

# Small Sample Predictive Modeling: Tips and Tricks Using SAS/STAT

Doug Thompson

June 20, 2018

## This Presentation

- Present a **practical, SAS/STAT-based** approach to **small sample predictive modeling**
  - Focus on **cross-validation** as the model evaluation method (which has advantages, but presents challenges)
  - Illustrate techniques using **example model** (binary logistic regression)
  - Overview **SAS macros** that enable automated, parallel construction of **multiple independent models**
  - Provide SAS tips and recommendations
    - Part of the art is knowing what parts of the process to run automatically vs. where to “manually” intervene and apply business judgment

## What Is Predictive Modeling?

- Predictive modeling is essential for cost-efficient operations in many industries
  - Healthcare
  - Telecommunications
  - Insurance
  - Financial services
  - Many others
- Often, the goal of predictive modeling is to estimate the likelihood of specific events before they happen (e.g., avoidable hospital readmissions, product purchase, service retention, loan default), in order to be able to influence those events and achieve desirable outcomes

## Predictive Modeling vs. Other Modeling

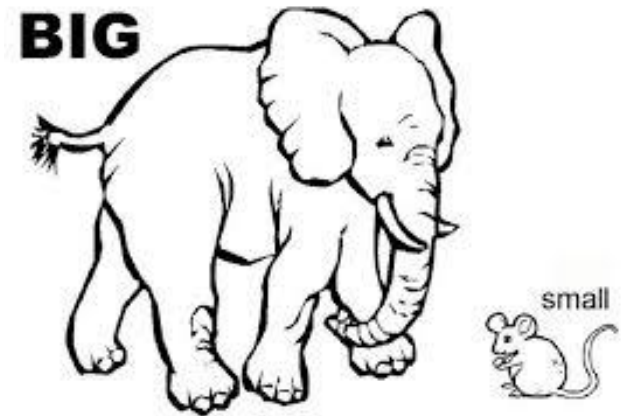
- Although they share some procedures, predictive modeling and other modeling have different goals and techniques
  - Predictive modeling
  - Hypothesis testing
  - Explanatory analysis
  - Forecasting
- In predictive modeling, the point is typically to use the model output (“score”), **at an individual person\* level**, to take some action; what matters most is that predictions are accurate and generalizeable
  - Other modeling typically focuses on results at an aggregate, summary level
- May intentionally violate important principles of other modeling (e.g., distributional assumptions, multicollinearity) and ignore central aspects of other modeling (e.g., p-values, confidence intervals, interpretation)

## Real Examples

1. Predict likelihood of high-speed Internet service churn
2. Predict likelihood of insurance policy lapse
3. Predict count of homeowners insurance claims caused by a hurricane projected to move through a specific area
4. Predict likelihood of having an elective surgery next year
5. Predict a person's likelihood of having an avoidable Emergency Room visit in the next 3 months

## Predictive Modeling and Sample Size

- In some contexts, available sample is **huge**:
  - Buying telecommunications services
  - Homeowners insurance claims from hurricane
- In other contexts, available sample is **small**:
  - Relatively rare healthcare events
  - Specific types of fraud
- Small samples pose challenges in both model building and evaluation



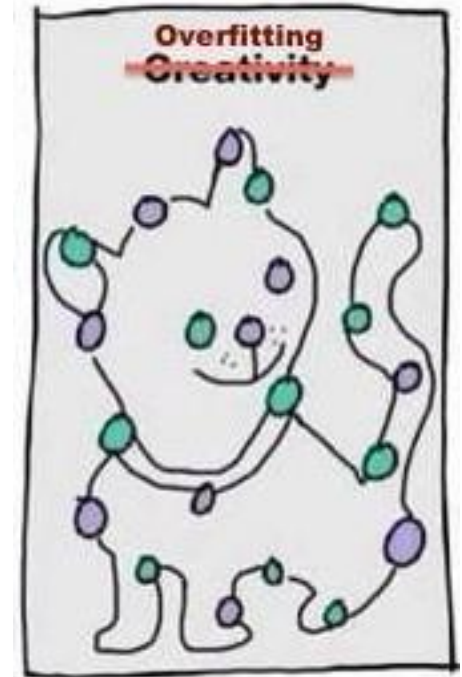
# Predictive Modeling With Small Samples



