converted to weight of evidence and, along with X1, are fitted to a logistic model. The weight of evidence code appears below. It was generated by running %NOD_BIN (see Lund (2017) for details).

```
Data Dummy_v_WOE_2; SET Dummy_v_WOE;
if X2 in ( 0 ) then X2_woe = -0.202853452 ;
if X2 in ( 1 ) then X2_woe = -0.021399128 ;
if X2 in ( 2 ) then X2_woe = 0.2459859434 ;
if X3 in ( 0 ) then X3_woe = -0.284533229 ;
if X3 in ( 1 ) then X3_woe = -0.038123571 ;
if X3 in ( 2 ) then X3_woe = 0.4104888684 ;
if X4 in ( 0 ) then X4_woe = -0.513885048 ;
if X4 in ( 1 ) then X4_woe = -0.192588173 ;
if X4 in ( 2 ) then X4_woe = 0.202062836 ;
if X4 in ( 3 ) then X4_woe = 0.5768195603 ;
run;
```

Finally, HPLOGISTIC is used to fit X1, X2_woe, X3_woe, X4_woe to a logistic model.

```
/* Refit model found in CLASS Model … no selections */
PROC HPLOGISTIC DATA= Dummy_v_WOE_2;
PARTITION FRACTION(VALIDATE=.3 TEST=.3 SEED=1);
MODEL Y (descending) = X1 X2_woe X3_woe X4_woe;
run;
```

The results of the CLASS MODEL and the WOE MODEL are shown in Table 18. For the TEST data set, the only data set not compromised by training and validating, the WOE MODEL performs as well (slightly better, in fact) than the CLASS MODEL. In this example (which, however, is not a basis for generalization), the WOE recoding can replace dummy variable coding in the final model without loss of predictive performance.

CLASS MODEL: Fit Statistics				WOE MODEL: Fit Statistics			
Description	Training	Validation	Testing	Description	Training	Validation	Testing
-2 Log Like	1588.49	1143.43	1169.77	-2 Log	1590.79	1142.65	1163.51
Model c	0.5977	0.6192	0.5777	Model c	0.599	0.6238	0.5833
ASE	0.2305	0.2274	0.2290	ASE	0.2309	0.2272	0.2274
Many More Statistics				Many More Statistics			

Table 18

DESCRIPTION OF LASSO METHOD FOR LOGISTIC MODELING

PROC HPLOGISTIC in release 15.1 does not support selection by LASSO (least absolute shrinkage and selection operator) as a SELECTION method but LASSO is available in HPGENSELECT.

Here is an outline of the LASSO process by HPGENSELECT for a logistic model.

First a positive value of λ is specified. Then LASSO fits a logistic model by solving for the coefficients in the objective function shown here:

$$F(b_0,\ \underline{b},\ \lambda) = \min_{(b0,\ b)} \{ -LL(b_0,\ \underline{\mathbf{b}})\ +\ \lambda * \textstyle\sum_{k=1}^{K} |b_k| \}$$

where LL denotes log likelihood, b_0 is the intercept, and $\underline{\mathbf{b}}$ = b_1, … b_K are the other coefficients.

Each λ gives rise to a model. Different λ's may be associated with models with the same predictors but with different values for their coefficients.

Let "maxλ" be the smallest λ where F(b_0, $\underline{b}$, λ) = -LL(b_0, 0, …, 0). As λ decreases from maxλ and approaches 0, predictors enter or exit the model 1-at-a-time or in small groups. (More accurately, the b's become non-zero or become zero.) At λ = 0 the LASSO model is the usual maximum likelihood estimation (MLE) model which would be found by HPLOGISTIC.

A sequence of λ's must be specified where the objective function will be evaluated. LASSORHO is an HPGENSELECT parameter (between 0 and 1) that starts this generation of λ values. The default for LASSORHO is 0.8.

The first λ in the sequence is LASSORHO * (maxλ)

The jth λ in the sequence is LASSORHO ^{j} * (maxλ)

The number of lambda steps is given by LASSOSTEPS with default = 20 … λ_1, λ_2, …, λ_{20}. Using a "CHOOSE" criterion, a final model is selection from among the models found at the LASSOSTEPS.

In order to simplify this outline of the LASSO algorithm, the preceding formulation does not include the case of classification variables.[30] The LASSO implementation in HPGENSELECT is actually "group lasso" which has a more general objective function than given above. In group lasso an entire classification variable enters or exits the model. Individual levels of the classification variable cannot enter or exit.[31]

Here are HPGENSELECT statements for selection of predictors by LASSO with SBC as the CHOOSE criterion.

```
PROC HPGENSELECT Data= <>
LASSORHO=.80 /* default */ LASSOSTEPS=20; /* default */
CLASS /*<some of the Var's>*/;
MODEL Y (descending) = /*<Var's>*/
/ DISTRIBUTION= BINARY; /*<= logistic*/
   SELECTION METHOD=LASSO (CHOOSE=SBC STOP=SBC) DETAILS=ALL;
```

From among the λ_1, λ_2, …, $\lambda_{20,}$ the λ that minimizes SBC gives the chosen logistic model.

