PharmaSUG 2014 - Paper SP04

%IC_LOGISTIC: A SAS® Macro to Produce Sorted Information Criteria (AIC/BIC) List for PROC LOGISTIC for Model Selection

Qinlei Huang, St. Jude Children's Research Hospital, Memphis, TN

Liang Zhu, St. Jude Children's Research Hospital, Memphis, TN

ABSTRACT

Model selection is one of the fundamental questions in statistics. One of the most popular and widely used strategies is model selection based on information criteria, such as Akaike Information Criterion (AIC) and Sawa Bayesian Information Criterion (BIC). It considers both fit and complexity, and enables multiple models to be compared simultaneously. PROC LOGISTIC is one of the most popular SAS procedures to perform logistic regression analysis on discrete responses including binary responses, ordinal responses, and nominal responses. However, there is no existing SAS procedure to perform model selection automatically based on Information Criteria for PROC LOGISTIC, given a set of covariates. This paper provides a SAS macro `%ic_logistic` to select a final model with the smallest value of AIC/BIC. Specifically, `%ic_logistic` will 1) produce a complete list of all possible model specifications given a set of covariates; 2) use do loop to read in one model specification every time and save it in a macro variable; 3) execute PROC LOGISTIC and use SAS/ODS to output AICs and BICs; 4) append all outputs and use SAS/DATA to create a sorted list of information criteria with model specifications; and 5) run PROC REPORT to produce the final summary table. Based on the sorted list of information criteria, researchers can easily identify the best model. This paper includes the macro programming language, as well as examples of the macro calls and outputs.

Keywords: Model Selection, Information Criterion, PROC LOGISTIC, SAS/ODS, SAS Macro

INTRODUCTION

This paper presents an accessible, flexible, and modifiable SAS macro `%ic_logistic` to perform model selection based on information criteria. Part I explains the framework and details of the macro program itself. It is intended for advanced users who wish to understand and/or modify the `%ic_logistic` code. Part II describes the usage of the macro program and provides hands-on examples for different data profiles. Using this macro requires basic knowledge of SAS and thus makes automatic model selection based on information criteria for PROC LOGISTIC available to any PC SAS user.

LOGISTIC REGRESSION

PROC LOGISTIC is one of the most popular SAS procedures to perform logistic regression analysis on discrete responses including binary responses, ordinal responses, and nominal responses.

Binary response

For a binary response *y* and an explanatory variable x, the logistic regression model is performed on the logit, Logit(p(Y=1))=log(p(Y=1)/(1-p(Y=1)))=α+βx. The following statements invoke PROC LOGISTIC to fit a model with `y1` as the binary response variable and two covariates `x1c` and `x2` as explanatory variables. `x1c` is a classification covariate and `x2` is a numerical covariate.

```
proc logistic data=work.a;
   class x1c;
   model y1 = x1c x2;
run;
```

Nominal response

For a nominal response *y* and an explanatory variable x, the logistic regression model is performed on the generalized logits. The following statements invoke PROC LOGISTIC to fit a model with `y2` as the nominal response variable and two covariates `x1c` and `x2` as explanatory variables. `x1c` is a classification covariate and `x2` is a numerical covariate.

```
proc logistic data=work.a;
   class x1c;
   model y2 = x1c x2 / link=glogit;
run;
```

INFORMATION CRITERION

Model selection based on information criteria, such as Akaike Information Criterion (AIC) and Sawa Bayesian Information Criterion (BIC), is a standard way and often recommended by reviewers. It considers both fit and complexity, and enables multiple models to be compared simultaneously.

The Akaike Information Criterion (AIC) is a measure of the relative quality of a statistical model, for a given set of data. AIC deals with the trade-off between the goodness of fit and the complexity of the model. For any statistical model, the AIC value is

AIC=2k-2ln(*L*),

where *k* is the number of parameters in the model, and *L* is the maximized value of the likelihood function for the model.

The Bayesian Information Criterion (BIC) or Schwarz Criterion (SC) is a criterion for model selection among a finite set of models. It is closely related to the Akaike information criterion (AIC) and introduces a larger penalty term for the number of parameters in the model to solve the problem of over-fitting.

PROC LOGISTIC reports AIC and BIC in the model fit statistics.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	96.486	99.461
SC	100.186	106.861
-2 Log L	92.486	91.461

Output 1. Model fit statistics output from a PROC LOGISTIC statement

MODEL SELECTION

There is no existing SAS procedure to perform model selection automatically based on information criteria for PROC LOGISTIC given a set of covariates. A SAS macro `%ic_logistic` is written to address the issue.

