Paper 3295-2015

Imputing Missing Data using SAS®

Christopher Yim, California Polytechnic State University, San Luis Obispo

ABSTRACT

Missing data is an unfortunate reality of statistics. However, there are various ways to estimate and deal with missing data. This paper explores the pros and cons of traditional imputation methods vs maximum likelihood estimation as well as singular versus multiple imputation. These differences are displayed through comparing parameter estimates of a known dataset and simulating random missing data of different severity. In addition, this paper utilizes the SAS® procedures PROC MI and PROC MIANALYZE and shows how to use these procedures in a longitudinal data set.

INTRODUCTION

At some point as a statistician, you will come across missing data. It is important to know what the common techniques are for handling missing data and what the benefits are to each method. In particular, this paper discusses list-wise deletion (also known as complete case analysis), regression imputation, stochastic regression imputation, maximum likelihood, and multiple imputation.

List-wise deletion is perhaps the easiest and simplest method to implement. With this method any observation with one more missing values is discarded from the data set. The benefit to this method is purely convenience. However, more often than not, there are more disadvantages than advantages. It can add a large amount of bias to the data if the “missingness” is related to the observation (i.e. a certain group of observations has a higher chance of having incomplete data). In addition, it can reduce sample size by a large amount and is particularly destructive in multivariate scenarios where maybe 1 or 2 out of 10 response are missing. In most cases, it is too much of a gamble to justify using it.

Arithmetic mean imputation is another simple method that is worth mentioning. Instead of deleting observations, you would fill in the missing values with the average of the available cases. To one somewhat inexperienced in the statistics world, this method would appear appealing. However, this method is a gamble and it is very data dependent. In general, this method can seriously distort the results by skewing the parameter estimates. Specifically, this method reduces the variability of the data which in turn reduces the effectiveness of measures of association. This method of imputation should be avoided. No simulation was done on this method due to the fact that this method’s results vary depending on the form of the data.

Regression imputation (also known as conditional mean imputation) fills missing values with predicted values that are generated from a regression equation. Variables tend to be related so it makes intuitive sense to use this information to fill in missing values. However, this method can be biased. It overstates the correlation between variables and also underestimates the variability of the data. In an attempt to resolve these issues, stochastic regression also uses regression equations to predict missing data, but it also adds a normally distributed residual term in an attempt to compensate for the natural variability in the data. In practice, this method has the highest potential of being unbiased out of the methods discussed so far. However, this method can also devalue the standard errors which can lead to a higher risk of type I error. In addition it does not reflect the uncertainty of the missing values.

Another method of dealing with missing data is using maximum likelihood (ML) parameter estimation. Rather than filling in data and then estimating parameters, maximum likelihood estimates the mean vector and covariance matrix to create a single imputed data set. The way that maximum likelihood estimates parameters will depend on the algorithm used. This paper uses the expectation maximization (EM) algorithm for ML parameter estimation. This algorithm first estimates the missing value, given the observed values in the data. The data is then processed using ML estimation, giving new mean and covariance estimates. Using the new estimates, the missing values are again estimated. This process is repeated until the maximum change in estimates does not exceed a certain criterion. This method requires missing at random values and a large sample size to be effective.

最大似然（ML）参数估计：最大似然估计均值向量和协方差矩阵以创建单个估算数据集，而不是填充数据然后估计参数。最大似然估计参数的方式将取决于所使用的算法。本文将期望最大化（EM）算法用于ML参数估计。该算法首先在给定数据中的观测值的情况下估计缺失值。然后使用ML估计对数据进行处理，给出新的均值和协方差估计。使用新的估计值，再次估计缺失的值。重复这一过程，直到估计值的最大变化不超过某一标准。这种方法需要随机缺失值和大样本量才能有效。

多重插补不是试图估计具体的值并使用这些估计来预测参数，而是从其分布中抽取缺失值的随机样本。该方法包括3个步骤：
1. 创建多个估算数据集；
2. 对每个估算数据集进行分析模型分析；
3. 组合得到的参数估计值

Multiple imputation is the last strategy that will be discussed. Instead of attempting to estimate each value and using these estimates to predict the parameters, this method draws a random sample of the missing values from its distribution. This method involves 3 steps, creating multiple imputed data sets, carrying out the analysis model on each of the imputed data sets, and then combining parameter estimates to get a single set of parameter estimates. Steps 2 and 3 are fairly mechanical (see documentation), however, step 1 requires some planning. The method of choice depends on the patterns of “missingness”. For data sets with monotone continuous missing patterns, one can use stochastic regression as discussed earlier. For data sets with arbitrary missing patterns, it is suggested to use the Markov Chain Monte Carlo (MCMC) method (“Multiple Imputation in SAS: part 1”). In short this is very similar to maximum likelihood. It estimates the missing values, obtains new parameter estimates and then uses those estimates to predict the missing values again. However, the parameter estimates are derived using Bayesian estimation of the mean vector and covariance matrix.