If the same predictors are in a LASSO model and a MLE model, then $\hat{\beta}$'s and LL are not the same because different objective functions are being solved.

It is important to select LASSORHO and LASSOSTEPS so that the CHOOSE criterion finds a model that is globally optimal and not an end-point optimum. For example, if CHOOSE=SBC is selected with LASSOSTEPS=20 and the chosen model occurs at λ_{20}, then the true optimum occurs at a point beyond λ_{20}. This argues for setting LASSORHO higher than the default and LASSOSTEPS much larger than the default of 20 in order to reach the optimum model according to the CHOOSE criterion. The downside might be long run-times.

COLLINEARITY: CHOOSE=SBC FOR LASSO VS. MLE FORWARD (SELECT=SBC)

Using CHOOSE=SBC, the performance of LASSO is compared to MLE FORWARD with SELECT=SBC on a data set with high multicollinearity. There are 10 binary predictors: B_In1 - B_In10, each of which have pairwise correlation with the others of at least 0.92 and another 10 binary predictors: B_Out1 - B_Out10, each of which is essentially independent of each other as well as the 10 B_In's. Strongly in the model are B_In1 - B_In10 and weakly, B_Out1 - B_Out10. Both HPLOGISTIC and HPGENSELECT are run on these 20 predictors.

```
DATA MULTI;
array B_In{10} B_In1 - B_In10;
array B_Out{10} B_Out1 - B_Out10;
do ID = 1 to 20000;
   xbeta = 0; RU = ranuni(1);
   do j = 1 to 10;
      B_Out{j} = (ranuni(1) < 0.3);
      xbeta = xbeta + 0.05*B_Out{j};
      end;
   /* High Multicollinearity is added here */
```

[30] SAS documentation: https://support.sas.com/documentation/cdl/en/stathpug/68163/HTML/default/viewer.htm#stathpug_hpgenselect_details29.htm

[31] But HPGENSELECT has a SPLIT statement which creates dummy variables from a classification predictor before the LASSO process begins. The dummies can enter or leave the LASSO model independently.

```
  do j = 1 to 10;
     B_In{j} = (.90*RU + .10*ranuni(1) < 0.50);
     xbeta = xbeta + 0.15*B_In{j};
     end;
  e = rannor(1);
  xbeta = xbeta + e;
  P_1 = exp(xbeta) / (1 + exp(xbeta));
  Y = (P_1 > 0.50);
  IF ID <= 10000 THEN PART = 1;
  ELSE IF ID <= 20000 THEN PART = 2;
  output;
  end;
```

SUMMARY OF FINDINGS

HPLOGISTIC with MLE FORWARD and SELECT=SBC, CHOOSE=SBC selects 4 B_In predictors and one B_Out. HPGENSELECT with LASSO and CHOOSE=SBC selects 8 B_In predictors and one B_Out.[32]

```
PROC HPLOGISTIC DATA= MULTI;
PARTITION ROLEVAR=part(TRAIN="1" VALIDATE="2");
MODEL Y(descending) = B_In: B_Out: ;
SELECTION METHOD=FORWARD (SELECT=SBC CHOOSE=SBC STOP=NONE);
run;
```

The HPLOGISTIC Procedure

Selected Effects:	Intercept B In1 B In2 B In9 B In10 B Out6

Partition Fit Statistics		
Statistic	Training	Validation
Area under the ROCC "c"	0.7701	0.7678
Average Square Error	0.1489	0.1456

Table 19. HPLOGISTIC

```
PROC HPGENSELECT DATA= MULTI LASSORHO=.80 LASSOSTEPS=20;
PARTITION ROLEVAR=part(TRAIN="1" VALIDATE="2");
MODEL Y(descending) = B_In: B_Out: / DISTRIBUTION = BINARY;
SELECTION METHOD=LASSO (CHOOSE=SBC STOP=NONE) DETAILS=ALL;
OUTPUT OUT= LASSO_OUT P=P_LASSO; ID Y PART;
run;
```

Optimal SBC occurred at step 13 for the LASSO selection.