The number of models to be run grows at an exponential rate with the increase of covariates. For example, for four covariates, there will be 16 (2^4) possible model specifications; seven covariates require 128 (2^7) model specifications; ten covariates need 1,024 (2^{10}) model specifications; and fifteen covariates demand 32,768 (2^{15}) model specifications.

# of covariates	# of models	# of covariates	# of models
1	1	**11**	2,048
2	4	**12**	4,096
3	8	**13**	8,192
4	16	**14**	16,384
5	32	**15**	32,768
6	64	**16**	65,536
7	128	**17**	131,072
8	256	**18**	262,144
9	512	**19**	524,288
10	1,024	**20**	1,048,576

Table 1. Number of model specifications for specific number of covariates

It is very time consuming to copy paste programs and change model specifications manually, even with an average size of covariates. Also, it may cause errors easily by typos, and thus, not reliable. The SAS macro `%ic_logistic` offers an efficient, automatic, and reliable tool to solve the problem. It is adopted to choose a subset of covariates based on the smallest AIC or BIC.

DEVELOPMENT OF THE SAS MACRO `%IC_LOGISTIC`

The macro introduced here, `%ic_logistic`, is actually a group of smaller macros (`%modelcomb, %modelsp, %modelreadin, %logistic_binary_base, %logistic_binary, %logistic_nominal_base, %logistic_nominal, %dataappend, %datafinal, %report_ic, and %ic_logistic`). All macros are

saved in a central directory. Following conventional use, each macro is defined in the SAS program file of the same name --- e.g., `%modelsp` is defined in `%modelsp.sas`. Some of these macros call other macros. Here, we introduce the workflow of `%ic_logistic` and main ideas behind each macro program.

WORKFLOW

Framework

`%ic_logistic` produces the sorted list of information criteria via five steps:

1) Execute `%modelsp` to produce a complete list of all possible model specifications given a set of covariates;
2) Execute `%modelreadin` to read in one model specification each time and save it in a macro variable;
3) Execute `%logistic_binary` etc. to run PROC LOGISTIC and use SAS/ODS to output AICs and BICs;
4) Execute `%dataappend` and `%datafinal` to create a sorted list of information criteria with model specifications; and
5) Execute `%report_ic` to PROC REPORT the final summary table.

Below is the SAS code of `%ic_logistic`. It considers four typical scenarios for logistic regression data analysis: binary response without force-in covariates, binary response with force-in covariates, nominal response without force-in covariates, and nominal response with force-in covariates.

```
%macro ic_logistic(datain=,      response=,
                   varlist=,     n=,     k=0,
                   classlist=,   n2=,
                   forceinvar=,  forceinclass=,
                   binary=,      forcein=);

   %modelsp();

   %let n_model=%eval(2**&n);
   %put total number of model specifications is &n_model;

   * scenario 1: binary response without force-in covariate;

   %if &binary=1 and &forcein=0 %then %do;
    %put scenario is binary response without force-in covariate;

      %do i=2 %to &n_model;

         %modelReadin();

              %if &i=2 %then %do;
                 %logistic_binary_base();
                 %dataAppend;
              %end;

         %logistic_binary();

         %dataAppend;

      %end;
   %end;

   * scenario 2: binary response with force-in covariate;

   %if &binary=1 and &forcein=1 %then %do;
    %put scenario is binary response with force-in covariate;

      %do i=1 %to &n_model;

         %modelReadin();
```

```
            %logistic_binary();

            %dataAppend;

         %end;
      %end;

      * scenario 3: nominal response without force-in covariate;

      %if &binary=0 and &forcein=0 %then %do;
      %put scenario is nominal response without force-in covariate;

         %do i=2 %to &n_model;

            %modelReadin();

                 %if &i=2 %then %do;
                       %logistic_nominal_base();
                  %dataAppend;
                 %end;

            %logistic_nominal();

            %dataAppend;

         %end;
      %end;

      * scenario 4: nominal response with force-in covariate;

      %if &binary=0 and &forcein=1 %then %do;
      %put scenario is nominal response with force-in covariate;

         %do i=1 %to &n_model;

            %modelReadin();

            %logistic_nominal();

            %dataAppend;

         %end;
      %end;


      %dataFinal();

      %report_ic();

   %mend ic_logistic;
```

