

A Method of Using Multiple Imputation in Clinical Data Analysis

Christina M. Gullion, Center for Health Research, Portland, OR
Chuhe Chen, Center for Health Research, Portland, OR
Gayle T. Meltesen, Center for Health Research, Portland, OR

ABSTRACT

Use of multiple imputation to replace missing outcomes in clinical research is a relatively new approach. Thus, there is little information published on how to obtain and combine the types of estimates needed to use multiple imputation in analysis of results from many clinical trials. This presentation summarizes procedures developed, or adapted from other sources, using SAS®, through the stages of preparation of the data for imputation, processing after imputation, setting up output of SAS PROCs (e.g., GLM, REG, MIXED) for processing by PROC MIANALYZE, and other routines we developed for output not processed by PROC MIANALYZE. This paper does not report on the efficiency or unbiasedness of the results. Rather, we focus on providing guidance on the use of SAS® by clinical investigators in order to yield meaningful results from analysis of a multiply-imputed dataset.

INTRODUCTION

Missing data are almost inevitable in multivariate clinical trials—especially in longitudinal studies. Subjects may decline to respond to certain questions at each time point, or fail to participate entirely at early stages of the study. A simple way to treat the missing data is to delete the incomplete data. This may produce biased inferences and reduced statistical power especially in multivariate modeling. A single imputation of missing data, either using a sample mean, an adjusted mean from a regression model, or a “Hot Deck” procedure, does offer a complete dataset and sometimes yields unbiased inferences. However, single imputation cannot provide valid standard error calculations and confidence intervals that account for the uncertainty caused by the imputed missing data. Using certain assumptions, however, multiple imputation can be used to provide valid inferences from a complex and incomplete clinical study dataset that account for the uncertainty involved in imputing missing values. This paper summarizes the procedures for accomplishing this task, using SAS® 9.13, that we have developed and adapted from other sources.

SUMMARY OF PROCEDURES FOR USE OF MULTIPLE IMPUTATION

1. Select list of variables to go in imputation model.
2. Use PROC MI to generate missing data pattern.
3. Generate descriptive statistics (PROC UNIVARIATE NORMAL PLOTS on all variables of interest), and obtain 4 or 5 highest correlations.
4. Identify potential co-linearities and revise list of variables to minimize identified co-linearities.
5. Define and apply transformations to normalize variables for the imputation.
6. Recode categorical variables to k-1 indicators, where k is the number of categories.
7. Obtain same set of correlations, missing data patterns, and univariates on the transformed variables, and verify no new issues have appeared. If adjustments need to be made return to step 4.
8. Create multiply-imputed datasets using PROC MI. If problems occur, it may be necessary to return to step 4.
9. Check distributions of the imputed variables. Depending on results, continue to step 10 or return to step 4.

10. Program and run back transformations, and create analysis dataset[s].
11. Carry out identical analyses on all copies of the imputed dataset.
12. Combine output for the analyses.

PREPARATION OF DATA FOR IMPUTATION

In the first step we prepare data for imputation by assembling a list of candidate variables. This step should include all variables that may be used in analyses of the study dataset, such as primary and secondary outcomes, covariates, design variables, and variables that code reason data is missing or that may predict dropout (Collins et al 2001, Schafer & Graham 2002). It is a good idea to include any available intermediate measures of the final outcome, in order to improve the accuracy and efficiency of the imputation. In a large clinical trial with a Data Safety and Monitoring Board and staff from funding agencies involved, this list may be requested and scrutinized by both groups long before the analysis will be done.

Once the list is approved, we examine univariate frequency and density distributions and missing data counts. Variables that are highly correlated may result in a singular covariance matrix, while not providing significant information, so one variable should be omitted. Binary variables with very unbalanced frequencies (<2% in a category) and no missing values should be identified and excluded from the imputation data, though they should be retained in the analysis dataset. Presence of these variables appears to increase the likelihood of failure to converge or of a singular covariance matrix.