The HPGENSELECT Procedure

Selected Effects:	Intercept B In1 B In2 B In3 B In4 B In5 B In8 B In9 B In10 B Out6

Partition Fit Statistics		
Statistic	Training	Validation
Average Square Error	0.1484	0.1446

Table 20. HPGENSELECT

HPGENSELECT does not compute the "c" (area under ROCC). It can be computed by running PROC LOGISTIC on the scored validation data set with the only predictor being the predicted probability (called P_LASSO). The fitted probability by PROC LOGISTIC will be a monotonic transform of the original

[32] For CHOOSE=SBC, SBC is computed on the Training data set.

probability P_LASSO so that the same “c” will be found. (The “c” can also be found by running CUM_LOGIT_SCREEN_2 on P_LASSO from the scored validation data set.)

```
PROC LOGISTIC DATA=LASSO_OUT(where=(PART=2)) desc;
MODEL Y= P_LASSO;
run;
```

Association of Predicted Probabilities and Observed Responses			
Percent Concordant	67.4	Somers' D	0.539
Percent Discordant	13.5	Gamma	0.666
Percent Tied	19.0	Tau-a	0.194
Pairs	18030400	c	0.769

Table 21

Two fit statistics, Average Squared Error and “c” (area under ROCC) on the validation dataset for HPLOGISTIC with FORWARD, SELECT=SBC are 0.1456 and 0.768. For LASSO these are 0.1446 and 0.769. The difference is small. Other statistics, including -2*Log(L) or R-Squares based on Log Likelihood, are not comparable.

Here are the coefficients from HPLOGISTIC and HPGENSELECT

Parameter	LASSO	MLE
Intercept	0.2413	0.1500
B In1	0.4224	0.6726
B In2	0.5466	0.7979
B In3	0.1827	
B In4	0.1891	
B In5	0.1546	
B In8	0.2415	
B In9	0.3332	0.5680
B In10	0.3343	0.5772
B Out6	0.0454	0.2112

Table 22.

Under LASSO the 8 highly correlated predictors share their coefficients The sum of LASSO coefficients for B_In1 B_In2 B_In3 B_In4 B_In5 B_In8 B_In9 B_In10 is 2.404

The sum of MLE coefficients B_In1 B_In2 B_In9 B_In10 is 2.616. Since all B_In’s are correlated at 0.92 or above, these MLE predictors, as a group, have similar combined effect on the probability calculation as the group of B_In’s selected as LASSO predictors.

In summary, both LASSO and MLE FORWARD, SELECT=SBC had similar fit statistics (ASE and c) but LASSO added 4 additional correlated predictors.

COMMENTS ON PROC LOGISTIC SELECTION=SCORE

In addition to FORWARD, BACKWARD, STEPWISE with p-value selection, PROC LOGISTIC offers “Best Subsets” through the selection option, SELECTION=SCORE. The statements to run SELECTION=SCORE for predictors X1-X4 are shown here:

```
PROC LOGISTIC DATA = MyData descending;
MODEL Y = X1-X4
/ SELECTION= SCORE START= 1 STOP= 4 BEST= 2;
run;
```

These values for START, STOP, BEST cause a total of 7 models to be selected as shown:

- From the four 1 variable models: {X1}, {X2}, {X3}, {X4}, take the 2 with highest score chi-square
- From the six 2 variable models: {X1 X2}, {X1 X3}, {X1 X4}, {X2 X3}, {X2 X4}, {X3 X4}, take the 2 with highest score chi-square
- From the four 3 variable models: {X1 X2 X3}, {X1 X2 X4}, {X1 X3 X4}, {X2 X3 X4}, take the 2 with highest score chi-square
- From the one 4 variable model: {X1 X2 X3 X4}, take the 1 model

The modeler then proceeds to study these 7 models.

SELECTION=SCORE is probably not going to be effective when the modeler has many classification variables. This topic is discussed in Lund (2016, pp. 4-8). Some summarizing remarks from this paper:

> The SELECTION=SCORE option does not support CLASS statements. One solution is to convert a classification variable into a collection of dummy variables. But with even a modest number of class variables the conversion to dummy variables could cause the total number of predictors to be 100 or more. Run time for PROC LOGISTIC begins to increase exponentially as the number of predictors for SELECTION= SCORE becomes 75 and greater. This makes large scale dummy variable conversion not practical when using SELECTON=SCORE.[33]
>
> One way to reduce the predictor count is to run a preliminary PROC LOGISTIC with SELECTION=BACKWARD FAST to reduce the count of predictors to the point where SELECTION=SCORE can be run. But BACKWARD FAST is not likely to select all the dummies associated with a classification variable. This distorts the results of final binning of NOD predictors. It amounts to unintended binning.

SUMMARY COMMENTS ON PREDICTOR SELECTION FOR LARGE SAMPLES

If sample size is sufficient for division of the data set in training, validation, and test, then PROC HPLOGISTIC provides the following attractive features:

- Multicollinearity in the model is avoided by CHOOSE=SBC.[34]
- CHOOSE=VALIDATE chooses the model by finding minimum ASE on the validation data set.
- SELECT=SBC and SELECT=AIC both avoid the problems of selection of predictors using p-values.
- Model fit statistics are computed on the training data set, the validation data set, and, as well, on the test data set. The test data set supports model fit analysis when CHOOSE=VALIDATE.