Step 1: model specification

Step 1a: %modelcomb

The SAS macro %modelcomb is used to produce a full combination sheet.

```
%modelcomb(n,k);
```

- n: the total number of covariates of interest for model selection
- k: the minimum number of covariates to be included for model selection (default value is 0)

For example, if there are four covariates of interest, `%modelcomb` will generate a combination sheet with 16 combinations and store it in a SAS dataset `comb`. M1 to M4 indicate the 4 covariates of interest. '1' indicates a covariate to be included; while '0' indicates a covariate NOT to be included. For example, the 1st line (0,0,0,0) defines an intercept only model with no covariate to be included; the 5th line (0,0,0,1) defines a model specification with M4 (the 4th covariate) to be included; the 10th line (0,1,0,1) defines a model specification with M2 (2nd covariate) and M4 (4th covariate) to be included; and the 16th line (1,1,1,1) defines a full model specification with all covariates of interests to be included.

```
%modelcomb(4,0);
```

M1	M2	M3	M4	id
0	0	0	0	1
1	0	0	0	2
0	1	0	0	3
0	0	1	0	4
0	0	0	1	5
1	1	0	0	6
1	0	1	0	7
1	0	0	1	8
0	1	1	0	9
0	1	0	1	10
0	0	1	1	11
1	1	1	0	12
1	1	0	1	13
1	0	1	1	14
0	1	1	1	15
1	1	1	1	16

Table 2. A combination sheet for four covariates of interest

Step 1b: `%modelsp`

The SAS macro `%modelsp` is implemented to transfer the combination sheet to a model specification sheet and store it in a SAS dataset `model`.

`%modelsp` first uses the above combination sheet and ARRAY statements to identify the covariates in each model specification. It next uses the CATX function to return a concatenated string of covariates with spaces as delimiters. It then stores the concatenated string in a column "modelvar". `&modelvar` is to be used in the PROC LOGISTIC MODEL statement.

`%modelsp` then uses the above combination sheet and ARRAY statements to identify the classification covariates in each model specification. It next uses the CATX function to return a concatenated string of classification covariates with spaces as delimiters. It then stores the concatenated string in a column "classvar". `&classvar` is to be used in the PROC LOGISTIC CLASS statement.

For example, `%modelsp` will transfer the above combination sheet, which is stored in a SAS data `comb`, to a model specification sheet and store it in a SAS data `model`, given four covariates (Table 3). Following is the SAS code to call `%modelsp`.

```
%modelsp(varlist=%str(x1c,x2,x3c,x4),
         n=4,
         classlist=%str(x1c,x3c),
         n2=2,
         k=0);
```

- varlist: the list of all covariates of interest for model selection, separated by comma
- n: the total number of all covariates of interest for model selection
- classlist: the list of all classification covariates of interest for model selection, separated by comma
- n2: the total number of all classification covariates of interest for model selection
- k: the minimum number of covariates to be included for model selection (default value is 0)

Combination sheet				id	Model specification sheet	
M1	M2	M3	M4		modelvar	classvar
0	0	0	0	1		
1	0	0	0	2	x1c	x1c
0	1	0	0	3	x2	
0	0	1	0	4	x3c	x3c
0	0	0	1	5	x4	
1	1	0	0	6	x1c x2	x1c
1	0	1	0	7	x1c x3c	x1c x3c
1	0	0	1	8	x1c x4	x1c
0	1	1	0	9	x2 x3c	x3c
0	1	0	1	10	x2 x4	
0	0	1	1	11	x3c x4	x3c
1	1	1	0	12	x1c x2 x3c	x1c x3c
1	1	0	1	13	x1c x2 x4	x1c
1	0	1	1	14	x1c x3c x4	x1c x3c
0	1	1	1	15	x2 x3c x4	x3c
1	1	1	1	16	x1c x2 x3c x4	x1c x3c

Table 3. A SAS dataset `model` stores a complete set of possible model specifications for four covariates, with classification variables indicated

Step 2: `%modelreadin` to read in model specification

`%modelreadin` is used to work with the `%do` loop to read in one model specification each time. It uses the CALL SYMPUT routine to assign the values in "modelvar" and "classvar" produced in step 1 to macro variables `&modelvar` and `&classvar`.