Highly skewed quantitative variables should be transformed using a mathematical function, if possible, or recoded to ordinal scale to achieve approximate normality. Generally, a skew statistic greater than $\text{abs}(1.0)$ is used as the criterion for attempting a normalizing transformation. Ordinal scales (e.g., income categories) may have categories combined to increase the density and approximate symmetry. We use the following naming conventions to keep track of variables that have been changed from the original during the imputation process:

- `_tr` as a suffix attached to a variable to indicate it has been transformed.
- `_btr` as a suffix attached to a variable with imputed values to indicate it has been back-transformed
- `_i` as a suffix attached to a variable with imputed values to indicate it has not been transformed.

Example 1. Variable transformation code

```
data transform;
  set all;
  **** mathematical transformations ****;
  var1_tr=var1**(1/4);
  var2_tr=var2**(1/3);
  var31_tr=0.1*(var3/10)**3;
  var4_T_tr=0.01*(var4/10)**5;

  **** transform quantitative measure to ordinal ****;
  var5_tr=.;
  if var5 = 9 then var5_tr=1;
  if 9 < var5 < 12 then var5_tr=2;
  if 11 < var5 < 16 then var5_tr=3;
  if 15 < var5 < 21 then var5_tr=4;
  if 20 < var5 < 27 then var5_tr=5;
  if 26 < var5 < 44 then var5_tr=6;

run;
```

Next, we examine intercorrelations of complete variables to determine whether any are redundant (i.e., correlation > .80) and hence likely to cause numerical problems (co-linearity) during estimation of posterior parameters. Co-linearity requires an invertible covariance matrix. We compute the rank of the final covariate matrix to assure that the matrix is full-rank (using PROC FACTOR). If the matrix is not full-rank, we make further adjustments to the variable to eliminate the redundancy.

We use PROC MI to check the missing data pattern of the final dataset, in part to see if there is a monotone missing pattern. We print the **misspattern** table into an .rtf file, then copy and paste this into a Microsoft Excel spreadsheet to analyze the missing pattern, including evaluating whether coding errors might have created some missing values.

Example 2. Code used to obtain missing data pattern

```
proc mi data=pattern_check
  nimpute=0 /*0 to get the missing pattern only*/;
  ods output misspattern=mispattern;
  var var1-var50;
run;
```

MULTIPLE IMPUTATION PROCEDURE

We carry out multiple imputations using SAS PROC MI, which implements algorithms given by Schafer, 1997. In multiple imputation, the parameters (means and covariances) of the joint distribution of observed and missing variables is estimated in an iterative fashion, starting with the observed data and plausible values for the missing values. After we obtain an estimate of the parameters for all the variables, we update imputed values of missing data points, and re-estimate parameters (Schafer, 1997, pp 68-87). Once we achieve stable parameter estimates, each missing value is sampled from the final ("posterior") joint distribution and then further perturbed with random error. Imputed values are drawn several times (we start with NIMPUTE=5), thus creating several "complete" versions of the dataset.

PROC MI carries out 2 iterative processes, and either can encounter a problem. These problems include failure to converge, a singular variance-covariance matrix in the first iteration process, or failure to generate a value that meets constraints for minima and maxima. We have learned to avoid putting constraints on the imputation process, and instead use post-processing (see below) to handle out-of-range values and rounding, since these constraints may result in failure to complete the imputation. Also, in response to these problems, further adjustments to the variables may be necessary, such as dropping low-frequency binary variables, recoding to eliminate low-frequency or empty cross-variable cells, or improving the transformation to normality. Starting values can be specified, but we have not tried this.

Example 3. Multiple imputation code

```
*** fixed seed to create replicable results, ***
***printed output to monitor process***;
proc mi data=for_imputation nimpute=5 seed=56986 out=imputed;
  mcmc initial=em(itprint) nbiter=1000 niter=500 timeplot(wlf)
    acfplot(wlf);
  var var1-var50 site_1 site_2 site_4;
run;
```

We examine the efficiency indicators in the output from PROC MI to determine whether enough imputations have been created. A key indicator of the efficiency of the imputation is the "df" for posterior parameter estimates.

