Logistic Modeling without Split-Samples v03b

by Bruce Lund

Statistical Trainer, Novi, MI

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

References: See, in particular, RMS and CPM for some topics

- 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., John Wiley & Sons, New York
- Siddiqi, N. (2017). Intelligent Credit Scoring, 2nd edition, Hoboken, NJ, John Wiley & Sons, Inc. Siddiqi influenced the design of SAS Enterprise Miner Credit Scoring application
- (CPM) Steyerberg, E. (2019). *Clinical Prediction Models* 2nd Ed., Springer, Cham, Switzerland
- Van Smeden, et. al. (2017) No rationale for 1 variable per 10 events criterion for binary logistic regression analysis, *BMC Medical Research Methodology*
- Van Smeden, et. al. (2019) Sample Size for binary logistic prediction models: Beyond events per variable criteria, *Statistical Methods in Medical Research*.

First Topics:

- Double Dipping,
- Split-Sampling,
- Bootstrap Sampling with Optimism Correction

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 and using the TRAINING dataset over again for computing model validation statistics (e.g. c-statistic, average squared error, etc.).

Can this problem be solved without a split-sample … a separate sample for VALIDATION ?

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

Dr. Witten was incorrect.
Remember the Seinfeld episode of 1993 where George double-dipped a chip!

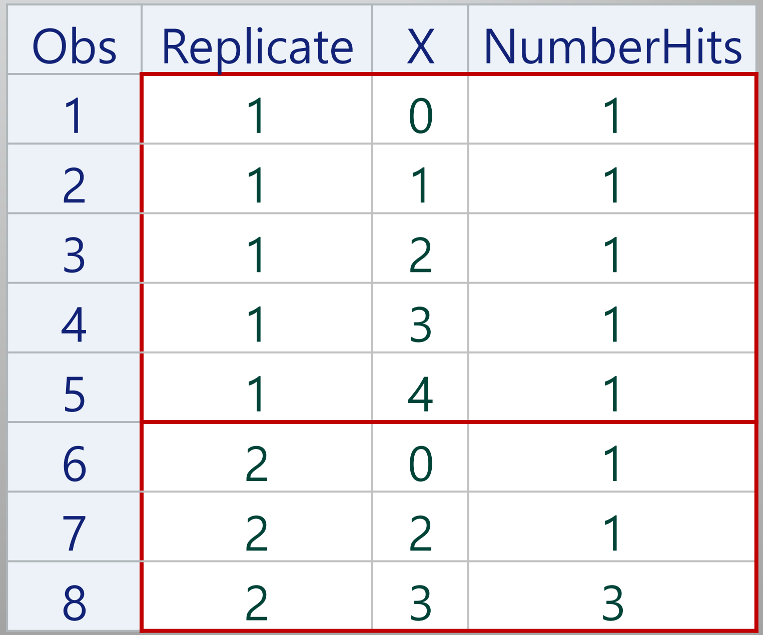
Bootstrap Sampling … appears in later slides

Suppose dataset BOOT has 5 observations.
e.g. if BOOT = {0, 1, 2, 3, 4}, then one possible bootstrap sample is {0, 0, 1, 3, 4}

A bootstrap sample from BOOT is formed by 5 random picks from BOOT with Replacement.

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

Model Fitting with Split-Sample Validation

In a Split-Sample, the Analysis Dataset (a sample from the population) is randomly split into Train and Validation

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

Model fitted on Train and validated on Validation

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

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

Then the chosen Model is validated on Test.

(PROC HPLOGISTIC has such an option)

A compelling rationale for split-sample is that "honest assessment" of performance is obtained.

Validation measures on a validation sample are not affected by work we do on Train ...

(a) exploratory data analysis (looking together at Y and X)

(b) transforming of predictors

(c) repeated attempts at model fitting.

NEXT

NEXT

Sample Size for fitting a Logistic Model

Notation: n1=# of events (cases where Y=1), n0=# of non-events, n = n1+n0 and n1 ≤ n0

Let K = number of X's (i.e. d.f.) estimated for a MODEL (the candidates) … not those selected for the Model.

Let EPP be "events per predictor" … e.g. if n1 =500 and K = 25, then EPP = 500/25 = 20

What is an acceptable minimum for EPP ?

- Often cited rule is n1 / K > EPP=10
  - Harrell [RMS p. 72] suggests n1 / K ≥ EPP=15
  - Steyerberg [CPM p. 55] cites studies that support n1 / K ≥ EPP=10 or 20
- Then we see active research:

  Van Smeden, et. al. (2017). No rationale for 1 variable per 10 events criterion for binary logistic regression analysis
  Van Smeden, et. al. (2019). Sample Size for binary logistic prediction models: Beyond events per variable criteria,

Conclusion: This is a complex subject with no consensus, much less, simple rules.

In some applications (database marketing, credit risk modeling) there are large databases, and minimum EPP is not a concern

Default minimum EPP might be 10.

Model Validation with Split-Sample

What is minimum number of events for Validation dataset?

[RMS, p. 112] and [CPM, p. 57] say Min(n1) = 100 … where n1 ≤ n0

Based on my modeling experience in automotive marketing, I'd set Min(n1) = 250.

When Min(n1) = 250, then a fairly good Lift Chart can be formed

Lift Charts are a basic validation measure for logistic models in marketing and credit risk … Lift Charts are illustrated later.

NEXT

Bootstrap Sampling for Optimism Correction

Harrell, Steyerberg, and others argue that Split-Sample wastes data which can be used to fit more X's or reduce error in $\hat{\beta}$'s ([RMS p. 114] and [CPM p. 107])

… Their alternative is:

Step (1) Model is fitted on the Analysis Dataset. (where TRAIN becomes ANALYSIS DATASET)
Step (2) Model is validated on Analysis Dataset by bootstrap sampling for "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 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
- Computing validation measures 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, honest validation of Model performance is obtained.

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

NEXT

How to do it: Bootstrap Sampling and Optimism Correction

A. Modeling Process is performed on the Analysis Dataset, including computing of Validation Measures.

The resulting Model is called the "Apparent Model".

Suppose we compute a validation measure called "M_{app}" for the Apparent Model

B. Bootstrap samples (assume 200 … recommended by Harrell) are drawn from Analysis Dataset

- The Modeling Process is applied to each Bootstrap sample … X preparation, selection, fitting.
  - Validation measure "M_{boot}" is computed for the bootstrap Model
- Each Model that was fitted to a Bootstrap sample (200) is used to score the Analysis Dataset
  - Validation measure "M_{full}" is computed on the scored full analysis dataset .. each 200
  - For each bootstrap sample: "Optimism" is computed as M_{boot} - M_{full}
  - Average Optimism: $M_{optimism} = \sum(M_{boot} - M_{full}) / 200$

C. Optimism Correction is: $M_{corrected} = M_{app} - M_{optimism}$

The performance measure to be reported is $M_{corrected}$

… An example will be given on later slides

NEXT

Reflection on the prior slide

NEXT

Real World Difficulties:

- Deciding on all steps in the Modeling Process ahead of time
- Coding the steps in a form that allows repeating the Modeling Process 200 times.

Compromises in defining the Modeling Process might be necessary:

Normally, the modeler omits some of the steps in preparing of predictors
But the modeler must seek to minimize these omissions.

Does Bootstrap Sampling with Optimism Correction really work?

