Survival Analysis with PHREG: Using MI and MIANALYZE to Accommodate Missing Data

Christopher F. Ake, SD VA Healthcare System, San Diego, CA
Arthur L. Carpenter, Data Explorations, Carlsbad, CA

ABSTRACT

Survival analyses based on a data collection process which the researcher has little control over are often plagued by problems of missing data. Deleting cases with any missing data will result in information loss and usually results in bias, while many analytic procedures that retain this information in some form underestimate the resulting uncertainty in parameter estimates and other output. SAS® Version 8 includes two new procedures that allow the researcher to generate "complete" data sets from incomplete data by multiple imputation and to analyze the resulting data in ways which adequately account for the uncertainty involved. This paper presents suggestions for optimal use of PROC MI to perform such multiple imputation and PROC MIANALYZE to conduct various statistical analyses of modeling output, in this case from PROC PHREG, including design of control macros, structure of multiply imputed datasets, generation of binary from non-binary categorical variables, and options for presentation of results.

Keywords: PROC MI, PROC MIANALYZE, missing data, multiple imputation

INTRODUCTION

MI and MIANALYZE are new procedures in SAS Version 8 whose usage is in the process of being established, especially when using large data sets, multiple models, or different forms of output and presentation of results. As suggested by the flowchart below, their use involves multiple stages of analysis: first creating several "complete" datasets by multiple imputation using MI, then performing individual analysis PROC runs (in our case, PHREG) on each of the individual "complete" (imputed) data sets, followed by combining the output from the individual analysis runs via MIANALYZE to produce final parameter estimates and other model results. While in general the number of imputed data sets required for this process is small (we created m=5 "complete" data sets, which is quite often sufficient for efficient model estimates) the added complexity of the multistage analysis puts a premium on prior planning and careful design to maximize program performance as well as accessibility and manipulability of results.

Modeling

We used proportional hazards models to examine whether patients in the Veterans Administration who had received VA care for HIV between January 1, 1993, and June 30, 2001 showed a higher risk for certain adverse side effects if they received some form of highly-active antiretroviral therapy (HAART) compared to those under VA care for HIV in the same period who did not receive such treatment. We examined various adverse events, such as all-cause mortality, mortality due to cardiovascular or cerebrovascular disease, and inpatient admissions for cardiovascular or cerebrovascular disease. For each such outcome event, we examined the effect of exposure to nucleoside analogs, non-nucleoside reverse transcriptase inhibitors, and protease inhibitors singly and in combination, both in the form of being ever exposed and of amount of cumulative exposure. In all models we adjusted for various patient characteristics; in some models we also included covariates to partially adjust for possible selection effects, whereby sicker patients might be more likely to receive the treatment in question. We therefore had a large number of models to analyze, together with a strong possibility that further models would suggest themselves as the analyses progressed.

Our survival analysis programming involved adaptation and development of a set of SAS macros developed at the Mayo Clinic in Rochester, Minnesota, under the leadership of Terry Therneau and Patricia Grambsch, and maintained by clinic staff. They have been thoroughly tested and used there, but the Mayo Clinic does not warrant their use in any way. They are discussed in Therneau and Grambsch's book *Modeling Survival Data: Extending the Cox Model,* and are available at www.mayo.edu/hsr/people/therneau/book/book.html. They include programs to calculate Kaplan-Meier survival curves together with standard errors, confidence limits, and median survival times, with numerous options to print and plot results; calculate logrank statistics; generate robust variance estimates; and examine functional form. Some of their macros can accommodate data that is in counting process format, which allows left-truncated survival or time-dependent calculations where a person can change state after their starting time. Their *schoen* macro, which uses scaled Schoenfeld residuals to produce plots and tests of proportional hazards assumptions, does not, as provided, permit data in counting process format to be used, and thus one of our adaptations was to expand this macro to accommodate our use of data in counting process format.

Among the covariates in all our models were four with missing values: age at first VA care for HIV (4 cases out of 36,766 or 0.01%), baseline severity of illness (181 cases or 0.49%), race (2684 cases or 7.30%), and risk group at first VA care for HIV (4632 cases or 12.60%). These totaled to slightly less than 1% of all entries in the various 36,766 x 22 to 36,766 x 27 data matrices used in our modeling.

Our decision to employ multiple imputation required a multi-stage analysis process, of which one portion needed to be iterated for each of the m=5 completed datasets used for each of our models. In our flowchart, the entire portion in the solid box is run by the analysis control program once per model. The sub-portion (within the smaller solid box) is run once per iteration, i.e., once for each of the m complete data sets for each distinct model.