Example 4. Excerpts from PROC MI output showing the DF and Relative Efficiency

The MI Procedure				
Multiple Imputation Parameter Estimates			Multiple Imputation Variance Information	
Variable	...	DF	...	Relative Efficiency
waisscore_tr	...	108.98	...	0.980373
LCPT_tr	...	170.62	...	0.991589
risk	...	86.86	...	0.975137
benefit	...	45.30	...	0.959009
risk_ben_r_tr	...	82.92	...	0.974056
rbversion_tr	...	156.87	...	0.989206
Jellybeans	...	198.09	...	0.997331
Q101_tr	...	109.21	...	0.980423
race_i1	...	204.19	...	0.999536

race_i2	...	204.55	...	0.999726
race_i3	...	204.64	...	0.999777

The DF is a direct function of the efficiency. In our experience, variables with smaller DF appear to be those that have other problematic features, e.g., unsatisfactory internal consistency, many missing values, low incidence, or are extremely skewed. Allison (2001) suggests as a rule-of-thumb that if the DF for a variable is less than 100, there is excessive variability among the imputations for that variable, and more imputations are needed to improve the efficiency of the imputation. For instance, after examining the output in example 4, we increased the number of imputations to 8.

Another consideration is the plausibility of imputed values. For example, large negative values for a variable that cannot be negative (e.g., weight) may indicate that the dataset does not sufficiently constrain the iterative estimation procedure. For instance, in a dataset with fewer than 5% missing primary outcomes we had included several binary indicators of reasons for missed final outcome, each of which had 5 positive instances out of over 1000 cases, but some imputed weights were negative and large. When these variables were dropped, the imputed weights were in an acceptable, positive range.

PROCESSING AFTER IMPUTATION

Once we are satisfied that we have enough imputations, we carry out back transformations of variables that have imputed values and had been transformed to normality solely to meet the needs of the imputation process. Variables transformed using a mathematical function (square root, log, exponential) are back-transformed with the inverse of that function.

Categorical variables are coded back to categories using a series of steps that take account of the most likely category over the several indicator variables—this is a bit complex. The code shown in Example 5 for back-transforming the race indicators takes account of the fact that three values have been imputed for persons whose race is missing. The highest value is identified and the person with missing race is assigned to that category.

Continuous variables transformed to binary or ordinal categories must stay that way after imputations; these typically were extremely skewed before recoding. After back transformations, we winsorize out-of-range values and round to the number of decimal points in the original measure. Labels show the status of the new variables (back transformed or just imputed).

Example 5. Code used to back transform variables after imputation.

```

title3 'back transformation';
data sasds.study2_master_mi(label='Study 2, 8 imputations') ;
set out_mi2;
* BACK TRANSFORMATIONS OF IMPUTED VARIABLES;
array trans ... waisscore_tr lcpt_tr risk benefit rbversion_tr
... Q101_tr race_i1 race_i2 race_i3 ...;
do i = 1 to dim(trans);
  if trans[i] < 0 then trans[i] = 0;
end;

Jellybeans_i = (round(((2*round(Jellybeans,0.5)+0.9)/2),1))-0.5;
  if jellybeans_i < -2.5 then jellybeans_i = -2.5;
  if jellybeans_i > 2.5 then jellybeans_i = 2.5;
q101_btr = round(((exp (q101_tr)-1)/11),.01);

do i = 17 to 19; *recode race category indicators;
  if trans[i] > 1 then trans[i] = 1;
end;
  other = 1 - race_i1 - race_i2 - race_i3;
  if other > race_i1 and other > race_i2 and other > race_i3
  then do;race_i1_btr=0;race_i2_btr=0;race_i3_btr=0;
  end;
  else if race_i1 > race_i2 and race_i1 > race_i3
  then do; race_i1_btr=1;race_i2_btr=0;race_i3_btr=0;
  end;
  else if race_i2 > race_i3
  then do; race_i2_btr=1;race_i3_btr=0;race_i1_btr=0;
  end;
end;

