

Combining PROC GENMOD Models with Multinomial Outcomes Using PROC MIANALYZE

Marie Gantz, Ph.D., RTI International, Research Triangle Park, NC

ABSTRACT

Using SAS® to analyze multiply imputed data involves analyzing each imputation separately and then combining those results using the MIANALYZE procedure. Individual imputations may be analyzed using a variety of SAS procedures, including PROC GENMOD. Resulting parameter estimates and associated covariance matrices are output to SAS datasets using ODS. The ODS tables are then input into PROC MIANALYZE to obtain results that apply to the multiply imputed data as a whole. When the dependent variable is binomial, the ODS tables output by PROC GENMOD are in a format recognized by PROC MIANALYZE. When the outcome is multinomial, however, modification of the ODS tables is necessary before PROC MIANALYZE will accept them as input. The macro described in this paper automates the process of analyzing each imputation with PROC GENMOD, outputting the parameter estimates and covariance matrices using ODS, restructuring the ODS tables, and combining the results into one overall inference for the multiply imputed data using PROC MIANALYZE.

INTRODUCTION

Analysis of multiply imputed data involves first analyzing each imputation separately and then combining the results into one overall inference. The first step can be accomplished using a wide variety of SAS® procedures, including GLM, LOGISTIC, MIXED and GENMOD. Parameter and covariance estimates from the separate imputation analyses are output to SAS data sets using ODS. These tables are then input into the MIANALYZE procedure, which combines the estimates and produces one overall analysis for the multiply imputed data.

When PROC GENMOD is used to analyze the individual imputations, three ODS tables must be output in order for PROC MIANALYZE to then combine the results. The ODS name of the first table is ParameterEstimates. This data set contains the parameter estimates, standard errors, confidence intervals, degrees of freedom, and chi-square tests of significance for each of the independent variables. Separate results are listed for each imputation. The first few observations from a sample ParameterEstimates data set are shown below. These estimates are from a model to predict smoking. The outcome is binomial, and the DIST=BIN option is specified in the MODEL statement of PROC GENMOD. The other predictors are treatment, which is binary, and age, which is continuous.

ParameterEstimates ODS Table for a Binomial Outcome

Obs	_Imputation_	Parameter	DF	Estimate	StdErr	Lower WaldCL	Upper WaldCL	ChiSq	Prob ChiSq
1	1	Intercept	1	2.1214	0.3919	1.3534	2.8895	29.31	<.0001
2	1	Treatment	1	-0.0764	0.1451	-0.3608	0.2080	0.28	0.5985
3	1	Age	1	-0.0349	0.0129	-0.0601	-0.0097	7.34	0.0067
4	1	Scale	0	1.0000	0.0000	1.0000	1.0000	—	—
5	2	Intercept	1	2.1726	0.3933	1.4017	2.9434	30.52	<.0001

The second ODS table is Covb, which contains the covariance matrix for the model parameters, including intercepts. The first ten observations from a sample Covb data set are shown below. All parameters are labeled Prm1, Prm2, etc., instead of using the actual variable names.

Covb ODS Table for a Binomial Outcome

Obs	_Imputation_	Row Name	Prm1	Prm2	Prm3
1	1	Prm1	0.1535654	-0.030477	-0.004076
2	1	Prm2	-0.030477	0.0210559	-0.000055
3	1	Prm3	-0.004076	-0.000055	0.0001658
4	2	Prm1	0.1546769	-0.0306	-0.004108

The third ODS table, named ParmInfo, links the names Prm1, Prm2, etc., that appear in the Covb table with the actual variable names.

ParmInfo ODS Table for a Binomial Outcome

Obs	<u>_Imputation_</u>	Parameter	Effect
1	1	Prm1	Intercept
2	1	Prm2	Treatment
3	1	Prm3	Age

The ODS tables shown above are structured in such a way that PROC MIANALYZE will recognize the data sets as input. A problem arises, however, when the DIST=MULT option is used instead of DIST=BIN. Consider an example in which the outcome has four levels. The model will include three intercept terms, as seen in the excerpt from the ParameterEstimates table, below. The ParameterEstimates table has the same structure as in the binomial example, however, the Covb and ParmInfo tables do not.