MODEL SPECIFICATION USING A CONTROL FILE

Due to the large number of outcome events and types of exposure of interest, as well as the various combinations of covariates, the number of models that had to be processed would have been unmanageable without some type of control system. We chose to build a SAS data table that contained one observation (row) for each model of interest. The columns in this control file contained sufficient information to completely specify, either directly or indirectly all aspects of the individual model.

During processing, the control file was read and macro variable arrays (lists of macro variables in the form of &&VAR&I) were generated following the techniques described in Carpenter and Smith (2002a). Once the specification of a given model was contained within these arrays, the analysis programs described below were executed for each model.

The models themselves were grouped according to various commonalities, such as covariates, transformations, event types, etc. Each model was assigned a unique control number, which reflected this association. The control number was also used to manage the thousands of graphs, charts, and tables that were generated as part of the analysis process. Carpenter and Smith (2002b) describe the output management process.

DATA CREATION

Variable creation: binary variables

The imputation process requires specification of a model with which you can derive imputed values from the observed values of a number of other variables in the original data set. For a given variable, the imputation model should ideally include all those variables that are potentially related to that variable, or potentially related to the missingness of that variable. But considerations of size or running time may put a premium on as parsimonious an imputation model as possible. Thus if two variables A and B are not only each related to variable C, but highly correlated with each other, then only having one of A or B in the imputation model for imputing values for C may suffice.

Unless the observed (non-missing) component of the data used in the imputation model has, under some ordering of the variables, a monotone structure, (whereby if the i^{th} variable for a given observation has a missing value then so does any j^{th} variable for that observation if $j > i$), then to use MI one must assume multivariate normality for data. In reality MI can accommodate some departure from this assumption as long as it is only a limited one. (see Schafer, pp. 147-148, 211-218 (sect. 6.4, which contains a simulation study examining the performance of various estimators using a normal model for imputation of what was more plausibly non-normal data))

In Version 8.2 MI features a TRANSFORM statement, which can be used to convert certain non-normal variables to normality, or approximate normality, via Box-Cox, exponential, logarithmic, logit, or power transformations prior to imputation and then reverse-transform them afterward. But it cannot transform categorical variables to normality. Thus in either Version 8.1 or 8.2, to be able to impute missing values to any categorical non-binary variable in the dataset so as to conform to the normality assumption, such a variable must be replaced by a corresponding set of binary dummy variables with 0-1 valuesThese dummy variables will then each be treated as individual normal variates for imputation purposes.

In treating each of these separate dummy binary variables as a normal variate, MI will impute a continuous value. So for a set of dummies corresponding to a single categorical covariate in the original dataset, you must program a routine to assign a 1 to one of the dummies and 0s to the other dummies based on their imputed values in that iteration (e.g., see Allison, pp 39-40). In the last stage of each round of the imputation process, the set of dummy variables can be replaced by the original categorical variable, which will have a 1 assigned to the category whose dummy received the 1 in that round.

Checking The Imputation Process

In imputing values, MI utilizes a Markov Chain Monte Carlo process designed to result in generating independent draws from the Bayesian posterior distribution of the missing values given the observed data and an assumed prior. For these draws to be representative of the posterior distribution the process must be run until it becomes stationary, but, as mentioned in MI documentation, verification of convergence of this iterative MCMC process to a stationary distribution is non-trivial, and is not implemented in the MI procedure itself. You can, however, create, for each variable to be imputed, a plot of the value imputed to the variable in that iteration versus the iteration index to check for time trends in the iterated values. As long as time trends are apparent, stationarity is unlikely to have been reached. Even with no such trends apparent, however, stationarity cannot be guaranteed, especially for data with high percentages of missing values. Thus it is valuable to perform other checks. You can, for example, check whether the correlations among the iterates in a single imputation run for a given variable have disappeared with a large enough lag, by plotting correlations for the first, say, 15 or 20 lags. (For suggestions on assessing convergence see Shafer, sect 4.4, pp. 118-136.)

