New Methods for Model Fitting and Validation for Logistic Modeling v09

by Bruce Lund

Statistical Modeling Consultant and Trainer, Novi, MI

blund_data@mi.rr.com and blund.data@gmail.com

Send an email for a copy of slides

***W**hen we raise money it's AI, when we hire it's machine learning, and when we do the work it's logistic regression.*

— *Juan Miguel Lavista* … Chief Data Scientist, Microsoft Corp.

References (see especially the reference in RED)

- Austin, P. and Steyerberg, E. (2017). Events per variable (EPV) and the relative performance of different strategies for estimating the out-of-sample validity of logistic regression models, *Stat Methods Med Res.*
- (RMS) Harrell, F. (2015) *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis, 2nd Edition*. New York: Springer.
- (HLS) Hosmer D., Lemeshow S., Sturdivant R. (2013). *Applied Logistic Regression*, 3rd Ed., Wiley, New York
- Riley, et. al. (2019). Minimum sample size for developing a multivariable prediction model: PART II - binary and time-to-event outcomes, *Statistics in Medicine.*
- (CPM) Steyerberg, E. (2019). *Clinical Prediction Models* 2nd Ed., Springer, Cham Switzerland.
- Stijacic-Cenzer, I, et. al. (2013) "Estimating Harrell's Optimism on Predictive Indices Using Bootstrap Samples", SAS Global Forum. … includes SAS macro for bootstrap optimism.
- Van Smeden, et. al. (2019) Sample Size for binary logistic prediction models: Beyond events per variable criteria, *Statistical Methods in Medical Research*.

Some Terminology related to Sampling, Model Fitting, and Model Validation

Terminology

An Analysis Dataset is randomly sampled (or possibly "oversampled") from the POPULATION

Oversampling might be used if the event is "rare" in the Population.

Example: Assume: 2,000,000 non-events and 5,000 events.

If the modeler samples 2,000 non-events and 2,000 events, this is an "Oversample" of events.

The TRAINING dataset is EITHER the full Analysis Dataset OR is a random sample from the Analysis Dataset. Either way, the TRAINING is the dataset where the Model coefficients are fitted.

In a Split-Sample, the Analysis Dataset is randomly split into Train and Validation

Typical splits are 50-50, 60-40, 70-30

Model is validated on Validation

Alternatively: Analysis Dataset is randomly split into Train, Validation, Test ... perhaps 40-30-30

Models are fitted on Train and the final Model is chosen with the use of Validation.

(PROC HPLOGISTIC has such an option)

Then the chosen Model is validated on Test

This talk is about training a model that Predicts. Focus is not on investigating effects (testing coefficients). Success of prediction is measured by validation metrics ... c-Stat, ASE, Lift Charts, etc.

NEXT

Double Dipping

Here is a Quote from: N. Kriegeskorte, et. al. (2009) "Circular analysis in systems neuroscience: the dangers of double dipping", *Nature Neuroscience* …

"Double Dipping is the use of the same dataset for selection and selective analysis. It gives distorted descriptive statistics and invalid statistical inference"

Double Dipping could arise if fitting a logistic model to TRAINING without a VALIDATION sample.

1. Preparation of predictors (X's) that involves screening, binning, or transforming runs the risk of double dipping when X's are prepared on the same dataset as is used for Validation.
2. Likewise, Validation of the final model using the same TRAINING dataset poses the challenge of how to avoid double dipping.

Can these two problems be solved without having a split-sample ??

It is a purpose of the talk today to answer this question.

Dr. Daniela Witten during a 2022 webinar hosted by Wake Forest University conjectured that the Kriegeskorte paper (cited above) was the first usage of "double dipping" as a statistical term

Don't Double Dip … Data or Chips

But don’t forget the Seinfeld episode of 1993 where, at a party, George double-dipped a chip!

Alternative to Split-Sample

Harrell, Steyerberg, et. al. argue that Split-Sample wastes data which can be used to fit more X's or reduce error in $\hat{\beta}$'s (see [RMS] and [CPM])

… The Alternative is:

Step (1) Model is fitted on the Analysis Dataset. (where TRAIN becomes ANALYSIS DATASET)
Step (2) Model is validated on Analysis Dataset using bootstrapping and "optimism correction"

"Optimism correction" is a process that provides honest validation of Model performance without a split-sample. It provides c-stat, ASE, Lift Charts that are NOT compromised by Double Dipping.

"Optimism correction" (in its purist form) is applied to the entire Modeling Process

- Exploratory analysis of X vs. Y is part of the Modeling Process
- Preparation of X's (screen, transform) is part of the Modeling Process
- Model fitting is part of the Modeling Process

ALL steps in Modeling Process are repeated on many bootstrap samples as part of optimism correction. As explained on Following Slides, an honest validation of Model performance is given.

It is focus of the talk today to explain how bootstrap sampling enables optimism correction

Reflections on the prior slide

Steps in preparing the predictors X's include:

- Missing data and imputation
- Exploratory analysis (tables and graphics)
- Screening out weak X's
- Transforming stronger X's
- Deciding on interactions
- And more

Optimism Correction, in its purist form, requires that these steps (above) be structured and included within the Modeling Process so that optimism (=bias) can be computed and corrected.

Real World Difficulties:

- Deciding on these steps ahead of time
- Programming the steps in a form that allows repeating the Modeling Process 100's of times.

Compromises in defining the Modeling Process might be necessary:

- Omit some of the steps in preparing X's from the Modeling Process

But do not omit Predictor Selection / Model Fitting from Modeling Process for optimism correction

See Austin and Steyerberg (2017) for study of split sample vs. alternatives

The German Bank Dataset

German Bank Dataset … used on slides to follow

- Dataset contains 1000 rows, each row has a binary target and 20 predictors.
- Each row gives information about a loan applicant who was approved by the bank for the loan.
- The 20 predictors contain information at the time of application.
    - 17 categorial and 3 continuous numeric X's
        - Categorial X includes nominal, ordered non-numeric, and numeric … but with "few" levels
        - If a Categorial X has L levels, then L-1 dummies are created by CLASS X … using L-1 d.f.
- Target was determined later in time. Had values “good” (loan paid as agreed) or “bad” (default).
- The bank uses this information to fit a “probability of default” (PD) model to assess future applicants for a loan.
- There are 300 Bad's (30% of total) and 700 Good's in the Dataset … an Oversample
- Source: UC Irvine Machine Learning Repository (or better yet, get CSV file from me)
  https://archive.ics.uci.edu/ml/datasets/Statlog+%28German+Credit+Data%29

Data Dictionary (4 Slides)

Attribute 1: (character) -- checking_status (ordered with missing)
Status of existing checking account
A11 : ... < 0 DM
A12 : 0 <= ... < 200 DM
A13 : ... >= 200 DM / salary assignments for at least 1 year
A14 : no checking account

Attribute 2: (numerical) -- duration
Duration of loan in month

Attribute 3: (character) – credit_history
Credit history
A30 : no credits taken/ all credits paid back duly
A31 : all credits at this bank paid back duly
A32 : existing credits paid back duly till now
A33 : delay in paying off in the past
A34 : critical account / other credits existing (not at this bank)

Attribute 4: (character) -- purpose
Purpose
A40 : car (new)
A41 : car (used)
A42 : furniture/equipment
A43 : radio/television
A44 : domestic appliances
A45 : repairs
A46 : education
A48 : retraining
A49 : business
A410 : others

Attribute 5: (numerical) -- credit_amount
Credit amount

Attribute 6: (character) – savings (ordered?)
Savings account/bonds
A61 : ... < 100 DM
A62 : 100 <= ... < 500 DM
A63 : 500 <= ... < 1000 DM
A64 : .. >= 1000 DM
A65 : unknown / no savings account

Data Dictionary (4 Slides)

Attribute 7: (character) – employment
Present employment since
A71 : unemployed
A72 : ... < 1 year
A73 : 1 <= ... < 4 years
A74 : 4 <= ... < 7 years
A75 : .. >= 7 years
ordered?

Attribute 8: (numerical) -- installment_rate
Installment rate in percentage of disposable income ... four levels 1, 2, 3, 4 ... might be ordered

Attribute 9: (character) -- personal_status
Personal status and sex
A91 : male : divorced/separated
A92 : female : divorced/separated/married
A93 : male : single
A94 : male : married/widowed
A95 : female : single

Attribute 10: (character) -- other_parties
Other debtors / guarantors
A101 : none
A102 : co-applicant
A103 : guarantor

Data Dictionary (4 Slides)

Attribute 11: (numerical) – residence_since
Present residence since
four levels 1, 2, 3, 4 … meaning uncertain, might be ordered

Attribute 12: (character) -- property_magnitude
Property
A121 : real estate
A122 : if not A121 : building society savings/ life insurance
A123 : if not A121/A122 : car or other, not in attribute 6
A124 : unknown / no property

Attribute 13: (numerical) -- age
Age in years

Attribute 14: (character) -- other_payment_plans
Other installment plans
A141 : bank
A142 : stores
A143 : none

Attribute 15: (character) -- housing
Housing
A151 : rent
A152 : own
A153 : for free

Data Dictionary (4 Slides)

Attribute 16: (numerical) -- existing_credits
Number of existing credits at this bank
… four levels, 1, 2, 3, 4, … meaning uncertain

Attribute 17: (character) -- job
Job
A171 : unemployed/ unskilled - non-resident
A172 : unskilled - resident
A173 : skilled employee / official
A174 : management/ self-employed/
highly qualified employee/ officer

Attribute 18: (numerical) -- num_dependents
Number of people being liable to provide maintenance
for only two levels

Attribute 19: (character) -- telephone
Telephone
A191 : none
A192 : yes, registered under the customers name

Attribute 20: (character) -- foreign_worker
foreign worker
A201 : yes
A202 : no

CLASS
Target: (numerical)
1: BAD Loan
0: GOOD Loan

To avoid confusion, it will be renamed to Y

Logistic Model to be fit to full German dataset (n=1000)

Notation: n1=# of events, n0=# of non-events, n = n1+n0 and n1 $\leq$ n0

Let K = number of X's (i.e. d.f.) to be considered for the MODEL (the candidate X's) ... not necessarily in the Model.

Restrictions on K ... Here is a Rule:

Van Smeden, et. al. (2019) propose an inequality:

$n \geq 10 * K / \text{event-rate}$ ... event-rate = n1 / n

For the German Bank Data: n = 1000 and event-rate is 300/1000 = 0.3

Set 1000 = 10*K / 0.3 ... This implies max(K) = 30

This Formula (above) implies the well-known rule of "At least 10 events per predictor (d.f.)"

$n \geq 10 * K / (n1 / n) \rightarrow n1 \geq 10 * K$ ... That is: require n1 to be at least 10 times K.