- **What is “small”?** No general definition, but some rules of thumb
  - Baesens et al (2015): >1,000 obs recommended for split sample validation
  - Rud (2001): Recommend 25+ obs per cell for predictive modeling
  - Harrell (2001): Binary target models –  $\min(n_1, n_2) = m$ ,  $p < m/10$
- Key point: “small” samples don’t prevent model building – they **prevent building complex models**, and **constrain model evaluation** approaches

# Model Evaluation

- When a model is trained on a set of data, the model may fit to idiosyncrasies of that training set, and not predict well on other data
  - This is called “**overfitting**”
- For a fair estimation of model performance (to ensure no overfitting), it is recommended that the model be evaluated on data other than the training set, e.g.:
  - Data from another organization
  - Data from another time period
  - A random sample of data held aside from the set used to build the model (“holdout”)





## Model Evaluation With Small Samples

- Holding aside data for evaluation means less data available to build the model
  - The smaller the sample, the less data available for validation
- *Solution:* Use same data for both training and evaluation, using appropriate techniques
  - Approaches: Cross-validation; bootstrapping (Hastie et al, 2008; Baesens et al, 2015; Harrell, 2001; Witten & Frank, 2005; Kuhn & Johnson, 2013)
    - Avoid sacrificing data available for training, while ensuring unbiased evaluation
  - Advantage of cross-validation: intuitive and relatively easy to explain

# What Is Cross-Validation?

- Randomly split the total data into K “folds” (say, 5 or 10)
- Evaluate the data on one fold, fit the model to all other folds combined
- Do this for all folds, then average the performance statistic
- 3-fold representation (inspired by Kuhn and Johnson, 2013)



Fold	Build on:	Evaluate on
1		
2		
3		

# Cross-Validation Considerations

## - Advantages

- Method is easy to describe
- Well-supported by literature, and commonly used in practice
- Allows all of the data to be used in model building (crucial for small sample problems), while enabling unbiased evaluation

## - Challenges

- Need to build model separately and independently for each fold (Hastie et al, 2008)
- Models usually differ somewhat between folds; need to choose what is the “final” model (problem discussed by Baesens)

## Summary of Key Points

- Small samples constrain model complexity and evaluation approaches
- To avoid over-complexity, use guidelines such as Harrell, 2001 ( $\min(n_1, n_2) = m$ ,  $p < m/10$ )
- To evaluate appropriately without sacrificing the data available for model build, use techniques such as cross-validation

# Example: Predict Emergency Room (ER) Frequent Utilization Next Year

## - Goal of predictive model

- Predict likelihood of having 3+ ER visits in the next year (among individuals with private insurance, under age 65)
- Reducing avoidable ER is crucial to save unnecessary expenses and to improve healthcare quality

## - Data

- Medical Expenditure Panel Survey (MEPS)
- Panel 19 (Year 1: 2014, Year 2: 2015)
- Panel/longitudinal with 5 rounds of data collection for each participant
- Complex survey design (e.g., weights, PSUs and strata) intentionally ignored in this illustration