MWSUG 2019, Chicago, v07

REFERENCES

Albert, A. and Anderson, J. (1984). On the existence of maximum likelihood estimates in logistic regression models. *Biometrika* 71 : 1 – 10

Allison, P. (2012a), *Logistic Regression Using SAS: Theory and Application 2nd Ed.,* Cary, NC, SAS Institute Inc.

Allison, P. (2012b). Logistic Regression for Rare Events, *Statistical Horizons*. Last accessed 6/19/2019 at path: https://statisticalhorizons.com/logistic-regression-for-rare-events

Cavanaugh, J. (2012a). 171:290 Model Selection Lecture V: The Bayesian Information Criterion, Lecture Notes, University of Iowa. Last accessed 6/18/2019 at path: https://s3.amazonaws.com/academia.edu.documents/33100913/ms_lec_5_ho.pdf?response-content-disposition=inline%3B%20filename%3D171_290_Model_Selection_Lecture_V_The_Ba.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20190618%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190618T170534Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Amz-Signature=5cde83ddccf5e1b248dad8853cf89553ca70beddfa23c323059af1f02120cffa

[33] Conversion to WOE coding instead of dummy variables causes issues related to d.f., as discussed in Lund (2016).

[34] If CHOOSE=SBC, even if a validation dataset is provided, the SBC calculation is performed on the training data set.

Cavanaugh, J. (2012b). 171:290 Model Selection Lecture II: The Akaike Information Criterion, Lecture Notes, University of Iowa. Last accessed 6/18/2019 at path:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.504.9466&rep=rep1&type=pdf

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

Firth, D. (1993). Bias Reduction of Maximum Likelihood Estimates, *Biometrika*, Vol. 80, No. 1, pp. 27-38

Harrell Jr, F. (2015) *Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis*. 2nd Ed. Springer-Verlag, New York.

Hosmer D., Lemeshow S., and Sturdivant R. (2013). *Applied Logistic Regression, 3rd Ed.,* John Wiley & Sons, New York.

Lund, B. (2016). Finding and Evaluating Multiple Candidate Models for Logistic Regression, SAS Global Forum, paper 7860-2016.

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

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

Lund, B. (2019a) Screening, Transforming, and Fitting Predictors for Cumulative Logit Model, SAS Global Forum, paper 2019-3067.

Lund, B. (2019b) Transforming Ordinal Predictors to Numeric for Binary Logistic Models, SAS Global Forum, paper 2019-3687.

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

O'Brien, R. (2007). A Caution Regarding Rules of Thumb for Variance Inflation Factors, *Quality & Quantity*, 41:673–690.

O'Brien, R. (2016). Dropping Highly Collinear Variables from a Model: Why it Typically is Not a Good Idea: Dropping Highly Collinear Variables from a Model, Social Science Quarterly, Vol. 98, No 1.

Rainey, C. and McCaskey, K. (2017). Estimating Logit Models with Small Samples, Forthcoming in *Political Science Research and Methods*. Last accessed 6/19/2019 at path: http://www.carlislerainey.com/papers/small.pdf

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

Santer, T. and Duffy, D (1989). A note on A. Albert and J. A. Anderson's conditions for the existence of maximum likelihood estimates in logistic regression models . *Biometrika* 73: 755 – 758

Siddiqi, N. (2017). Intelligent Credit Scoring, 2nd edition, Hoboken, NJ, John Wiley & Sons, Inc.

CONTACT INFORMATION

Your comments and questions are valued and encouraged.

Contact the author with comments or for copies of SAS macro programs that were discussed in this paper.

Bruce Lund at blund_data@mi.rr.com or blund.data@gmail.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

APPENDIX

TOPIC 1: Mathematical description of convergence of an MLE $\widehat{\beta}_j$ to a normal distribution with mean β_j and variance, obtained from the inverse of the information matrix associated with the logistic model.

Suppose there are K predictors X_j where j = 1 to K in a logistic model so that there are K+1 parameters β_j where j = 0 to K and with a design matrix [$\underline{X}$] with n observations. The observed information matrix, $\mathrm{I}(\underline{\beta})$ has the j,l entry (j = 0 to K and l = 0 to K) given by:

$$\text{j,l entry} = \mathrm{I}_{j,l}(\underline{\beta}) = \sum_{i=1}^{n} x_{i,j}\, x_{i,l}\, p_i\, (1 - p_i)$$

where $x_{i,0} = 1$ and $p_i = \exp(\text{xbeta}_i) / (1 + \exp(\text{xbeta}_i))$ for i = 1 to n … no Y's required

The formula for variance $\sigma^2_{\hat{\beta}_j}$ of MLE $\widehat{\beta}_j$ is given by j^th^ entry in the diagonal of the inverse of $\mathrm{I}(\underline{\beta})$. ALR p. 38.