See also Riley, et. al. (2019) for advanced discussion of planning and evaluating sample size for Logistic Models

Preparing X's when fitting Model to full Analysis Dataset

Screening and Preparing X's:

NEXT

Categorial X's:

1. If the modeler "looks" at X via an analysis of X v. Y, then is X a candidate predictor? ... count X against K?
   Depends on how the X list was identified:
   - If "purposeful" selection by modeler, then X is a candidate for Model
   - If "compilation" of X's, then OK (I think) to screen out weak X's (and not count against K)
     - Regard the 17 categorial X's from German Bank as "compiled"
2. It's common to use CLASS X in PROC LOGISTIC to create Dummies for all levels of X but a reference level
   - For SELECTION= FORWORD (BACKWARD/STEPWISE.) ... then ALL or NONE dummies are in Model (*)
     ALL or NONE avoids "unintended binning" and I think this is good. ... Let X1 have levels A, B, C
     ... Else, if dummy for B is in Model, but dummy for A is not, then A is "binned" with reference C
     ... Do A and C have compatible meanings? ... might be OK or might not
3. If X1 is ordered, then possible for P(Y=1|X1=A), P(Y=1|X1=B), P(Y=1|X1=C) to be unordered ... other X's fixed
   - Is this undesirable for your model?
     - Simple Fix: Make X1 numeric and treat as a linear term in Model
     - If 5+ levels of X1, then consider Monotonic Binning of X1 (**)

(*) HPGENSELECT allows CLASS X / SPLIT when fitting a logistic model

(**) See: https://statcompute.wordpress.com/2017/09/24/granular-monotonic-binning-in-sas/
or See: http://support.sas.com/resources/papers/proceedings17/0969-2017.pdf

Screening and Preparing X's:

NEXT

Categorial X's , continued:

4. Low freq. level of X … create a dummy variable for this level?

    No. Doesn’t help prediction, coefficient meaningless, might cause "separation" (MLE doesn't converge)
    It's OK to combine low freq. level with some other level of X … if not looking at Y
        If X is ordered, then combine the low freq. level with an adjacent level
        If X is not ordered, then look for another level with "similar" meaning

It would be hard to include Points 1-4 (above) in a Modeling Process to be repeated automatically
We will not do this when we begin the modeling and validation of the German Bank Data model.

Continuous numeric X's … (e.g. (i) use X or Log(X) … (ii) should X^2 be added to X?)

- Same comments, as above, about "looking".
- How to decide on a transformation for X without an exploratory analysis of relationship of X to Y?
    - Regression Splines provide flexible transforms, are created at time of model fitting. … Will discuss later
        - Consider using splines for "strong" predictors where non-linearity seems possible
        - Otherwise, for "weak" X, enter X only (as linear)

Interactions: Use subject matter expertise to choose interactions in candidate list

An advantage of Decision Trees over Logistic Regression is automatic creation of interactions

Screening and Preparing categorial X's

when fitting Model to Analysis Dataset

IV (Information Value) as Screener of categorial X

X	Y = 0	Y = 1	Col % Y=0 "b_k"	Col % Y=1 "g_k"	$\text{Log}(g_k/b_k)$ = X_woe	D =(g_k – b_k)	D * X_woe
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

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 $\geq$ 0.3	"Strong"

IV is "gold standard" for measuring X and to eliminate weak X

IV is not defined if zero in a freq cell

Siddiqi (2017, p. 179). *Intelligent Credit Scoring*, 2nd edition, John Wiley & Sons, Inc., Hoboken, NJ

NEXT

Screening Categorial X's from German Bank using IV

%CUM_LOGIT_SCREEN_2 (IOWA23.bank_german_data, Y, &NUMVAR, &CHARVAR, NO, YES);

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 $\geq$ 0.3	"Strong"

There were no zero-cells

- Screened out 12 weak predictors
- Leaving only 5 categorial
- There are 3 continuous numeric X's
- … so now there are 8 X's left.

NEXT

VAR_NAME	Levels	Character	IV
checking_status	4	YES	0.666
credit_history	5	YES	0.293
~~employment~~	~~5~~	~~YES~~	0.086
~~existing_credits~~	~~4~~	~~NO~~	0.013
~~foreign_worker~~	~~2~~	~~YES~~	0.044
~~housing~~	~~3~~	~~YES~~	0.083
~~installment_rate~~	~~4~~	~~NO~~	0.026
~~job~~	~~4~~	~~YES~~	0.009
~~num_dependents~~	~~2~~	~~NO~~	0.000
~~other_parties~~	~~3~~	~~YES~~	0.032
~~other_payment_plans~~	~~3~~	~~YES~~	0.058
~~personal_status~~	~~4~~	~~YES~~	0.045
property_magnitude	4	YES	0.113
purpose	9	YES	0.150
~~residence_since~~	~~4~~	~~NO~~	0.004
savings	5	YES	0.196
~~telephone~~	~~2~~	~~YES~~	0.006

The predictor "purpose" ... has several low frequencies

"purpose" is unordered. Cells were subjectively combined using similarity of definitions. There was no cross-tabs of "purpose" vs. Y.

purpose	Freq	Meaning
A40	234	A40 : car (new)
A41	115	A41 : car (used)
A42	181	A42 : furniture/equipment
A43	280	A43 : radio/television
A44	**12**	A44 : domestic appliances
A45	22	A45 : repairs
A46	50	A46 : education
A48	**9**	A48 : retraining
A49	97	A49 : business

Combines	%Y=1
A40	234
A41	115
A42_A44	**193**
A43	280
A45	22
A46_A48	**59**
A49	97

This is the Now the working dataset

```
DATA IOWA23.bank_german_data_v2;
SET IOWA23.bank_german_data;
if purpose in ("A42" "A44") then purpose = "A42_44";
if purpose in ("A46" "A48") then purpose = "A46_48";
run;
```

"purpose" now reduced to 7 levels.

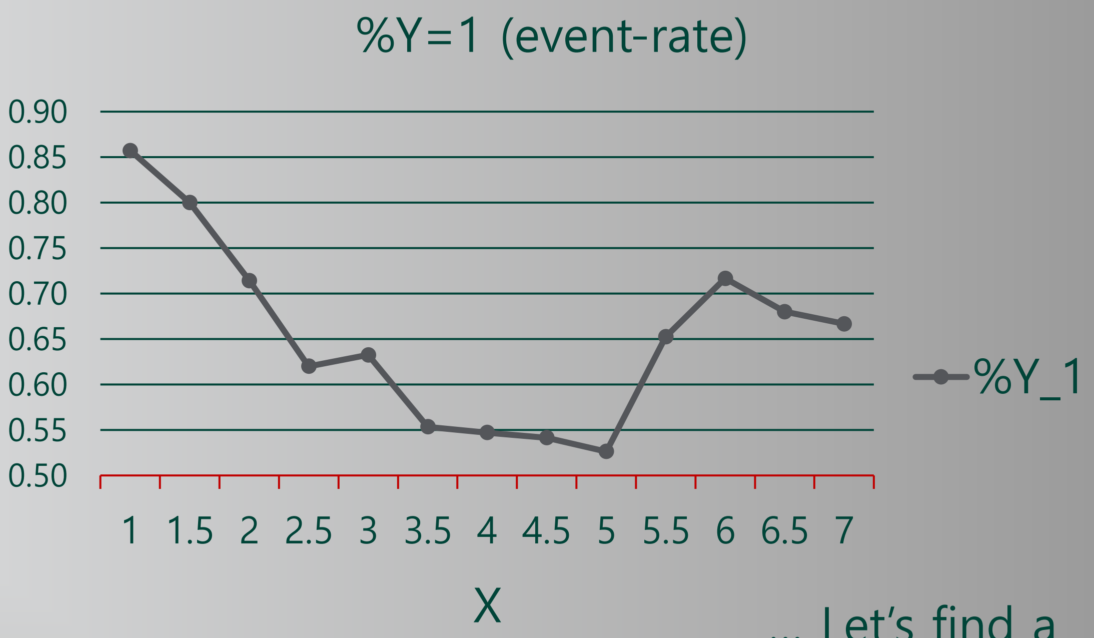
NEXT

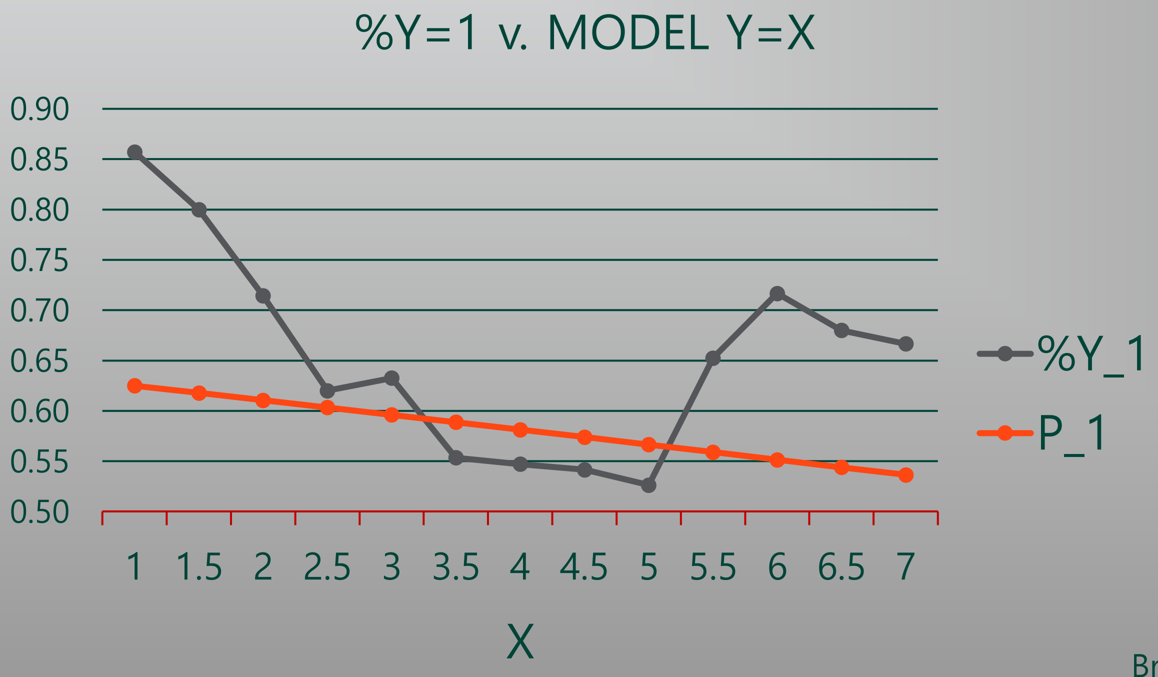
Splines as Transforms for Continuous Numeric X

Event-Rate is "U" shaped versus X

```
DATA SPLINEDATA;
Do ID = 1 to 4000;
   cumLogit = ranuni(2);
   e = 1*log( cumLogit/(1-cumLogit ));
   X = round(rannor(9),.5);
   X = max(min(X,3),-3) + 4;
   xbeta = 1 + (X-4)**2 + 6*e;
   Ps = exp(xbeta) / (1 + exp(xbeta));
   Y = (Ps > 0.50);
   output;
   end;
run;
PROC LOGISTIC
DATA = SPLINEDATA DESC;
MODEL Y = X;
SCORE DATA=SPLINEDATA
OUT=SCORE;
run;
```

Use of linear X is not the best choice of transform.