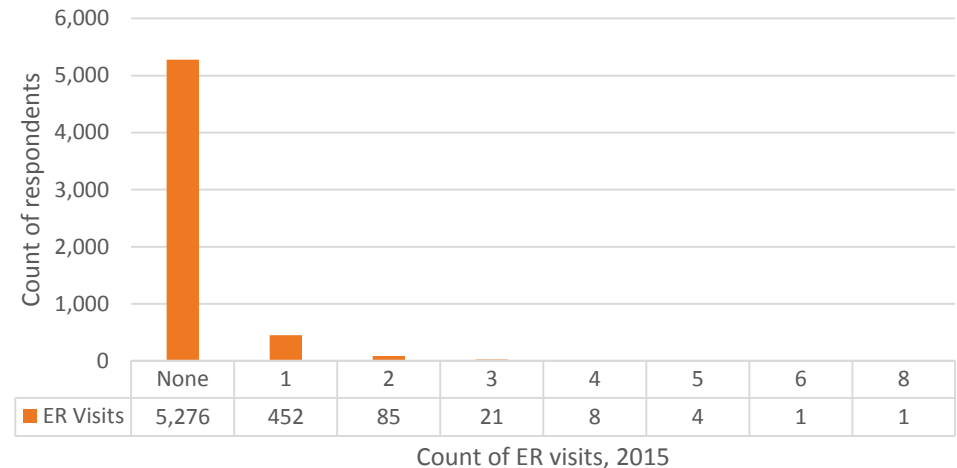


Example is realistic – similar model built recently by Rush Health

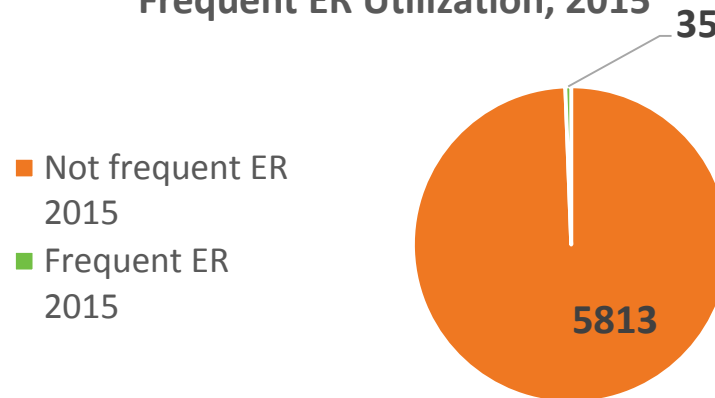
## Small Sample

- 5,848 observations
- **35 total target cases** (ER frequent utilizers in 2015)
- This is “small” by some definitions:
  - Can support one cell (Rud)
  - Can support 3 predictors (Harrell)

### Frequency of ER Visits, 2015



### Frequent ER Utilization, 2015



# Target and Potential Predictors

- **Target** (what we are trying to predict)
  - **2015** ER frequent utilization
- **Predictors** (measured in **2014**)
  - Medical expenditures
  - ER utilization
  - Has primary care physician (or not)
  - Self-reported health status
  - Geographic region
  - Income & employment
  - Demographics (race, ethnicity, age, gender, marital status, born in US)



## (Frequently Used) Steps in Predictive Modeling

0. Define scope, sample and variables

1. Split sample

2. Bivariate screening

3. Variable clustering

4. Multivariate variable selection

5. Transformations

6. Interactions

7. Create final model

8. Evaluate model performance

For good discussions of details of these steps, see Harrell (2001), Kuhn & Johnson (2013), Rud (2001)



# Step 1: Create Cross-Validation Folds

Randomly put each observation into one of five “folds”

```
data inscope2;  
set inscope;  
rand=ranuni(34573);  
run;
```

---

```
proc rank data=inscope2 out=inscope_crossval groups=5;  
ranks crossval_fold;  
var rand;  
run;
```

Create flags indicating for which fold each observation is “test” (for all others it is “train”)

```
data inscope_crossval2;  
set inscope_crossval;  
train1=0; train2=0; train3=0; train4=0; train5=0;  
test1=0; test2=0; test3=0; test4=0; test5=0;  
  
if crossval_fold=0 then test1=1; else train1=1;  
if crossval_fold=1 then test2=1; else train2=1;  
if crossval_fold=2 then test3=1; else train3=1;  
if crossval_fold=3 then test4=1; else train4=1;  
if crossval_fold=4 then test5=1; else train5=1;  
run;
```

## Result: Cross-Validation Samples

- Each of the 5,848 total observations is in the test (prediction) set for only one fold; it is in the training set for all other folds

Set	Outcome category	Cross-validation fold				
		1	2	3	4	5
Training set	Non-target cases	4,651	4,652	4,648	4,646	4,655
	Target cases	28	26	30	32	24
Prediction set	Non-target cases	1,162	1,161	1,165	1,167	1,158
	Target cases	7	9	5	3	11
<b>Total</b>		<b>5,848</b>	<b>5,848</b>	<b>5,848</b>	<b>5,848</b>	<b>5,848</b>

Each obs  
repeated 4x

Each obs  
only once

## Step 2: Bivariate Screening

Goal: Remove predictors with no association with target

Collected results from separate logistic regression for each predictor; done separately via macro for each fold

```
data screen;
set meplib.alltab;
if probchisq<0.05;
keep var trainset;
run;

%global vars_m1 vars_m2 vars_m3 vars_m4 vars_m5;

%macro varlists1_by_trainset;
%do i=1 %to 5;
data vars&i;
set screen(where=(trainset=&i));
run;

proc sql noprint;
select distinct var into:vars_m&i separated by " " from vars&i;
quit;

%put vars from trainset&i -- &&vars_m&i .;

proc delete data=vars&i; run;
%end;

%mend varlists1_by_trainset;
%varlists1_by_trainset;
```

Macro variable vars\_m(i) lists variables “surviving” bivariate screening for each fold i