```

```

else do; race_i1_btr=0;race_i2_btr=0;race_i3_btr=1;
end;
drop other;
Race_cat_i = sum(1*(race_i1_btr=1), 2*(race_i2_btr=1),
3*(race_i3_btr=1));
if race_cat_i = 0 then race_cat_i = 4;
...
label
Jellybeans_i = 'Jellybeans Q051 (i)'
Q101_btr = 'Q101: mammography ratio (btr)'
Race_cat_i = 'Race (S03c) (i):1-4'
race_i1_btr = 'Race (S03c) 1=white(btr)'
race_i2_btr = 'Race (S03c) 1=asian (btr)'
race_i3_btr = 'Race (S03c) 1=Haw/PI (btr)'
...;
run;

```

SETTING UP OUTPUT OF SAS PROCs FOR PROC MIANALYZE

Once a dataset is set up as needed, we run analyses. Every analysis is repeated in exactly the same way in each of the imputed datasets (usually by using `BY _IMPUTATION_` as a statement in any SAS PROC). In most cases where model selection is involved, we arbitrarily chose one of the imputations to run the selection process, then run the resulting model in all copies.

Combine the results often requires extra steps, typically using ODS or other output statements to create an output dataset that matches PROC MIANALYZE input specifications. In some cases, output that doesn't match any of the input specifications is combined in a DATA step, with the equations programmed by us. This is shown in example 6.

Appendix 1 contains a summary of how output from various SAS PROCs is handled. We have found that the newer PROCs like MIXED and GLIMMIX for modeling given their compatibility with PROC MIANALYZE. When doing model selection with PROC GLMSELECT or PROC REG we use one of the 5 imputations and re-run the final model with all 5 imputations.

Finally, in order to produce results that can be used in papers or presentations by the study investigators, we combine data over imputations. Point estimates can be simply averaged over imputations, whereas statistics are combined according to Rubin's (1987) rules, which involve adjusting the standard error and degrees of freedom for between imputation variance.

We use either SAS PROC MIANALYZE or code developed in-house for situations not covered by PROC MIANALYZE. Examples of the kinds of output requested by investigators included frequency counts, percentages with chi-square test, correlations, F tests, contrasts between levels of a CLASS variable, predicted values for tables or figures.

Example 6 provides an example of the steps needed from coding the model effects through printing the resulting combined F-test results and least-squares means. PROC GLMMOD is used to set up effect parameterization for the CLASS variables. This could be done in a DATA step, but is more tedious. Use the NOINT option because GLM will add an intercept term, and the presence of two intercepts produces an error. The output from GLMMOD is used in PROC GLM, where no CLASS statement is necessary. The ODS tables are then input into PROC MIANALYZE, which produces combined F-tests for main effects and contrasts between age groups. Note that reference effects (parameter estimate=0) are omitted from the PROC MIANALYZE MODELEFFECTS statement. The MULT option on the TEST statement produces the F test. For single effects, this is just the square of the t-test on the combined parameter that is automatically produced by PROC MIANALYZE, but these are included for convenience in displaying the results. The resulting table can be dropped from the .rtf file directly into a paper.

Example 6. Combined F tests from PROC GLM analysis

```

ods listing close;
proc glmmmod noprint data = medsplit outparm=Parm outdesign=Design;
by _imputation_;
class CAT agegrp ;
model Jellybeans_i = CAT agegrp / noint;
run;
ods listing;title5 'Parameter list for ANCOVA model';
proc print data=parm;where _imputation_=1;run;
ods listing close;
proc glm data = Design;