The following SAS code excerpt indicates how GPLOT and AUTOREG can be used in this way, when using Version 8.1, as we did. (Cf. Example 6 in the MI documentation in Version 8.1.) In Version 8.2, PROC MI itself can produce such plots using the TIMEPLOT option in the MCMC statement in place of using PROC GPLOT, and using the ACFPLOT option in the MCMC statement instead of having to call PROC AUTOREG (although ACFPLOT does not furnish Durbin-Watson statistics, which we produced with the DWPROB option in the MODEL statement in PROC AUTOREG).

```
* create a separate plot & analysis
* for each of the missing vars;
%do m = 1 %to &allmisscnt;

  * define the symbols used in the plots;
  symbol1 v=triangle c=black i=none;
  symbol2 v=star c=blue i=none;
  symbol3 v=square c=red i=none;
  symbol4 v=circle c=brown i=none;
  symbol5 v=plus c=green i=none ;
```

```
  * plot of iteration number vs each
  * var with missing values;
  title1 h=1.5 "Imputation: MEAN of
&&allmiss&m vs Iteration.  RunID: &runid";
  proc gplot
data=work.outmean1(where=(_type_='MEAN'));
   plot
&&allmiss&m*_Iteration_ = _Imputation_ /
         href = 0
         name = "&runid.mn&m";
    label &&allmiss&m = ' ';
    run ;
  title1 "AutoReg of Imputed Data.
Dependent Variable: MEAN &&allmiss&m...
RunID: &runid";
  proc autoreg data=work.outmean2
      (where=(_type_='MEAN')) ;
    by _imputation_;
    model &&allmiss&m = /nlag=18 dwprob;
    run ;
. . .
%end;
```

Data set creation

There are likely to be components of your various data creation and analysis programs (macros) that will contain differences in the way they handle those covariates with some imputed values versus those without any imputed values. As a result you may want to create two separate macro variables, one containing a list of the imputed covariates, the other containing a list of the nonimputed ones, which could then be used, for example, to create a macro variable whose value is the list of names of the imputed covariates.

Using the OUT= option in the PROC MI statement creates a SAS data set which contains all the variables in the input data set with missing values replaced by imputed values. Each observation in this output data set also contains an additional variable, _Imputation_, whose value identifies which of the m imputations produced the given observation.

Depending on the size of such a data set and the proportion of variables with missing values, you may want to consider various options for storage of your data. For example, all those covariates without missing values might be kept in one data set with an ID variable which would allow you to merge each observation in this data set with the corresponding m observations in a separate data set containing only those covariates that had to have some values imputed. You will thereby save the space that m-1 copies of the data set with the variables without missing values would occupy. Each required merge could then be run immediately before the combined data is input to PHREG.

The data that is actually input to PHREG can itself be structured in various ways. One choice would be to construct a separate dataset for each of the m runs of PHREG for a given model. Alternatively, one could create one data set containing all m values for each imputed variable, and then simply pull out the i^{th} value of each imputed covariate for the i^{th} of the m runs for that model. To run our analyses, we chose the most compact form: to create a single data set for each model with all m versions of the imputed values for any imputed covariate in the same observation, indexed so that we could pull out the i^{th} value for each imputed variable for the i^{th} run for that model of PHREG. So for our risk variable, for instance, a given observation contains values for RISK_I1, RISK_I2, RISK_I3, RISK_I4, and RISK_I5. From this observation the first run for a given model uses the values of the non-imputed covariates together with the RISK_I1 value for risk, the RACE_I1 value for race, the SEVERITY_I1 value for severity, etc.

DATA EXPLORATION

Using multiple imputation raises a question as to how any exploratory data analysis (EDA) conducted prior to or separate from statistical modeling, e.g., examining frequencies, crosstabs, or correlations, is to be done. The multiple imputation process is not intended to provide estimates per se of missing values but rather to produce a set (a random sample) of values whose variability captures the uncertainty that exists with respect to missing data values. No single one of the m imputed data sets is the "real" data, and while MIANALYZE combines analysis output parameters, for example, it is not meant to combine data values themselves. For any variable with continuous values, one could combine the m values for a given observation by taking the mean, but such a "combining" technique would not work for those variables with ordinal or categorical values.

One option for EDA is to select one of the m imputed data sets to use; differences in results from use of another of the m data sets are likely to be small if the percentages of missing data in the original data set were small. This is the choice we made to produce Kaplan-Meier curves, for instance. But the larger the percentages of missing data are, the more advisable it may be to conduct EDA on each of the imputed data sets and examine the variations from each of the m outputs to the others.

ANALYSIS

With m runs of PHREG for each model, running time can become much more of a consideration than otherwise, especially if a number of different models are being examined. We recommend careful consideration prior to running PHREG or any other analysis procedure(s) of exactly what output might be wanted. Wald tests for individual coefficients or subsets of coefficients, for example, are straightforward to program into an analysis run using the TEST statement in PHREG. But if you decide after PHREG has been run that you need to conduct such tests, substantial programming may be required, since you will need to specify the proper variance-covariance matrix and then pull out certain values from it to calculate the Wald test statistic, which can require extensive manipulations with indices and variable names to succeed.