## Step 3: Variable Clustering

Goal: Remove redundant variables (e.g., correlation with own cluster >0.8); this is a good place to apply some business judgment

```
%macro var_clustering;  
%let depvar = freq_ERTOTY2;  
  
%do i=1 %to 5;  
ods select none;  
proc varclus data=inscope_crossval2 (where=(train&i=1));  
var &&vars_m&i ;  
ods output RSquare=RSquare;  
run;
```

Cluster 1:  
Choose \_RTHLTH2

Cluster 2:  
Leave both

	Correlation		Prob chisq	
Variable	OwnCluster	Cluster	full model	Fold
_RTHLTH2	0.8082	Cluster1	0.0029	1
_MNHLTH2	0.8082	Cluster1	0.5225	1
TOTEXPY1	0.6364	Cluster2	0.3394	1
_ERTOTY1	0.6364	Cluster2	<.0001	1

## Step 3: Variable Clustering (Cont'd)

- After weeding out redundant variables based on PROC VARCLUS, ideally the resulting VIFs would be less than 2

```
* Check VIFs, make sure they are <10, ideally <2;
%macro check_vifs;
%let depvar = freq_ERTOTY2;

%do i=1 %to 5;
ods select default;
proc reg data=inscope_crossval2(where=(train&i=1));
model &depvar = &&vars_m&i / tol vif;
run;
quit;
%end;
%mend check_vifs;


---


%check_vifs;


---


* VIFs all <2, so looks good;
```

# Step 4: Multivariate Variable Screening

Goal: Identify predictive *sets* of variables (weed out predictors that are in no set)

Full model,  
stepwise, and  
backward  
selection

```
%macro variable_selection;
%let depvar = freq_ERTOTY2;

%do i=1 %to 5;

ods select none;
proc logistic data=inscope_crossval2 (where=(train&i=1)) descending namelen=100;
model &depvar = &&vars_m&i ;
ods output parameterestimates=parm1;
run;

* Keep sig<0.05;
data _parm1;
set parm1;
keep variable;
if probchisq ne . and probchisq<0.05;
run;

proc logistic data=inscope_crossval2 (where=(train&i=1)) descending namelen=100;
model &depvar = &&vars_m&i / selection=stepwise;
ods output parameterestimates=parm2;
run;

data _parm2;
set parm2;
keep variable;
if probchisq ne .;
run;
```

## Step 4: Multivariate Variable Screening (Cont'd)

- Update &vars(i) to reflect surviving variables in the set

```
* Update the macro variable lists to reflect the variables surviving at this point;
%macro varlists2_by_trainset;
  %do i=1 %to 5;
    data vars&i;
    set all_allparm(where=(trainset=&i));
    if variable ne 'Intercept';
    run;

    proc sql noprint;
    select distinct variable into:vars_m&i separated by " " from vars&i;
    quit;

    %put vars from trainset&i -- &&vars_m&i .;

    proc delete data=vars&i; run;
  %end;

%mend varlists2_by_trainset;
%varlists2_by_trainset;
```

## Step 5: Test Transformations

Goal: Optimize form of variables for predictive accuracy

- Go with a simple form (e.g., linear for continuous predictors), unless a transformation gives materially better fit
- Ones to try:
  - Categories
  - Squared
  - Inverse
  - Could try MARS (in R) to identify optimal set of non-linear predictors
- Really complex transformations (e.g., complex splines) may be dangerous in small-sample models – may overfit and not generalize
- In this model, no transformations materially improved fit



## Step 6: Test Interactions

Goal: Identify predictive *combinations* of variables

- Various approaches – multiplicative interactions; combinations of variables from exploratory decision trees (PROC HPSPLIT)
  - Note: HPSPLIT may not split in small, skewed samples without additional sampling to make the target categories more even

```
|%macro interaction_test(trainset=,int=);  
%let depvar = freq_ERTOTY2;  
  
proc logistic data=inscope_crossval2 (where=(train&trainset=1)) descending namelen=100;  
model &depvar = (&int @2) / selection=backward;  
run;  
  
%mend interaction_test;  
  
%put &vars_m1;  
* %interaction_test(trainset=1,int=_ERTOTY1|_RTHLTH2);  
* Conclusion: No interactions;
```

## Step 7: Final Model Equations


```
%global equation1 equation2 equation3 equation4 equation5;

options nomprint;
%macro final_model;
%let depvar = freq_ERTOTY2;

%do i=1 %to 5;
proc logistic data=inscope_crossval2 (where=(train&i=1))
descending namelen=100;
model &depvar = &&vars_m&i;
ods output parameterestimates=parm;
output out=scores pred=_score;
run;

data parm2;
set parm end=lastrec;
eqn = compress(variable|| "*" || put(estimate,20.8) || "+");
eqn = tranwrd(eqn, 'Intercept*', '');
if lastrec then eqn=tranwrd(eqn, '+', '');
keep eqn;
run;
```