ParameterEstimates ODS Table for a Multinomial Outcome

Obs	<u>_Imputation_</u>	Parameter	DF	Estimate	StdErr	Lower WaldCL	Upper WaldCL	ChiSq	Prob ChiSq
1	1	Intercept1	1	0.2586	0.3456	-0.4187	0.9359	0.56	0.4543
2	1	Intercept2	1	1.7521	0.3508	1.0646	2.4395	24.95	<.0001
3	1	Intercept3	1	3.2321	0.3653	2.5160	3.9481	78.26	<.0001
4	1	Treatment	1	-0.3875	0.1269	-0.6361	-0.1388	9.33	0.0023
5	1	Age	1	-0.0153	0.0115	-0.0379	0.0072	1.78	0.1827

In the Covb table below, the intercept terms are labeled as such, while only the parameters treatment and age are denoted by Prm1 and Prm2. Also, the ParmInfo table includes records for Prm1 and Prm2 (treatment and age), but not the intercepts. If these tables are used as input for PROC MIANALYZE, the procedure will not recognize the Covb and ParmInfo table structures and will fail to run. The macros described in this paper reformat the tables so that PROC MIANALYZE will accept them as input. This is not a difficult task, but it is tedious, and automation of the process saves valuable time.

Covb ODS Table for Multinomial Outcome

Obs	<u>_Imputation_</u>	RowName	Intercept1	Intercept2	Intercept3	Prm1	Prm2
1	1	Intercept1	0.1194121	0.1182776	0.1186284	-0.023516	-0.003216
2	1	Intercept2	0.1182776	0.1230276	0.1226553	-0.024198	-0.003244
3	1	Intercept3	0.1186284	0.1226553	0.133476	-0.024556	-0.003263
4	1	Prm1	-0.023516	-0.024198	-0.024556	0.0160937	-0.000021
5	1	Prm2	-0.003216	-0.003244	-0.003263	-0.000021	0.0001325

ParmInfo ODS Table for a Multinomial Outcome

Obs	<u>_Imputation_</u>	Parameter	Effect
1	1	Prm1	Treatment
2	1	Prm2	Age

DATA STRUCTURE

Several requirements for the structure of the data set are necessary in order for the macros described in this paper to run properly. First, all of the imputed data sets must be contained in the same SAS data file, and there must be a variable named _IMPUTATION_ which identifies the observations in each imputation. The _IMPUTATION_ variable must take on the values 1, 2, ..., m , where m is the total number of imputations. Second, the dependent variable must be categorical, and if there are more than two levels, it must be ordinal. Furthermore, the order of the categories must correspond to the order of the unformatted values of the variable. Finally, the independent variables must be either continuous, or binary indicator variables.

THE MACROS

The macro COMB contains all the code needed to create PROC GEMOD models for each imputation, reformat the ODS tables output from those analyses, and combine the results using PROC MIANALYZE. The macro FIX, which is embedded in the COMB macro, does the work of reformatting the output from PROC GENMOD so that the tables can be input into PROC MIANALYZE.

There are four required parameters for the COMB macro, as described below.

```
%macro comb(ds,a,b,i);
```

Parameters:

- ds Name of the analysis dataset, including SAS library if applicable.
- a The categorical dependent variable, which must be binary or ordinal.
- b A list of the independent variables, which must be either continuous numeric variables or binary variables.
- i An integer which will be appended to the names of SAS data sets containing the analysis results.

The macro code is described below, step by step.

