

Recreating the SELECTION=SCORE Model Specification with the BEST= n Effect Selection Option for PROC SURVEYLOGISTIC

Sara R. Adams, David R. Mink, Dave P. Miller, ICON Clinical Research, San Francisco, CA

ABSTRACT

The SURVEYLOGISTIC procedure in SAS® 9 provides a way to perform logistic regression with survey data. However, some options frequently used with the LOGISTIC procedure, such as stepwise and score model selection, were not included in PROC SURVEYLOGISTIC. One such option is SELECTION=SCORE BEST= n , which is used to identify the best subsets of covariates, allowing the user to select the number of models displayed for each model size with the highest score chi-square statistics. Two methods are described here for recreating this procedure option in PROC SURVEYLOGISTIC. The first method employs macros that run PROC SURVEYLOGISTIC once for each combination of covariates; for example, there are 10,660 possible combinations of 3 covariates from a candidate set of 41 variables, resulting in 10,660 runs of PROC SURVEYLOGISTIC. The macro call is nested within multiple DO loops. ODS is used to output the statistic of interest. The second method uses data step programming to generate a data set that repeats the observations for each combination of covariates. PROC SURVEYLOGISTIC is executed only once using a *by* command to test each combination of covariates separately. Although some important options are missing from PROC SURVEYLOGISTIC, careful programming can recreate the desired results.

INTRODUCTION

This paper provides two novel solutions to recreate the score model selection method that is unavailable in PROC SURVEYLOGISTIC. The SELECTION=SCORE option is used in PROC LOGISTIC to select the best subsets of covariates from a candidate set for a given model size, where the best subsets are those with the highest score chi-square statistics. With SELECTION=SCORE, the BEST= n option controls the number of models displayed for every model size. For example, if the only options specified are SELECTION=SCORE BEST=10, then the 10 best models for each model size are presented (i.e. the 10 best models with 1 covariate, the 10 best models with 2 covariates, etc. up to the 10 best models with the total number of candidate variables). If not every model size is desired, then the minimum and maximum model size can be specified with the START= n and STOP= n options. For example, the SELECTION=SCORE BEST=10 START=4 STOP=5 option only displays the 10 best models for model size 4 and the 10 best models for model size 5.

Our basic approach to recreate score model selection in PROC SURVEYLOGISTIC is to (1) run the model for each combination of covariates, (2) store the score chi-square statistic for each model, and (3) report the models with the highest score chi-square statistics. The two solutions presented here differ in how they run the model for each combination of covariates: one solution uses two macros while the other solution utilizes data step programming without using macros. In our example, we have a candidate set of 41 variables, and we use score selection to identify the 5 best combinations (analogous to BEST=5) of 3 covariates (analogous to START=3 STOP=3). The code could easily be adapted to select a different number of best models reported or a different model size (see the Adapting the Code section below). In PROC LOGISTIC, the SELECTION=SCORE method is not available for models that include categorical variables in a class statement. In our example, all the candidate variables are categorical and are included in a class statement in the SURVEYLOGISTIC procedure; however, the code could be modified to allow for all continuous variables or a mix of categorical and continuous variables.

BACKGROUND

Throughout this paper we will make use of a real world example: an asthma study for which ICON Clinical Research performed the analysis and first encountered some of the strengths and limitations of PROC SURVEYLOGISTIC. A goal of the study was to identify three questions that physicians could quickly ask an asthma patient to determine if the patient is likely to have uncontrolled asthma. We used data from a Web based survey administered to a representative sample of asthma patients. The data were weighted to mirror the US population of asthma sufferers. In addition to demographic and clinical information, the survey collected information about each patient's asthma control status and responses to 41 questions about the patient's attitudes toward medical professions and their asthma care (these were the questions that could potentially be asked by physicians to identify patients likely to have uncontrolled asthma). Examples of the candidate questions are:

Q1. I sometimes need to take my asthma medication more frequently than prescribed.

Q2. I am extremely satisfied with my current treatment regimen.

In our analysis, we simplified the original responses (strongly disagree, disagree, neither agree nor disagree, agree, and strongly agree) into three levels (disagree, neither agree nor disagree, and agree). This ordinal scale could be treated as either continuous (assuming a linear relationship on the logit scale) or categorical (making no assumption about the shape of the effect). Results showing that a large proportion of asthma patients were poorly controlled in spite of receiving multiple medications were recently published in the *Journal of Allergy and Clinical Immunology* (Peters et al., 2007).