Harrell [RMS, p114] references work by Bradley Efron as the original research on this topic:

B. Efron (1983) Estimating error rate of a prediction rule: Improvement on cross validation. JASA.
B. Efron (1986) How biased is the apparent error rate of a prediction rule? JASA.

See Efron and Tibshirani (1993) *An Introduction to the Bootstrap*, pp 247-252 (difficult to read)

Austin and Steyerberg (2017) simulate Optimism Correction applied to the c-Statistic.
Only one dataset and one Model on the dataset. But the simulation was extensive and definitive.
Study showed (for their example):

(i) Bootstrap sampling for optimism correction provided correct c-Statistics when EPP ≥ 20
(ii) Need 2*EPP when 50-50 split-sampling to obtain same c-Statistic as optimism correction (for 1*EPP)

See "Self-Study“ later in the slides, which gives details of A-S study

I performed a much less ambitious simulation study and obtained results consistent with A-S.
See the Appendix for discussion.

A SAS Global Forum paper gives an example of Optimism Correction and provides a SAS Macro

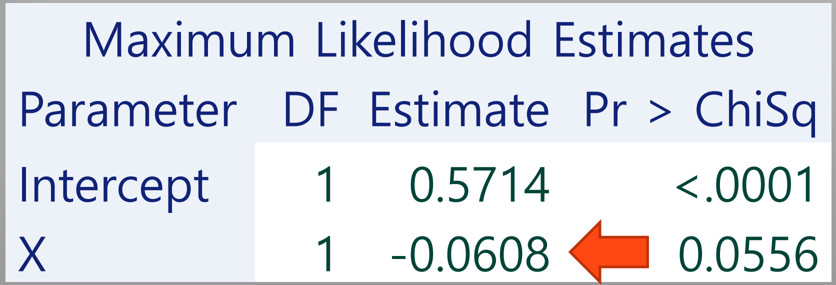
I. Stijacic Cenzer, Y. Miao, K. Kirby, W. J. Boscardin (2013) "Estimating Harrell’s Optimism on Predictive Indices Using Bootstrap Samples", SAS Global Forum.

Now a very different topic …
Splines as Transforms for Continuous Numeric X

- Splines do not require an exploratory examination of X vs. Y
- This is consistent with performing Model Validation by bootstrap sampling and optimism correction

