Reporting Results of Multiple Logistic Regression Models Depending on the Availability of Data

Richard M. Mitchell, Westat, Rockville, MD

ABSTRACT

This paper discusses a process of developing multiple logistic regression models based on the availability of data, as well as the presentation of corresponding results. The process was developed for an individual patient data meta-analysis (IPD-MA) because the author's team was faced with the problem of creating a standard, uniform model that would push all available covariates through SAS/STAT® software's PROC LOGISTIC allowing results to be compared among 17 international studies with similar information. Studies did not necessarily collect the same set of variables nor use identical definitions, and data that were collected sometimes included high percentages of missing values. Since PROC LOGISTIC requires uniform coding and does not accommodate missing data, a program was needed to enhance the process of input to and output from the PROC LOGISTIC procedure.

The most important steps in implementing this process are not in running the PROC LOGISTIC model, but in utilizing SAS® software tools to filter the data in and out of the procedure so that meaningful results may be presented. These steps include the following:

- Determining the availability of data
- Selecting variables for the regression model
- Running PROC LOGISTIC
- Calculating additional statistics
- Presenting the data

INTRODUCTION

Reporting results of multiple logistic regression models can be performed easily and quickly with an automated process that directs data through PROC LOGISTIC based on its evaluation of the availability of data. The author's analysis team developed and implemented such a process where a standard model could be used for an infinite number of studies. As many variables as possible would be pushed through PROC LOGISTIC to adjust for covariates and examine odds ratios for specific variables. Since PROC LOGISTIC excludes entire records when one of the covariates is missing, criteria were established to select independent variables that contained low percentages of missing values. The outcome of this process is a single SAS program that produces regression models by considering all covariates, but only retaining those with certain degrees of missing values.

IMPLEMENTATION OF THE PROCESS

Determining the Availability of Data

Since PROC LOGISTIC eliminates entire records from a regression analysis if any one variable in the model statement contains a missing value, you may choose to exclude variables from the model depending on the availability of data. The author's analysis team chose an arbitrary number of 25% such that all variables whose missing proportion exceeded this limit would be excluded. Any larger proportion than this may potentially cause bias, and conclusions drawn from the remaining data may not be reliable or useful. By utilizing this approach to focus on the exclusion of variables rather than observations, we are avoiding the elimination of data that may otherwise have improved the power of the model. In Figure 1 below, an example of 3 covariates (the author's team used 27 in the actual analysis) are examined to determine their percentage of missing values. After table information from PROC FREQ is directed to an output file (MOUT&I), a data set is created for each variable with a corresponding flag (MISSFLAG) to indicate whether the variable should later be excluded because of its high percentage of missing values. Since PROC FREQ provides missing data first in its output file, it is only necessary to examine the first record by using the option OBS=1.

```
/* Identify all variables, 1 to n */

%let fname1=var1;
%let fname2=var2;
%let fname3=var3;

/* Assess missing pct. for each variable using PROC FREQ */

%macro nummiss (dsn=, missval=);
  %do i=1 %to 3;
    proc freq data=ipd.&dsn noprint;
      tables &&fname&i / missing out=mout&i;
    data missing&i;
      set mout&i (obs=1);
      if &&fname&i=. and percent >= &missval then missflag=1;
      else missflag=0;
  %end;
%mend nummiss;

%nummiss (dsn=study1, missval=25.00);
```

Figure 1

Selecting Variables For the Regression Model

After the missing exclusion criteria is applied to all variables for a given data set, a selection process is implemented such that two strings of variables are generated for:

1. the regression model for PROC LOGISTIC
2. documenting the excluded variables and reporting these results