... Let's find a better transformation of X using SPLINES

Maximum Likelihood Estimates

Parameter	DF	Estimate	Pr > ChiSq
Intercept	1	0.5714	<.0001
X	1	-0.0608	0.0556

Natural Cubic Splines

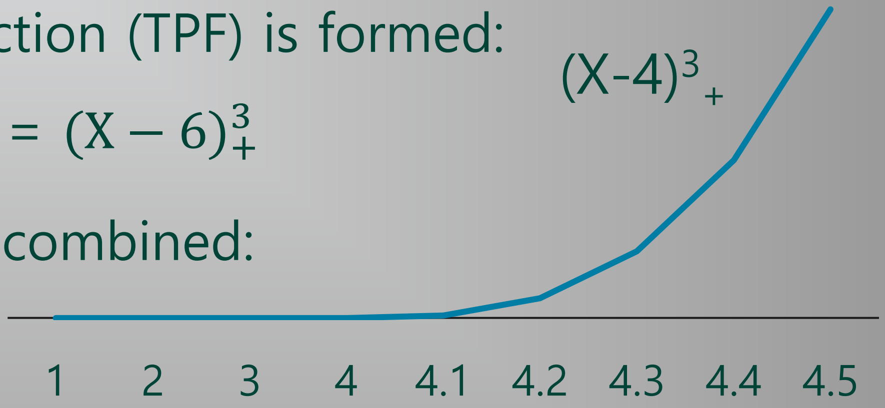
Perhaps try polynomials $X1=X$, $X2=X^2$, …, $X10=X^{10}$ … bad endpoint behavior, which power? overfit?

Alternative to polynomials is "natural (=restricted) cubic splines" (NCS) … needs explaining.

- A subjective feature of NCS's is the decision regarding the number and location of "knots"
- Knots are points inside the domain of X (not end points)
- For the SPLINEDATA with X and Y the knots will be at 2, 4, 6 … this is good for our example but is not necessarily the best number or location.

To begin: For each of the 3 knots, a truncated (cubic) power function (TPF) is formed:

TPF(2) = max(0, $(X\text{-}2)^3$) = $(X-2)_+^3$ TPF(4) = $(X-4)_+^3$ TPF(6) = $(X-6)_+^3$



For a Natural Cubic Spline: The 3 truncated power functions are combined:

N1(X) = [TPF(2) - TPF(6)]/2 - [TPF(4) - TPF(6)]

Why N1(X)?

Natural Cubic Splines

N1(X) = [TPF(2) - TPF(6)]/2 - [TPF(4) - TPF(6)]

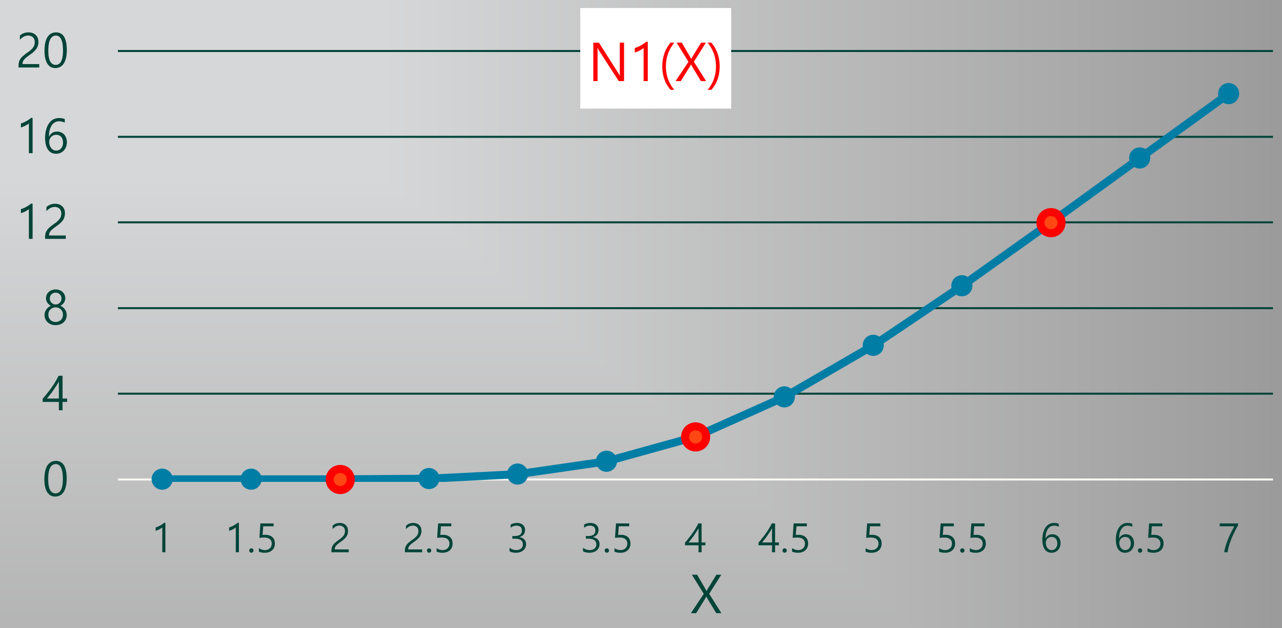
This simplifies to N1(X) = ($(X - 2)_+^3$ - 2*$(X - 4)_+^3$ + $(X - 6)_+^3$) / 4

N1(X) uses only 1 d.f.

Because of clever construction,

N1(X) has these properties:

- Linear to the left of 2
- Linear to the right of 6 (= 6*X – 24)
- Cubic polynomial between the knots
- Twice differentiable across all X.



NOTE: So far, Y is not involved.

Can 1, X, N1(X) provide good fit to the Y from SPLINEDATA in a logistic regression?

PROC LOGISTIC with NCS

NEXT

```
PROC LOGISTIC DATA = SPLINEDATA desc;
EFFECT X_spl = Spline ( X / details
   Naturalcubic
   basis=TPF(noint) /* always use "noint" */
   knotmethod=LIST(2, 4, 6));
MODEL Y = X_spl;
SCORE DATA = SPLINEDATA OUT=SCORED;
run;
```

Analysis of Maximum Likelihood Estimates						
Parameter		df	Estimate	Std Error	Wald Chi-Sq	Pr > ChiSq
Intercept		1	2.0697	0.2704	58.6	<.0001
X_spl = X	1	1	-0.5658	0.0851	44.2	<.0001
X_spl = N1(X)	2	1	0.1683	0.0261	41.8	<.0001

xbeta = 2.0697 - 0.5658*X + 0.1683*($(X-2)_+^3$ - 2*$(X-4)_+^3$ + $(X-6)_+^3$) / 4

Now: Compute P_1 = exp(xbeta) / (1 + exp(xbeta)) ... next slide

SAS Notation: "Raw" X is the first spline ... X = X_spl1

N1(X) is the second spline ... N1(X) = X_spl2

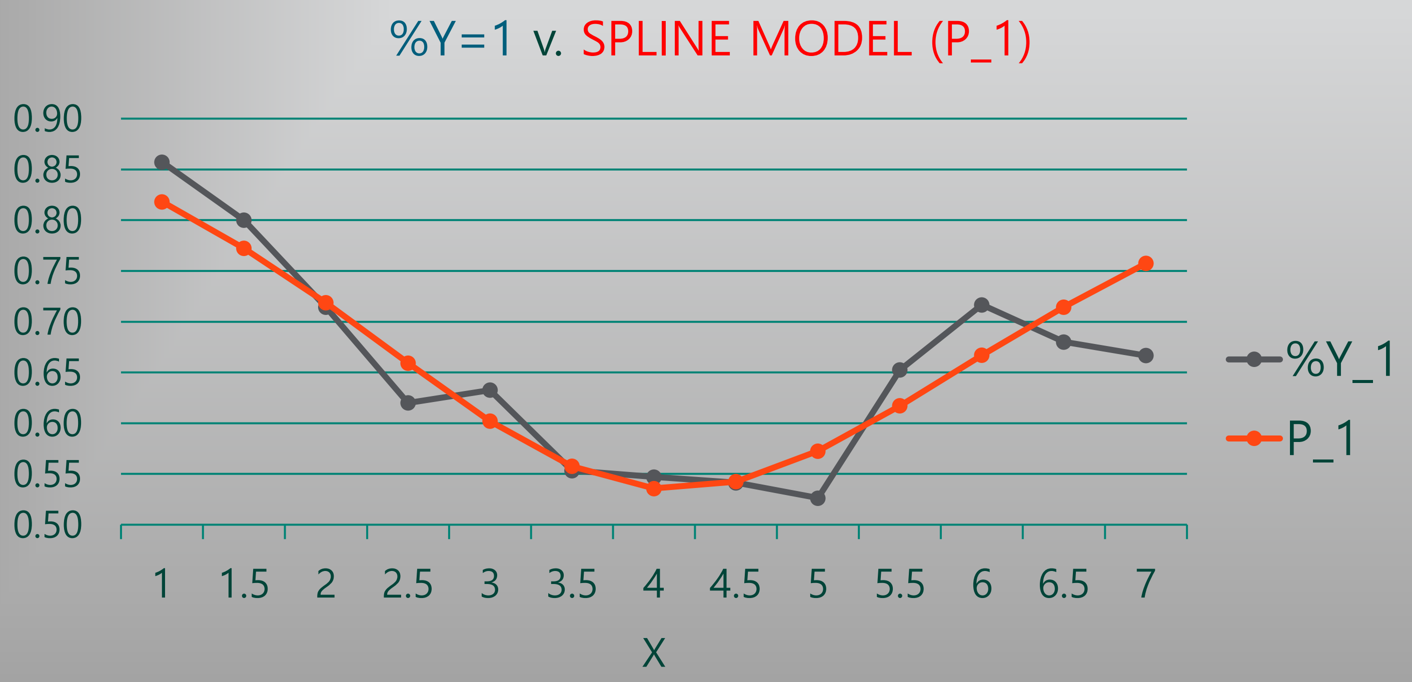
If "noint" is omitted, then X_spl1=1, X_spl2=X, X_spl3=N1(X) ... but X_spl1 is redundant with the intercept ... adds a predictor to the report above with 0 coefficient and other columns blank.

Spline Transformation of X vs. %Y=1 for Example

The Spline transform tracks %Y=1 quite well.

SPLINE: P_1 for spline model

%Y=1 the proportion of Y=1 at X



X	COUNT
1	7
1.5	40
2	119
2.5	300
3	479
3.5	739
4	786
4.5	639
5	475
5.5	259
6	120
6.5	25
7	12

NEXT

Natural Cubic Splines - More than 3 Knots

In the SPLINEDATA example there were KN=3 knots, giving one spline (added to 1 and X)

In general, if there are KN ($\geq$ 3) knots, then KN-2 splines (in addition to 1 and X)

If KN=5, then, using SAS notation, there are: 1, X, X_spl2, X_spl3, X_spl4

Formula for splines: X_spl2, X_spl3, X_spl4 depends on location of knots … See Appendix

Normally, KN $\leq$ 5 is adequate for a predictor X when fitting a Logistic Model.