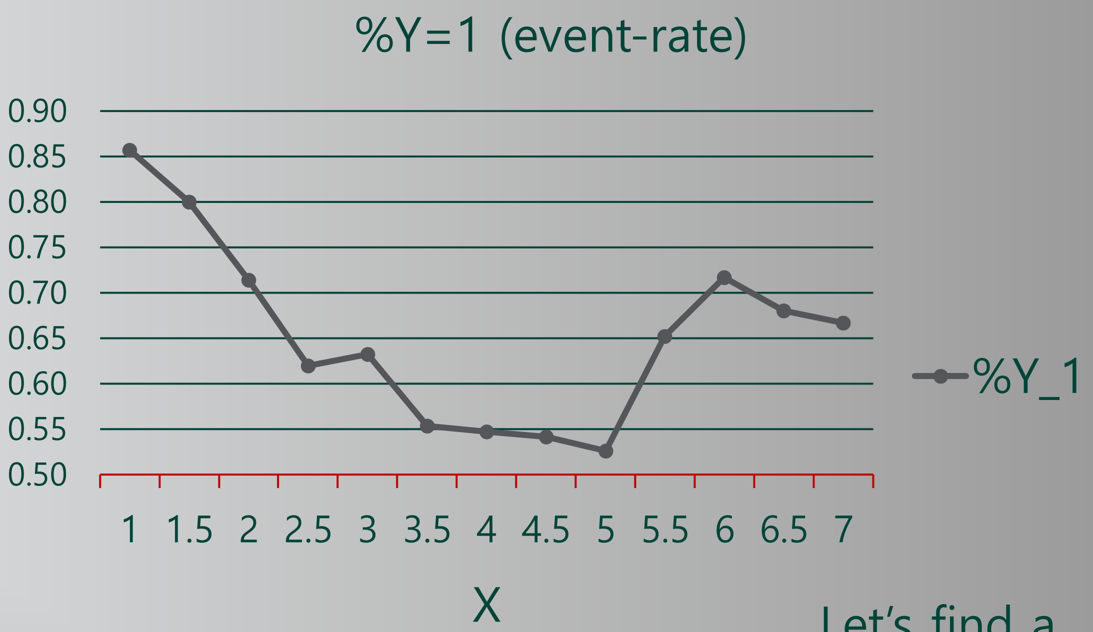
Event-Rate is "U" shaped versus X

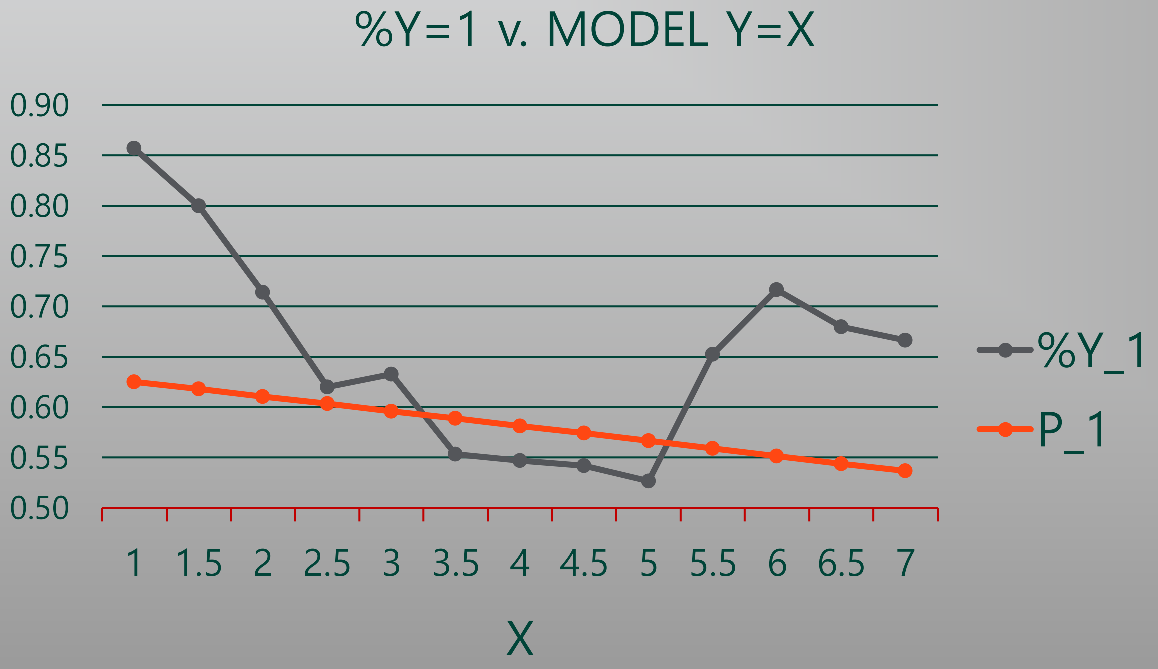
```
DATA SPLINEDATA;
Do ID = 1 to 4000;
   cumLogit = ranuni(2);
   e = 1*log( cumLogit/(1-cumLogit ));
   X = round(rannor(9),.5);
   X = max(min(X,3),-3) + 4;
   xbeta = 1 + (X-4)**2 + 6*e;
   Ps = exp(xbeta) / (1 + exp(xbeta));
   Y = (Ps > 0.50);
   output;
   end;
run;
PROC LOGISTIC
DATA = SPLINEDATA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



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

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





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

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.

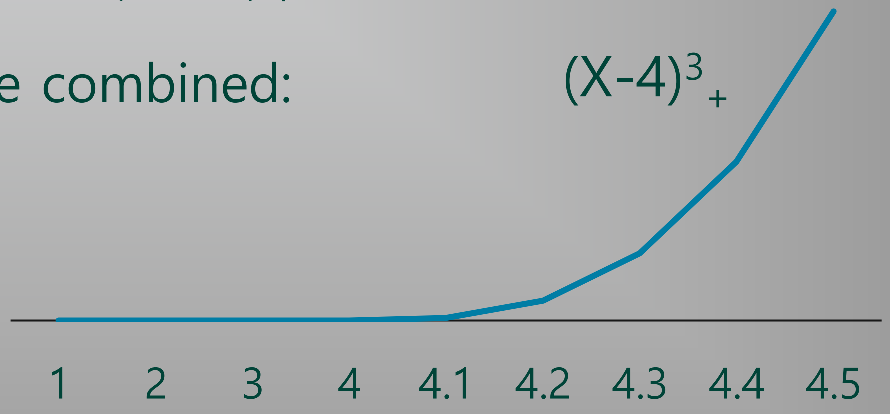
To begin: For each of the 3 knots, a truncated (cubic) power function (TPF) is formed:

TPF(2) = max(0, $(X-2)^3$) = $(X-2)_+^3$ 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)?



NEXT

Natural Cubic Splines

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

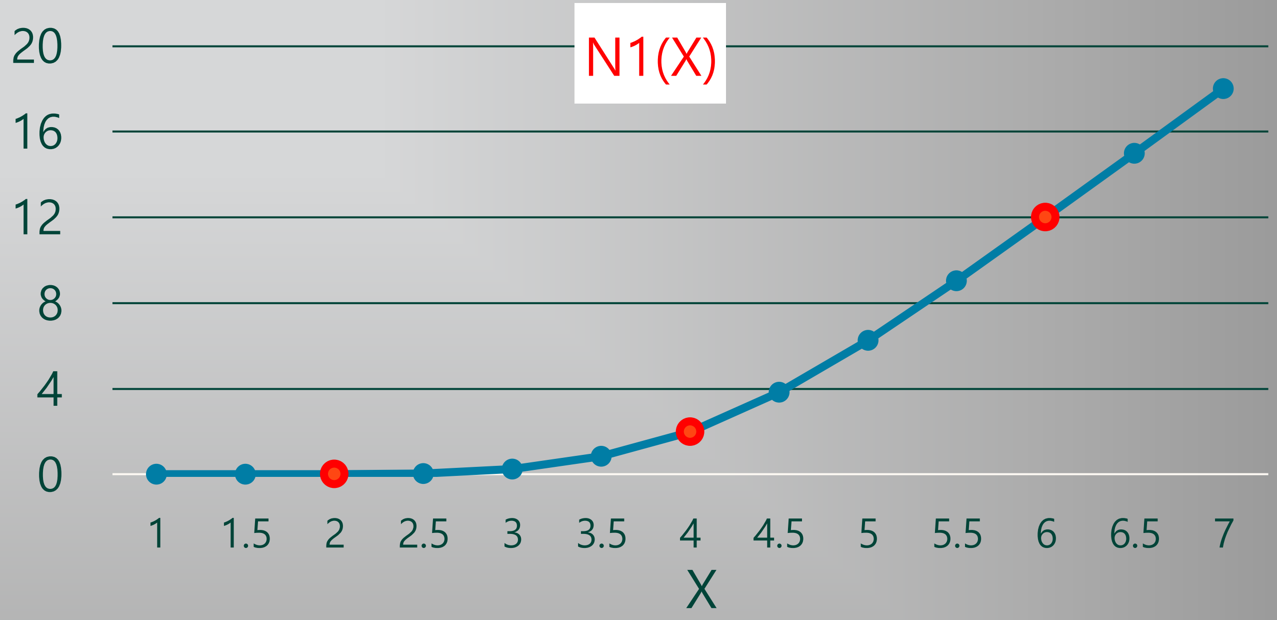
This simplifies to N1(X) = ($(X-2)_+^3$ - 2*$(X-4)_+^3$ + $(X-6)_+^3$) / 4

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.

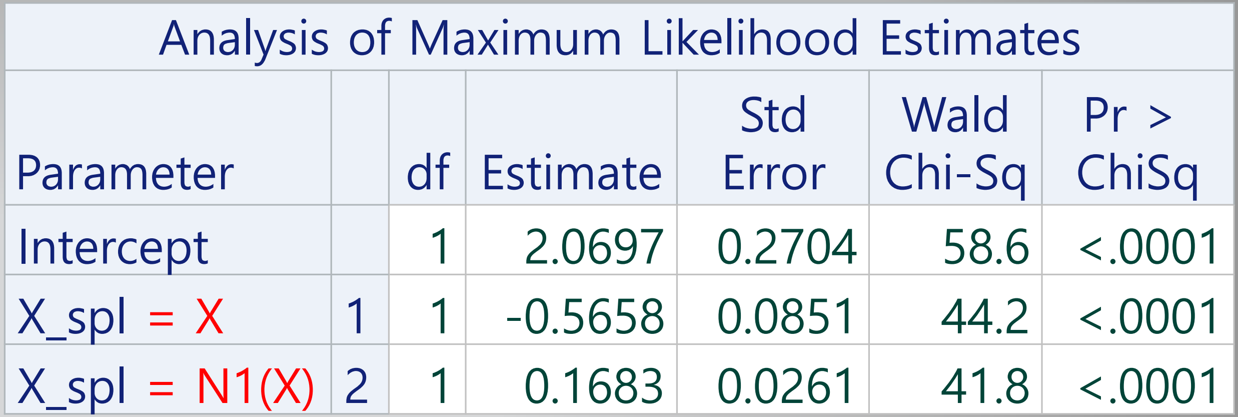
Can 1, X, N1(X) provide good fit to the Y from SPLINEDATA in a logistic regression?

NEXT

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

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

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)

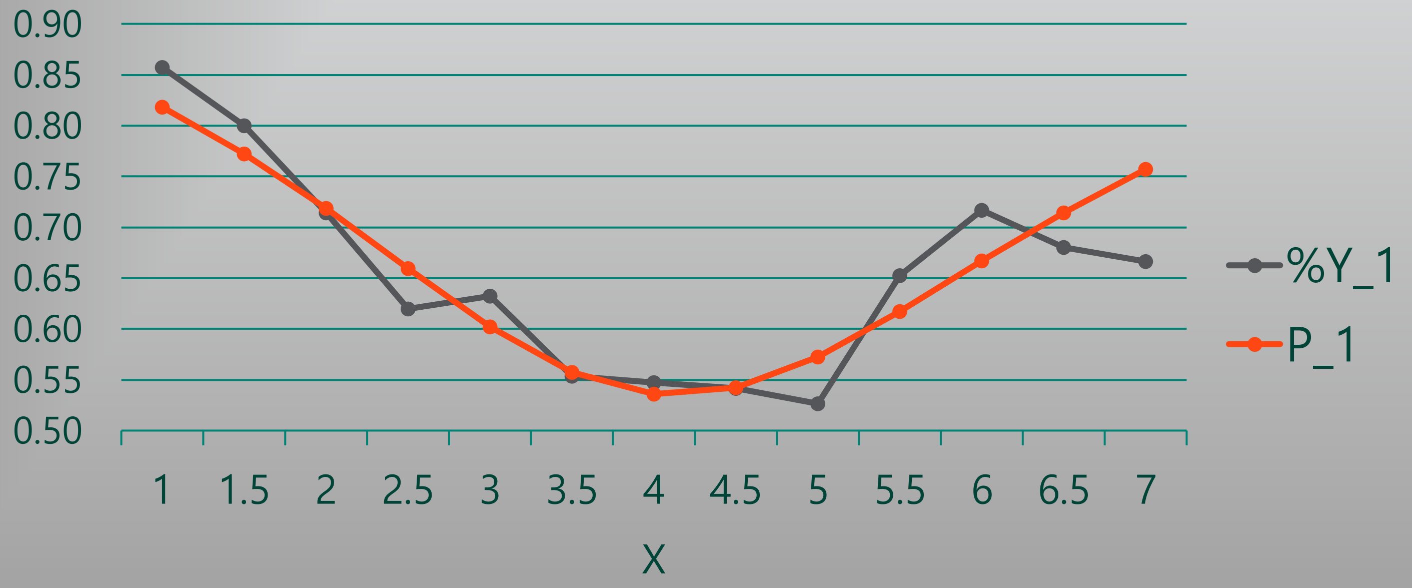
The Spline Transformation of X vs. %Y=1

The Spline transform tracks %Y=1 quite well.

SPLINE: P_1 for spline model

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

%Y=1 v. SPLINE MODEL (P_1)



Distribution of 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 (in addition to 1, X)
- In general, if there are KN (≥ 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 ≤ 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 complex.

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

NEXT

How Many Knots and Locations?

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

NEXT

The German Bank Dataset

This will be the example for Bootstrap Sampling with Optimism Correction

German Bank Dataset

- 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 classification and 3 continuous numeric X's
        - classification X includes nominal, ordered non-numeric, and numeric … but with "few" levels
        - If a classification 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

NEXT

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

SELF STUDY

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

SELF STUDY

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

SELF STUDY

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

Screening and Preparing classification X's when fitting Model to German Bank

IV (Information Value) as Screener of classification 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

NEXT

Screening classification X's from German Bank using IV

%CUM_LOGIT_SCREEN_2 (GERMAN.Bank, 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 ≥ 0.3	"Strong"

There were no zero-cells

- Screened out 12 weak predictors
- Leaving only 5 classification.

There are 3 continuous numeric X's ... so now there are 8 X's in total for potential usage in a Model.

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" cells were subjectively combined using similarity of definitions.

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

Did not look at Y ... No DD

This is the Now the working dataset