As each variable is examined in Figure 2 below (the dataset MISSING1 represents VAR1, MISSING2 represents VAR2, etc.), the value of MISSFLAG directs the program to assign that variable to one of the two strings: the regression model (LOGLINE) or the listing of excluded variables (MISSLINE). Still, extra code is needed to enhance the look of the output. Although only the actual dummy variable name is needed for the regression model, the labels of each excluded variable are more descriptive in the final presentation of the data. Note that yes/no indicator variables have already been created for PROC LOGISTIC where each variable level is noted by an A, B, etc. For example, our original variable VAR2 has been separated into two levels where VAR2A indicates whether or not the data are represented in level 2 (e.g. range of 10-20), and VAR2B indicates whether or not the data are represented in level 3 (e.g. range of 21 and higher). Each of these levels are ultimately compared to their reference group level 1 (e.g. range of 0-9).

```
data _null_;
  set missing1(in=a) missing2(in=b) missing3(in=c) end=last;

  /* Define specifications for string variables and components */

  length logline missline $ 200;
  retain logline missline;
  set1='var1a'; set2='var2a var2b'; set3='var3a var3b';
  name1='Variable 1'; name2='Variable 2'; name3='Variable 3';

  /* Define arrays of names, set strings, and dsn identifiers */

  array namevars{*} name1-name3;
  array setvars{*} set1-set3;
  array allsets{*} a b c;

  /* Implement loop – assign variables to appropriate string */

  do i=1 to dim(allsets);
    if missflag=0 and allsets{i} then logline=trim(logline) || " " ||
      setvars{i};
    else if missflag=1 and allsets{i} then do;
      if (trim(missline) ne " ") then missline=trim(missline) || ", ";
      missline=trim(missline) || " " || namevars{i};
    end;
  end;
  if last then do;
    call symput("logline",trim(logline));
    call symput("missline",trim(missline));
  end;
```

Figure 2

In the final preparation before running and reporting the results of the regression models through PROC LOGISTIC, global macro variables are produced by utilizing the SYMPUT function on the LAST record (assigned as END=LAST). The string, LOGLINE, represents all variables with < 25% missing values that are included in the regression model, while the string, MISSLINE, documents the names of those variables that are excluded from the analysis.

A dilemma of interest encountered by our analysis team was that some studies collected one of two common variables, but not always both. For example, a COUNT and PERCENTAGE of a given variable may represent similar thresholds and meanings, however, STUDY 1 may have collected only COUNT while STUDY 2 collected only PERCENTAGE. An additional criteria was needed such that only VAR2 or VAR3 would be analyzed in the model, but never both because of their high correlation to each other. To be consistent across studies, it was determined to always use VAR2 unless this variable was excluded because of a high percentage of missing values, otherwise, VAR3 would be used, assuming of course that it also met the missing criteria. Flags were assigned during the selection process of common variables such as these (code not shown in this condensed version).

Running PROC LOGISTIC

After determining the variables that will be included in the regression model for each study, PROC LOGISTIC is run (Figure 3 below) where the parameter estimates and covariances are directed to an output file (COEFF). This same code would be used with all of the participating studies (&DSN) where the selected variables in the regression model (&LOGLINE) change depending on their availability. Thus, a regression is run on the maximum number of variables for each study as they are related to a dependent variable (DEP1). By adjusting for the covariates in the model, odds ratios are examined for specific variable attributes.

```
proc logistic data=&dsn outest=coeff covout;
  model dep1 = &logline;
  output out=allout;
```

Figure 3

As mentioned previously, PROC LOGISTIC only processes records that contain non-missing data for all variables that are included in the MODEL statement, thus the reason why we have excluded variables with high percentages of missing values. Given that there are 100,000 records in a file, if only 20,000 were used, then little reliability could be placed on the results of the analysis. The number of records that are actually processed may be of interest and is therefore determined in Figure 4 on the following page. The PROC LOGISTIC output file, ALLOUT, is used here to count the number of records that have no missing values from any of the variables included in the macro variable &LOGLINE. The SYMPUT function is again utilized to convert this result to a global macro variable, &INCL, and later presented as a footnote in the final report. Note that the SET statement

in the macro NUMMISS is never actually executed since "IF 0" is never true, and we are able to determine the number of records in the data set without any compilation.

```
data ipdout.lognum allmiss;
  set allout;
  array misschk &logline;
  do over misschk;
    if misschk=. then do;
      output allmiss;
      return;
      end;
    end;
  output ipdout.lognum;

%macro nummiss(dsn);
  %global include;
  data _null_;
    if 0 then set &dsn nobs=count;
    call symput("include",trim(left(put(count,4.))));
    stop;
%mend nummiss;

%nummiss(dsn=ipdout.lognum);
```