There are several options for SPLINES in PROC LOGISTIC and the full details are confusing.
For discussion: SAS/STAT® 14.2 User's Guide Shared Concepts and Topics, Ch 19 Shared Concepts and Topics, pp 405-413. https://support.sas.com/documentation/onlinedoc/stat/142/introcom.pdf

SELF-STUDY

Suppose KN=5 and SELECTION = FORWARD is used.

Modeler can require ALL X_spl2, X_spl3, X_spl4 to be <u>either</u> IN <u>or</u> OUT of Logistic Model

OR

Allow FORWARD to select some of these for the Model

For ALL IN or ALL OUT, do this:
Add an EFFECT statement: EFFECT <your collection name> = COLLECTION (your var-list);
MODEL Y = <your collection name> <other vars> / SELECTION = etc. ;

How Many Knots and Locations?

NEXT

KNOTMETHOD: In addition to "LIST" there are other options:

KNOTMETHOD=PERCENTILES(**KN**) where KN is number of knots

- For KN=4: Knots are placed at 20th, 40th, 60th, 80th percentiles … 2 Splines and X
- For KN=5: Knots are placed at 16.7th, 33.3th, 50th, 66.7th, 83.3th percentiles … 3 Splines and X

KNOTMETHOD=PERCENTILELIST(list of numbers with format nn.n)

F. Harrell recommends [RMS ch. 2]:

PERCENTILELIST(5 35 65 95) for 4 knots

PERCENTILELIST(5 27.5 50 72.5 95) for 5 knots

See R. Wicklin "Regression with restricted cubic splines in SAS" for discussion

https://blogs.sas.com/content/iml/2017/04/19/restricted-cubic-splines-sas.html

Fit German Bank Data using all 1000 rows for the Analysis Dataset

- Splines for continuous numeric X's (age, credit_amount, duration)
- CLASS statement for 5 remaining categorial X's

Preliminary Step: Create "Spline Design" for later usage

```
PROC LOGISTIC DATA = IOWA23.bank_german_data_v2 desc
OUTDESIGN = Spline_Design; /* design matrix */
EFFECT age_spl = spline( age / details naturalcubic basis=tpf(noint)
knotmethod=PERCENTILES(4));
EFFECT credit_amount_spl = spline( credit_amount / details naturalcubic basis=tpf(noint)
knotmethod=PERCENTILES(4));
EFFECT duration_spl = spline( duration / details naturalcubic basis=tpf(noint)
knotmethod=PERCENTILES(4));
MODEL Y = age_spl credit_amount_spl duration_spl;
run;
DATA IOWA23._6_Data; MERGE IOWA23.bank_german_data_v2 Spline_Design; /* No BY statement needed */
PROC PRINT DATA = IOWA23._6_Data(obs=2); var age: credit_amount: duration: ;
run;
```

OUTDESIGN saves Splines for the 3 X's

4 percentile knots creates 2 cubic splines plus raw predictor [and use "noint"]

The "splined" X's are put in MODEL

extreme value

Extreme's will be a problem if using LASSO

Obs	age	age_spl1	age_spl2	age_spl3	credit_amount	credit_amount_spl1	credit_amount_spl2	credit_amount_spl3	duration	duration_spl1	duration_spl2	duration_spl3
1	67	67	940	540	1169	1169	0	0	6	6	0	0
2	22	22	0	0	5951	5951	14341802	7923549	48	48	936	675

NEXT

Why create Spline Design and Merge to master file?

HPLOGISTIC and HPGENSELECT do not create splines.

- Use PROC LOGISTIC to create Spline Design dataset
- MERGE Spline Design dataset to master file before running HPLOGISTIC or HPGENSELECT.
- Enables SELECT and CHOOSE features of HPLOGISTIC/HPGENSELECT to be used with splines.

Another Reason to create Spline Design:

```
PROC LOGISTIC DATA = <your data> desc;
EFFECT X_spl = spline( X / details naturalcubic basis=tpf(noint)
knotmethod=PERCENTILES(4));
MODEL Y = X_spl;
output out = scored p = predict;
score data= <your data> out = scored2;
run;
```

Splines not Saved

Splines not Saved

Here are the X's and d.f.'s that are available for Model Fit

VAR_NAME	Levels
checking_status	4
credit_history	5
property_magnitude	4
purpose (collapsed)	7
savings	5
TOTAL=	25

predictors	5
levels	25
degrees of freedom	**20**

\+

age spline	**3**
duration spline	**3**
credit_amount spline	**3**

=

29 coefficients

Barely satisfies K in: 1000 ≥ 10 * K / 0.3 ➔ max K = 30

Fit German Bank using PROC LOGISTIC BACKWARD, SLS=0.05

```
%LET C_VARS = checking_status credit_history property_magnitude purpose savings;
PROC LOGISTIC DATA = IOWA23._6_Data desc;
CLASS &C_VARS;
MODEL Y = &C_VARS age_spl: credit_amount_spl: duration_spl:
/ SELECTION=BACKWARD SLS=.05;
SCORE DATA = IOWA23._6_Data OUT=SCORED FITSTAT;
TITLE1 "_6_Logistic_Backward_with_Splines.sas";
run;
```

":" Include all VARs with prefix

FITSTAT Creates Report see next slide

SCORED includes Model Probability called P_1 and Y

NEXT

BACKWARD is good because it gives the full model as a reference point. … F. Harrell

The "Apparent Model" ... Is it any good? Needs validation !!!

NEXT

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	-2.0936	0.3755	31.0924	<.0001
credit_amount_spl1		1	-0.00039	0.000127	9.6815	0.0019
credit_amount_spl3		1	1.992E-7	5.304E-8	14.1079	0.0002
duration_spl1		1	0.1038	0.0211	24.2721	<.0001
duration_spl2		1	-0.00254	0.000788	10.3569	0.0013
checking_status	A11	1	0.7594	0.1429	28.2450	<.0001
checking_status	A12	1	0.3879	0.1455	7.1103	0.0077
checking_status	A13	1	-0.2170	0.2502	0.7521	0.3858
credit_history	A30	1	0.6847	0.3165	4.6817	0.0305
credit_history	A31	1	0.6914	0.2864	5.8288	0.0158
credit_history	A32	1	-0.1734	0.1498	1.3402	0.2470
credit_history	A33	1	-0.3360	0.2378	1.9969	0.1576
purpose	A40	1	0.4979	0.1740	8.1834	0.0042
purpose	A41	1	-1.0168	0.2766	13.5137	0.0002
purpose	A42	1	0.0772	0.1897	0.1656	0.6840
purpose	A43	1	-0.2814	0.1781	2.4969	0.1141
purpose	A45	1	0.2892	0.4381	0.4359	0.5091
purpose	A46	1	0.5684	0.2958	3.6913	0.0547
savings	A61	1	0.5446	0.1611	11.4303	0.0007
savings	A62	1	0.3262	0.2342	1.9393	0.1637
savings	A63	1	0.0758	0.3212	0.0557	0.8134
savings	A64	1	-0.5502	0.3873	2.0185	0.1554
TOTAL parameters =		**21**				

SELECTIONS by BACKWARD

- For credit_amount, spline1 and spline3 entered.
- For duration spline3 did not enter.
- No spline for age entered
- Four categorical X's entered.

Fit Statistics for SCORE Data

Data Set	Total Freq	AUC(*)	Brier Score(**)
IOWA23._6_DATA	1000	0.808	0.156

* AUC is c-Statistic ... 0.808 is good (**too good !!**)

** Brier Score is Average Squared Error

Quick Review of Bootstrap Sampling

Explain Bootstrap Sampling

Suppose dataset BOOT has 5 observations.
A bootstrap sample from BOOT is formed by 5 random picks from BOOT with Replacement.
e.g. if BOOT = {0, 1, 2, 3, 4}, then one possible bootstrap sample is {0, 0, 1, 3, 4}

PROC SURVEYSELECT can perform bootstrap sampling. Here are two bootstrap samples:

```
DATA BOOT;
   DO X = 0 to 4;
   OUTPUT;
   END;
PROC SURVEYSELECT DATA=BOOT
OUT=BootSamples
NOPRINT /* Don't print a summary of sampling */
SEED=111
METHOD=URS /* with replacement */
SAMPRATE=1 /* Sample size = 100% (size of Boot) */
REPS=2 /* Create two bootstrap samples */
;
PROC PRINT DATA=BootSamples;
run;
```

Obs	Replicate	X	NumberHits
1	1	0	1
2	1	1	1
3	1	2	1
4	1	3	1
5	1	4	1
6	2	0	1
7	2	2	1
8	2	3	3

NEXT

Validate the Modeling *Process*

"Optimism-Corrected Performance" has the objective of giving an unbiased performance measurement of entire Modeling Process.

Our Modeling Process (*):

- Predictor selection via Backward (predictor selection bias)
- Fitting coefficients of final model (coefficient overfitting bias)

➔ Look at c-Statistic …

… Use Optimism Correction to correct the c-Statistic

(*) Add any other steps that occur in the Modeling Process.

Optimism Correction for German Bank Model … 0.808 is too good!

1. **c-Stat$_{app}$** for the Apparent Model on analysis dataset: Backward with SLS=0.05

```
%LET C_VARS = checking_status credit_history property_magnitude purpose savings;
PROC LOGISTIC DATA = IOWA23._6_Data desc;
CLASS &C_VARS;
MODEL Y = age_spl: credit_amount_spl: duration_spl: &C_VARS / SELECTION=BACKWARD SLS=.05;
SCORE DATA = IOWA23._6_Data OUT=SCORED FITSTAT;
```

c-Stat$_{app}$ = 0.808333 (where "app" = apparent)

2. Compute 200 c-Stat$_{boot}$ from fitting Models to 200 bootstraps using Backward with SLS = 0.05 (MODELs use FREQ statement to count number of “hits” in the bootstrap samples)

Average of c-Stat$_{boot}$ = 0.827444

3. Compute 200 c-Stat$_{orig}$ by scoring 200 models from #2 on the original (1000) analysis dataset.

Average of c-Stat$_{orig}$ = 0.798250

4. c-Stat$_{optimism}$ = c-Stat$_{boot}$ - c-Stat$_{orig}$ = 0.827444 - 0.798250 = 0.029194

5. Optimism-Corrected Performance = c-Stat$_{app}$ - c-Stat$_{opt}$ = 0.808333 - 0.029194 = **0.779139**

[RMS] recommends that Optimism-Corrected **0.779139** be reported as the validation statistic. Same process can be applied to ASE and other validation stats. We'll try this on Lift Charts later.

The 200 bootstrap Models have different X's

The bootstrap models did not always select the same number of predictors for the 200 models
... see report from BACKWARD SLS=0.05 ➔

Number of Effects In Model		
Number In Model	Frequency	Percent
6	2	1
7	8	4
8	34	17
9	45	22.5
10	56	28
11	39	19.5
12	15	7.5
13	1	0.5

References: Optimism Correction

F. Harrell references theoretical work by Bradley Efron to justify this methodology [RMS, p114]:

B. Efron. Estimating the error rate of a prediction rule: Improvement on cross validation. J Am Stat Assoc, 78:316–331, 1983.
B. Efron. How biased is the apparent error rate of a prediction rule? J Am Stat Assoc, 81:461–470, 1986.