Now the goal is to show that $\sigma^2_{\hat{\beta}_j}$ is O[1/n]) as n → ∞. The number of predictors K is set to 1 to simplify the discussion. Then $\mathrm{I}(\underline{\beta})$ is a 2 x 2 matrix. The 1,1 entry of the inverse of $\mathrm{I}(\underline{\beta})$ is:

$$\sum_{i=1}^{n} 1 * 1 * p_i * (1 - p_i) \text{ / determinant.}$$

The determinant is given by:

$$\sum_{i=1}^{n} x_1^2 * p_i * (1 - p_i) * \sum_{i=1}^{n} 1 * 1 * p_i * (1 - p_i) - \left(\sum_{i=1}^{n} 1 * x_1\, p_i\, (1 - p_i) \right)^2$$

The argument will now become motivational. If the design [$\underline{X}$] for this sample is now repeated N-1 times (N-1 replications of the sample), then for this new large sample the numerator has a factor of N and the denominator has factor of N^2 and, otherwise, all terms remains the same. This shows that under this replication the variance of $\widehat{\beta_1}$ tends to zero as 1/N.

To approximate $\mathrm{I}(\underline{\beta})$ from a sample, the $\hat{\underline{\beta}}$ replace the $\underline{\beta}$

TOPIC 2: Simulations of conceptual logistic model populations, the mathematical justification.

Let ε be a random variable following a **logistic** distribution. The probability density for ε is given by $f(\varepsilon) = \exp(\varepsilon) / (1 + \exp(\varepsilon))^2$. Cumulative distribution function is $F(\varepsilon) = \exp(\varepsilon) / (1 + \exp(\varepsilon))$.

The SAS code included:

```
c = ranuni(s);
e = log(c / (1-c));
```

Then: $P(e < e_0) = P(\log(c / (1\text{-}c)) < e_0) = P(c / (1\text{-}c) < \exp(e_0)) = P(c < \exp(e_0) / (1 + \exp(e_0)))$

Because "c" is distributed as uniform [0,1], the cumulative distribution function for c is F(c) = c for c in [0,1].

Therefore, $P(e < e_0) = P(c < \exp(e_0) / (1 + \exp(e_0))) = \exp(e_0) / (1 + \exp(e_0))$ and "e" has the cumulative distribution of a logistic random variable.

Suppose there is a "latent" variable "Z" such as $Z = \alpha_0 + \alpha_X X + \varepsilon$ … where ε is logistic

Set Y = 1 if $Z \ge \lambda$ and Y = 0 if $Z < \lambda$ for a number λ.

The probability that Y = 1 is given by $P(Y = 1) = P(\alpha_0 + \alpha_X X + \varepsilon \ge \lambda)$. This leads to:

$$= P(\varepsilon \ge \lambda - (\alpha_0 + \alpha_X X)) = \int_{\lambda - \alpha_0 - \alpha_1 X}^{\infty} f(\varepsilon) d\varepsilon = 1 - \exp(\lambda - (\alpha_0 + \alpha_X X)) / (1 + \exp(\lambda - (\alpha_0 + \alpha_X X)))$$

$$= 1 / (1 + \exp(\lambda - (\alpha_0 + \alpha_X X))) = \exp(-\lambda + \alpha_0 + \alpha_X X) / (1 + \exp(-\lambda + \alpha_0 + \alpha_X X)) \ldots (*)$$

The final expression, denoted (*), is the logistic model probability for Y=1 with parameters α_X for X and intercept of $(-\lambda + \alpha_0)$. Using β's, these parameters are $\beta_0 = (-\lambda + \alpha_0)$ and $\beta_X = \alpha_X$.

The pairs (X, Y) that are created from the SAS code can be regarded as having been observed from a conceptual logistic model population with design [$\underline{X}$] and parameters β_0 and β_X

Although only one predictor was in this example, the result extends to multiple predictors.

TOPIC 3: Theoretical standard deviation $\sigma_{\hat{\beta}_j}$ from population with [$\underline{X}_S$], β0=1, β1=5, and *N*=5000

See Topic 1 for formula for variance $\sigma^2_{\hat{\beta}_1}$ Here is the SAS code which does the computation of the standard deviation. The variables I00, I01, I10, I11 are the elements of the observed information matrix.

```
Data StdErr;
   retain I00 I01 I11 0;
   do ID = 1 to 5000;
      X = (mod(ID,5) - 2)/10;
      P = 1 / (1 + exp(-(1 + 5*X)));
      I00 = I00 + 1*1*P*(1-P);
      I01 = I01 + 1*X*P*(1-P);
      I11 = I11 + X*X*P*(1-P);
      end;
      Det = I00*I11 - I01*I01;
      StdErr_beta1 = sqrt(I00 / Det);
      output;
run;
PROC PRINT DATA= StdErr;
VAR StdErr_beta1;
run;
```

Obs	StdErr_beta1
1	0.24595

TOPIC 4: Theoretical standard deviation $\sigma_{\hat{\beta}_j}$ from population with [$\underline{X}_S$], β0=1, β1=5, and *N*=100

In the code for Topic 3, replace 5000 with 100.