```

```

ods output ParameterEstimates=glmparms1 InvXPX=glmjspxi1 ;
by _imputation_;
model Jellybeans_i = col1-col5 /inverse solution ;
run;quit;
proc mianalyze parms=glmparms1 xpxi=glmjspxi1 edf=604;
ods output TESTMULTSTAT=Jellybeans_i_TEST;
modeleffects Intercept Col1 Col3 Col4 ;
Numeracy: test col1/mult;
Agegrp: test col3,col4/mult;
Yvs00: test col3 /mult;
Y0vs00: test col4 /mult;
YvsY0: test col3 - col4/mult;
run;
ods rtf file="\\&dir\output\JBftests.rtf";ods listing;
title4 " F-tests on combined results";
proc print data=jellybeans_i_test label;
id test;
var numdf dendif fvalue probf;
format probf pvalue8.4;
label fvalue = 'F' test='Hypothesis test' probf = 'p <';
run;

```

Example 7 shows the sequence of steps needed for obtaining the least squares means from imputed data. The PROC MIXED output can be input directly into PROC MIANALYZE, but the combined results will not contain data for cells (defined by the combination of CLASS variables) that contain no imputed values. These are identified in the PROC MEANS output as having a standard deviation of the estimate equal to 0. Results for the two types of data are combined into a single table and printed, for use in papers or presentations.

Example 7. Obtaining least squares means from imputed data

```

proc mixed data = medsplit;
ods output lsmeans=ls;
by _imputation_;
class CAT agegrp ;
model Jellybeans_i = CAT agegrp ;
lsmeans agegrp ;
lsmeans CAT ;
run;quit;
proc sort data=LSM; by cat agegrp ; run;
proc mianalyze data=LSM edf=602;
ods output parameterestimates=ls_means;
by cat agegrp ;
modeleffects estimate;
stderr stderr;
run;
** fill in missing values for non-imputed cells;
proc means data=LSM noprint;
by cat agegrp ;
var estimate stderr DF ;
output out=ls_std mean = std=sd_est;
run;
data ls_std;set ls_std;
where sd_est = 0;
label sd_est = 'Imputed? (1/0)';
run;
data ls_means;
merge ls_means ls_std(drop=_TYPE_ _FREQ_);
by cat agegrp ;
drop parm min max theta0;
if sd_est=. then sd_est=1;
run;
ods listing;
ods rtf file="\\&dir\output\Study2_29_JB_lsmeans.rtf";

```

```
title5 "Least squares means: Combined results";
proc print data=ls_means label;
  id cat;by cat;
  var agegrp estimate stderr sd_est;
  format estimate stderr 8.2;
run;
ods rtf close;
ods listing close;
```

OTHER ROUTINES

We developed several macros in order to combine quantities in cases that PROC MIANALYZE does not handle. We tested these in a variety of analyses, including analysis of variance, regression analysis, correlation analysis, and generating data for tables and figures.

CONCLUSION

Multiple imputation can be very useful for estimating missing data in multivariate clinical trial data. One of the advantages of the procedure laid out in this paper is that the process, when done correctly, generates valid inferences from a complex and incomplete clinical study dataset that are able to account for the uncertainty involved in imputing missing values. However, users of these methods must be aware of the extra data preparation needed to successfully use multiple imputation, and of the extra processing needed to analyze the imputed multiple data sets. In this paper we have provided a detailed version of the steps required to perform such an analysis.

REFERENCES

- Allison, P. D. 2001. Missing Data. *Sage University Papers Series on Quantitative Applications in the Social Sciences*. Thousand Oaks, CA, 07-136. Sage.
- Collins, L. M., Schafer, J. L., and C. M. 2001. "A comparison of inclusive and restrictive strategies in modern missing data procedures." *Psychological Methods* 6:330-351.
- Rubin DB. 1987. *Multiple Imputation For Nonresponse In Surveys*. New York: John Wiley & Sons, Inc.
- Schafer J. L., and J. W. Graham. 2002. Missing data: Our view of the state of the art. *Psychological Methods* 7:147-177.
- Schafer J. L. 1997. *Analysis of Incomplete Multivariate Data*. Boca Raton: Chapman & Hall/CRC.

CONTACT INFORMATION

Dr. Christina M. Gullion
The Center for Health Research
3800 N. Interstate Ave
Portland OR 97227
Work Phone: 503-335-6506
Fax: 503-335-2428
E-mail: christina.gullion@kpchr.org