Turn ODS  
OUTPUT into  
easy to  
copy/paste  
equation



## Step 7: Final Model Equations (Cont'd)

```
%put &equation1;  
%put &equation2;  
%put &equation3;  
%put &equation4;  
%put &equation5;
```

---



```
749  
750  
751 %put &equation1;  
-6.88940337+_ERTOTY1*0.74205865+_RTHLTH2*0.61805228  
752 %put &equation2;  
-7.58897673+TOTEXPY1*0.00001412+_ERTOTY1*0.58517296+_RTHLTH2*0.81842377  
753 %put &equation3;  
-8.08444343+_ERTOTY1*0.75295317+_RTHLTH2*0.70183469+white*1.26051802  
754 %put &equation4;  
-7.00691566+_ERTOTY1*0.74074131+_RTHLTH2*0.71103709  
755 %put &equation5;  
-7.26705584+_ERTOTY1*0.82930886+_RTHLTH2*0.67099690
```

The size of each equation  
is consistent with Harrell's  
rule of thumb  
( $\min(n_1, n_2) = m$ ,  $p < m/10$ )

## Step 8: Evaluate Final Model on Test Samples

Note, this is the first time that we have used the “test” sets!

Performance statistics:

C-statistic (AUC)

Decile analysis

```
%macro crossval_test_perform;
%let depvar = freq_ERTOTY2;

%do i=1 %to 5;
data test&i;
set test&i;
weight=1;
logit=&&equation&i ;
score=1/(1+exp(-1*logit));
weight=1;
_p_hat=score;
run;

%wtc (
ds = test&i,
outds = trn,
weight = weight,
depvar = &depvar
) ;

%weighted_decile(
infile=test&i,
score=_p_hat,
target=&depvar,
seed=74572,
weight=weight
);
```

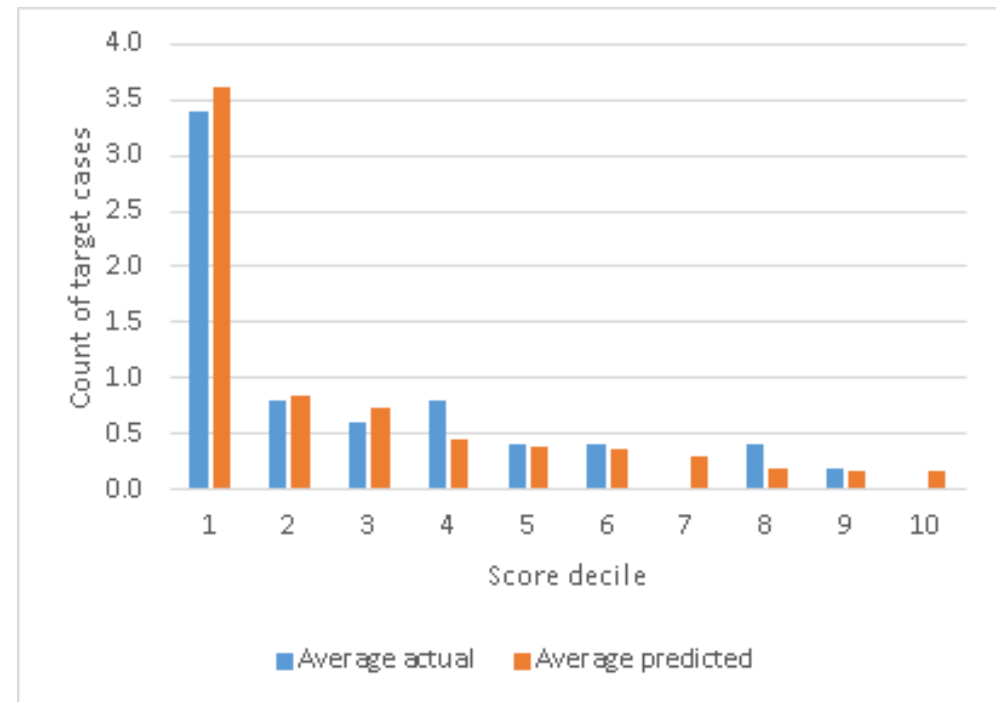
# Step 8: Evaluate Final Models on Test Samples