I looked at these papers briefly … both utilize advanced mathematical statistics … I quickly gave up.

This SAS Global Forum paper discusses the Optimism Correction and provides a Macro for computing Optimism Correction

I. Stijacic Cenzer, Y. Miao, K. Kirby, W. J. Boscardin (2013) "Estimating Harrell’s Optimism on Predictive Indices Using Bootstrap Samples", SAS Global Forum.

Validate the Final Modeling "Process"

Illustrating the Optimism-Corrected Lift Charts

Review the construction of LIFT CHART

Let's review how to construct a Lift Chart … recall the Apparent Model:

```
PROC LOGISTIC DATA = IOWA23._6_Data desc;
CLASS &C_VARS;
MODEL Y = age_spl: credit_amount_spl: duration_spl: &C_VARS / SELECTION=BACKWARD SLS=.05;
SCORE DATA = IOWA23._6_Data OUT=SCORED
```

RANKP (groups)	_FREQ_	P_1	meanY =event rate
ALL	1000	0.3	0.3
0	125	0.735	0.728
1	125	0.532	0.576
2	125	0.404	0.352
3	125	0.291	0.320
4	125	0.198	0.168
5	125	0.128	0.144
6	125	0.078	0.072
7	125	0.035	0.040

1. **SCORED**: Includes P_1 (model probability) and Y
2. Use P_1 to put the observations in 8 groups (Why "8" … see the Appendix for guidelines)
   - SORT by descending P_1 (actually use PROC RANK)
   - Slice into 8 equal groups
3. Ranks are called RANKP = 0 … RANKP = 7
   - Highest P_1 are in RANKP = 0
   - Lowest P_1 are in RANKP = 7
4. Column "meanY" = "event rate" measures how well Model "discriminates" between events and non-events
   - This Looks Good … 0.728 >> 0.040
   - But it is TOO GOOD !!!
   - Need to correct for optimism (bias)

NEXT

SAS Code for Lift Chart … Leave as Self Study

```
%LET C_VARS =
checking_status credit_history property_magnitude purpose savings;
ods exclude all;
PROC LOGISTIC DATA = IOWA23._6_Data desc;
CLASS &C_VARS;
MODEL Y =
age_spl: credit_amount_spl: duration_spl: &C_VARS
/ SELECTION=BACKWARD SLS=.05;
SCORE DATA = IOWA23._6_Data
OUT=SCORED FITSTAT;
TITLE1 "_6_Logistic_Backward_with_Splines.sas";
run;
ods exclude none;
PROC RANK DATA= SCORED OUT= RANKOUT
GROUPS=8 DESCENDING;
VAR P_1; /* variable that is ranked */
RANKS RANKP; /* name of ranks */
run;
PROC MEANS DATA= RANKOUT NOPRINT;
CLASS RANKP; VAR P_1 Y;
OUTPUT OUT= MEANOUT
MEAN= PREDICT meanY;
run;
PROC PRINT DATA= MEANOUT;
run;
```

Fit Model as before (same results). Output **P_1** (probabilities) into dataset SCORED

See Appendix for the reason why **8** Groups were selected.
Put highest **P_1** in RANKP=0, next highest in RANKP=1, … lowest **P_1** in RANKP=7

This is the "LIFT CHART" … MEANOUT

Correct for Optimism in Lift Charts - German Bank Model

Reuse the 200 bootstrap samples to compute Lift Charts and take Averages.

- "Bootstrap" MINUS "Scored" gives Optimism for Y

Bootstrap Models		
RANKP	PREDICT_B	Y_B
ALL	0.298	0.298
0	0.768	0.759
1	0.552	0.565
2	0.404	0.415
3	0.279	0.273
4	0.183	0.177
5	0.115	0.111
6	0.065	0.066
7	0.026	0.026

Scored Boot on All Data		
RANKP	PREDICT	Y
ALL	0.299	0.300
0	0.769	0.709
1	0.552	0.546
2	0.404	0.413
3	0.279	0.279
4	0.183	0.190
5	0.114	0.130
6	0.065	0.090
7	0.026	0.043

Optimism	
PREDICT	Y
-0.00058	-0.00155
-0.00036	0.04932
-0.00022	0.01952
-0.00013	0.00163
0.00034	-0.00634
0.00021	-0.01256
0.00034	-0.01847
0.00028	-0.02374
0.00006	-0.01726

NEXT

Optimism Corrected Lift Charts - German Bank Model

Apparent Lift Chart (original model on full data)

Subtract Optimism … this gives the Optimism Corrected Lift Chart

- Discrimination (Y column) is a little less but still good (0.679 >> 0.057)
- Calibration (agreement between PREDICT and Y) is still fairly good

Overall, good!

Apparent Lift		
RANKP	PREDICT	Y
ALL	0.3	0.3
0	0.735	**0.728**
1	0.532	0.576
2	0.404	0.352
3	0.291	0.320
4	0.198	0.168
5	0.128	0.144
6	0.078	0.072
7	0.035	**0.040**

minus

Optimism		
RANKP	PREDICT	Y
ALL	-0.00058	-0.00155
0	-0.00036	0.04932
1	-0.00022	0.01952
2	-0.00013	0.00163
3	0.00034	-0.00634
4	0.00021	-0.01256
5	0.00034	-0.01847
6	0.00028	-0.02374
7	0.00006	-0.01726

=

Optimism Corrected		
RANKP	PREDICT	Y
ALL	0.301	0.302
0	0.735	**0.679**
1	0.532	0.556
2	0.404	0.350
3	0.291	0.326
4	0.198	0.181
5	0.128	0.162
6	0.078	0.096
7	0.035	**0.057**

Optimism Corrected - German Bank Model

We decide optimism corrected performance is good. … this Model is accepted !!

My very unpolished SAS code for Optimism-Corrected Performance calculation (C-statistic and LIFT) is in Appendix

NEXT

German Bank data was oversampled … has biased Intercept

Weight the Apparent Model (i.e. final X's) back to Population to correct the biased intercept (*)

But can't run the Weighted Model unless we "make up" facts about GERMAN BANK Population.

Suppose population is size 50,000 and default rate = 0.05 & non-default rate = 0.95

Then defaults = 50000*0.05 = 2500 and non-defaults = 50000*0.95 = 47500.

To weight the Analysis Dataset sample back to the Population:

A. Weight for defaults in sample: (2500)/300 = 8.33
= (Defaults in POP)/(Defaults in Sample) … 1 default projects back to 8.33 in POP

B. Weight for non-default in sample: (47500)/700 = 67.86

If Y=1 then wgt = 8.33 and if Y=0 then wgt = 67.86

```
PROC LOGISTIC DATA = IOWA23._6_Data ;
WEIGHT wgt;
MODEL Y = <the final X's as selected by the apparent model>;
```

NOW: Use the Weighted Model for scoring new datasets … MOVE TO PRODUCTION.

(*) See HLS p. 231 for theory regarding oversampling and weighting a Logistic Model

SELF_STUDY … Rescale Optimism Corrected Lift Chart after weighting

		Optimism Corrected Before Weighting		Optimism Corrected After Weighting	
RANKP	_FREQ_	PREDICT	MeanY	PREDICT	MeanY
	(A)	(B)	(C)	(D)	(E)
ALL	1000	0.301	0.302	0.050	0.050
0	125	0.735	0.679	0.254	0.206
1	125	0.532	0.556	0.123	0.133
2	125	0.404	0.350	0.077	0.062
3	125	0.291	0.326	0.048	0.056
4	125	0.198	0.181	0.029	0.026
5	125	0.128	0.162	0.018	0.023
6	125	0.078	0.096	0.010	0.013
7	125	0.035	0.057	0.004	0.007

Formula to Compute PREDICT and MeanY After Weighting

PREDICT Numerator = A*B*(2500/300)

PREDICT Denominator = A*B*(2500/300) + A*(1-B)*(47500/700)

D = Numerator / Denominator

MeanY Numerator = A*C*(2500/300)

MeanY Denominator = A*C*(2500/300) + A*(1-C)*(47500/700)

E = Numerator / Denominator

Optimism-Corrected, after Weighting, is reported as the unbiased Lift Chart

Optimism-Corrected c-Statistic is essentially unchanged due to weighting.

Get the Slides

We have now met the Core Goal of the talk

… to illustrate optimism correction of validation metrics

If TIME permits,

then let's begin to discuss LASSO for Logistic Models

PROC HPGENSELECT provides LASSO for Logistic Models

… PROC's LOGISTIC, HPLOGISTIC do not provide LASSO.

LASSO for fitting logistic models

One “Penalized Maximum Likelihood” method for fitting a logistic model is LASSO (least absolute shrinkage and selection operator). Here is a description:

If there are predictors X_1 to X_K, then …

Given any $\lambda \geq 0$ there is a LASSO model where the coefficients are found as follows:

Let b_0 be an intercept and $\underline{\mathbf{b}} = (b_1, \ldots, b_K)$ be the coefficients for the X's

Vary (b_0, $\underline{\mathbf{b}}$) in order to **minimize:** $-\text{Log}(L) + \lambda * \sum_{j=1}^{K} |b_j|$

… NOTE: the sum $\lambda * \sum_{j=1}^{K} |b_j|$ does not include the intercept.

Minimum gives us the LASSO $\hat{\beta}$’s for this λ … a model for each λ … an infinite number of models!

if $\lambda=0$, then $\hat{\beta}$’s are MLE (… maximized Log(L))

Which λ gives "best" model? Need criterion … some choices: minimum AIC, BIC ASE, max c-Stat

AIC = -2*Log-Likelihood + 2*(K+1) … K = d.f. in model excluding intercept
BIC = -2*Log-Likelihood + log(n)*(K+1)

LASSO finds biased $\hat{\beta}$’s for a logistic model … $\hat{\beta}$’s forced down in absolute value by penalty term.
But may provide better P’s on VALIDATION since variability of the $\hat{\beta}$’s is decreased.

More about LASSO method and HPGENSELECT parameters

RECALL: For $\lambda > 0$ ... the (b_0, **b**) are found which **minimize** $-\text{Log}(L_{(b0,\ \underline{\mathbf{b}})}) + \lambda * \sum_i^K | b_i |$

- For HUGE λ the only way to minimize LASSO objective is to set $\sum_i | b_i | = 0$ ... i.e. each $b_i = 0$
  - Let "Λ" be the *smallest* λ where $b_i = 0$ for all $j > 0$
  - As $\Lambda \to \lambda \to 0$, some of the b's become non-zero (one at a time or in groups)

The GOAL is to find the λ giving "best model" ... Today "best" will be Minimum AIC

A sequence of λ's is needed where the LASSO objective function is evaluated (can't be infinite!!)
... λ_1, λ_2, ..., λ_{20}, ... λ_{end}

Use LASSORHO to start a sequence ... allowed values 0 < LASSORHO < 1

- The first λ in the sequence is LASSORHO * Λ
- The j^{th} λ in the sequence is given by $\text{LASSORHO}^{j} * \Lambda$

LASSOSTEPS = Number of steps ... default = 20

Then here is the sequence of lambda's: $\text{LASSORHO}^{1} * \Lambda$ ... $\text{LASSORHO}^{20} * \Lambda$

The Chosen Model among these 20 models is the one giving minimum AIC

See Appendix for Discussion of some HPGENSELECT Options

A more complex algorithm called "**Group Lasso**" is actually used by HPGENSELECT.
Group LASSO handles X's in CLASS X's. See documentation.