SELECTION=SCORE IN PROC LOGISTIC

First, we provide a brief example of the SELECTION=SCORE option in PROC LOGISTIC. Note that we could not specify that the candidate variables were categorical by including them in a class statement because the SELECTION=SCORE option in PROC LOGISTIC is not compatible with a class statement. Therefore, we ran the model with asthma control status as the dependent variable and 41 candidate covariates as continuous variables, specifying score selection (SELECTION=SCORE) and requesting that the 5 best models (BEST=5) of size 3 (START=3 STOP=3) be identified. Prior to running any of the models, we renamed the candidate variables so that they could easily be used in an array and DO loops (i.e. q1-q41).

```
proc logistic data=studypop01 descending;
  weight wt1812;
  model dependvar = q1-q41 / SELECTION=SCORE BEST=5 START=3 STOP=3;
run;
```

For each of the five best models identified, the "Regression Models Selected by Score Criterion" output reports the number of variables used (i.e. the model size), the score chi-square statistic, and the three variables included in the model. Out of all combinations of three covariates, the model with the variables q15, q19, and q27 produced the highest score chi-square statistic in PROC LOGISTIC: 377.9889. However, PROC LOGISTIC does not have a mechanism to take study design into consideration for weighted survey data so we needed to be able to use PROC SURVEYLOGISTIC to identify the best models.

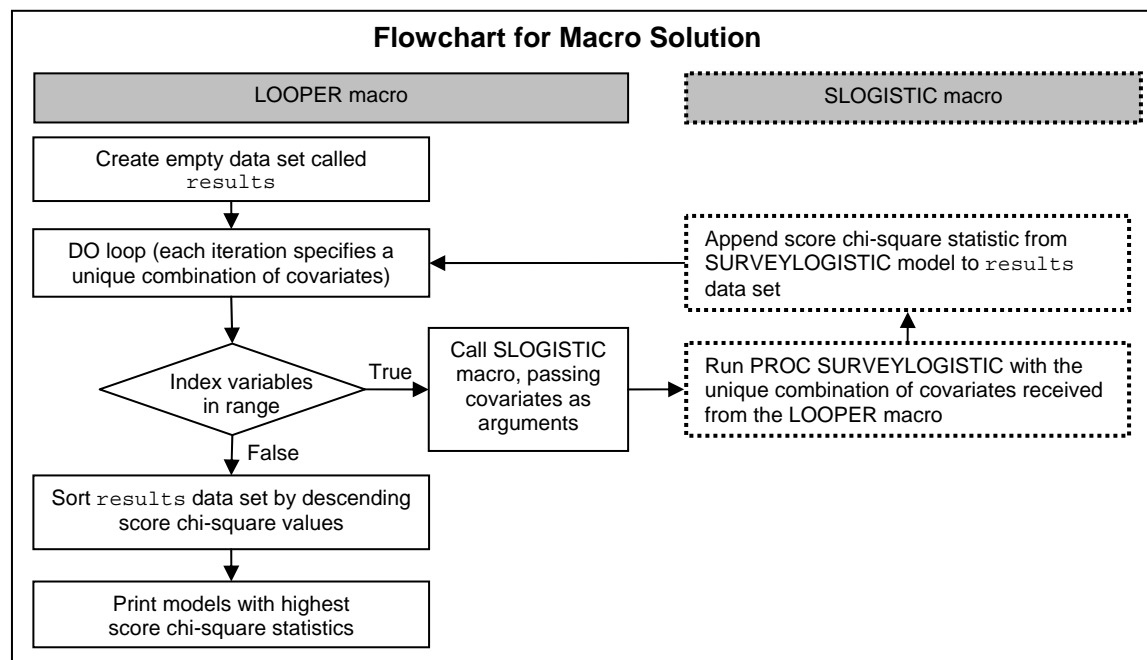
Regression Models Selected by Score Criterion		
Number of Variables	Score Chi-Square	Variables Included in Model
3	377.9889	q15 q19 q27
3	376.0321	q15 q19 q23
3	375.3048	q19 q25 q27
3	371.3191	q15 q19 q25
3	365.4339	q19 q23 q25