Step 3: `%logistic_binary` to run PROC LOGISTIC and output AIC/BIC using SAS/ODS

`%logistic_binary` and `%logistic_nominal` are used to run PROC LOGISTIC. `%logistic_binary` is written for binary responses and `%logistic_nominal` is written for nominal responses. ODS OUTPUT is used to output the fit statistics AIC and BIC and stores them in SAS datasets `AIC` and `BIC`. `%logistic_binary_base` and `%logistic_nominal_base` are used to run PROC LOGISTIC and output AIC and BIC for intercept only models. Below is the code for `%logistic_binary`. All other SAS macros for this step share the same strategy with minor differences.

- datain: the name of the input SAS dataset, including the name of the SAS library where the input dataset is located.
- response: the name of the response variable
- modelvar: the concatenated string of the covariates in a specific model specification, separated by space
- classvar: the concatenated string of the classification covariates in a specific model specification, separated by space
- forceinvar: the list of force-in covariates, separated by space
- forceinclass: the list of classification force-in covariates, separated by space

```
ods output fitstatistics=AIC(keep=Criterion InterceptAndCovariates
                             rename=(InterceptAndCovariates=AIC)
                             where=(Criterion='AIC'))
           fitstatistics=BIC(keep=Criterion InterceptAndCovariates
                             rename=(InterceptAndCovariates=BIC)
                             where=(Criterion='SC'));
proc logistic data=&datain;
   class &forceinclass &classvar;
   model &response = &forceinvar &modelvar;
run;
```

Step 4: `%dataappend & %datafinal` to create a sorted list of information criteria with model specifications

`%dataappend` is used to append the SAS datasets `AIC` and `BIC`, which are created by PROC LOGISTIC and ODS OUTPUT, to the base datasets `FINALAIC` and `FINALBIC`.

```
proc append base=finalAIC
            data=AIC;
run;

proc append base=finalBIC
            data=BIC;
run;
```

`%datafinal` is implemented to sort and merge the SAS datasets `finalAIC` and `finalBIC` to create a summary SAS dataset `finalIC`. `finalIC` contains the complete set of model specifications and corresponding AICs and BICs. `finalIC` is sorted by the values of AIC.

Step 5: `%report_ic` to PROC REPORT the final summary table

`%report_ic` is used to PROC REPORT the final summary table `finalIC`.

```
ods listing close;
proc report data=finalic nowindows headskip center split='#';
     column id aic bic modelvar;

     define id / display "No.";
     define aic / analysis "AIC" format=8.3;
     define bic / analysis "BIC" format=8.3;
     define modelvar / display "Model Specification";
run;
```

FOUR SCENARIOS

PROC LOGISTIC is one of the most popular SAS procedures to perform logistic regression analysis on discrete responses including binary responses, ordinal responses, and nominal responses. Binary and ordinal responses share the same MODEL statement; while the nominal response uses glogit as the link function.

In real data analysis, researchers are often interested in one or more specific covariates out of theoretical consideration. They thus want to always keep them in the model. In practice, we can force these covariates in when we execute SAS procedures such as PROC LOGISTIC.

The SAS macro `%ic_logistic` considers different scenarios in terms of response type and force-in status. `%ic_logistic` is designed for four typical scenarios for logistic regression data analysis: a binary response without force-in covariates, a binary response with force-in covariates, a nominal response without force-in covariates, and a nominal response with force-in covariates.

In the below section, we are going to describe the usage of the SAS macro `%ic_logistic` and provide hands-on examples.

USAGE OF THE SAS MACRO `%IC_LOGISTIC`

INSTALLATION

The macro introduced here, `%ic_logistic`, is actually a group of smaller macros (`%modelcomb, %modelsp, %modelreadin, %logistic_binary_base, %logistic_binary, %logistic_nominal_base, %logistic_nominal, %dataappend, %datafinal, %report_ic, and %ic_logistic`). To use it within a SAS program, put these macros (all found in the IC directory) into a central directory --- e.g., c:\SAS\ic. Then add the following to the SAS program before calling `%ic_logistic`.

```
%let icroot = c:\SAS\IC;
options mautosource sasautos=("&icroot",sasautos);
```

This tells SAS to look at the contents of the &icroot directory to find new macro definitions (in particular, `%ic_logistic` and its component macros).