OUTPUT MANIPULATION AND USE

We also suggest you consider carefully before running your analyses whether you should save direct PHREG or other SAS analysis PROC output, e.g, model diagnostics (in our case, such things as scaled Schoenfeld residuals) in addition to PHREG output datasets and MIANALYZE output. If, for example, you anticipate that questions of model adequacy might arise subsequently, then direct output from the m individual runs for each model may need to be saved. You could possibly save output from only one of the m PHREG runs if, say, only

general questions about model adequacy were expected. Again, storage structure should be determined beforehand with an eye on questions of network access for potential users, storage capacity, ease of transfer or file reclassification, and accommodation to creation of additional output files and directories--which includes naming and tracking of models. We suggest you develop from the start a model output indexing system that is capable of expansion, particularly if it is likely that modifications to models may be made after or even during the process of running initial models.

The parameter estimates resulting from the individual runs of PHREG on each of the m individual imputed data sets will be combined by MIANALYZE, which computes a mean and appropriate variance estimate as a function of the within and between variance estimates for the parameter in question. MIANALYZE creates a data Table named PARMEST, for example, to hold these resulting parameter estimates. For example, output is routed to SAS data sets with the statement

```
ods output parmest=libname.dataname;
```

If multiple models (e.g., with different outcome variables or covariates) are being run from which you want to package together all the resulting parameter estimates that MIANALYZE produces for each such model, one option would be to run a loop indexed by model version number with a call to MIANALYZE inside the loop and then saving the estimates all to the same overall data set by creating a new data set after the loop. The following is an example for hazard estimates rather than the parameter estimates themselves, in which the hazard data set has been created using the ODS PARMEST= option with the MIANALYZE procedure:

```
%if  %existfunc(projdata.hazardEST)  %then
%do;
data projdata.hazardEST;
  modify projdata.hazardest
         hazard;
...
  run;
%end;
%else  %do:
  data projdata.hazardEST;
  set hazard;
...
  run;
%end;
```

A macro similar to %EXISTFUNC is discussed in Carpenter (2002).

For displaying your final output from MIANALYZE the advantages of using HTML format include the ability to package it attractively and post it easily. But you should remember that in doing so you would not be able to extract specific parameter or other output values from HTML files. Therefore may want to send such output to EXCEL spreadsheets, for example, instead of or in addition to your HTML packaging. If it is anticipated that specific calculations or manipulations of some of these values will need to be done subsequently, then you may want to create SAS data sets. These data sets can then function as input for the calculation of such further results as hazard ratios, hazard ratio plots, odds ratios, or particular confidence intervals.

CONCLUSION

The MI and MIANALYZE procedure represent powerful tools to handle data with missing values. Their use, however, can add to the complexity of your programming, via multiple analysis data sets, repeated analyses, and a two-stage structure to the overall analysis of PHREG or other analysis procedures followed by MIANALYZE. This puts a premium on planning beforehand for modification, expansion, well-designed storage and access, and optimal presentation. Such planning and flexibility is greatly facilitated by the use of control files and SAS macro language.

ACKNOWLEDGEMENTS

Dr. Jacinte Jean was instrumental in initial development of the programmatics discussed here. Dr. Samuel A. Bozzette has directed the entire modeling project. Dr. Thomas A. Louis of the Johns Hopkins Bloomberg School of Public Health has served as a statistical consultant.

REFERENCES

Allison, Paul D., 2001, *Missing Data*, Sage University Papers Series on Quantitative Applications in the Social Sciences, 07-136. Thousand Oaks, CA: Sage.

Carpenter, Arthur L. and Richard O. Smith, 2002a, "Library and File Management: Building a Dynamic Application", Proceedings of the Twenty-Seventh Annual SAS® Users Group International Conference, Cary, NC: SAS Institute Inc., paper 21.

Carpenter, Arthur L. and Richard O. Smith, 2002b, "ODS and Web Enabled Device Drivers: Displaying and Controlling Large Numbers of Graphs", Proceedings of the Tenth Annual Western Users of SAS® Software Conference, Cary, NC: SAS Institute Inc.

Carpenter, Arthur L., 2002, "Macro Functions: How to Make Them - How to Use Them", Proceedings of the Twenty-Seventh Annual SAS® Users Group International Conference, Cary, NC: SAS Institute Inc., paper 17.

Schafer, J.L., 1997, *Analysis of Incomplete Multivariate Data*. Boca Raton: Chapman & Hall/CRC.

Therneau, Terry M. and Patricia M. Grambsch, 2000, *Modeling Survival Data: Extending the Cox Model*. New York: Springer-Verlag.

CONTACT INFORMATION

Chris Ake
Health Services Research & Development
Veterans Affairs San Diego Healthcare System
3350 La Jolla Village Drive, 111N-1
San Diego, CA 92161
(858) 552-8585 x 5492
cake@ucsd.edu

Art Carpenter
Data Explorations
2270 Camino Vida Roble, Suite L
Carlsbad, CA 92009
(760) 945-0613
art@caloxy.com