METHOD 1 – MACRO SOLUTION

The first method for recreating score selection in PROC SURVEYLOGISTIC employs two macros (LOOPER and SLOGISTIC) that execute PROC SURVEYLOGISTIC once for each combination of covariates. Since we are testing each combination of 3 variables from 41 candidate variables, PROC SURVEYLOGISTIC is run

$$\binom{41}{3} = \frac{41!}{38!3!} = \frac{41 \cdot 40 \cdot 39}{3 \cdot 2 \cdot 1} = 10,660$$

separate times. The following flowchart provides the sequence in which the macro solution is executed; the LOOPER macro is called first.



THE SLOGISTIC MACRO

The SLOGISTIC macro receives three arguments from the LOOPER macro: the names of three variables. First, PROC SURVEYLOGISTIC is run with asthma control status as the dependent variable and the three variables passed from the LOOPER macro as covariates. A weight statement is included to apply survey weights. Since our candidate variables are all categorical, the three covariates are included in a class statement. ODS is used to capture the output from the tests for global null hypothesis, which includes the score, likelihood ratio, and Wald chi-square statistics. In a subsequent data step, the score chi-square statistic is selected and three new variables (*var1-var3*) are created to store the names of covariates in the model. Finally, for each call to the SLOGISTIC macro, the data set containing the three covariates and their corresponding score chi-square statistic from the SURVEYLOGISTIC model are appended to a temporary data set called *results*.

```
%macro slogistic (ind1,ind2,ind3); *number of arguments equals model size;

proc surveylogistic data=studypop01;
  weight wt1812;
  class &ind1. &ind2. &ind3.;
  model dependvar = &ind1. &ind2. &ind3.;
  ods output GlobalTests=chisq1; *ODS captures global tests (including score chisq);
run;

data chisq2 (keep=ChiSq var1-var3);
  set chisq1 (where=(test="Score")); *keep only the score chi-square value;
  var1="&ind1. ";
  var2="&ind2. ";
  var3="&ind3. ";
run;

data results;
  set results chisq2;
run;

%mend slogistic;
```

THE LOOPER MACRO

The LOOPER macro is called first and contains the nested DO loops that are used to call the SLOGISTIC macro once for each combination of covariates. The LOOPER macro creates the empty *results* data set, which is later appended in each call to the SLOGISTIC macro. Since we are using a model size of three, we use a three-layer DO loop to cycle through every combination of covariates from the candidate set of 41 – passing a unique combination of three covariates to the SLOGISTIC macro in each loop.

For the first DO loop, we set the index variable (*i*) to have a lower bound equal to one and an upper bound equal to two less than the number of candidate variables, i.e. 39. For the second DO loop, we set the index variable (*j*) to have a lower bound equal to one more than the index variable from the first DO loop (*i*) and an upper bound equal to one less than the number of candidate variables, i.e. 40. Finally in the third DO loop, we set the index variable (*k*) to have a lower bound equal to one more than the index variable from the second DO loop (*j*) and an upper bound equal to the number of candidate variables, i.e. 41.

Since the SLOGISTIC macro is called multiple times within the DO loop, the DO loop could not be contained within a data step and was created as a stand-alone macro DO-loop (%do) inside the LOOPER macro instead. The macro DO loop also allows the index variables to be used as macro variables. With each iteration of the DO loop, the SLOGISTIC macro is called and the names of three variables are passed as arguments. The three arguments are specified using the *i*, *j*, and *k* index (macro) variables from the three DO loops: %slogistic(*q&i.*, *q&j.*, *q&k.*). For example, in the first iteration of the DO loop, *i*=1 *j*=2 and *k*=3 so the macro variables resolve to *q1*, *q2*, and *q3*, and those three variables are sent to the SLOGISTIC macro as arguments. In the final iteration of the DO loop, *i*=39 *j*=40 and *k*=41 so *q39*, *q40*, and *q41* are the three variables sent to the SLOGISTIC macro as arguments.