PARAMETERS

In this section, we explain how to use `%ic_logistic`. Examples are provided for each scenario. Some of the parameters might best be understood via the examples that follow.

```
%ic_logistic(datain= ,
             response= ,
             varlist= ,
             n= ,
             k= ,
             classlist= ,
             n2= ,
             forceinvar= ,
             forceinclass= ,
             binary= ,
             forcein= );
```

- datain: the name of the input SAS dataset, including the name of the SAS library where the input dataset is located
- response: the name of the response variable
- varlist: the list of all covariates of interest for model selection, separated by comma
- n: the total number of all covariates of interest for model selection
- k: the minimum number of covariates to be included for model selection (default value is 0)
- classlist: the list of classification covariates of interest for model selection, separated by comma
- n2: the total number of classification covariates of interest for model selection
- forceinvar: the list of force-in covariates, separated by space
- forceinclass: the list of classification force-in covariates, separated by space
- binary: an indicator indicating if the response is a binary variable or a nominal variable ('1' indicates binary response; '0' indicates nominal response)
- forcein: an indicator indicating if there is any force-in covariate ('1' indicates yes; '0' indicated no force-in covariate)

EXAMPLES

Our examples will be based on an artificial dataset with six variables: y1, y2, x1c, x2, x3, and x4c. y1 is a binary response variable with values of 1 and 2. y2 is a nominal response variable with values of 1, 2 and 3. x1c and x4c are classification covariates with values of 0 and 1. x2 and x3 are numeric covariates with values ranging from 0 to 1. Table 4 shows the first ten observations of the dataset.

Obs	y1	y2	x1c	x2	x3	x4c
1	2	3	0	0.49	0.45	0
2	1	2	1	0.51	0.73	0
3	2	2	1	0.83	0.37	1
4	1	2	1	0.82	0.83	1
5	2	3	0	0.09	0.91	0
6	2	3	1	0.77	0.42	1
7	2	3	1	0.26	0.92	1
8	2	2	1	0.94	0.88	0
9	1	1	1	0.73	0.38	0
10	1	2	0	0.76	0.38	0

Table 4. Dataset for model selection using PROC LOGISTIC (first ten observations)

Example 1: binary response without force-in covariates

The first example addresses a binary response (`binary=1`) without force-in covariates (`forcein=0`). The input dataset is `ic` data in the `work` library (`datain=%str(work.ic)`). All four covariates (`x1c, x2, x3, x4c`) are to be considered for model selection (`varlist=%str(x1c,x2,x3,x4c), n=4`). Among them, `x1c` and `x4c` are

classification covariates (`classlist=%str(x1c,x4c), n2=2`). There is no force-in covariates to be included (`forceinvar=%str(), forceinclass=%str()`).The minimum number of covariates to be selected is 0, the default number for k, and thus no need to specify. The total number of possible model specifications is 16 (2^4).

```
%ic_logistic(datain=%str(work.ic),
             response=%str(y1),
             varlist=%str(x1c,x2,x3,x4c),
             n=4,
             classlist=%str(x1c,x4c),
             n2=2,
             forceinvar=%str(),
             forceinclass=%str(),
             binary=1,
             forcein=0);
```

No.	AIC	BIC	Model Specification
8	65.558	71.109	x1c x4c
14	65.739	73.140	x1c x3 x4c
5	65.897	69.598	x4c
11	66.051	71.602	x3 x4c
1	66.109	67.960	
2	66.139	69.839	x1c
4	67.129	70.829	x3
7	67.247	72.797	x1c x3
13	67.557	74.958	x1c x2 x4c
16	67.588	76.839	x1c x2 x3 x4c
15	67.848	75.248	x2 x3 x4c
10	67.891	73.441	x2 x4c
3	68.040	71.740	x2
6	68.089	73.639	x1c x2
9	68.810	74.360	x2 x3
12	68.997	76.398	x1c x2 x3

Table 5. Sorted information criteria (by AIC) for model selection: binary response without force-in covariates

In the first scenario, the SAS macro `%ic_logistic` produces a summary dataset `finalIC` including sorted AIC and BIC for the 16 model specifications, from an intercept only model (#1) to a full model (#16) with all four covariates included (Table 5). Based on the table, we can reach the conclusion that model #8 (the model with predictive covariates x1c and x4c) is the best model considering the smallest AIC (65.558). If BIC is the criterion you prefer, then model #1 (the intercept only model) is the best model considering the smallest BIC (67.960). To decide which information criteria to use is out of the scope of this paper. Audience may read statistical methods articles about model selection for reference.

