Paper TT-04-2015

Multiple Imputation for Arbitrary Missing Data: SAS® and R

Kaushal Raj Chaudhary, Sanford Research, Sioux Falls, South Dakota

Deanna Naomi Schreiber-Gregory, National University, Moorhead, Minnesota

Abstract

Missing data values are a common problem in the vast majority of real world data analysis situations. With this problem being as prominent as it is, it is important to have a well-rounded arsenal of skills to combat it. Multiple imputation is one such tactic that is well-supported as an effective resolution to this issue, especially when the missing data has an arbitrary component to it. Multiple imputation can be conducted through several different programs, but SAS® and R are by far the most popular choices. If both of these programs are available to the analyst, which should you choose? In this paper, the authors review different methods for conducting multiple imputation on arbitrary missing values in both SAS and R. These techniques are explored within the context of SAS 9.4 and presented in a way that would benefit beginning and moderate level SAS users, especially those versed in both SAS and R.

Introduction

Missing data is a common problem in all areas of analytics and research. It can take on many forms and can be attributed to several causes. For example, data that is derived from survey administration usually has missing data in the form of items that the respondent refuses, or neglects, to answer, resulting in a “don’t know” or “refused” response. This failure to respond to requested information can be driven by the desire to elude answering questions that are sensitive in nature, intrusive, confusing, or beyond the scope of knowledge of the respondent. As for observational and experimental research, missing observations can be due to missed clinical appointments, equipment failures, or other unforeseen or inconvenient circumstances. In other disciplines such as epidemiology (Schenker et al., 2010) and genetics (Bobb et al., 2011), missing data takes on another form and therefore receives another title, that of “unobserved data”. Unobserved data is a more appropriate label for missing values in these fields as their presence is normally due to a lack of opportunity for observation. Given this new label, unobserved data is especially appropriate as a promising candidate for powerful statistical analyses such as multiple imputation, given that the concept of multiple imputation is based on strong associations within variables that are observed in the same or similar studies.

Missing Data

As a direct result of our need to understand the meaning behind our missing data problem and to assist us in choosing the correct imputation method to handle this problem, devoted statisticians and analysts have managed to develop a two-part classification system for the most common missing data problems. The first dimension of this taxonomy is referred to as the “missing data pattern” and is comprised of the particular distribution of the missing observations across the data cases within our data set. The most common missing data pattern is termed as ordinary or “arbitrary”. This pattern relates to the idea that there is no particular pattern in the missing data structure. The second type of missing data pattern is termed monotonic. This type of data pattern is a much more structured or systematic pattern of missingness than that observed in datasets with arbitrary missing data. It is commonly seen in clinical or experimental trials through which the cause of missingness can be traced back to a specific identifiable point in time. The third missing data pattern arises within studies that incorporate randomization procedures to allow item missing data on selected variables for subsets of study observations. This type of technique is otherwise referred to as “matrix sampling” or “missing by design” sampling. This type of missingness is non-monotonic in nature and generally a prime candidate for multiple imputation utilization. Before moving on, it is worthy to note that in considering imputation implementation, it is important to understand the pattern of missing data before choosing the type of imputation method, as choosing the wrong method could have drastically negative effects on the integrity of your final “analysis-ready” dataset. In this paper, we will cover the most common type of missing data, arbitrary.

The second dimension of our previously mentioned devoted statisticians’ taxonomy work is the missing data mechanism. There are three understood types of missing data mechanisms: 1) missing completely at random (MCAR), 2) Missing at random (MAR), and 3) Not missing at random (NMAR). Data that is termed “missing completely at random” or MCAR holds the assumption that the variable Y has some missing values. These values are MCAR if the probability of the missing data on Y is unrelated (ie. independent or arbitrary) to the value of Y itself or to the values of any other variable within the data set. However, this assumption does still allow for the possibility that “missingness” on Y could be related to the “missingness” on some other variable X (Briggs et al., 2003; Allison,

2001). As for data that is termed “missing at random” or MAR, the assumption is that the probability of missing data on Y is completely unrelated to the value of Y after controlling for other variables within in the analysis (Allison, 2001). Lastly, data that is termed “not missing at random” or NMAR, follows the assumption that these missing values do, in fact, depend on the impact of unobserved values. A common rule surrounding the concept of MAR and the potential utilization of missing data modeling is thus employed: 1) if the MAR assumption is fulfilled, then the missing data mechanism is said to be ignorable. This essentially means that there is no need to employ a model of the missing data mechanism as a part of the estimation process. Multiple imputation, among other approaches, is one way to assess these types of missing values; 2) if the MAR assumption is not fulfilled, then the missing data mechanism is said to be non-ignorable and, therefore, the missing data mechanism must be modeled in order to obtain appropriate estimates of the chosen parameters. This type of analysis requires an in-depth and thorough understanding of the missing data process and causes behind the presence of the missing values. Given this consideration, the topic of this paper will be covering instances in which the MAR assumption has been fulfilled.