Once the DO loops have finished executing, the *results* data set (which contains 10,660 score chi-square statistics) is sorted by score chi-square in descending order. Finally, the five best subsets (i.e. the five models with the highest score chi-square statistics) are printed.

```

%macro looper();

data results; *create empty data set to store chi-square values;
  set _null_;
run;

%do i = 1 %to 39; *set upper bound to number of candidate variables minus 2;
  %do j = &i.+1 %to 40; *set upper bound to number of candidate variables minus 1;
    %do k = &j.+1 %to 41; *set upper bound to the number of candidate variables;
      %slogistic(q&i.,q&j.,q&k.); *number of arguments equals model size;
    %end;
  %end;
%end;

proc sort data=results out=results_sorted; *sort by descending chi-square;
  by descending ChiSq;
run;

title3 "Regression Models Selected by Score Criterion";
proc print data=results_sorted (obs=5); *obs= controls the # of models displayed;
run;

%mend looper;
%looper();

```

OUTPUT

The output reports the three variables included in each of the five models with the highest score chi-square statistics. Out of the 10,660 unique combinations of three covariates, the model with the variables q19, q25, and q27 produced the highest score chi-square statistic in PROC SURVEYLOGISTIC: 341.6988. The results from the macro solution using SURVEYLOGISTIC are similar to the output for the PROC LOGISTIC model with the SELECTION=SCORE option; however, the score chi-square statistics differ due to both the weighting and the fact that we decided to fit the model using categorical variables rather than assuming linear effects.

Regression Models Selected by Score Criterion

Obs	ChiSq	var1	var2	var3
1	341.6988	q19	q25	q27
2	327.6492	q15	q19	q27
3	322.3556	q19	q23	q25
4	320.7365	q23	q25	q27
5	319.1081	q15	q19	q25

METHOD 2 – DATA STEP PROGRAMMING SOLUTION

The second method for recreating score selection in PROC SURVEYLOGISTIC uses data step programming to test each combination of covariates. The first step is to generate a new data set that repeats the observations for each combination of covariates. In our example, the original data set contains 1,812 records, and there are 10,660 possible combinations of covariates; therefore, the new data set created has $1,812 * 10,660 = 19,315,920$ observations. The observations are repeated using nested DO loops that are similar to the DO loops in the macro solution, with two major differences: (1) rather than calling a macro in each iteration, a copy of the data set is output and (2) rather than using macro variables to name the covariates (i.e. `q&i.`), the index variables are used to refer to the position of the candidate variable in an array (i.e. `indvar{i}`). Prior to the output statement, three new variables are created and set equal to the values of the covariates using the array of candidate variables. The only variables kept in this new data set (`studypop02`) are the dependent variable, the weight variable, the three variables that were set equal to the values of the covariates, and the three index variables that indicate which covariates were used in that iteration of the DO loop. If you are using formats with any of your candidate variables, check to make sure that the format information is not lost when the `valpred` variables are created.

```
data studypop02(keep=dependvar wt1812 valpred1 valpred2 valpred3 i j k);
  set studypop01;
  array indvar {1:41} q1-q41; *array of candidate variables;
  do i = 1 to 39; *set max value to number of candidate variables minus 2;
    do j = i+1 to 40; *set max value to number of candidate variables minus 1;
      do k = j+1 to 41; *set max value to number of candidate variables;
        valpred1=indvar{i};
        valpred2=indvar{j};
        valpred3=indvar{k};
        output;
      end;
    end;
  end;
end;
```

The following excerpt from the log file demonstrates that this data step converts a short (and wide) file into a tall and thin file.

```
NOTE: There were 1812 observations read from the data set WORK.STUDYPOP01.
NOTE: The data set WORK.STUDYPOP02 has 19315920 observations and 8 variables.
NOTE: DATA statement used (Total process time):
      real time           1:44.23
      cpu time            8.45 seconds
```

The `studypop02` data set is then sorted by the three index variables so that they can be used in the `by` statement in PROC SURVEYLOGISTIC. PROC SURVEYLOGISTIC is executed only once (as opposed to 10,660 times in the macro solution) using a `by` command to test each combination of covariates separately. Similar to the model in the macro solution, the weight variable is specified and the class statement includes the three covariates. ODS is used to output the tests for global null hypothesis, from which the score chi-square statistic is selected.

```
proc sort data=studypop02 out=studypop03;
  by i j k;
run;

proc surveylogistic data=studypop03;
  by i j k;
  weight wt1812;
  class valpred1 valpred2 valpred3;
  model dependvar = valpred1 valpred2 valpred3;
  ods output GlobalTests = chisq1;
run;

data chisq2 (keep=ChiSq i j k);
  retain chisq;
  set chisq1 (where=(test="Score"));
run;
```