使用多重填补时，必须指定填补数据集的数量，并且三到五个数据集就足够了。然而，缺失数据的百分比越大，就越需要进行更多的估算才能得到准确的估计

When using multiple imputation, the number of imputed data sets must be specified and as few as three to five data sets can be adequate. However, the larger the percentage of missing data, the more imputations are necessary to get an accurate estimate. Unfortunately, there is not a simple criterion to determine when MCMC has converged. The technique that is commonly used is to visually inspect the parameters from successive iterations to determine if there is a clear trend. If there is no trend, this suggests that the model has reached a stationary distribution. In addition, autocorrelation may be possible between imputed datasets. It is best to make sure that the new dataset does not have this feature. This method requires a fairly large sample size. After the multiple imputation is complete, the predetermined model is then used on each set and the parameter estimates are combined to obtain a singular set of estimates.

METHODS

The primary goal of this paper is to compare and contrast the previously discussed methods of imputation for missing data. To do this, various versions of the same data set were simulated with random values set to missing. The data used for the simulations was based on a fabricated example data set that measured a group of subject’s seizure occurrences within a time period of 2 weeks. Three predictor variables (treatment, baseline seizure count, and age) were recorded for 100 observations, where each had 4 time periods. For the sake of the simulations, no censoring was incorporated in the fabrication of the data set. The general trend of the example data set responses was linear, but this may not be the case for all datasets. The results of these simulations are obviously dependent on the dataset and may differ for other kinds of data. For example, if responses increase with time, then time points near the end and the beginning of the time will be drastically off course.

The first step to comparing the methods was deciding on a model to use for the data. The 4 time period seizure counts were modeled using the 3 predictor variables: what treatment they were assigned, their baseline seizure count, and their age. After the model was fit, the parameter estimates were recorded as the “true” parameter values (as a baseline comparison). After that, a certain percentage of responses were dropped randomly at the following rates: 5%, 10%, 15%, 20%, and 25%. After imputing, the same model previously used was fit, and the new parameter estimates were record and then compared across data sets. This was done 1000 times for each of the 5 missing rates.

Some of these imputation methods are easy to incorporate in SAS, demonstrating their appeal of convenience. List-wise deletion can be done simply by using the DELETE statement as shown in the following program.

```
DATA ListWiseExample;
   SET test;
   ARRAY var[*] Response1 Response2 ;
   DO i=1 TO dim(var);
      IF missing(var(i)) THEN DELETE;
   END;
RUN;
```

The remainder of the methods, stochastic regression, maximum likelihood estimation and multiple imputation can be all run using the MI procedure. 其余方法、随机回归、最大似然估计和多重填补都可以使用MI程序运行

PROC MI has the following structure:

```
PROC MI <options>
   BY variables;
   CLASS variables;
   [Imputation method];
   FREQ variable;
   TRANSFORM transform (variables </options>);
   VAR variables;
```

The BY statement specifies groups to stratify imputations analysis. The CLASS statement lists which variables are categorical variables. If the variable exists in the data set, the FREQ statement specifies the frequency of occurrence. TRANSFORM specifies the variables to be transformed before imputing. The VAR statement specifies the numeric variables to be analyzed/imputed.

To choose which imputation method you want, you have 4 options. If the data is missing at random, you would use EM (expectation maximization - MLE), FCS (fully conditional specification - Regression), or MCMC (Markov Chain Monte Carlo). If you know that your data has monotone missingness, you would use the MONOTONE statement to impute. The simulation data example is assumed to be missing at random and thus EM, FCS and MCMC are the options that are to be used.

To impute data using stochastic regression, the FCS statement is used. Within the FCS statement, you can impute using a discriminant function, logistic regression, regression, or predictive mean matching. Including the FCS statement will automatically use regression for continuous variables and discriminant functions for categorical variables. For each imputed variable, all other variables in the VAR statement are used as covariates. An error term is added to each missing observation based on the observed variability of the data. As an option, you can specify the set of effect to impute the variables.

PROC MI is a procedure designed to do multiple imputations; however, you can use it to do single imputations by specifying in the options that NIMPUTE (number of imputations) is equal to 1. In addition, in the FCS statement, make sure to include that NBITER (number of burn in iterations) equals 1. This will ensure 1 iteration of imputation and 1 set of imputed data. Lastly, to output the imputed data, use the OUT = option. The imputation for the example simulation data using stochastic regression is shown in the following code.

```
PROC MI DATA = test NIMPUTE = 1 OUT = example3;
   CLASS Treat;
   FCS NBITER = 1;
   VAR Treat B_Count Age Response1 Response2;
RUN;
```

To use maximum likelihood estimation, instead of using the FCS statement, you would use the EM statement. In addition, specify the OUT = option to save the imputed data set. You can also output the mean and covariance matrix estimates from the MLE using the OUT = option. Do be aware that as of now, PROC MI does not utilize categorical data in MLE estimation, and a dummy variable needs to be created if you want to include that information.