Multiple Imputation

Multiple imputation is a statistical technique that was developed for analyzing incomplete data sets. It is the process of “filling in” missing data points with probable values derived from the existing dataset. Multiple imputation, otherwise known as MI, is a Monte Carlo technique through which the missing values are replaced by $m>1$ simulated versions, where m is typically small in size (Rubin, 1987). The application of this technique can be accomplished in three steps: 1) imputation, 2) analysis, and 3) pooling. In the imputation phase, the missing data points are filled in m times to generate m complete data sets. In the analysis phase, the m complete data sets are analyzed using standard procedures, and during the third and last phase the results from the m complete data sets are combined for inference. In this paper, we will cover two ways that this technique can be performed in SAS and one way that it can be performed in R.

Fully Conditional Specification (FCS)

The FCS method is another component of this paper that needs to be addressed. The Fully Conditional Specification (FCS) method is quite commonly used for imputation of missing data in large mixed sets of continuous, nominal, ordinal, count, and semi-continuous variables. This complex array of variable types make the process of identification and maintenance of missing data a difficult process. This process is thankfully simplified through use of the FCS procedure. Given the type of approached used through this procedure, FCS has otherwise been known as the sequential regression algorithm in IVEware of the “chained equations” approach in Stata and R. Loosely described, these algorithms are based on an overarching iterative algorithm. In this algorithm, each iteration of the algorithm moves one at a time through the sequence of variables within the imputation model. Throughout this process and at each of these iterations and for each variable used, there is a P-Step and I-Step. Within the P-Step, the current values of not only the observed, but also the imputed values for the variables in the imputation model are used to derive the predictive distributions of the missing values within the target variable. Next, in order to model the conditional predictive distribution of the individual target values, the multiple imputation procedure employs the same regression or discriminant function methods used to address monotone missing data patters by utilizing the linear regression or regression-based mean matching approach to impute missing values for continuous variables, ordinal logistic regression to generate imputations needed for binary or ordinal classification variables, and lastly, the discriminant function methods for nominal classification variables. These updated imputations are then generated by stochastic draws derived from the predictive distribution that are defined by the updated regression model. Once the last variable in this sequence is imputed, the algorithm cycle s again through each variable, thus repeating the chain of regression estimation and imputation draw steps. This is repeated until the cycles converge on the user-defined algorithm or the system default has been met.

Methods

We know that missing data is a problem in pretty much every “real world” data set. We also know that there are several different methods to approach the issue of missing data. Our question now is, how does SAS and R approach this subject? What options do they offer? We will now touch on some of the differences between the two programming languages in not only their approach to missing data representation, but also in the procedures through which we can handle this very necessary step in the data preparation process.

Missing Data in SAS

First, we will touch on how SAS handles missing data. SAS will identify a missing value as missing depending on how it is represented in the data set. For example 1) if you have a numeric variable, a single decimal point or “.” is used to denote a missing value in numeric data, 2) if you have a character variable, a blank that is enclosed in quotes or “ ‘ ‘ “ is used to denote a missing value in character data, 3) if you have a variable with special characters, a missing value can either be represented by a decimal point followed by a letter (such as .B) or by a decimal point followed by an underscore (such as ._). SAS will identify each of these entries as a missing variable, so long as they are represented within a variable of the appropriate type.

When approaching a new dataset, one must consider the possibility of missing data. In order to obtain an initial grasp as to how much missing data is already presented in a dataset, it is useful to employ a procedure that would help check for values that are already presented appropriately as missing. There are several ways to do, many of which are available for implementation solely through use of the DATA Step. This makes it very convenient to check and control for missing data early on in the data preparation process.