Figure 4

Calculating Additional Statistics

Although PROC LOGISTIC does not directly provide odds ratios, p-values, and confidence intervals for the model as a whole in its output files, it does provide a mechanism for you to calculate these statistics. In Figure 5 below, the output file COEFF (generated earlier by the PROC LOGISTIC option OUTEST) is utilized to extract the variances and the parameter estimates for all of the variables in the regression model. For the parameter estimates (_TYPE_=PARMS), data for all beta coefficients are included in the first row of the data set, while the variances are extracted from the diagonal of the covariance matrix (_TYPE_=COV). This is accomplished through the DATA step, although SAS/IML® software may provide a more optimal approach for the more skilled programmer.

Sample PROC PRINT of COEFF:

```
OBS  _LINK_  _TYPE_  _NAME_     INTERCEP    VAR1    VAR2A     VAR2B      LNLIKE
 1   LOGIT   PARMS   ESTIMATE  -2.96925   0.96522  0.62067   0.41379   -2341.77
 2   LOGIT   COV     INTERCPT   0.03327  -0.03033 -0.00500  -0.00444   -2341.77
 3   LOGIT   COV     VAR1      -0.03033   0.03145  0.00102   0.00044   -2341.77
 4   LOGIT   COV     VAR2A     -0.00500   0.00102  0.01226   0.00403   -2341.77
 5   LOGIT   COV     VAR2B     -0.00444   0.00044  0.00403   0.00704   -2341.77
```

Code Used to Extract Data From COEFF:

```
data cov(keep=_name_ variance varname) coeffb;
  set coeff;

  length varname $ 5;
  array allvars{*} var1a var2a var2b var3a var3b;
  do i=1 to dim(allvars);
     call vname (allvars{i}, varname);
     /* Read the diagonal */

     if _type_='cov' and _name_=varname then do;
       variance=allvars{i};          /* assign variance                */
       output cov;
     end;

     /* Read the first row */

     else if _type_='parms' then do;
       _name_=varname;               /* assign variable name           */
       llab=i;                       /* assign variable level format */
       bcoeff=allvars{i};            /* assign beta coefficient        */
       if bcoeff ne . then output coeffb;
     end;
  end;
```

Figure 5

After extracting the relevant information from the PROC LOGISTIC output files, you are now ready to begin calculating the additional statistics. In Figure 6 below, the parameter coefficients (COEFFB) and variances (COV) are merged together resulting in 1 record of data per variable. These statistics are then used to compute the odds ratio, and the lower and upper confidence intervals. To calculate the p-value, the function PROBCHI is utilized after the z value is calculated based on the beta coefficient and the square root of the variance computed by PROC LOGISTIC. Additional statements (not shown) are utilized in a final step to tailor the appearance of the data for presentation using PROC REPORT. These statements include converting numeric variables to character variables so that a variety of additional footnotes can be added, as well as accommodation of special formats (e.g. p < 0.001 for highly significant results).

```
data covar;
  merge coeffb cov;
  by _name_;
  or=exp(bcoeff);                          /* odds ratio             */
  lci=exp(bcoeff-(1.96*sqrt(variance)));   /* lower conf interval    */
  uci=exp(bcoeff+(1.96*sqrt(variance)));   /* upper conf interval    */
  zval=(bcoeff/sqrt(variance))**2;         /* calculate z squared    */
  pval=1-probchi(zval,1);                  /* identify p-value       */
```