Example 2: binary response with force-in covariates

The second example addresses a binary response (`binary=1`) with force-in covariates (`forcein=1`). The input dataset is `ic` data in the `work` library (`datain=%str(work.ic)`). Three out of the four covariates (`x2, x3, x4c`) are to be considered for model selection (`varlist=%str(x2,x3,x4c), n=3`). Among them, `x4c` is a classification covariate (`classlist=%str(x4c), n2=1`). There is one force-in covariate `x1c` and it is a classification covariate (`forceinvar=%str(x1c), forceinclass=%str(x1c)`). The minimum number of covariates to be selected is 0, the default number for k, and thus no need to specify. The total number of possible model specifications is 8 (2^3).

```
%ic_logistic(datain=%str(work.b),
             response=%str(y1),
             varlist=%str(x2,x3,x4c),
             n=3,
             classlist=%str(x4c),
             n2=1,
             forceinvar=%str(x1c),
             forceinclass=%str(x1c),
             binary=1,
             forcein=1);
```

No.	AIC	BIC	Model Specification
4	65.558	71.109	x4c
7	65.739	73.140	x3 x4c
1	66.139	69.839	
3	67.247	72.797	x3
6	67.557	74.958	x2 x4c
8	67.588	76.839	x2 x3 x4c
2	68.089	73.639	x2
5	68.997	76.398	x2 x3

Table 6. Sorted information criteria (by AIC) for model selection: binary response with force-in covariates

In the second scenario, the SAS macro `%ic_logistic` produces a summary dataset `finalIC` including sorted AIC and BIC for the 8 model specifications, from a model with force-in covariate only (#1) to a full model (#8) with all covariates of interest included (Table 6). Based on the table, we can reach the conclusion that model #4 (the model with force-in covariate x1c and predictive covariate x4c) is the best model considering the smallest AIC (65.558). If BIC is the criteria you prefer, then model #1 (the model with fore in covariate x1c only) is the best model considering the smallest BIC (69.839).

Example 3: nominal response without force-in covariates

The third example addresses a nominal response (`binary=0`) without force-in covariates (`forcein=0`). The input dataset is `ic` data in the `work` library (`datain=%str(work.ic)`). All four covariates (`x1c, x2, x3, x4c`) are to be considered for model selection (`varlist=%str(x1c,x2,x3,x4c), n=4`). Among them, `x1c` and `x4c` are classification covariates (`classlist=%str(x1c,x4c), n2=2`). There is no force-in covariates to be included (`forceinvar=%str(), forceinclass=%str()`). The minimum number of covariates to be selected is 0, the default number for k, and thus no need to specify. The total number of possible model specifications is 16 (2^4).

```
%ic_logistic(datain=%str(work.b),
             response=%str(y2),
             varlist=%str(x1c,x2,x3,x4c),
             n=4,
             classlist=%str(x1c,x4c),
             n2=2,
             forceinvar=%str(),
             forceinclass=%str(),
             binary=0,
             forcein=0);
```

No.	AIC	BIC	Model Specification
1	96.486	100.186	
5	97.498	104.899	x4c
3	98.444	105.845	x2
2	99.461	106.861	x1c
10	99.750	110.851	x2 x4c
4	99.872	107.272	x3
11	100.102	111.203	x3 x4c
8	100.529	111.630	x1c x4c
9	100.867	111.968	x2 x3
15	100.987	115.788	x2 x3 x4c
6	101.312	112.413	x1c x2
13	102.690	117.491	x1c x2 x4c
7	102.783	113.884	x1c x3
14	103.129	117.930	x1c x3 x4c
12	103.393	118.194	x1c x2 x3
16	103.691	122.193	x1c x2 x3 x4c

Table 7. Sorted information criteria (by AIC) for model selection: nominal response without force-in covariates

In the third scenario, the SAS macro `%ic_logistic` produces a summary dataset `finalIC` including sorted AIC and BIC for the 16 model specifications, from an intercept only model (#1) to a full model (#16) with all four covariates included (Table 7). Based on the table, we can reach the conclusion that model #1 (the intercept only model) is the best model considering the smallest AIC (96.486) and the smallest BIC (100.186).