One way to check for missing values is through use of the N and NMISS functions. These functions return the number of nonmissing and missing values, in respective order, from a list of numeric arguments. If you are needing to check for ordinary missing numeric values, you can use a coding structure similar to the following:

```
If numvar=. then do;
```

However, if your data contains special missing values, then another approach is needed. When checking for either ordinary or special missing values, a statement similar to the following can be used:

```
If numvar<=.z then do;
```

For missing character values, a statement such as this can be used:

```
If charvar=' ' then do;
```

Another option to check for either character or numeric missing values is through the MISSING function. In order to employ this alternative, a statement such as the following could be used:

```
If missing(var) then do;
```

In any of these cases, SAS will check the value of the variable in the current observation to see if it satisfies the condition specified. If it does, then SAS will execute the DO loop as specified in the respective statement. However, there are times when a missing value is coded as something other than what SAS will recognize as missing. If this occurs, it is important to identify these types of values and recode them to missing so that SAS handles them appropriately. SAS enables us to set defined values to missing within the DATA step by using programming statements such as:

```
If status<0 then status=.;
```

This statement will set the stored value of STATUS to a numeric missing value if STATUS has a value that is less than 0. Another option is to display a missing numeric value with a character other than a period by using the DATA step's MISSING statement or the MISSING= system option. In the case of a character value, we can identify missing values by using a programming statement such as:

```
If gender="none" then gender=' ';
```

Another option would be to set a missing value for one or more variable values through use of the CALL MISSING routine. This sets identified variable values to missing. This option is also especially useful when a dataset contains both numeric and character values as the CALL MISSING routine is able to control for both types. The programming statement to perform this is as follows:

```
call missing(university, degree);
```

So far we have covered a brief introduction to what multiple imputation is and how SAS identifies and approaches missing data. By now, we should be ready to implement a multiple imputation procedure to control for the missing data in our data set through SAS procedures. Before we get into the different ways of handling missing data in SAS, we will touch on how to identify and control for missing data in R and how it differs from the approach that SAS uses.

Missing Data in R

Now, we will touch on how R approaches missing data. In R, missing values in a dataset are represented by the symbol "NA" which stands for "not available". Any other impossible values (such as dividing by zero or the concept of imaginary numbers) are represented by the symbol "NAN" which stands for "not a number". Also, unlike SAS, R uses the same symbol for both character and numeric data, thus disregarding the actual type of data being analyzed. There are several different ways to approach missing values in R. First, we need to make sure that all missing values are accounted for and presented in the same format. One way to do this is to take an inventory of not only the array of values that are presented in the dataset, but also the number of values that have already been coded as missing. One way to do this is through the following code:

```
Is.na(x) # returns TRUE of x is missing
```

```
Y <- c(1,2,3,NA)
Is.na(y) # returns a vector (F F F T)
```

Another step in the data preparation process may be the need to recode certain values to missing. For example, if a particular questionnaire coded a missing value as 99, we would want to make sure that this value is taken into consideration as missing in the analysis, versus being considered a valid value in and of itself. In order to do this, we must recode this value of 99 to missing. One way to do this is through the following code:

```
# recode 99 to missing for variable v1
# select rows where v1 is 99 and recode column v1
Mydata$v1[mydata$v1==99] <- NA
```

Now that we have accounted for all of the missing data points in our dataset, our next step is to decide how to handle them. Luckily, most modeling functions available in R offer several options for dealing with missing values. However, multiple imputation offers an alternative to these options that is much more advanced and sophisticated than simple pairwise or listwise missing value deletion. R offers good multiple imputation implementation methods through its Amelia II, mice, mi, and mitools procedures. In this paper, we will concentrate mainly on the R Mice package as an option for multiple imputation in R.