```
DATA GERMAN.Bank_v2;
SET GERMAN.Bank;
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

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

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

Preliminary Step: Create "Spline Design" for later usage

```
PROC LOGISTIC DATA = GERMAN.Bank_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 GERMAN.Bank_v3; MERGE GERMAN.Bank_v2 Spline_Design; /* No BY statement needed */
PROC PRINT DATA = GERMAN.Bank_v3 (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

HPGENSELECT can fit Logistic by LASSO. But using NCS may cause problems.

extreme value

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

NEXT

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

EPP = 300/29 = 10.3

NEXT

Fit German Bank using PROC LOGISTIC BACKWARD, SLS=0.05

```
%LET C_VARS = checking_status credit_history property_magnitude purpose savings;
PROC LOGISTIC DATA = GERMAN.Bank_v3 desc;
CLASS &C_VARS;
MODEL Y = &C_VARS age_spl: credit_amount_spl: duration_spl:
/ SELECTION=BACKWARD SLS=.05;
SCORE DATA = GERMAN.Bank_v3 OUT=SCORED FITSTAT;
run;
```

":" Include all VARs with prefix

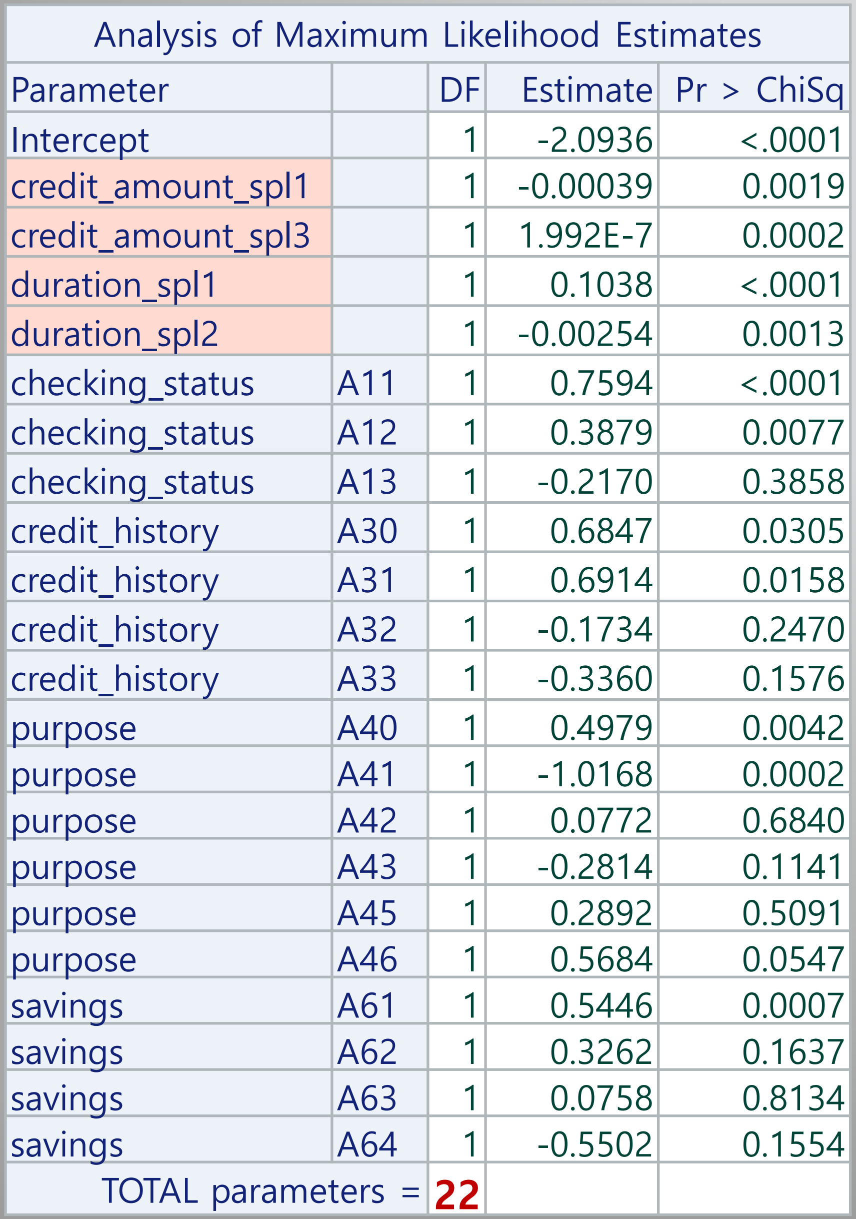
FITSTAT Creates Report see next slide

SCORED includes Model Probability called P_1 and Y

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

NEXT

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



Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Pr > ChiSq
Intercept		1	-2.0936	<.0001
credit_amount_spl1		1	-0.00039	0.0019
credit_amount_spl3		1	1.992E-7	0.0002
duration_spl1		1	0.1038	<.0001
duration_spl2		1	-0.00254	0.0013
checking_status	A11	1	0.7594	<.0001
checking_status	A12	1	0.3879	0.0077
checking_status	A13	1	-0.2170	0.3858
credit_history	A30	1	0.6847	0.0305
credit_history	A31	1	0.6914	0.0158
credit_history	A32	1	-0.1734	0.2470
credit_history	A33	1	-0.3360	0.1576
purpose	A40	1	0.4979	0.0042
purpose	A41	1	-1.0168	0.0002
purpose	A42	1	0.0772	0.6840
purpose	A43	1	-0.2814	0.1141
purpose	A45	1	0.2892	0.5091
purpose	A46	1	0.5684	0.0547
savings	A61	1	0.5446	0.0007
savings	A62	1	0.3262	0.1637
savings	A63	1	0.0758	0.8134
savings	A64	1	-0.5502	0.1554
TOTAL parameters =		**22**		

SELECTIONS by BACKWARD

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

(Can have ALL-IN or NONE-IN for splines if using the COLLECTION statement.)

Fit Statistics for SCORE Data			
Data Set	Total Freq	AUC(*)	Brier Score(**)
German	1000	0.808	0.156

* AUC is c-Statistic ... 0.808 is good (**<u>too</u> good**)

** Brier Score is Average Squared Error

Bootstrap Sampling with Optimism Correction

Plan for validating Apparent Model with Optimism Correction

NEXT

This was the MODELING PROCESS for the German Bank Model:

1. Preparing predictors X's:
   a) IV screening of classification X's … involves looking at Y
   b) Combined levels of PURPOSE … did not look at Y
   c) Using SPLINES for age credit_amount duration … does not look at Y
2. BACKWARD selection of X's with SLS = 0.05 and fitting coefficients
3. Computing validation measures

As part of optimism correction the MODELING PROCESS would run 200 times on 200 bootstrap samples. But we are making some compromises:

Here is our PLAN for optimism correction: Include (1c), (2), and (3), Exclude (1a and 1b)

REASONS (excuses?) to exclude (1a and 1b):

- Never really regarded the dropped 12 X's as part of the candidate predictors
- A lot of complicated programming to replicate (1a and 1b) for repeated bootstrap samples.
- Assume that it would not effect the optimism correction process.

In the slides that follow: Optimism Correction is applied to compute (1) c-Statistic, (2) Lift Charts. But it would be easy to extend to other measures such as ASE.

Optimism Correction for c-Statistic … 0.808 is too good!

1. **c-Stat$_{app}$** for the Apparent Model