Back to GERMAN BANK

Fit a LASSO Model ... Using same X's as used when fitting the Apparent Model:
Splines for AGE, CREDIT_AMOUNT, DURATION and 5 Categorical X's

```
%LET C_VARS = checking_status credit_history property_magnitude purpose savings;
PROC HPGENSELECT Data= IOWA23._6_Data
LASSORHO=.8 LASSOSTEPS=60;
CLASS &C_VARS / PARAM=REF REF=FIRST;
MODEL Y (descending) = &C_VARS age_spl: credit_amount_spl: duration_spl:
/ DISTRIBUTION= BINARY;
SELECTION METHOD=LASSO (CHOOSE=AIC STOP=NONE)
DETAILS= ALL; ID Y;
OUTPUT OUT = SCORED P=PREDICT;
run;
PROC LOGISTIC DATA = SCORED desc;
MODEL Y = PREDICT;
run;
DATA SCORED; SET SCORED;
ASE = (PREDICT - Y)**2;
run;
PROC MEANS DATA = SCORED MEAN; VAR ASE;
run;
```

Without PARTITION, no FITSTAT report
← c-Stat and Average Squared Error computed here

AIC = -2*Log(L) + 2(K+1)
Theory says: Best Model has minimum AIC

A Terrible Model … !!!!

Selection Details					
Step	Description	Effects In Model	Lambda	AIC	BIC
0	Initial Model	1	1	1223.729	1228.636
1	credit_amount_spl2 entered	2	0.8	1218.169	1227.985
2		2	0.64	1213.011	1222.827
3	coefficient of	2	0.512	1211.390	1221.205
4	credit_amount_spl2	2	0.4096	1209.562	1219.378
5	changes at each step	2	0.3277	1208.061	1217.877
6		2	0.2621	1206.973	1216.788
7		2	0.2097	1206.230	1216.046
8		2	0.1678	1205.739	1215.554
9	Stop here for AIC (also BIC)	2	0.1342	**1205.418***	1215.233
10	credit_amount_spl3 entered	3	0.1074	1207.208	1221.931

- LASSO is sort-of like FORWARD.
- As “Lambda” decreases, the X’s appear
- Coefficients change for each Lambda
- “Choose” the model at Minimum AIC

← THIS MODEL
c = 0.553
ASE = 0.205

Our First Model
c = 0.808
ASE = 0.156

```
/* Out First Model */
PROC LOGISTIC DATA = IOWA23._6_Data desc;
CLASS &C_VARS;
MODEL Y
= age_spl: credit_amount_spl: duration_spl: &C_VARS
/ SELECTION=BACKWARD SLS=.05;
run;
```

Natural Cubic Spline is Problem …

… the Spline "credit_amount_spl2" is skewed and with a big spike … distorts LASSO Penalty.

- Standardization doesn’t correct shape sufficiently. Preliminary Log Transformations didn't help
- For discussion of the problem, see Appendix "Splines and LASSO"

Appendices

- Appendix 1: Rules for Constructing the Spline Formulas
- Appendix 2: Guideline as to number of ranks in Lift Chart
- Appendix 3a-3d: SAS code for Optimism-Corrected Performance
- Appendix 4a-4d: Discussion of more HPGENSELECT options
- Appendix 5a-5h: Splines and LASSO

blund_data@mi.rr.com and blund.data@gmail.com

Send an email for a copy of slides

Appendix 1: Rules for Constructing the Spline Formulas

Let X be a continuous predictors with numerous levels

If KN knots are selected, there are KN-1 splines created by PROC LOGISTIC with the EFFECT statement. Here is how to construct them:

Call the knot values: $\xi_1, \xi_2, \ldots, \xi_{KN}$

First, $X\_spl_1 = X$. Then the formulas for $X\_spl_2, \ldots, X\_spl_{KN-1}$ are below:

$$X\_spl_k = [(X - \xi_{k-1})_+^3 - (X - \xi_{KN})_+^3]/(\xi_{KN} - \xi_{k-1}) \;-\; [(X - \xi_{KN-1})_+^3 - (X - \xi_{KN})_+^3]/(\xi_{KN} - \xi_{KN-1})$$

for k = 2 to KN

Example: Assign knots: 1, 3, 4, 7 so KN=4. For example, look at X_spl_2 and X_spl_3:

$$X\_spl_2 = [(X - 1)_+^3 - (X - 7)_+^3]/(7 - 1) \;-\; [(X - 4)_+^3 - (X - 7)_+^3]/(7 - 4)$$

$$X\_spl_3 = [(X - 3)_+^3 - (X - 7)_+^3]/(7 - 3) \;-\; [(X - 4)_+^3 - (X - 7)_+^3]/(7 - 4)$$

X_spl_2 and X_spl_3 are linear after 7 and both have 2nd derivatives across all X

Appendix 2: Guideline as to number of ranks in Lift Chart

Lift Chart with "too many" Ranks makes Lift Separation look "too good" … Unrealistically high %Y's in top rank

Below is an ad-hoc guideline to flag when a Lift Chart has excess ranks. To avoid "too many ranks" the number of ranks should satisfy these 2 heuristics:

A. For each rank in a Lift Chart, P lies inside (Y - 1.28*SD(Y), Y + 1.28*SD(Y)).

where SD(Y) = SQRT(Y*(1-Y) / Freq)

B. Each Y should be less than 1.05 times the preceding Y: That is: $Y_{r+1} / Y_r < 1.05$ (to avoid serious flip-flops)

There are two reasons why (A) or (B) might fail.

(1) BAD LUCK: %Y=1's vary randomly within a rank when scoring on the Validation dataset
(2) Model is POORLY FIT

The conditions (A) and (B) rule out (2) but still allow for some (1) Bad Luck

RANKP	_FREQ_	P	Y	SD(Y)	LOW	HIGH	In or Out	Y Ratio
ALL	1000	0.3	0.3		Y +/- 1.28*SD			
0	125	0.735	0.728	0.040	0.677	0.779	P IN	
1	125	0.532	0.576	0.044	0.519	0.633	P IN	0.79
2	125	0.404	0.352	0.043	0.297	0.407	P IN	0.61
3	125	0.291	0.32	0.042	0.267	0.373	P IN	0.91
4	125	0.198	0.168	0.033	0.125	0.211	P IN	0.53
5	125	0.128	0.144	0.031	0.104	0.184	P IN	0.86
6	125	0.078	0.072	0.023	0.042	0.102	P IN	0.50
7	125	0.035	0.04	0.018	0.018	0.062	P IN	0.56

8 ranks are suitable for this Lift Chart

Appendix 3a: SAS code for Optimism-Corrected Performance

```
* _09_Lift_Optimism_of_5_Stepwise;
%LET C_VARS =
checking_status
credit_history
property_magnitude
purpose
savings
;
PROC SURVEYSELECT DATA=IOWA23._6_Data
OUT=BootSamples
NOPRINT SEED=111
METHOD=URS /* Sample with replacement */
SAMPRATE=1 /* Sample rate 100% */
REPS=200; /* Number of boot strap samples */
TITLE1 "_09_Lift_Optimism_of_5_Stepwise";
run;
/* Macro parameter R gives the number of bootstrap samples for computing Optimism */
/* This code does not finish the job of computing Optimism-Corrected Performance */
/* The code only computes Optimism … leaving final step to Modeler */
/* This requires subtracting Optimism from the Apparent Model performance statistics */
```

If interested, contact me for a TEXT file. Easier than trying to copy the SAS code from the PowerPoint.

Appendix 3b: SAS code for Optimism-Corrected Performance

```
%MACRO REP(R);
/* Clean-up Datasets that appear in PROC APPEND */
%IF %SYSFUNC(EXIST(BASE1)) = 1 %THEN %DO;
PROC DELETE DATA = BASE1;
run;
%END;
%IF %SYSFUNC(EXIST(BASE2)) = 1 %THEN %DO;
PROC DELETE DATA = BASE2;
run;
%END;
%IF %SYSFUNC(EXIST(BASE3)) = 1 %THEN %DO;
PROC DELETE DATA = BASE3;
run;
%END;
%IF %SYSFUNC(EXIST(BASE4)) = 1 %THEN %DO;
PROC DELETE DATA = BASE4;
run;
%END;
%IF %SYSFUNC(EXIST(BASE5)) = 1 %THEN %DO;
PROC DELETE DATA = BASE5;
run;
%END;
%IF %SYSFUNC(EXIST(ScoreFitStat1)) = 1 %THEN %DO;
PROC DELETE DATA = ScoreFitStat1;
run;
%END;
%IF %SYSFUNC(EXIST(ScoreFitStat2)) = 1 %THEN %DO;
PROC DELETE DATA = ScoreFitStat2;
run;
%END;
%IF %SYSFUNC(EXIST(NumberinModel)) = 1 %THEN %DO;
PROC DELETE DATA = NumberinModel;
run;
%END;
%IF %SYSFUNC(EXIST(Lift_Chart1)) = 1 %THEN %DO;
PROC DELETE DATA = Lift_Chart1;
run;
%END;
%IF %SYSFUNC(EXIST(Lift_Chart2)) = 1 %THEN %DO;
PROC DELETE DATA = Lift_Chart2;
run;
%END;
```

```
/* */
%DO I = 1 %TO &R;
ods exclude all;
/* Save MODEL information using OUTMODEL = OM */
PROC LOGISTIC DATA = BootSamples(where=(replicate=&I))
desc OUTMODEL = OM;
FREQ NumberHits;
CLASS &C_VARS;
MODEL Y = age_spl: credit_amount_spl: duration_spl: &C_VARS
/ SELECTION=BACKWARD SLS=.05;
SCORE DATA = BootSamples(where=(replicate=&I)) FITSTAT
OUT=SCORED1(keep = Replicate Y P_1 NumberHits);
ods output ScoreFitStat = ScoreFitStat1;
ods output ConvergenceStatus = ConvergenceStatus;
ods output ModelBuildingSummary=ModelBuildingSummary;
run;
* Create Lift Charts for Boot Sample Models;
PROC RANK DATA= SCORED1
OUT= RANKOUT1
GROUPS=8 DESCENDING;
VAR P_1; /* variable that is ranked */
RANKS RANKP; /* name of ranks */
PROC MEANS DATA= RANKOUT1 NOPRINT;
FREQ NumberHits;
CLASS RANKP; VAR P_1 Y Replicate;
OUTPUT OUT= Lift_Chart1 MEAN= PREDICT_B Y_B Replicate;
run;
* END Create Lift Charts for Bootstrap Sample Models;
/* For information: Record number of predictors in each Boot Strap Model */
DATA NumberinModel(keep=NumberinModel); Set ModelBuildingSummary end=eof;
if eof then output;
run;
ods exclude none;
```