STEP 1: SORT BY IMPUTATION

The analysis data set must be sorted by imputation so that a BY statement can be used in the PROC GENMOD code to generate separate analyses for each imputation.

```
proc sort data=&ds;  
  by _imputation_;  
run;
```

STEP 2: FIND NUMBER OF IMPUTATIONS

The MEANS procedure is used to obtain the number of imputations in the analysis data set. It is assumed that the maximum value of the _IMPUTATION_ variable is the total number of imputations. A DATA step with a CALL SYMPUT routine creates a macro variable to hold the number of imputations.

```
proc means data = &ds noprint;  
  var _imputation_;  
  output out=maximp max=numimp;  
run;  
  
data _null_;  
  set maximp;  
  call symput ("nimp",trim(left(numimp)));  
run;
```

STEP 3: FIND NUMBER OF OBSERVATIONS PER IMPUTATION AND NUMBER OF DEPENDENT VARIABLE CATEGORIES

In order for the macro to run, the number of levels in the categorical outcome variable must be known. The FREQ procedure calculates the number of observations in each category of the dependent variable and outputs those counts to a data set with one observation for each level of the variable. PROC MEANS obtains the total number of observations in the multiply imputed dataset with non-missing values of the dependent variable. Missing outcomes may occur if the variable does not apply to all subjects. The same PROC MEANS code also obtains the number of outcome categories. Finally, it creates another macro variable to hold the number of categories minus one, which will be the number of intercept terms in the statistical model.

A DATA step calculates the number of observations with a non-missing outcome per imputation and creates a macro variable to hold the value. If the number of observations in each imputation varies, the value will be an average over the imputations. The DATA step also creates a macro variable to hold the number of outcome categories. Finally, it creates another macro variable to hold the number of categories minus one, which will be the number of intercept terms in the statistical model.

```
proc freq data=&ds;  
  where &a ne .;  
  tables &a / out= count;  
  title1 "Frequency for &a";  
run;  
  
proc means data = count noprint;  
  var count;  
  output out=total sum=numobs n=numcat;  
run;  
  
data _null_;  
  set total;  
  call symput ("nobs",trim(left(numobs/&nimp)));  
  call symput ("ncat",trim(left(numcat)));  
  call symput ("nint",trim(left(numcat-1)));  
run;
```

STEP 4: PERFORM STATISTICAL ANALYSES FOR EACH ITERATION

PROC GENMOD tests the effect of the independent variables on the categorical dependent variable for each imputation. A multinomial distribution is specified for the dependent variable, which is appropriate for either binomial or ordinal multinomial outcomes. The RORDER=INTERNAL option is used so that the dependent variable will be ordered according its unformatted values. PROC GENMOD will model the probability of being in a category with a lower ordered value. Three ODS tables are output by PROC GENMOD, as described previously. The FIX macro will be used to reformat the tables so that PROC MIANALYZE will accept them as input.

```
proc sort data=&ds;
  by _Imputation_;
proc genmod data=&ds rorder=internal;
  model &a=&b / dist=mult covb;
  by _Imputation_;
  ods output ParameterEstimates=gmparms&i
             Covb=gmcovb&i
             ParmInfo=pinfo&i;
  title "Individual Imputation Analysis";
run;
```

STEP 5: OBTAIN THE NUMBER OF INDEPENDENT VARIABLES

In order to implement the FIX macro, the number of independent variables must be known. The ODS table ParmInfo contains one record for each independent variable in the model, excluding intercepts, for each imputation. The MEANS procedure is used to sum the values of the _IMPUTATION_ variable for ParmInfo records where _IMPUTATION_=1. This total will be the number of independent variables the user has included in the model. A DATA step with a CALL SYMPUT routine creates a macro variable to hold this value. Another macro variable is created to hold the complete data degrees of freedom that will be used by PROC MIANALYZE in combining the individual imputation analyses. The complete data degrees of freedom will equal the number of observations per imputation with a non-missing value for the outcome, minus the number of intercepts and other parameters in the model.

```
proc means data=pinfo&i noprint;
  var _imputation_;
  output out=sumvar sum=numvar;
  where _imputation_=1;
run;

data _null_;
  set sumvar;
  call symput ("nvar",trim(left(numvar)));
  call symput ("ndf",trim(left(&nobs-&nint-numvar)));
run;
```