```
%LET C_VARS = checking_status credit_history property_magnitude purpose savings;
PROC LOGISTIC DATA = GERMAN.Bank_v3 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 Splines and Backward with SLS = 0.05

A bootstrap sample has ~63% of rows from Analysis with some repeats … use FREQ in PROC LOGISTIC for repeats

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

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

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

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

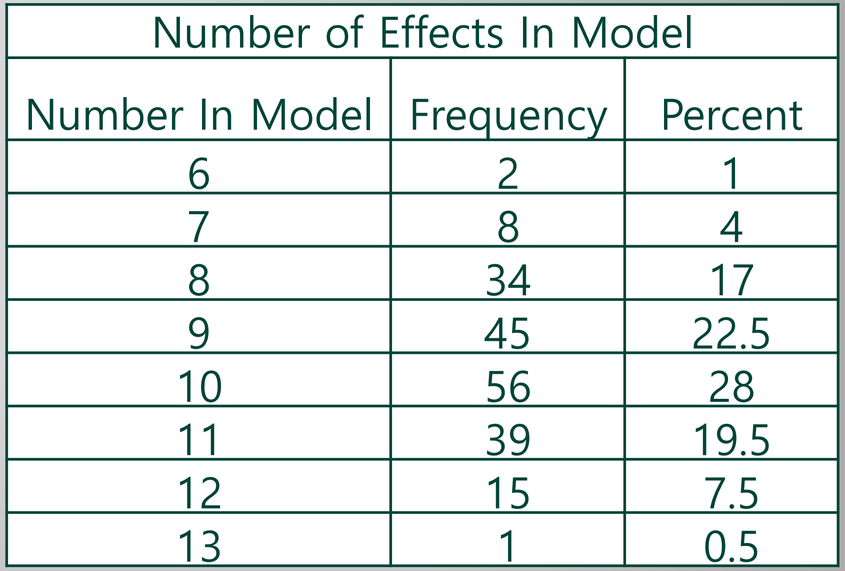
5. Optimism-Corrected Performance = c-Stat$_{app}$ - c-Stat$_{opt}$ = 0.808333 - 0.029194 = **0.779139**

Optimism-Corrected **0.779139** is reported as the validation statistic.
Same process can be applied to ASE etc. We'll do 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

Review the construction of LIFT CHART

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

```
PROC LOGISTIC DATA = GERMAN.Bank_v3 desc;
CLASS &C_VARS;
MODEL Y = age_spl: credit_amount_spl: duration_spl: &C_VARS / SELECTION=BACKWARD SLS=.05;
SCORE DATA = GERMAN.Bank_v3 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 … Self Study

```
%LET C_VARS =
checking_status credit_history property_magnitude purpose savings;
ods exclude all;
PROC LOGISTIC DATA = GERMAN.Bank_v3 desc;
CLASS &C_VARS;
MODEL Y =
age_spl: credit_amount_spl: duration_spl: &C_VARS
/ SELECTION=BACKWARD SLS=.05;
SCORE DATA = GERMAN.Bank_v3
OUT=SCORED FITSTAT;
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

Put highest **P_1** in RANKP=0, next highest in RANKP=1, … lowest **P_1** in RANKP=7

This is the "LIFT CHART" … Dataset 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	P_1	Y
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	P_1	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	
P_1	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 Model 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 P_1 and Y) is still fairly good

Overall, good!

Apparent Model Lift

RANKP	P_1	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	P_1	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	P_1	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**

Finish Up …
the German Bank Model

German Bank Model … Accepted

We decide optimism corrected performance for the Apparent Model is good (*).
… this Model is accepted !!

(*) My very unpolished SAS code for Optimism-Corrected Performance calculation (c-Statistic and Lift Chart) is in Appendix

But, German Bank data was oversampled … What to do?