Appendix 3c: SAS code for Optimism-Corrected Performance

```
/* Perform steps below only if LOGISTIC MODEL on a bootstrap sample converges */
DATA _NULL_; Set ConvergenceStatus;
call symput('converged', status);
run;
%IF &converged = 0 %THEN %DO;
PROC APPEND BASE = BASE1 DATA = ScoreFitStat1;
run;
PROC APPEND BASE = BASE3 DATA = NumberinModel;
run;
PROC APPEND BASE = BASE4 DATA = Lift_Chart1;
run;
ods exclude all;
PROC LOGISTIC INMODEL = OM;
SCORE DATA = IOWA23._6_Data FITSTAT
OUT=SCORED2(keep = Y P_1);
ods output ScoreFitStat = ScoreFitStat2;
run;
ods exclude none;
PROC APPEND BASE = BASE2 DATA = ScoreFitStat2;
run;
* Create Lift Charts from Scoring Full Sample with Boot Model;
PROC RANK DATA= SCORED2 OUT= RANKOUT2
GROUPS=8 DESCENDING;
/* "8" was determined by an external process
... would be wise to make it a macro parameter */
VAR P_1; /* variable that is ranked */
RANKS RANKP; /* name of ranks */
PROC MEANS DATA= RANKOUT2 NOPRINT;
CLASS RANKP; VAR P_1 Y;
OUTPUT OUT= Lift_Chart2 MEAN= PREDICT Y;
DATA Lift_Chart2; SET Lift_Chart2;
Replicate = &I;
run;
* END: Create Lift Charts from Scoring Full Sample with Bootstrap Model;
PROC APPEND BASE = BASE5 DATA = Lift_Chart2;
%END;
%END;
%MEND;

%REP(200);
```

Appendix 3d: SAS code for Optimism-Corrected Performance

```
/* Merge Performance Stats from Bootstrap Model and scoring on full Model */
DATA BOTH; MERGE
BASE1(RENAME = (AUC=c_B BrierScore=ASE_B) DROP=Dataset)
BASE2(RENAME = (AUC=c_F BrierScore=ASE_F) DROP=Dataset)
;
d_B = 2*c_B - 1;
d_F = 2*c_F - 1;
Diff_d = d_B - d_F;
Diff_c = c_B - c_F;
Diff_ASE = ASE_B - ASE_F;
run;
PROC MEANS DATA = BOTH NOPRINT;
VAR Diff_d Diff_c Diff_ASE d_B d_F c_B c_F ASE_B ASE_F;
OUTPUT OUT = MEANOUT
MEAN = Diff_d Diff_c Diff_ASE d_B d_F c_B c_F ASE_B ASE_F;
run;
PROC PRINT DATA = MEANOUT;
VAR Diff_c Diff_ASE c_B c_F ASE_B ASE_F;
TITLE2 "MEANOUT .. Optimism Performance Statistics";
run;
PROC FREQ DATA = BASE3; TABLES NumberinModel;
TITLE2 "Number of predictors in Bootstrap Models";
run;
/* Compute optimism for Lift Charts */
DATA Base4_5; Merge Base4 Base5; by replicate _type_ rankP;
DROP _FREQ_ _TYPE_;
Optimism_Y = Y_B - Y;
Optimism_Predict = Predict_B - Predict;
If RankP = . then RankP = -9;
run;
PROC MEANS DATA = Base4_5 NOPRINT;
Class RankP;
Var Optimism_Y Optimism_Predict Y_B Y Predict_B Predict;
OUTPUT OUT = Optimism_Lift(where=(_TYPE_ = 1))
N = N
MEAN = Optimism_Y Optimism_Predict Y_B Y Predict_B Predict;
PROC PRINT DATA = Optimism_Lift;
Var N _TYPE_ RankP Optimism_Y Optimism_Predict Y_B Y Predict_B Predict;
TITLE2 "Optimism for Lift Chart";
run; TITLE; run;
```

Appendix 4a: HPGENSELECT with LASSO: PARTITION and CHOOSE=VALIDATE

```
PROC HPGENSELECT Data= <your data>
LASSORHO=.8 /* default */ LASSOSTEPS=20; /* default = 20 */
PARTITION ROLEVAR= role (TRAIN="1" VALIDATE="2");
CLASS <C1 cvar> / PARAM=REF REF=LAST;
MODEL Y (descending) = <X1 xvar> <C1 cvar>
/ DISTRIBUTION= BINARY; /*<= specifies logistic */
SELECTION METHOD=LASSO (CHOOSE=VALIDATE STOP=NONE)
DETAILS=ALL; /* No "SELECT=" */
run;
```

PARTITION: Required that <your data> has a variable (here called "role") with values "1" and "2" which identify an observation either as TRAIN or VALIDATE

The MODEL is fitted on TRAIN.

CHOOSE=VALIDATE (using VALIDATE='2')
On VALIDATE, ASE is computed:

- If min ASE is achieved, STOP ... this is the Model
- Performance statistics are computed on VALIDATE

METHOD=LASSO:
For each STEP

- X's might appear, disappear, or not change

... but coefficients do change.

A third TEST dataset is not supported in PARTITION with LASSO ... it is supported for HPGENSELECT with SELECT=SL and also for HPLOGISTIC

Appendix 4b: HPGENSELECT LASSO with CHOOSE=VALIDATE

```
PROC HPGENSELECT Data= <your data>
LASSORHO=.8 /* default */ LASSOSTEPS=20; /* default = 20 */
PARTITION ROLEVAR= role (TRAIN="1" VALIDATE="2");
CLASS <C1 cvar> / PARAM=REF REF=LAST;
MODEL Y (descending) = <X1 xvar> <C1 cvar>
/ DISTRIBUTION= BINARY; /*<= specifies logistic */
SELECTION METHOD=LASSO (CHOOSE=VALIDATE STOP=NONE)
DETAILS=ALL; /* No "SELECT=“ when using LASSO */
run;
```

In this hypothetical example, the final model is reached at STEP 13 … the Validation ASE begins to (slightly) increase at STEP 14.

Selection Details

Step	Description	Effects In Model	Lambda	Validation ASE
0	Initial Model	1	1	0.194
1	C1 entered	2	0.8	0.193
2		2	0.64	0.191
3	coefficients	2	0.512	0.190
4	keep changing	2	0.4096	0.190
5		2	0.3277	0.189
6		2	0.2621	0.189
7		2	0.2097	0.189
8		2	0.1678	0.188
9		2	0.1342	0.188
10		2	0.1074	0.188
11		2	0.0859	0.188
12		2	0.0687	0.188
13	X1 entered	3	0.0550	**0.188***
14		3	0.0440	0.188

Appendix 4c: LASSO … PARAM in CLASS affects Model

See dataset TEST_HPGENSELECT on next slide

```
PROC HPGENSELECT Data= TEST_HPGENSELECT
LASSORHO=.8  LASSOSTEPS=20;
PARTITION ROLEVAR= role (TRAIN="1" VALIDATE="2");
CLASS C34 / PARAM=REF REF=LAST;
MODEL Y (descending) = X7 C34
/ DISTRIBUTION= BINARY; /*<= specifies logistic */
SELECTION METHOD=LASSO (CHOOSE=VALIDATE STOP=NONE)
/* No SELECT= */ DETAILS=ALL;
run;
```

Parameter Estimates		
Parameter	DF	Estimate
Intercept	1	2.058719
C34 0	1	-1.237838
C34 1	1	-0.672698
X7	1	1.583068

Different Models !!

```
PROC HPGENSELECT Data= TEST_HPGENSELECT
LASSORHO=.8  LASSOSTEPS=20;
PARTITION ROLEVAR= role (TRAIN="1" VALIDATE="2");
CLASS C34; /* No PARAM statement !!! */
MODEL Y (descending) = X7 C34
/ DISTRIBUTION= BINARY; /*<= specifies logistic */
SELECTION METHOD=LASSO (CHOOSE=VALIDATE STOP=NONE)
/* No SELECT= */ DETAILS=ALL;
run;
```

Parameter Estimates		
Parameter	DF	Estimate
Intercept	1	1.450512
C34 0	1	-0.649308
C34 1	1	-0.084633
C34 2	1	0.751179
X7	1	1.602003

Without PARAM=REF, there is coefficient for each level of C34

Appendix 4d: Dataset for Appendix 4c

```
DATA TEST_HPGENSELECT;
do ID = 1 to 10000;
If mod(ID,2)=0 Then role=1; Else role=2;
cumLogit = ranuni(2);
e = 1*log( cumLogit/(1-cumLogit ));
X1 = rannor(9);
X2 = rannor(9);
X3 = rannor(9);
X4 = rannor(9);
X5 = rannor(9);
X6 = rannor(9);
X7 = X6*X5;
B1 = (ranuni(1) < .4);
B2 = (ranuni(1) < .6);
B3 = (ranuni(1) < .5);
B4 = (ranuni(1) < .5);
C12 = B1 + B2;
C34 = B3 + B4;
xbeta = X1**2 + log(X2+8) + .01*X3 + 2*X7 + 0.1*B1 + B2 + B3 + B4 + e;
P_1 = exp(xbeta) / (1 + exp(xbeta));
Y = (P_1 > 0.95);
output;
end;
```

Appendix 5a: X's are standardized by HPGENSELECT before LASSO

```
DATA WORK1;
Do i = 1 to 1000;
X1 = rannor(1);
X2 = 5*rannor(1);
Y_Star = X1 + X2 + 2*rannor(1);
Y = (Y_Star > 1);
Output;
End;
/* MODEL 1 */
PROC LOGISTIC
DATA = WORK1 desc;
MODEL Y = X1 X2;
DATA WORK2; SET WORK1;
X2 = X2/2.2;
/* MODEL 2 */
PROC LOGISTIC
DATA = WORK2 desc;
MODEL Y = X1 X2;
run;
```

Changed pounds to kilograms for X2 in WORK2.

MODEL 1	
Parameter	Estimate
Intercept	-0.7917
X1	0.7822
X2	0.8992

-2 Log L	507.021

MODEL 2	
Parameter	Estimate
Intercept	-0.7917
X1	0.7822
X2	1.9782

For LASSO … X's are standardized before fitting:

LASSO objective function is not "scale invariant"

$$\text{Minimize: } -\text{Log}(L) + \lambda * \sum_{j=1}^{K} |b_j|$$

- Log(L) is scale invariant
- $\lambda * \sum_{j=1}^{K} |b_j|$ is not scale invariant … consider

  $\lambda * (b_1 + b_2))$ if X2 not transformed

  versus

  $\lambda * (b_1 + 2.2*b_2))$ if X2 is transformed

To avoid scaling disparities:

HPGENSELECT standardizes numeric X's to mean=0 and standard deviation=1 before running LASSO.