TOPIC 5: Firth Estimation

To find the minimum over b of FLog(L(b)) = Log(L(b)) + Log(A(b)), the derivatives of FLog(L) with respect to b_j are set to zero. The derivatives of Log(L(b)) are given by the likelihood equations (see ALR, p 37).

The derivatives of Log(A(b)) are another matter. The formula for A(b) equals {det $\mathbf{I}$(b) }$^{0.5}$ where $\mathbf{I}$(b) is the observed information matrix evaluated at b, as given in Topic 1. The formula for these derivatives is stated in ALR on p. 392. In ALR the formula is attributed to Heinze and Schemper (2002 p. 2412) who give the formula without supporting calculations. The derivatives of FLog(L) (being set to zero for maximization) are shown below:

$$\sum_{i=1}^{n}\{(y_i - p_i) * x_{i,j} + h_i * (0.5 - p_i) * x_{i,j}\} = 0 \text{ for } j = 0 \text{ to } K$$

where p_i = P(Y_i=1 | $\underline{x_i}$ b_j) and where h_i is the i^{th} diagonal element of the hat matrix (see ALR, p. 187)

The diagonals of the hat matrix also depend on b.

With some additional algebraic manipulation the equations above become:

$$\sum_{i=1}^{n}(y_i - p_i) * x_{i,j} + \sum_{i=1}^{n} 0.5 * h_i * (y_i - p_i) * x_{i,j} + \sum_{i=1}^{n} 0.5 * h_i * (1 - y_i - p_i) * x_{i,j} = 0 \text{ for } j = 0 \text{ to } K$$

For every $\underline{x_i}$ = ($x_{i,1}, \ldots, x_{i,k}$) there are 2 observations in each term of the sum, each with a weight. The observation with y_i has weight 1 + 0.5*h_i and the observation with $1 - y_i$ has weight 0.5*h_i. These h_i are positive (ALR pp. 187-188). Without going into further detail, it is this property of the Firth estimation method that insures that there are always Firth parameter estimates, even in the event of complete separation.

The correction term $h_i * (0.5 - p_i) * x_{i,j}$ in the derivative of FLog(L) tends to zero if the sample is replicated many times. This is because h_i tends to zero. See ALR p. 188 “any diagonal element in the hat matrix has an upper bound of 1/k” where k = number of cases with same predictor values.

TOPIC 6: The relationship between x-statistic and c-statistic

Suppose X has k = 1, …, J levels. Let G_k = count of Y=1 when X= x_k, and let B_k = count of Y=0 when X= x_k. Let G = $\sum_{k=1}^{J} G_k$ and B = $\sum_{k=1}^{J} B_k$ and g_k = G_k / G, b_k = B_k / B. Let M = G * B.

A refinement of the formula for the c-statistic is given below where the concordance pairs and ties are computed explicitly in terms of G_k and B_k.

Refined formula for c-statistic:

$$C = \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} Bi * Gj \text{ and } T = \sum_{k=1}^{J} B_k * G_k \text{ and c-statistic} = \max((C + 0.5*T) / M, 1 - (C + 0.5*T) / M)$$

A new formula for the x-statistic will be derived:

To begin, consider the k's of X_k's to be re-labelled so that $G_k / (G_k + B_k)$ is non-decreasing vs. k. The x-statistic is the c-statistic where the Y's are compared to the newly re-labelled X_k's, re-labelled so that $G_k / (G_k + B_k)$ is non-decreasing.

Because $G_k / (G_k + B_k)$ is non-decreasing, it follows by a short calculation that

$$(C + 0.5*T) / M \geq 1 - (C + 0.5*T) / M$$

so that the formulation of the x-statistic is

$$\text{x-statistic} = (C + 0.5*T) / M = \{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} Bi * Gj + 0.5 * \sum_{k=1}^{J} B_k * G_k \} / M$$

To simply the notation, without loss of generality, let J=3.

(*) The formula for x-statistic = $\{B_1*G_2 + B_1*G_3 + B_2*G_3\} / M + 0.5 * \{B_1*G_1 + B_2*G_2 + B_3*G_3\} / M$

M can be written as: $M = \{B_1*G_2 + B_1*G_3 + B_2*G_3\} + \{B_1*G_1 + B_2*G_2 + B_3*G_3\} + \{B_2*G_1 + B_3*G_1 + B_3*G_2\}$

x-statistic = $0.5 * \{ B_1*G_2 + B_1*G_3 + B_2*G_3 \} / M$ **+** $0.5 * \{ -B_2*G_1 - B_3*G_1 - B_3*G_2 + M \} / M$

x-statistic = $0.5 * \{ \sum_{i=1}^{2}\sum_{j=i+1}^{3} [B_i*G_j - B_j*G_i] / M + 1 \}$

Because $G_k / (G_k + B_k)$ is non-decreasing, then $B_i * G_j \geq B_j * G_i$ when i < j. This allows absolute signs to be inserted without affecting the sum and this leads to (**).