We have to “make up” facts about GERMAN BANK Population (I don’t know the real facts)

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

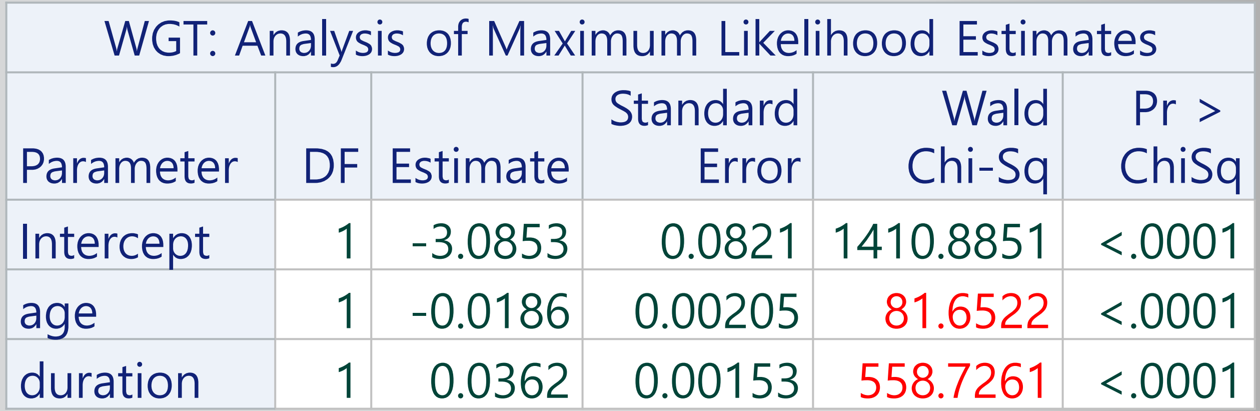
B. Weight for non-default in sample: (47500)/700 = 67.86

What is the effect of Over-sampling on a Logistic Model? … see Next Slide

NEXT

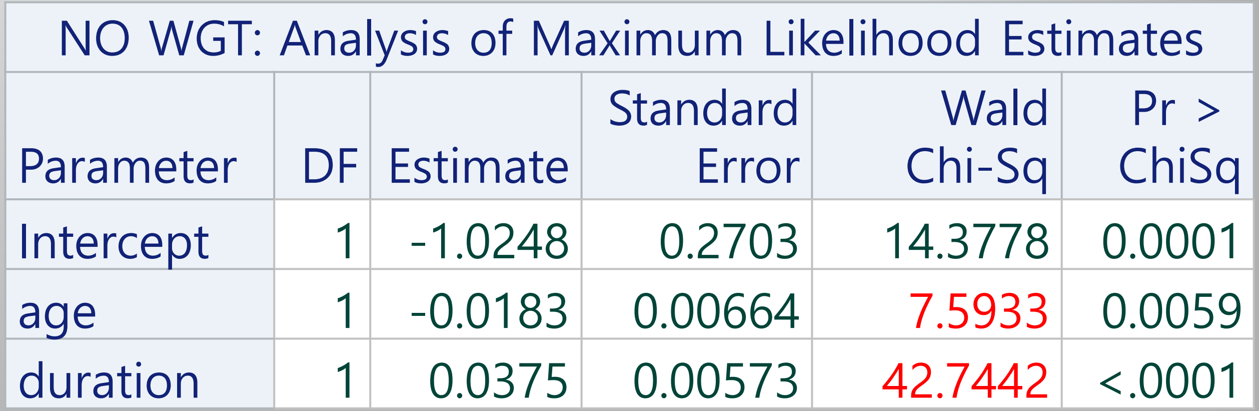
German Bank data was oversampled … has biased Intercept

```
DATA TEMP; SET GERMAN.Bank_v3;
If Y=1 then wgt = 8.33;
if Y=0 then wgt = 67.86;
run;
PROC LOGISTIC DATA = TEMP DESC;
WEIGHT wgt;
MODEL Y = age duration;
title "WGT";
run;
PROC LOGISTIC DATA = TEMP DESC;
MODEL Y = age duration;
title "No WGT";
run;
```



WGT: Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Sq	Pr > ChiSq
Intercept	1	-3.0853	0.0821	1410.8851	<.0001
age	1	-0.0186	0.00205	81.6522	<.0001
duration	1	0.0362	0.00153	558.7261	<.0001



NO WGT: Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Sq	Pr > ChiSq
Intercept	1	-1.0248	0.2703	14.3778	0.0001
age	1	-0.0183	0.00664	7.5933	0.0059
duration	1	0.0375	0.00573	42.7442	<.0001

- Approximately, same b's for age and duration.
- But quite different estimates for Intercept.
- Notice: Higher Chi-Sq's for weighted Logistic

Comment: c-statistic is essentially unaffected by weighting … P's are ranked ~same by both weighted and unweighted models

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

Finalize the German Bank Model

To weight the Analysis Dataset sample back to the Population:

If Y=1 then wgt = 8.33

If Y=0 then wgt = 67.86

Now update the Final X's (from Apparent Model) with weights:

```
PROC LOGISTIC DATA = GERMAN.Bank_v3 ;
WEIGHT wgt;
MODEL Y = <the final X's as selected by the Apparent Model>;
```

Use the Weighted Model for scoring new datasets

… **MOVE TO PRODUCTION !**

NEXT

Why not use weights when selecting X's?

Can we use the Weights when fitting our Logistic Model ? … not recommended

… the weights create a risk of overfitting or, at least, increases complexity.

Right-tail p-values for the X's in BACKWARD … are decreased … i.e. made more significant.

All (or many) X's become significant.

But the weights do not reflect "more information" … same information as unweighted.

Modeler must guess at an appropriate SLS to offset the decrease in p-values to avoid overfitting.

```
PROC LOGISTIC; WEIGHT Wgt;
MODEL Y = X's / SELECTION=BACKWARD SLS=???;
```

Similar problem if using AIC or BIC in HPLOGISTIC options SELECT and CHOOSE

NEXT

SELF STUDY

Rescaled 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
After Weighting: PREDICT Numerator = A*B*(2500/300)
After Weighting: PREDICT Denominator = A*B*(2500/300) + A*(1-B)*(47500/700)
Column D = Numerator / Denominator
After Weighting: MeanY Numerator = A*C*(2500/300)
After Weighting: MeanY Denominator = A*C*(2500/300) + A*(1-C)*(47500/700)
Column E = Numerator / Denominator

c-Statistic is essentially not changed due to oversampling and weighting.

Lift Chart, should be computed and reported with adjustment for weights

Alternatively: Take final X's from Apparent Model, do not use BACKWARD, and completely rerun bootstrap now with using weights.

Here is how to re-compute for RANKP= 0

Numerator of D is:

125*0.735*(2500/300) = 765.625

Denominator of D is

765.625 + 125*(1-0.735)*(47500/700)

= 3013.393

D = 765.625/ 3015.759 = 0.254

In Numerator: (2500/300) weights the number of events in RANKP=0.

In Denominator: (47500/700) weights number of non-events in RANKP=0.

Bruce Lund MSUG 2023

Self-Study

The Paper by Austin and Steyerberg (2017)

SELF STUDY

Paper by Austin and Steyerberg (2017)

The ANALYSIS dataset from the EFFECT Study (Enhanced Feedback for Effective Cardiac Treatment) had with **16,237** patients. The response was binary with an event-rate of **0.319**

This dataset was used to compare 3 approaches to computing the validation c-Statistic for a Logistic Model with 10 predictors (**10 d.f.**). The approaches were: (1) Compute c-Stat for Apparent Model (validation dataset is TRAIN, (2) Split sample 50-50, (3) Bootstrap sample with optimism correction (with 100 repetitions)

Note: Model was pre-selected … The MODELING PROCESS included ONLY the fitting of the coefficients.