Figure 6

Presenting the Data

PROC REPORT is utilized in Figure 7 on the following page to produce a final presentation of the regression results. By this stage of the process the data have been manipulated extensively through a fully automated process that has been carefully planned out between the study statistician and the programmer. The results are now presented in a simple manner that may be repeated for any number of studies and for any select group of variables. Provided in the title (&NUMSUBS) and footnote (&INCL) are the population sizes for all records before and after running PROC LOGISTIC. These counts give you a better idea of the population for each study that was actually utilized based on the variables selected. Note that while FOOTNOTE10 is centered, FOOTNOTE1

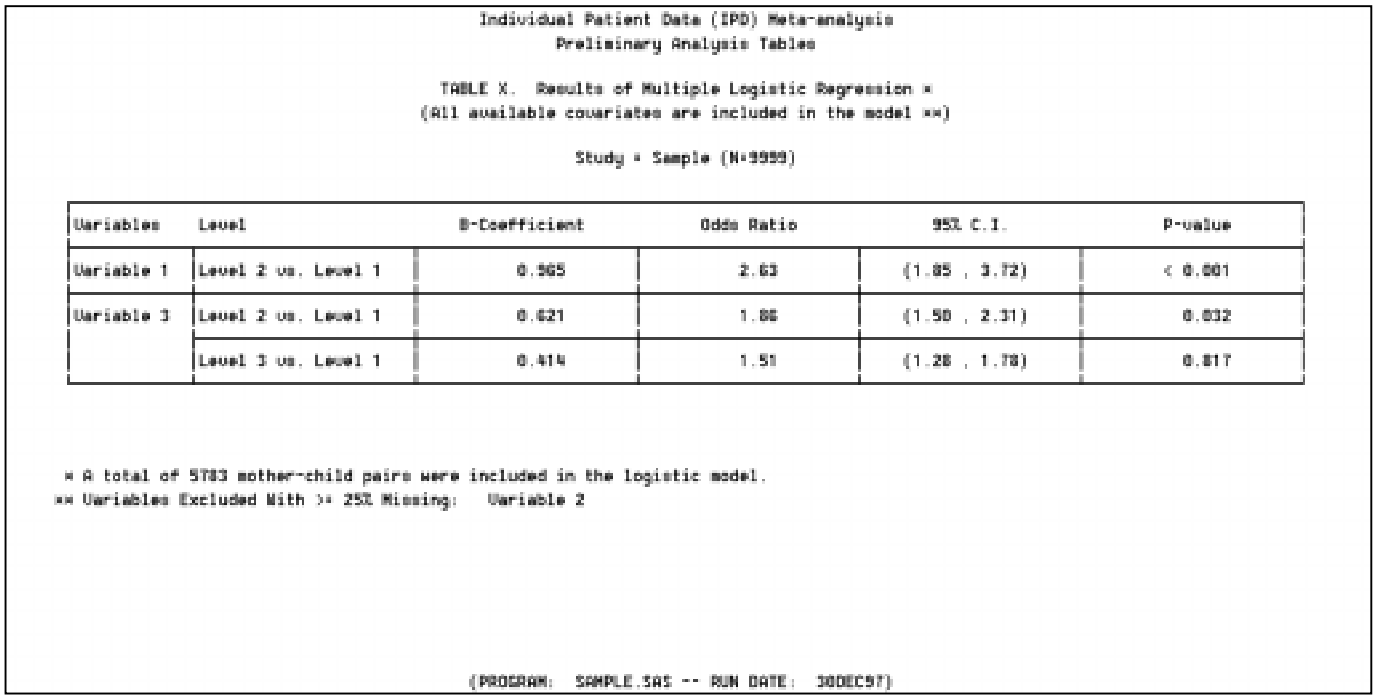
and FOOTNOTE2 are left justified by placing additional blank spaces in quotes to exceed the maximum linesize.

```
proc report data=ipdout.table5 nowindows spacing=1 split='*' box;
  format _name_ $varname. llab llab.;
  column _name_ llab bcoeff or ci pv;
  define _name_   / group width=12 left order=internal 'variables';
  define llab     / group width=22 left order=data 'level';
  define bcoeff   / display width=22 center;
  define or       / display width=22 center;
  define ci       / display width=22 center;
  define pv       / display width=22 left '    P-value';
  title1 'Individual Patient Data (IPD) Meta-analysis';
  title2 'Preliminary Analysis Tables
  title4 'Table X:  Results of Multiple Logistic Regression *';
  title5 '(All available covariates are included in the model **)';
  title7 "study = sample (n=&numsubs)*";
  footnote1 "   * A total of &incl mother-child pairs were      "
    " included in the logistic model.                          "
    "                                                          " ;
  footnote2 "   ** Variables excluded with >= 25% missing:    "
    "&missline                                                 "
    "                                                          ";
  footnote10 "(PROGRAM:  SAMPLE.SAS – RUN DATE:  "
    "&sysdate)";
```