(**) x-statistic = $0.5 * \{ \sum_{i=1}^{2}\sum_{j=i+1}^{3} | B_i*G_j - B_j*G_i | / M + 1 \}$

This gives to a new formula for the x-statistic:

$$\text{x-statistic} = 0.5 * \{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} | B_i*G_j - B_j*G_i | / M + 1 \}$$

With this formulation, the x-statistic is invariant (has unchanging value) under any re-labelling of the X values. Specifically, every pair i,j (where i < j) appears exactly once in an expression $| B_i*G_j - B_j*G_i |$. So any re-labelling will produce the same terms.

Under any re-labelling of x_k's:

$$\text{x-statistic} = 0.5 * \{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} | B_i*G_j - B_j*G_i | / M + 1 \}$$

The x-statistic equals the model c for PROC LOGISTIC desc; CLASS X; MODEL Y=X;

For this logistic model, P(Y=1 | X=x_k) = $G_k / (G_k + B_k)$. The original definition of x-statistic equals this model c.

Theorem: $G_k / (G_k + B_k)$ is monotonic in the ordering of x_k's if and only if x-statistic = c-statistic

Outline of proof:

Left to Right: Assume monotonic non-decreasing and follow the steps from equation (*) to (**). A similar proof applies under the assumption that monotonic non-increasing

Right to Left:

Assume $\{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} \mathrm{Bi} * \mathrm{Gj} + 0.5 * \sum_{k=1}^{J} \mathrm{B_k} * \mathrm{G_k} \} / \mathrm{M} \geq 0.5.$

Then c-statistic = $\{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} \mathrm{Bi} * \mathrm{Gj} + 0.5 * \sum_{k=1}^{J} \mathrm{B_k} * \mathrm{G_k} \} / \mathrm{M}.$

Since x-statistic = c-statistic, this assumed equality gives this equation:

$$0.5 * \{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} | \mathrm{B_i}*\mathrm{G_j} - \mathrm{B_j}*\mathrm{G_i} | / \mathrm{M} + 1 \} = \{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} \mathrm{Bi} * \mathrm{Gj} + 0.5 * \sum_{k=1}^{J} \mathrm{B_k} * \mathrm{G_k} \} / \mathrm{M}$$

The right hand expression can be recast as $0.5 * \{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} [\mathrm{B_i}*\mathrm{G_j} - \mathrm{B_j}*\mathrm{G_i}] / \mathrm{M} + 1 \} \ldots$ (see expression (*) and the steps which follow.

But then $0.5 * \{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} | \mathrm{B_i}*\mathrm{G_j} - \mathrm{B_j}*\mathrm{G_i} | / \mathrm{M} + 1 \} = 0.5 * \{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} [\mathrm{B_i}*\mathrm{G_j} - \mathrm{B_j}*\mathrm{G_i}] / \mathrm{M} + 1 \}.$

This implies: $B_i*G_j - B_j*G_i \geq 0$ for all (i, j) with i < j. Therefore, $G_k / (G_k + B_k)$ is monotonic non-decreasing.

Otherwise, assume $\{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} \mathrm{Bi} * \mathrm{Gj} + 0.5 * \sum_{k=1}^{J} \mathrm{B_k} * \mathrm{G_k} \} / \mathrm{M} < 0.5.$

Then c-statistic = $\{ \sum_{i=1}^{J-1}\sum_{j=i+1}^{J} \mathrm{Bj} * \mathrm{Gi} + 0.5 * \sum_{k=1}^{J} \mathrm{B_k} * \mathrm{G_k} \}$ / M (… indices in double sum are reversed)

An argument similar to that given above shows that $G_k / (G_k + B_k)$ is monotonic non-increasing.

What remains to be proved is that c-statistic < x-statistic if $G_k / (G_k + B_k)$ is not monotonic.

Again, assume J=3. This outline below motivates the proof.

Assume $G_1 / (G_1 + B_1) > G_2 / (G_2 + B_2)$, and $G_1 / (G_1 + B_1) \leq G_3 / (G_3 + B_3)$.

This implies: $G_2 / (G_2 + B_2) \leq G_3 / (G_3 + B_3)$.