- EPP of 5 to 100 in increments of 5 were studied. For each EPP level, random samples from the Master File were taken until there was 1000 samples … 1000 samples for each EPP (massive amount of computing!)
- For each EPP level: The average c-Statistic was computed for each iteration across the 1000 samples

(1) Apparent Model on TRAIN. [For EPP=5: If 5=n1/10, then n1=50 and n=50/0.319 = 157]
(2) Split-Sample on the 50% Validation Sample [For EPP=5, the split sample TRAIN has n = 157/2 = 78
(3) Optimism-Corrected with 100 bootstrap repetitions.
(4) Apparent Model validated on ANALYSIS minus TRAIN. [For EPP=5, this is 16,080=16,237-157]

- The c-Statistic from (4) would be regarded as the true c-Statistic performance of the Apparent Model.
- If (3) equals (4) for each EPP, this shows that (3) *accurately* estimates the c-Statistic for Apparent Model

- The c-statistic from fitting the 10 predictor model to the full 16,237 ANALYSIS dataset was 0.741

At EPP=20

- Apparent Model has upward bias.
- Split sample strong downward bias
- Bootstrap (100) essentially = External Validation (= true c-Statistic for apparent model)
- Blue Dots at 20 equals Split-Sample at 40. Blue Dots at 20 is ~ equal to Bootstrap and External at 20.
- CONCLUSION: Split-sample at 2*EPP is comparable to Bootstrap at EPP.

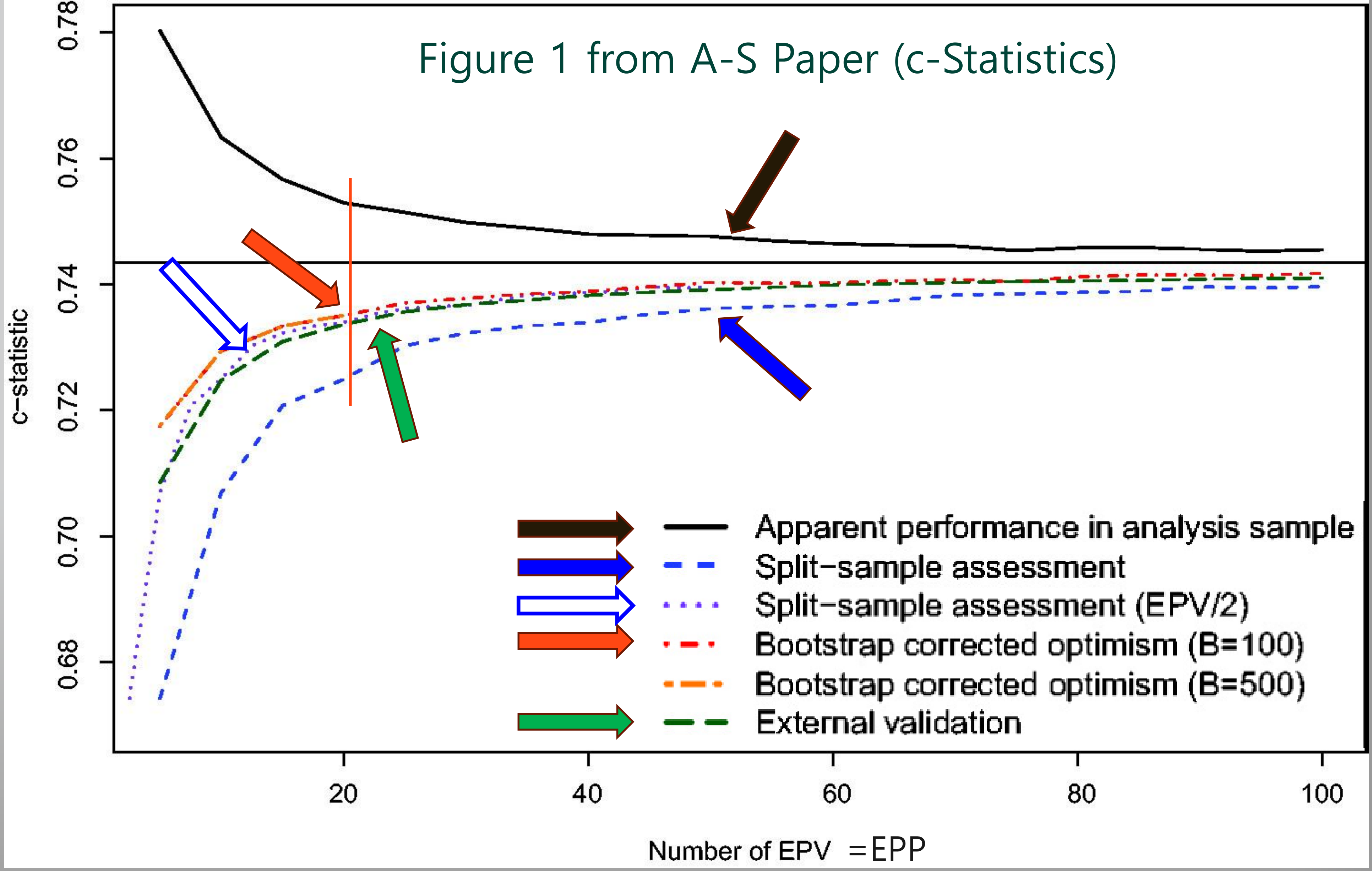


Figure 1 from A-S Paper (c-Statistics)

Blue Dot .... derived from Split-Sample by dividing EPP by 2. Blue dot at 20 EPP is Split-Sample at 40, etc.

Appendices

- Appendix 1: Rules for Constructing the Cubic Spline Formulas
- Appendix 2: Guideline as to number of ranks in Lift Chart
- Appendix 3a-3d: SAS code for Optimism-Corrected Performance (unpolished)
- Apparent 4a-4d: My Simulation Study of Optimum Correction

Logistic Modeling without Split-Samples

by Bruce Lund

Statistical Trainer, Novi, MI

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

Appendix 1: Rules for Constructing the Cubic 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) should 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

```
%LET C_VARS =
checking_status
credit_history
property_magnitude
purpose
savings
;
PROC SURVEYSELECT DATA= GERMAN.Bank_v3
OUT=BootSamples
NOPRINT SEED=111
METHOD=URS /* Sample with replacement */
SAMPRATE=1 /* Sample rate 100% */
REPS=200; /* Number of boot strap samples */
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 = GERMAN.Bank_v3 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;
TITLE1 "MEANOUT … Optimism Performance Statistics";
run;
PROC FREQ DATA = BASE3; TABLES NumberinModel;
TITLE1 "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;
TITLE1 "Optimism for Lift Chart";
run; TITLE; run;
```

Appendix 4a: My Simulation Study of Optimum Correction

My Conclusion:

Study shows that the optimism corrected c-statistic gives a correct validation measure, provided bootstrap sampling and optimism corrected calculations include all steps of the Modeling Process.

Methodology:

A dataset is generated in this simulation which has "n" observations in the analysis dataset plus an additional 22,000 observations from an "external" sample. The values of n are 1000, 2000, 3000, 4000, 5000, 6000.

A total sample consists of the n + 22,000 observations

For each "n" there are 50 independent instances of a total sample. That is, n + 22,000 observations are randomly generated 50 times. All together, as n goes from 1000 … 6000 in increments of 1000, there are 6*50 = 300 instances.

Target variable Y (with levels 0 or 1) and 11 predictors are randomly generated within each instance of the n + 22,000 observations.