STEP 6: BEGIN THE FIX MACRO

The FIX macro has two parameters: the number of categories in the dependent variable, and the number of independent variables. These values will be fed into the macro using the macro variables defined in steps 3 and 5.

```
%macro fix(numcat,numvar);
```

STEP 7: REFORMAT THE COVB DATASET

The FIX macro reformats both the Covb and ParmInfo ODS tables. The Covb dataset will be in the WORK library and will be named GMCOVB*i* where *i* is the integer value specified by the user in the COMB macro. Restructuring of the Covb data set is accomplished by two do loops.

Recall that, in the Covb data set, the values of the RowName variable start with either Intercept or Prm. For PROC MIANALYZE to recognize the table, those values all need to be Prm1, Prm2, etc. Likewise, the names of the columns of Covb start with either Intercept or Prm. These columns all need to be renamed Prm1, Prm2, etc.

The first do loop accomplishes both of those things. First, it renames the columns that have names starting with Prm. The column named Prm*j* (*j*=1,2,...) is renamed Prm*n*, where *n* is equal to *j* plus the number of intercepts. So, if the model has three intercepts, Prm1 is renamed Prm4, Prm2 is renamed Prm5, and so on. Then, the do loop changes the values of RowName in a similar fashion. The value Prm*j* will become Prm*n* where *n* is the number of intercepts plus *j*. The macro variable that holds the number of independent variables determines how many loops are executed.

```

data gmcovb&i;
set gmcovb&i;
%do j=1 %to &numvar;
  %let newnum=%eval(&j+&numcat-1);
  rename Prm&j=Prm&newnum;
  if RowName="Prm&j" then do;
    Rowname="Prm&newnum";
  end;
%end;

```

The second do loop renames the Intercept columns so that Intercept1 is renamed Prm1, Intercept2 is renamed Prm2, and so on. The values of the RowName variable are likewise changed so that Intercept j becomes Prm j .

```

%do j=1 %to &nint;
  rename Intercept&j=Prm&j;
  if RowName="Intercept&j" then do;
    Rowname="Prm&j";
  end;
%end;
run;

```

STEP 8: REFORMAT THE PARMINFO DATASET

Similar do loops are used to reformat the ParmInfo ODS table. The table will be in the WORK library and will be named Pinfo i , where i is the integer value specified by the user in the COMB macro. The first do loop changes the value of the Parameter variable so that Prm j becomes Prm n where n is the number of intercepts plus j . The macro variable that holds the number of independent variables determines how many loops are executed.

```

data Pinfo&i;
set Pinfo&i;
%do j=1 %to &numvar;
  %let newnum=%eval(&j+&numcat-1);
  if Parameter="Prm&j" then do;
    Parameter="Prm&newnum";
  end;
%end;
run;

```

Next, a DATA step creates a new SAS file called INT which will be appended to the ParmInfo dataset. Nested do loops create one record for each intercept in the model. For each intercept j ($j=1,2,\dots$) the Parameter variable will have the value Prm j , and the Effect variable will have the value Intercept j . Another DATA step adds the INT file to the ParmInfo data set.

```

data int;
length Parameter $30. Effect $30.;
%do k=1 %to &nimp;
  %do j=1 %to &nint;
    _Imputation_=&k;
    Parameter="Prm&j";
    Effect="Intercept&j";
    output;
  %end;
%end;
run;

data Pinfo&i;
set int Pinfo&i;
run;

%mend fix;

%fix(&ncat,&nvar);