By assumption, | B1*G2 - B2*G1 | = - B1*G2 + B2*G1 > 0

x-statistic = .5 * { B1*G2 + B1*G3 + B2*G3 } / M **+** .5 * {-B2*G1) - B3*G1) - B3*G2 + M} / M **+** Q

where Q = .5 * {-2* B1*G2 + 2* B2*G1 } / M. Note that Q > 0

The terms to the left of Q can be arranged to give the c-statistic, therefore, x-statistic = c-statistic + Q

TOPIC 7: Collinear Predictors, Sample Size, and Standard Errors for Estimated Coefficients

The first DATA Step creates bivariate normal variables X1 and X2, standardized, and with correlation of 0.995. The data set has 500 observations. In the second DATA Step the correlated normals, X1 and X2, are used to generate a logistic model population with target Y, parameters $\beta_1 = \beta_2 = 1$ for X1 and X2, and intercept 0.

```
DATA A;
  mu1=0; mu2=0; var1=1; var2=1; rho=.995;
  do i = 1 to 500;
   X1  = mu1+sqrt(var1)*rannor(123);
   X2  = (mu2+rho*(sqrt(var2)/sqrt(var1))*(x1-mu1)) +
         sqrt(var2*(1-rho**2))*rannor(123);
   output;
   end;
run;
PROC CORR DATA= A noprob; Var X1 X2;
run;
```

Pearson Correlation Coefficients, N = 500		
	x1	x2
x1	1.00000	0.99466
x2	0.99466	1.00000

Table 1, Topic 7

```
DATA B; SET A;
    c = rand("Uniform");
    e = 1*log(c / (1-c));
    Z = 0 + 1*X1 + 1*X2 + e;
    Y = (Z > 0);
run;
```

Then PROC LOGISTIC is run. The model is significant, with right tail probability < .0001 for the likelihood ratio test.

```
PROC LOGISTIC DATA= B desc;
model y= X1 X2;
run;
```

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	203.8015	2	<.0001

Table 2, Topic 7

But neither X1 nor X2 is significant as seen in Table 3, Topic 7.

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.0287	0.1115	0.0662	0.7970
x1	1	0.7129	1.1342	0.3950	0.5297
x2	1	1.1499	1.1303	1.0350	0.3090

Table 3, Topic 7

Now the sample size is changed from 500 to 5000. Both predictors X1 and X2 are significant. This shows the effect of sample size on the calculation of standard errors.

Testing Global Null Hypothesis: BETA=0			
Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	2394.0288	2	<.0001

Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	0.00867	0.0368	0.0556	0.8136
x1	1	0.8109	0.3654	4.9251	0.0265
x2	1	1.2505	0.3652	11.7259	0.0006

Table 4, Topic 7

TOPIC 8: Code to Rank Predictors by X-statistic and Remove Collinearity Based on Ranking

```
DATA Correlation_Data;
do i = 1 to 1000;
   random1 = ranuni(1);
   random2 = ranuni(1);
   random3 = ranuni(1);
   random4 = ranuni(1);
   random5 = ranuni(1);
   random6 = ranuni(1);
   X1 = floor(10*random1);
   X2 = floor(13*(random1*(3/5) + random2*(2/5)));
   X3 = floor(13*(1.8*random1 + random3 + 0.2*random5)/3);
   X4 = floor(20*(0.1*random1 + 1.9*random4 + 2.0*random5)/4);
   X5 = (X1 - 4.5)**2;
   xbeta = -.20 + 1.5*X1 + .1*X2 + .15*X3 + .10*X4 + 0.1*X5 + 12*rannor(1);
   Y = ( xbeta > 0);
   output;
   end;
run;
%CUM_LOGIT_SCREEN_2(Correlation_Data, Y, X1 X2 X3 X4 X5,  ,   ,  );

* SET CORR_LIMIT;
%let CORR_LIMIT = .80;
* __IV_FINAL is the report data set from %CUM_LOGIT_SCREEN_2;
PROC SORT DATA = __IV_FINAL; BY descending X_STAT;
where CHARACTER = "NO";
run;
DATA Var_Name_Ordered; SET __IV_FINAL end = eof;
length Var_Name_Ordered $5000;
retain Var_Name_Ordered;
retain count 0;
count + 1;
Var_Name_Ordered = trim(COMPBL(Var_Name_Ordered || " " || Var_Name));
if eof then do;
     count_c = trim(left(put(count,4.)));
     call symput('V_N_O', Var_Name_Ordered);
     call symput('count_inputx', count_c);
     call symput('Dataset', Dataset);
     output;
     end;
run;
PROC CORR DATA = &Dataset pearson outp = Pearson;
var &V_N_O;
run;
DATA Removed_Inputs(keep=step removed correlation);
   SET Pearson end = eof;
length removed $32;
retain i 0;
retain ___1 - ___&count_inputx 1;
array ___x {*} ___1 - ___&count_inputx;
array inputx {*} &V_N_O;

if _type_ = "CORR" then i = i+1;
else delete;

do j = i+1 to &count_inputx;
```

```
   if abs(inputx{j})*___x{i} >= &CORR_LIMIT
   then
   do;
      ___x{j} = .;
      removed = vname(inputx{j});
      correlation = inputx{j};
      step = i;
      output;
      end;
   end;
if eof
then do;
     removed = "End_File";
     output;
     end;
run;
PROC PRINT DATA = Removed_Inputs; var Removed step correlation;
title1 "Variables to be removed due to correlation";
title2 "Absolute Correlation Max = &Corr_Limit";
run;
```