Finally, the results are sorted by descending score chi-square statistic, and the five combinations with the highest score chi-square statistics are printed.

```
proc sort data=chisq2 out=results_sorted;
  by descending ChiSq;
run;

title3 "Regression Models Selected by Score Criterion";
proc print data=results_sorted (obs=5); *obs= controls the # of models displayed;
run;
```

OUTPUT

The results from the data step programming method are identical to those of the macro method. The only difference in output is that the variable number is printed (e.g. 19, 25, and 27) rather than the variable name (e.g. q19, q25, and q27).

Regression Models Selected by Score Criterion

Obs	ChiSq	i	j	k
1	341.6988	19	25	27
2	327.6492	15	19	27
3	322.3556	19	23	25
4	320.7365	23	25	27
5	319.1081	15	19	25

PROS AND CONS

On our system, speed was not a differentiator between the macro and data step programming methods; we did not observe a meaningful difference in the amount of time required to implement the two methods. The macro solution may be preferable if you have space constraints, because the data step programming method could create a very large data set. The data step programming method may be preferable if you are not comfortable with macros or prefer not to use macros because of the loss of transparency in the log file. We believe either set of code could be readily modified.

ADAPTING THE CODE

Our example is specific to selecting the five best models with three covariates from a set of 41 candidate variables. Strategies for adapting the code to different situations are provided below.

TO CHANGE THE NUMBER OF BEST MODELS REPORTED

To change the number of best models reported, modify the `obs=` option in the print statement at the end. For example, to print the 25 best models (analogous to `BEST=25`), the code would be adapted in the following way.

```
title3 "Regression Models Selected by Score Criterion";
proc print data=results_sorted (obs=25); *obs= controls the # of models displayed;
run;
```

TO CHANGE THE NUMBER OF CANDIDATE VARIABLES

To change the number of candidate variables, set the upper bound for the last (innermost) loop equal to the number of candidate variables. Set the upper bound for the second-to-last layer equal to one less than number of candidate variables, the upper bound for the third-to-last layer equal to two less than the number of candidate variables, and so on. The following code from the macro solution has been adapted to a situation with 99 candidate variables. The `DO` loop in the data step programming solution would be modified in the same way, but the array would also need to be adjusted: `array indvar {1:99} q1-q99`.

```
%do i = 1 %to 97; *set max value to number of candidate variables minus 2;
  %do j = &i.+1 %to 98; *set max value to number of candidate variables minus 1;
    %do k = &j.+1 %to 99; *set max value to the number of candidate variables;
      %slogistic(q&i.,q&j.,q&k.);
    %end;
  %end;
%end;
```

TO USE ALL CONTINUOUS VARIABLES OR A MIX OF CATEGORICAL AND CONTINUOUS VARIABLES

If all your candidate variables are continuous, simply remove the class statement from the SURVEYLOGISTIC procedure. We provide an easy way to allow for a mix of categorical and continuous variables in the macro solution but not the data step programming solution. In the macro solution, populate the class statement with all the categorical variables in the candidate set (this is possible because PROC SURVEYLOGISTIC does not complain when variables are included in the class statement but do not appear in the model statement). In our example, if all of our 41 variables were categorical except q5 and q10, then the code in the macro solution could be adapted in the following way.

```
proc surveylogistic data=studypop01;
  weight wt1812;
  class q1-q4 q6-q9 q11-q41;
  model dependvar = &ind1. &ind2. &ind3.;
  ods output GlobalTests=chisq1;
run;
```

TO CHANGE THE MODEL SIZE

Another modification likely to be made is to use a different model size than three. In both solutions, the DO loop needs to have as many layers as variables in the model; additionally, every layer of the DO loop needs to have a unique index variable. Assuming that the candidate variables are numbered sequentially starting at one, set the lower bound for the first DO loop to one. The lower bounds for the rest of the loops are set to one more than the index variable from the previous DO loop. The upper bounds are set according to the rules described in the section, Changing the Number of Candidate Variables.