```

STEP 9: MULTIPLE IMPUTATION INFERENCE

Once the ODS tables are restructured, they can be input into PROC MIANALYZE so that the results of the individual imputation analyses can be combined. PROC MIANALYZE requires that all independent variables in the model be listed in the MODELEFFECTS statement, including all intercepts. The LISTINT macro is used to list the appropriate number of intercepts, as determined by the number of outcome categories. The required parameter for the LISTINT macro is the number of intercept terms in the model. This value is fed into the macro by means of the

macro variable created in step 3. The EDF option specifies the complete data degrees of freedom. This value is plugged in using the macro variable created in step 5. PROC MIANALYZE will use the complete data degrees of freedom to calculate the adjusted degrees of freedom recommended by Barnard and Rubin (1999). The parameter and variance estimates from the multiple imputation analysis are output to the SAS data sets COMBINED i and VAR i , respectively, where i is the integer value specified by the user in the COMB macro.

```
%macro listint(k);
  %do j=1 %to &k;
    Intercept&j
  %end;
%mend listint;

proc mianalyze parms=gmparms&i covb=gmcovb&i parminfo=pinfo&i edf=&ndf;
  modeleffects %listint(&nint) &b;
  title "Combined Analysis";
  ods output ParameterEstimates=combined&i VarianceInfo=var&i;
run;
```

STEP 10: CALCULATE ODDS RATIOS

The final step of the COMB macro exponentiates the parameter estimates to obtain odds ratios, odds ratio confidence intervals, and minimum and maximum odds ratios from the individual imputations. These variables are added to the COMBINED i data set.

```
data combined&i;
  set combined&i;
  se=stderr;
  OR=put(exp(estimate),8.2);
  ORlowbound=put(exp(lclmean),8.2);
  ORupbound=put(exp(ucldata),8.2);
  minimpOR=put(exp(min),8.2);
  maximpOR=put(exp(max),8.2);
  label OR="Odds Ratio"
         ORlowbound="Lower 95% CL for OR"
         ORupbound="Upper 95% CL for OR"
         minimpOR="Min OR from a single imp"
         maximpOR="Max OR from a single imp";
run;

%mend comb;
```

GEE MODELS

The macros described in this paper are designed for handling generalized linear models without random effects. PROC GENMOD, however, can also be used to fit generalized estimating equations (Liang and Zeger, 1986). This is done by including a REPEATED statement which models the covariance between observations in the data set. With slight modification, this macro can be used for GEE analysis of multiply imputed data. First, a REPEATED statement must be added to the PROC GENMOD code. Note that the independent working correlation structure is the only type currently available for use with multinomial outcomes. The REPEATED statement requires a SUBJECT variable which must also appear in the CLASS statement. Finally, the names of the ODS datasets output by PROC GENMOD must be changed to those that correspond to the GEE parameters and covariance estimates. The example below outputs the empirical covariance estimates; model based covariance estimates can also be specified.

```
proc genmod data=&ds rorder=internal;
  class id;
  model &a=&b / dist=mult covb;
  repeated subject=id / ecovb;
  by _Imputation_;
  ods output GEEEmpPEst=gmparms&i
             GEERCov=gmcovb&i
             ParmInfo=pinfo&i ;
run;
```

CONCLUSION

Programming tasks that are simple in nature, such as restructuring data sets, can be tedious to execute. This is especially true if the task must be repeated for multiple analyses, and if the code must be altered to accommodate variations in those analyses, such as outcome variables with differing numbers of categories. The code described in

this paper automates the reformatting necessary to read PROC GENMOD output into PROC MIANALYZE when the DIST=MULT option is specified. With the COMB macro, modeling of categorical outcomes from multiply imputed data can be done quickly and efficiently.

REFERENCES

Barnard, J. and Rubin, D.B. 1999. "Small-Sample Degrees of Freedom with Multiple Imputation." *Biometrika*, 86:948-955.

Liang, K.Y. and Zeger, S.L. 1986. "Longitudinal Data Analysis Using Generalized Linear Models." *Biometrika*, 73:13-22.

CONTACT INFORMATION

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

Marie Gantz, Ph.D.
RTI International
PO Box 12194
Research Triangle Park, NC 27709-2194
Work Phone: 919-485-7780
Fax: 919-485-7762
Email: mgantz@rti.org
Web: www.rti.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.