Appendix 5b: Recall that the LASSO model with splines was terrible

Selection Details					
Step	Description	Effects In Model	Lambda	AIC	BIC
0	Initial Model	1	1	1223.729	1228.636
1	credit_amount_spl2 entered	2	0.8	1218.169	1227.985
2		2	0.64	1213.011	1222.827
3		2	0.512	1211.390	1221.205
4		2	0.4096	1209.562	1219.378
5		2	0.3277	1208.061	1217.877
6		2	0.2621	1206.973	1216.788
7		2	0.2097	1206.230	1216.046
8		2	0.1678	1205.739	1215.554
9		2	0.1342	**1205.418***	1215.233
10	credit_amount_spl3 entered	3	0.1074	1207.208	1221.931

← THIS MODEL
c = 0.553
ASE = 0.205

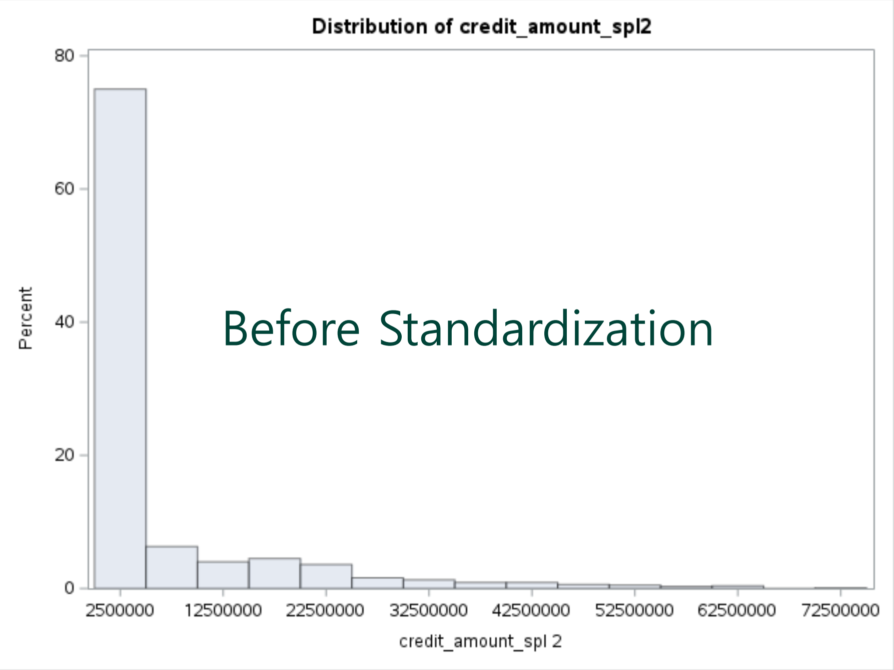
Appendix 5c: credit_amount_spl2 is highly skewed

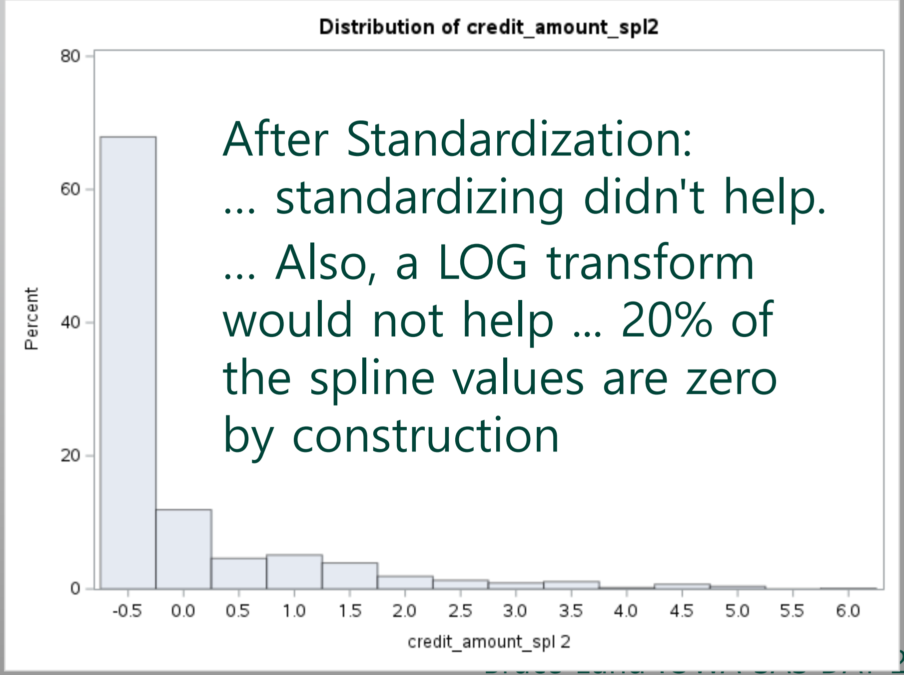
What is going on with credit_amount_spl2?

- Look at histograms of credit_amount_spl2 both before and after standardization.
- Standardization still leaves a strong rightward skew.
- A LOG transform would not help.
- credit_amount_spl2 has 20% zeros, by design, and then rises as a cubic polynomial.

LASSO is thrown off.

```
PROC STANDARD DATA=IOWA23._6_Data (keep=credit_amount_spl2)
MEAN=0 STD=1
OUT=TEMP(rename=(credit_amount_spl2=z_credit_amount_spl2));
VAR credit_amount_spl2;
DATA BOTH; MERGE TEMP IOWA23._6_Data (keep=credit_amount_spl2);
PROC UNIVARIATE DATA=BOTH;
VAR credit_amount_spl2 z_credit_amount_spl2;
HISTOGRAM credit_amount_spl2 z_credit_amount_spl2;
```





Appendix 5d: Try replacing splines with "spline equations"

```
PROC LOGISTIC DATA = IOWA23.bank_german_data_v2 desc;
EFFECT age_spl = spline( age / details naturalcubic basis=tpf(noint)
knotmethod=PERCENTILES(4));
MODEL Y = age_spl;
```

... and run two more PROC LOGISTICS for credit_amount and for duration

Here are splines for AGE:						
Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > Chi Sq
Intercept		1	0.8080	0.8821	0.8391	0.3597
age_spl	1	1	-0.0539	0.0338	2.5402	0.1110
age_spl	2	1	-0.00007	0.0120	0.0000	0.9952
age_spl	3	1	0.00351	0.0188	0.0350	0.8516

Full Spline Equations:

```
age_eqn=-0.0539428429*age_spl1+-0.00007196284*age_spl2+0.00351157992*age_spl3;
ca_eqn=-0.00018549584*credit_amount_spl1+0.00000020615*credit_amount_spl2+-0.0000022312*credit_amount_spl3;
dur_eqn=0.11652691985*duration_spl1+-0.02801529989*duration_spl2+0.03450814874*duration_spl3;
```

Appendix 5e: Use Spline Equations … is this double dipping?

```
DATA TEMP; SET IOWA23._6_Data;
age_eqn=-0.0539428429*age_spl1+-0.00007196284*age_spl2+0.00351157992*age_spl3;
ca_eqn=-0.00018549584*credit_amount_spl1+0.00000020615*credit_amount_spl2+-0.00000022312*credit_amount_spl3;
dur_eqn=0.11652691985*duration_spl1+-0.02801529989*duration_spl2+0.03450814874*duration_spl3;
%LET C_VARS = checking_status credit_history property_magnitude purpose savings;
PROC HPGENSELECT Data= TEMP
LASSORHO=0.8 LASSOSTEPS=60;
CLASS &C_VARS / PARAM=REF REF=FIRST;
MODEL Y (descending) = &C_VARS age_eqn ca_eqn dur_eqn:
/ DISTRIBUTION= BINARY; /*<= specifies logistic */
SELECTION METHOD=LASSO (CHOOSE=AIC STOP=NONE) DETAILS= ALL;
run;
```

Is this double dipping? I don't think so, if the original analysis plan, all along, included this step. …
In this case the relationship of X's to Y did not influence the predictor selection.

Appendix 5f: LASSO with LASSORHO=0.8, with 60 steps

AIC essentially reaches a minimum at step 39. AIC=998.182 from step 39 to the final step 60.

At step 39 the lambda is 0.0002.

So, declare this lambda to give the optimal AIC model

Refit LASSO with LASSORHO=0.0002 and LASSOSTEPS=1 to obtain coefficients and fit statistics.

... see next slide.

Step			Description	Effects In Model	Lambda	AIC
0			Initial Model	1	1.0000	1223.729
1			checking_status entered	3	0.8000	1203.976
			dur_eqn entered	3	0.8000	1203.976
2				3	0.6400	1163.774
3				3	0.5120	1135.040
4				3	0.4096	1113.852
5			ca_eqn entered	4	0.3277	1099.960
6			credit_history entered	5	0.2621	1089.945
7				5	0.2097	1074.931
8			purpose entered	8	0.1678	1078.997
			savings entered	8	0.1678	1078.997
			age_eqn entered	8	0.1678	1078.997
9				8	0.1342	1058.372
10			property_magnitude entered	9	0.1074	1047.637
11				9	0.0859	1034.080
			OMIITTED ROWS			
39				9	0.0002	998.182
			OMIITTED ROWS			
60				9	0.0000	998.182

Appendix 5g: Set LASSORHO= 0.0002 to finalize the Model

```
DATA TEMP; SET IOWA23._6_Data;
age_eqn=-0.0539428429*age_spl1+-0.00007196284*age_spl2+0.00351157992*age_spl3;
ca_eqn=-0.00018549584*credit_amount_spl1+0.00000020615*credit_amount_spl2+-0.00000022312*credit_amount_spl3;
dur_eqn=0.11652691985*duration_spl1+-0.02801529989*duration_spl2+0.03450814874*duration_spl3;
%LET C_VARS = checking_status credit_history property_magnitude purpose savings;
PROC HPGENSELECT Data= TEMP
LASSORHO=0.0002 LASSOSTEPS=1;
CLASS &C_VARS / PARAM=REF REF=FIRST;
MODEL Y (descending) = &C_VARS age_eqn ca_eqn dur_eqn:
/ DISTRIBUTION= BINARY; /*<= specifies logistic */
SELECTION METHOD=LASSO (CHOOSE=AIC STOP=NONE) DETAILS= ALL;
ID Y; OUTPUT OUT = SCORED P=PREDICT;
run;
PROC LOGISTIC DATA = SCORED desc;
MODEL Y = PREDICT;
run;
DATA SCORED; SET SCORED;
ASE = (PREDICT - Y)**2;
run;
PROC MEANS DATA = SCORED MEAN; VAR ASE;
run;
```

Appendix 5h: With LASSORHO set to 0.0002

Selection Details				
Step	Description	Effects In Model	Lambda	AIC
0	Initial Model	1	1	1223.729
1	checking_status entered	9	0.0002	998.720*
	credit_history entered	9	0.0002	998.720*
	property_magnitude entered	9	0.0002	998.720*
	purpose entered	9	0.0002	998.720*
	savings entered	9	0.0002	998.720*
	age_eqn entered	9	0.0002	998.720*
	ca_eqn entered	9	0.0002	998.720*
	dur_eqn entered	9	0.0002	998.720*

All 8 candidates predictors are in this final model

Very small LAMBDA … LASSORHO=**0.0002**
Basically gives the MLE solution

LASSO lambda=0.018	
c-Stat	ASE
0.809	0.157

Compare to solution from BACKWARD SLS=0.05

Logistic Model with BACKWARD SLS=0.05	
c-Stat	ASE
0.808	0.156