SAS Proc Mi

The MI procedure in SAS was developed as a direct result of the need for a procedure that performs multiple imputation of missing data. The MI procedure creates multiple imputed data sets for incomplete multivariate data. It then uses methods that incorporate appropriate variability across the *m* imputation. The method used is dependent on the patterns of missingness within the dataset and chosen by the programmer. These methods can range from nonparametric, to Markov chain Monte Carlo (MCMC) and FCS (Fully conditional specification). We'll be using FCS method in this paper. Regardless of the method chosen, once the *m* complete data sets are analyzed using the standard SAS procedures, additional steps may be utilized in order to generate valid statistical inferences about these parameters by combining results from the *m* analysis, such that is possible through the MIANALYZE procedure. In our presentation we will demonstrate how SAS Proc MI can be implemented as a multiple imputation algorithm in SAS and as a means of handling arbitrary missing data values. An example code is shown below.

```
proc mi data=lwbs_fcs seed=1234 out=lwbs_impute;
   class lwbs age_cat insur_2 num_v_cat study_site triage4 dist_cat race1 sex timing1;
   fcs plots=trace
   nbiter=10
   discrim(race1/ classeffects=include details)
   logistic(age_cat/  details order=internal)
   logistic(insur_2/ link=logit  details)
   logistic(triage4/ details order=internal details)
   logistic(dist_cat/ link=logit details)
   logistic(lwbs/link=logit  details descending);
   var dist_cat age_cat insur_2 triage4 race1 num_v_cat study_site sex timing1 lwbs ;
run;

proc glimmix data= lwbs_impute;
 class race1 age_cat insur_2 triage4 urban sex dist_cat study_site timing1/ ref=first;
 model lwbs (ref=first)= race1 age_cat insur_2 triage4 urban sex dist_cat timing1
race1*triage4/link=logit dist=binary oddsratio solution covb ddfm=residual;
 random intercept / subject=study_site solution;
 lsmeans race1*triage4/ e pdiff oddsratio cl ilink slice=triage4 slicediff=triage4;
 by _imputation_;
 ods output parameterestimates=gparms;
run;

proc mianalyze parms=gparms;
   class race1 age_cat insur_2 triage4 urban sex  dist_cat study_site timing1;
   modeleffects intercept race1 age_cat insur_2 triage4 urban sex dist_cat timing1
race1*triage4;
run;
```

SAS IVEware

IVEware, or Imputation and Variance Estimation Software, was developed by the researchers at the Survey Methodology Program at the University of Michigan. This software performs five very distinct functions: 1) it performs single or multiple imputations of missing values while using the Sequential Regression Imputation Method, 2) it performs a variety of descriptive and model based analyses that account for complex design features such as clustering, stratification, and weighting, 3) it performs multiple imputation analyses for both descriptive and model-based survey statistics, 4) it is able to create either partial or full synthetic data sets using the Sequential Regression Approach to protect confidentially and limit statistical disclosure, and 5) it is able to combine information from multiple sources by vertically concatenating data sets and multiply the process of imputing the missing portions to create a larger rectangular data set. As an added feature, IVEware includes the option of six modules: IMPUTE, DESCRIBE, REGRESS, SASMOD, SYNTHESIZE, and COMBINE. In our presentation we will demonstrate how SAS IVEware can be implemented as a multiple imputation algorithm in SAS and as a means of handling arbitrary missing data values. An example code follows below.

```
options set = srclib "c:\iveware\srclib" sasautos = ('!srclib' sasautos) mautosource;

data chart_abstraction_mwsug;
  set abs.exclude_not_stated;
  keep recordid smokingduring_prior_preg smoking_during_pregnancy smoking_prior_preg
       alcoholuseanytime alcoholusepriorpreg alcoholuse_indexpreg druguseanytime
       drugusepreg drugusepriorpreg indicationsdisability prenatalcare
       prenatalcare_startedtrimester prenatalcare_visit_category
       total_birth_weight_pounds birth_weight_category gestational_age_at_birth
       gestational_age_category;
run;

%impute(name=mwsug, dir =c:\iveware\tutorial, setup = new)
    datain chart_abstraction_mwsug ;
    dataout chart_abstraction_imputed all;
    default continuous;
    categorical smokingduring_prior_preg  smoking_during_pregnancy  smoking_prior_preg
alcoholuseanytime alcoholusepriorpreg alcoholuse_indexpreg druguseanytime drugusepreg
drugusepriorpreg indicationsdisability prenatalcare prenatalcare_startedtrimester
prenatalcare_visit_category birth_weight_category gestational_age_category;
    transfer recordid;
    multiples 5;
    seed 876;
run;
```

R Mice Package

The R package, mice, imputes incomplete multivariate data by chained equations. It first appeared as an S-PLUS library in the year 2000 and was given the name, mice 1.0. It appeared a year later in 2001 as an R package. In our presentation we will demonstrate how mice can be implemented as a multiple imputation algorithm in R and as a means of handling arbitrary missing data values. An example code of the use of mice package is shown below.

```
library('mice')
library(mitools)
md.pattern(df)
imp=mice(df, m=5, seed=456)
mydata <- imputationlist(lapply(1:5, complete, x=imp))
summary(mydata)

fit<- with(imp,lm(total_birth_weight_pounds~smokingpriortopreg+smokingduringpregnancy+smokingduringpriorpreg+
   alcoholuseanytime+alcoholusepriorpreg+alcoholuseindexpreg+druguseanytime+druguseindexpreg+prenatalcare+
prenatalcarestartedtrimester))
pool(fit,2)
```