In the macro solution, the number of arguments passed to the SLOGISTIC macro will need to be adjusted to match the model size.

```
%do h = 1 %to 38; *first layer—from 1 to (# of candidate variables - model size + 1)
  %do i = &h.+1 %to 39; *subsequent layers start at 1 + the previous index variable;
    %do j = &i.+1 %to 40;
      %do k = &j.+1 %to 41;
        %slogistic(q&h.,q&i.,q&j.,q&k.); *number of arguments equals the model size;
      %end;
    %end;
  %end;
%end

...

%macro slogistic (ind1,ind2,ind3,ind4); *number of arguments equals the model size;

proc surveylogistic data=studypop01;
  weight wt1812;
  class &ind1. &ind2. &ind3. &ind4.;
  model dependvar = &ind1. &ind2. &ind3. &ind4.;
  ods output GlobalTests=chisq1;
run;

* Keep only the score chi-square value;
data chisq2 (keep=ChiSq var1-var4); *keep statement adjusted to number of variables;
  set chisq1 (where=(test="Score"));
  var1="&ind1. "; *number of var variables needs to match the model size;
  var2="&ind2. ";
  var3="&ind3. ";
  var4="&ind4. ";
run;

data results;
  set results chisq2;
run;

%mend slogistic;
```

In the data step programming solution, the number of `valpred` variables also needs to be modified to match the model size. The following example provides the modifications required for a model size of 4.

```
array indvar {1:41} q1-q41;
do h = 1 to 38; *first layer-from 1 to (# of candidate variable - model size + 1);
  do i = h+1 to 39;
    do j = i+1 to 40;
      do k = j+1 to 41;
        valpred1=indvar{h}; *num of valpred variables needs to match model size;
        valpred2=indvar{i};
        valpred3=indvar{j};
        valpred4=indvar{k};
        output;
      end;
    end;
  end;
end;
end;
```

TO USE THE LIKELIHOOD RATIO OR WALD CHI-SQUARE STATISTIC INSTEAD OF THE SCORE CHI-SQUARE STATISTIC

We used the score chi-square statistic in order to recreate the `SELECTION=SCORE` option as it is implemented in `PROC LOGISTIC`. The score statistic is commonly used in model selection algorithms because of computational efficiencies relative to using other statistics such as the likelihood ratio. Because our algorithm is more basic and does not recognize any of those computational efficiencies, it would be very reasonable to use another statistic to rank the best models. For instance, you could use the likelihood ratio or Wald chi-square statistics by replacing the `where=(test="Score")` statement with `where=(test="Likelihood Ratio")` or `where=(test="Wald")`.

TO USE SCORE SELECTION WITH CLASS VARIABLES IN PROC LOGISTIC

Since the `SELECTION=SCORE` option in `PROC LOGISTIC` is not compatible with categorical variables in a class statement, you could adapt the solutions provided here to perform score selection in `PROC LOGISTIC` with categorical variables. The only change necessary would be to replace `PROC SURVEYLOGISTIC` with `PROC LOGISTIC` in the code.

CONCLUSION

This paper describes two methods for recreating the `SELECTION=SCORE` model specification with `PROC SURVEYLOGISTIC` and could be used as a starting point for adding other functionality to `SURVEYLOGISTIC` that is currently only available in `LOGISTIC`.

REFERENCES

Peters SP, Jones CA, Haselkorn T, Mink DR, Valacer DJ, Weiss ST. "Real-World Evaluation of Asthma Control and Treatment (REACT): Findings from a National Web-Based Survey." *Journal of Allergy and Clinical Immunology*, June 2007, Vol. 119, No. 6, pp. 1454-1461.

SAS Institute Inc., *SAS/STAT® User's Guide, Version 9*, Cary, NC: SAS Institute Inc., 2003.

CONTACT INFORMATION

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

Sara R. Adams
ICON Clinical Research
Formerly Ovation Research Group
188 Embarcadero, Suite 200
San Francisco, CA 94105
Work Phone: 415-371-2130
Fax: 415-856-0840
Email: sadams@ovation.org
Web: www.ovation.org

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