Figure 7

The final output from PROC REPORT is shown below in Figure 8. After all of the careful planning, manipulation of data, and "packaging" of the data, you are provided with a single page presenting a wealth of information. For each level comparison of the corresponding variables, you are able to see the parameter coefficients and odds ratios. Lower and upper confidence intervals for these odds ratios are also presented along with the probability that each variable is significant after controlling for the effects of other covariates.

While the main focus of the report is the tabular presentation of results, other items of interest are provided. Note that the variable VAR2 (Variable 2) is excluded from the analysis because 25% or more of its data were missing. As indicated in the title, the sample data set utilized to perform this analysis included 9,999 records. Even though VAR2 was excluded, missing values for other variables included in the regression model still decreased the number of records that were actually included in the analysis (5,783 records). In this example, 80% of the data are missing for VAR2. If VAR2 had not been excluded, PROC LOGISTIC would have eliminated at least 7,999 records (9999*.80). Only 3 sample variables have been included in this process to simplify its presentation. In a real example, one may more appreciate the complexity of this process as the number of potential variables that are both included and excluded from the regression model are expanded.



Individual Patient Data (IPD) Meta-analysis
Preliminary Analysis Tables

TABLE X. Results of Multiple Logistic Regression *
(All available covariates are included in the model **)

Study = Sample (N=9999)

Variables	Level	B-Coefficient	Odds Ratio	95% C.I.	P-value
Variable 1	Level 2 vs. Level 1	0.965	2.63	(1.85 , 3.72)	< 0.001
Variable 3	Level 2 vs. Level 1	0.621	1.86	(1.50 , 2.31)	0.032
	Level 3 vs. Level 1	0.414	1.51	(1.28 , 1.78)	0.017

* A total of 5783 mother-child pairs were included in the logistic model.
** Variables Excluded With >= 25% Missing: Variable 2

(PROGRAM: SAMPLE.SAS -- RUN DATE: 30DEC97)

Figure 8

CONCLUSION

Once the automated process of filtering data into and out of PROC LOGISTIC was complete, the author's analysis team was provided with a fast and easy mechanism to better understand multiple data sets individually during the preliminary analysis stage. Before implementing such a process, the time and effort needed to write the code should be weighed against the number of datasets and variables involved. An approach of dumping all available covariates into a regression model may not be suitable for all analysis scenarios and the results from such an approach should be interpreted cautiously. The determination of which variables should be included in the final model will occur in a later stage that involves careful planning and extensive examination of the variable relationships. By utilizing the proper SAS tools, and establishing well-defined criteria, the results of complex statistical procedures such as PROC LOGISTIC may be presented to others in a format that is simple, yet informative.

ACKNOWLEDGEMENTS

Many thanks to Dr. Parivash Nourjah (Westat – Rockville, MD) not only for her invaluable contributions in all phases of this process, but for taking the time to explain the how's and why's of the related statistical concepts. Also much appreciation to Duke Owen (Westat – Rockville, MD) for encouraging me to write this paper.

Corresponding Author:

Richard M. Mitchell
Westat
1650 Research Boulevard, WB 496
Rockville, MD 20850
(301) 251-4386 (voice)
(301) 738-8379 (fax)
MITCHER1@WESTAT.COM