Conclusion

Missing data is a common problem in all areas of analytics and research. In review of the nature of missing data, we find out that missing data itself can take on one of three different forms: 1) missing completely at random (MCAR), 2) Missing at random (MAR), and 3) Not missing at random (NMAR). If the MAR assumption is fulfilled, then multiple imputation is able to be employed in order to control for the incidence of missing values.

In this paper the authors explored several different ways in which to check for missing values as well as to recode appropriate missing values in both SAS and R. We then touched on three separate ways to employ multiple imputation through either SAS or R as a way of working with the presence of missing values in an otherwise complete dataset. Please contact the authors of this paper for questions, examples, and further information on the use of these procedures.

References

IVEware: Imputation and Variance Estimation Software. Retrieved Jan 2015 from http://www.isr.umich.edu/src/smp/ive/.

Allison, P., 2001. Missing data —Quantitative applications in the social sciences. Thousand Oaks, CA: Sage. Vol. 136.

Bobb, J. F., D. O. Scharfstein, M. J. Daniels, F. S. Collins, and S. N. Kelada. 2011. Multiple Imputation of missing phenotype data for QTL mapping. *Statistical Applications in Genetics and Molecular Biology*, 10, 1, 1-27.

Briggs, A., Clark, T., Wolstenholme, J., Clarke, P., 2003. Missing.... presumed at random: cost-analysis of incomplete data. Health Economics 12, 377–392.

Bryk, A.S. and Raudenbush, S.W. (1992) *Hierarchical Linear Models*. Sage, Newbury Park.

Gilks, W.R., Richardson, S. & Spiegelhalter, D.J. (Eds.) (1996) *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London.

IVEware: Imputation and Variance Estimation Software. (2014). Retrieved Oct 2015 from http://www.isr.umich.edu/src/smp/ive/

Little, R.J.A. and Rubin, D.B. (1987) *Statistical Analysis with Missing Data*. J. Wiley & Sons, New York.

Meng, X.L.(1995) Multiple-imputation inferences with uncongenial sources of input (with discussion). *Statistical Science*, 10, 538-573.

Rubin, D.B. (1976) Inference and missing data. *Biometrika*, 63, 581-592.

Rubin, D.B. (1987) *Multiple Imputation for Nonresponse in Surveys*. J. Wiley & Sons, New York.

Rubin, D.B. (1996) Multiple imputation after 18+ years (with discussion). *Journal of the American Statistical Association*, 91, 473-489.

Schafer, J.L. (1997) *Analysis of Incomplete Multivariate Data*. Chapman & Hall, London.

Schafer, J.L. (1999) Multiple imputation: a primer. *Statistical Methods in Medical Research*, in press.

Schafer, J.L. and Olsen, M.K. (1998)

Schafer, J. L. *The multiple imputation FAQ page*. Retrieved Jan 2015 from http://sites.stat.psu.edu/~jls/mifaq.html.

Schenker, N., T. E. Raghunathan, and I. Bondarenko, I. 2010. Improving on analyses of self-reported data in a large-scale health survey by using information from an examination-based survey. *Statistics in Medicine*, 29, 533-545.

Berglund, P. and Heeringa, S. (2014). *Multiple Imputation of Missing Data Using SAS.* Cary, NC: SAS® Institute Inc.

Field, A., & Miles, J. (2012). *Discovering Statistics Using SAS*® , Thousand Oaks, CA: Sage Publications.

SAS® Institute Inc. 2008. SAS /STAT® 9.2 *User's Guide*. Cary, NC: SAS® Institute Inc.

Contact Information

Your comments, questions, and suggestions are valued and encouraged. Contact the authors at:

Kaushal Raj Chaudhary
Sanford Research
Sioux Falls, SD
Email: kaushal.chaudhary@SanfordHealth.org

Deanna Naomi Schreiber-Gregory
National University
La Jolla, CA / Moorhead, MN
E-mail: d.n.schreibergregory@gmail.com