Example 4: nominal responses with force-in covariates

The fourth example addresses a nominal response (`binary=0`) with force-in covariates (`forcein=1`). The input dataset is `ic` data in the `work` library (`datain=%str(work.ic)`). Three out of the four covariates (`x2, x3, x4c`) are to be considered for model selection (`varlist=%str(x2,x3,x4c), n=3`). Among them, `x4c` is a classification covariate (`classlist=%str(x4c), n2=1`). There is one force-in covariate `x1c` and it is a classification covariate (`forceinvar=%str(x1c)`, `forceinclass=%str(x1c)`). The minimum number of covariates to be selected is 0, the default number for k, and thus no need to specify. The total number of possible model specifications is 8 (2^3).

```
%ic_logistic(datain=%str(b),
             response=%str(y2),
             varlist=%str(x2,x3,x4c),
             n=3,
             classlist=%str(x4c),
             n2=1,
             forceinvar=%str(x1c),
             forceinclass=%str(x1c),
             binary=0,
             forcein=1);
```

No.	AIC	BIC	Model Specification
1	99.461	106.861	
4	100.529	111.630	x4c
2	101.312	112.413	x2
6	102.690	117.491	x2 x4c
3	102.783	113.884	x3
7	103.129	117.930	x3 x4c
5	103.393	118.194	x2 x3
8	103.691	122.193	x2 x3 x4c

Table 8. Sorted information criteria (by AIC) for model selection: nominal response with force-in covariates

In the fourth scenario, the SAS macro `%ic_logistic` produces a summary dataset `finalIC` including sorted AIC and BIC for the 8 model specifications, from a model with force-in covariate only (#1) to a full model (#8) with all covariates of interest included (Table 8). Based on the table, we can reach the conclusion that model #1 (the model with force-in covariate x1c only) is the best model considering the smallest AIC (99.461) and the smallest BIC (106.861).

DISCUSSION

We often get questions from the users of the SAS macro `%ic_logistic`. The following part discusses two of the most frequently asked questions.

The first one is, when the number of covariates of interest goes above 10, the number of models to be run increases dramatically and thus consumes a lot of computation resources. For example, with 15 covariates, 32,768 models need to be run to produce the sorted list of information criteria. Though the SAS macro `%ic_logistic` can still produce the results automatically, it takes time and occupies considerable computation resources. One solution is to group the covariates of interest into smaller groups, with each group containing less than 10 covariates. First run `%ic_logistic` on each smaller group to get the best subset of covariates for each group. Then run `%ic_logistic` again based on the covariates selected in the first step. If the number of covariates is still large, repeat the above two steps until obtaining a reasonable size of covariates.

The second one is in fact related to the first one. When there are many models to be run, sometimes, the SAS log and SAS output reach the limit of the SAS program and stop the execution of the macro. One solution is using the PROC PRINTTO statement to route the SAS log and output into an external file.

We encourage the request and use of the SAS macro from all legal SAS users. We appreciate all comments and questions brought up by our users.

CONCLUSION

This paper introduces a SAS macro `%ic_logistic` to perform model selection based on information criteria for PROC LOGISTIC in an automatic, efficient, and reliable way. `%ic_logistic` is actually a group of smaller macros. `%ic_logistic` works for discrete responses including binary responses, ordinal responses, and nominal responses. It accounts for models with or without force-in covariates. It can also be easily modified and extended to support users own needs.

REFERENCES

Bozdogan, Hamparsum. 1987. “Model Selection and Akaike’s Information Criterion (AIC): The General Theory and Its Analytical Extensions.” *Psychometrika*, 52(3): 345-370.

Menard, Scott. 2001. *Applied Logistic Regression Analysis*. Sage Publications, CA: Thousand Oaks.

ACKNOWLEDGMENTS

My special thanks to Dr. James Boyett and the department of Biostatistics at the St. Jude Children’s research hospital for the exceptionally supportive working environment and great learning resources. My appreciation dedicates to Dr. Sean Phipps and Dr. Hui Zhang for their continuous support to my work. I also owe my sincere gratitude to Dr. Kumar Srivastava, experienced Biostatisticians Catherine Billups, Yinmei Zhou, and Wei Liu for their valuable comments on my work.

RECOMMENDED READING

- Base SAS® Procedures Guide
- SAS® Certification Prep Guide: Advanced Programming for SAS® 9

CONTACT INFORMATION

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

Name: Qinlei Huang
Enterprise: St. Jude Children’s Research Hospital
Address: 262 Danny Thomas Place, Mail Stop 723
City, State ZIP: Memphis, TN 38103
Work Phone: 901.595.2027
Fax: 901.595.4585
E-mail: qinlei.huang@stjude.org
Web: www.stjude.org