The 22,000 will be "held out" as an external sample to give an approximation to the population. In the discussion below, all descriptions refer to the 50 samples of "n" as n goes from 1000 … 6000 in increments of 1000. The are 50*6 = 300 such samples.

For each of the 50 samples of an "n", the event rate (%Y=1) varies randomly. The events per predictor (EPP) for a sample is given by: EPP = (number of events) / 11.

Appendix 4b: My Simulation Study of Optimum Correction

Here is a table showing the average EPP for each n:

n	EPP (average over 50)
1000	6.88
2000	13.82
3000	20.60
4000	27.50
5000	34.54
6000	41.16

Four modeling processes are applied to each of the 300 samples.

(1) The Apparent Model: The Apparent Model is the logistic model shown below.

```
PROC LOGISTIC DATA = Sample_of_n; MODEL Y = X1 - X11;
```

The c-statistic is computed each logistic model. All together there are 50 c-statistics for each n and 300 in total. The c-statistic is called A_c-stat.

(2) The Split-Sample Model: Each sample of "n" is split 50-50 into TRAIN and VALIDATE. The logistic model is fit on TRAIN. The c-statistic is computed on VALIDATION. All together there are 50 c-statistics for each n and 300 in total. The c-statistic is called S_c-stat.

(3) Bootstrap Sampling with Optimism Correction: For each sample: 50 bootstrap samples are taken. (This is a different 50. Harrell recommends 200 bootstrap samples but due to computing resources, I reduced this number of bootstrap samples to 50.)

The Logistic Model for each bootstrap sample:

```
PROC LOGISTIC DATA =  BootSample_of_Sample_of_n OUTMODEL = OM;
                      MODEL Y = X1 - X11;
```

The c-statistic is computed for this model. Call it Boot_c-stat

Appendix 4c: My Simulation Study of Optimum Correction

Then, using model dataset OM, the full Sample_of_n is scored and c-statistic is computed. SAS code is below:

```
PROC LOGISTIC INMODEL = OM;
SCORE DATA = Sample_of_n FITSTAT;
```

The c-statistic is found in the FITSTAT report. Call it Full_c-stat

The average of (Boot_c-stat - Full_c-stat) over the 50 bootstrap samples is computed.

Optimism corrected c-statistic for each sample of n is:

O_c-stat = A_c-stat - Average(Boot_c-stat - Full_c-stat);

For each n there are 50 O_c-stat's.

(4) External: The External c-statistic comes from scoring the Apparent Model on the sample of 22,000.

```
PROC LOGISTIC DATA = Sample_of_n; MODEL Y = X1 - X11;
SCORE DATA = Sample_of_22000 FITSTAT;
```

The c-statistic is computed from the scores on the Sample_of_22000. All together 50 c-statistics for each n and 300 in total. The c-statistic is called E_c-stat.

The E_c-stat is the best approximation to a population c-statistic and it taken as the gold standard.

The three other c-statistics: A_c-stat, S_c-stat, and O_c-stat are compared to E_c-stat.

This comparison is made for each of the sample sizes: 1000 to 6000 with special emphasis on EPP.

Appendix 4d: My Simulation Study of Optimum Correction

Here is a Table that summarizes the results of the simulation study:

	Average c-statistic for the 50 samples					
E_c-stat	0.716	0.728	0.731	0.733	0.734	0.735
A_c-stat	0.762	0.745	0.744	0.744	0.743	0.742
S_c-stat	0.700	0.712	0.720	0.724	0.728	0.729
O_c-stat	0.724	0.724	0.730	0.732	0.734	0.735
n	1000	2000	3000	4000	5000	6000
Average EPP	6.88	13.82	20.60	27.50	34.54	41.16

Contact the author for a SAS macro. The macro has about 250 lines of code

Summary:

- For EPP at 13.82 (n=2000) the average O_c-stat was only 0.004 below the E_c-stat.
- At EPP of 20.60 (n=3000) and higher, the average c-statistics for E and O were essentially equal.
- Average A_c-stat, is consistently too optimistic. At EPP of 20.60 the over-estimate was 0.013.
- Average c-stat for Split Sampling is consistently too pessimistic. At EPP of 20.60 the under-estimate was 0.011.
- S_c-stat at EPP = 27.50 is the same as O_c-stat at EPP = 13.82. Likewise, S_c-stat at EPP = 41.16 is the same as the O_c-stat at EPP=20.60. This is consistent with an finding by Austin and Steyerberg (2017) that says that c-statistic of split-sample at 2*EPP is roughly equal to the optimism corrected c-statistic at 1*EPP.

Unused Slides

COLLECTION: Force “ALL IN” or “NONE IN”

```
%LET C_VARS = checking_status credit_history property_magnitude purpose savings;
PROC LOGISTIC DATA = GERMAN.Bank_v3 desc;
EFFECT Xspl_age = COLLECTION (age_spl2 age_spl3);
EFFECT Xspl_duration = COLLECTION (duration_spl2 duration_spl3);
CLASS &C_VARS;
MODEL Y = &C_VARS age duration Xspl_age credit_amount_spl: Xspl_duration
/ SELECTION=BACKWARD SLS=.05 DETAILS;
SCORE DATA = GERMAN.Bank_v3 OUT=SCORED FITSTAT;
run;
```

- age_spl2, age_spl3 are “collected” under label “Xspl_age”.
- “Xspl_age” is entered as a predictor in MODEL Y=
- This “Collection” will either Enter or Not.
- Likewise for “Xspl_duration”

“age” and “duration”
are in MODEL Y= as “free” X’s
… not part of a Collection.

NEXT

SELF STUDY

Splines: Force ALL IN or NONE IN

- Age Splines 2 and 3 were ALL or NONE. Did not enter.
- Also “age” did not enter. As before: No “ages”
- Duration splines 2 and 3 were ALL or NONE. Did enter.
- Also “duration” entered (not part of ALL or NONE)..
- Fit Statistics on the ANALYSIS dataset are essentially equal to those of the Apparent Model but with 1 additional d.f. (added “duration_spl3”)

Fit Statistics for SCORE Data

Data Set	Total Freq	AUC	Brier Score
German	1000	0.810	0.156

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Pr > ChiSq
Intercept		1	-2.5312	<.0001
checking_status	A11	1	0.7584	<.0001
checking_status	A12	1	0.3838	0.0084
checking_status	A13	1	-0.212	0.398
credit_history	A30	1	0.6721	0.0342
credit_history	A31	1	0.6998	0.0148
credit_history	A32	1	-0.1775	0.2368
credit_history	A33	1	-0.3254	0.1713
purpose	A40	1	0.497	0.0044
purpose	A41	1	-0.9873	0.0004
purpose	A42	1	0.0734	0.6987
purpose	A43	1	-0.2805	0.1157
purpose	A45	1	0.2557	0.5612
purpose	A46	1	0.5714	0.0537
savings	A61	1	0.539	0.0008
savings	A62	1	0.3255	0.1653
savings	A63	1	0.0831	0.7955
savings	A64	1	-0.5479	0.1568
duration		1	0.1453	0.0004
credit_amount_spl1		1	-0.00039	0.002
credit_amount_spl3		1	1.98E-07	0.0002
Xspl_duration	duration_spl2	1	-0.021	0.1777
Xspl_duration	duration_spl3	1	0.0233	0.2351