Average C-stat (good)

	C-stat
Fold	(AUC)
1	0.88
2	0.72
3	0.63
4	0.83
5	0.85
<b>Avg.</b>	<b>0.78</b>

Average decile analysis (good top decile)

Score decile	Sum		Average (/5)	
	Actual	Expected	Actual	Expected
1	17	18.0	3.4	3.6
2	4	4.3	0.8	0.9
3	3	3.7	0.6	0.7
4	4	2.3	0.8	0.5
5	2	2.0	0.4	0.4
6	2	1.8	0.4	0.4
7	0	1.5	0.0	0.3
8	2	1.0	0.4	0.2
9	1	0.9	0.2	0.2
10	0	0.8	0.0	0.2



## Model Equation For Application

- But now there are **5 final models** – how to score the equation on other/future data?
- Baesens et al discuss options:
  1. choose one at random
  2. build final on entire data, report performance on cross-validation
  3. **weighted ensemble**

## Model Equation For Application (Cont'd)

- Ensemble model (equally weighted)

```
* Final composite model scoring equation;
data scored;
set inscope;

logit1=(-6.88940337+_ERTOTY1*0.74205865+_RTHLTH2*0.61805228);
logit2=(-7.58897673+TOTEXPY1*0.00001412+_ERTOTY1*0.58517296+_RTHLTH2*0.81842377);
logit3=(-8.08444343+_ERTOTY1*0.75295317+_RTHLTH2*0.70183469+white*1.26051802);
logit4=(-7.00691566+_ERTOTY1*0.74074131+_RTHLTH2*0.71103709);
logit5=(-7.26705584+_ERTOTY1*0.82930886+_RTHLTH2*0.67099690);

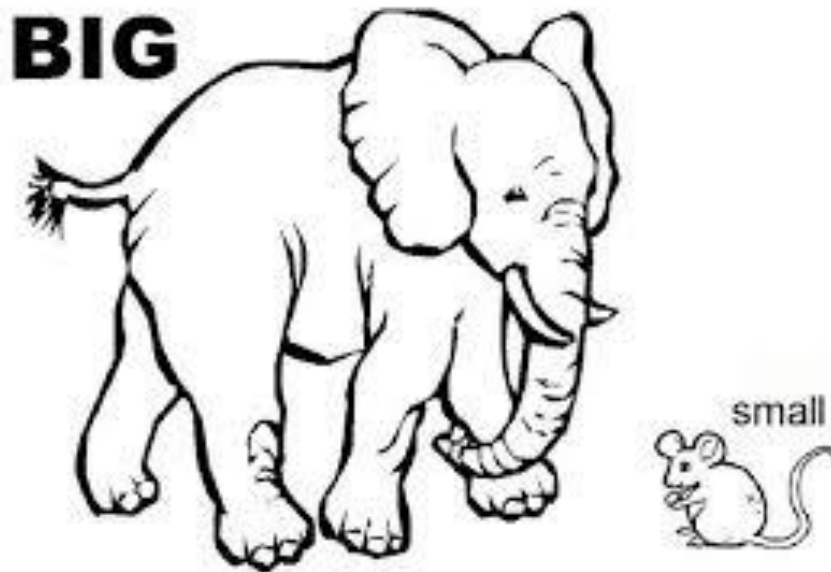
score1=1/(1+exp(-1*logit1));
score2=1/(1+exp(-1*logit2));
score3=1/(1+exp(-1*logit3));
score4=1/(1+exp(-1*logit4));
score5=1/(1+exp(-1*logit5));

average_score=mean(score1,score2,score3,score4,score5);

run;
```

## Concluding Thoughts

- Don't let (small) sample size be a barrier to predictive modeling – SAS/STAT provides all of the tools needed to handle the job!
  - But use them thoughtfully with awareness of limitations and consequences





## References

- Baesens, Van Vlasselaer & Verbeke. 2015. *Fraud Analytics*. Wiley.
- Harrell. 2001. *Regression Modeling Strategies*. Springer.
- Hastie, Tibshirani & Friedman. 2009. *Elements of Statistical Learning* (2<sup>nd</sup> Ed.). Springer.
- Kuhn & Johnson. 2013. *Applied Predictive Modeling*. Springer.
- Rud. 2001. *Data Mining Cookbook*. Wiley.
- Witten & Frank. 2005. *Data Mining* (2<sup>nd</sup> Ed.). Elsevier.