Use the following PROC MI step to utilize MLE:

```
PROC MI DATA = test;
   EM OUT = example4;
   VAR Treat B_Count Age Response1 Response2;
RUN;
```

The last method is multiple imputation using MCMC. Even though there are various options for the MCMC statement, only a few were used as a focused for this paper. You can specify whether or not a single chain is used for all imputations or separate chains are used for each imputation. You can also

include a PLOTS option to view both the trace and autocorrelation function plots. This allows you to ensure convergence and independence in your iterations. You can specify the number of imputed data sets in the PROC MI statement using the NIMPUTE option. To output the imputed data sets, specify OUT = option in the PROC MI statement. The default number of burn-in iterations is 200. The MCMC imputation for the simulation example is shown in the following code.

```
PROC MI DATA = test NIMPUTE = 6 OUT = example5;
   MCMC CHAIN = MULTIPLE PLOTS = TRACE PLOTS = ACF;
   VAR Treat B_Count Age Response1 Response2;
RUN;
```

Now that you have your missing data imputed multiple times, you can run your model and obtain your estimates for the model parameter estimates. For this paper, a multivariate regression model was used. You need to specify the OUTEST option so that a data set containing parameter estimates is output. Another way to output a data set of parameter estimates would be to use the ODS OUTPUT option. It is important to also include the COVOUT option for the next step, as shown in the following code.

```
PROC REG DATA = example5 OUTEST = regout COVOUT;
   MODEL Response1 Response2 = Treat B_Count Age;
   BY _imputation_;
RUN;
```

For the multiple imputation a new procedure needs to be used, the MIANALYZE procedure. This procedure summarizes the multiple imputations and provides parameter estimates.

PROC MIANALYZE has the following structure:

```
PROC MIANALYZE <options>;
   BY variables;
   CLASS variables;
   MODELEFFECTS effects;
   <label:> TEST equation1<,...,<equationk>></options>;
   STDERR variables;
```

The BY and CLASS statements have the standard uses. The required MODELEFFECTS statement lists the effects to be analyzed. The TEST statement tests linear hypothesis around the parameters. If the data analyzed includes both parameter estimates and standard errors as variables, use STERR statement to list the standard errors associated with effects. For the simulation in this paper, because there are multiple responses, some alterations are necessary for the MCMC imputation analysis to work. After PROC MI and PROC REG, you must use PROC MIANALYZE separately on each response variable. If no BY statement is used, an error statement about within-imputation COV matrix not being symmetric will appear. In addition, after PROC REG is finished, the dataset is sorted by imputation, not by the response variables. Therefore, a PROC SORT is also necessary as shown in the following code.

```
PROC REG DATA = MCMCimp OUTEST = regMCMCout COVOUT;
   MODEL Response1 Response2 = Treat B_Count Age;
   BY _imputation_;
RUN;

PROC SORT DATA = regMCMCout;
   BY _DEPVAR_;
RUN;

PROC MIANALYZE DATA = regMCMCout;
   BY _DEPVAR_;
   MODELEFFECTS Treat B_Count Age;
Run;
```

It is important to note that both PROC MI and PROC MIANALYZE are only appropriate with single-level observations. That is, if data has observations that are nested within clusters, these methods are not appropriate as they may result in variance estimates based toward zero and possible other biased parameters (Mistler, 2013).

When comparing methods, two things were of interest for this paper. How far the simulated parameter estimates were from the "true" parameter estimates and how far the simulated standard deviation estimates were from the "true" standard deviation. Bias was calculated in the following way.

Parameter Estimate Bias = $\beta - b_i$

Standard Deviation Bias = $\sigma_\beta - s_{b_i}$

In addition, the mean square error of the parameter estimate bias for each percentage and overall were investigated. MSE was calculated in the following way.

MSE for each percentage = $\sum_{j=1}^{4}\sum_{i=1}^{1,000}\left[\left(b_{ij}-\beta\right)^2\right]/(4\ x\ 1{,}000)$

MSE Overall = $\sum_{k=1}^{5}\sum_{j=1}^{4}\sum_{i=1}^{1,000}[\left(b_{ijk}-\beta\right)^2]/(5\ x\ 4\ x\ 1{,}000)$

Where i is simulation iterations, j is dependent variable, and k is percentage.

Lastly, how the methods were related to each was also of interest. Because each method was used on the same simulated missing dataset, methods that are similar to each will most likely both estimate either high or low on the same dataset. A scatterplot matrix of the parameter bias is provided to further explore this.

RESULTS

MEAN SQUARE ERROR

After completing the simulations, the measured mean square error (MSE) was obtained for each method. Table 1 displays the MSE for each method by percentage of missing observations as well as overall MSE.

Percentage	List-Wise	Stochastic Re-gression	MLE	MCMC
5	0.319177	0.154360	0.072803	0.082472
10	0.712898	0.265906	0.150776	0.170686
15	1.253279	0.437186	0.244857	0.277702
20	2.144380	0.616171	0.354181	0.399573
25	3.232556	0.784034	0.451530	0.517979
Overall	1.532458	0.445531	0.254829	0.289682

Table 1. Mean Square Error for Each Method at Various Percentages of Missing Data

For each of the methods investigated, the MSE increased as the percentage of missing observations increased. List-Wise had the highest MSE for each percentage and overall. In addition, MLE had the lowest MSE of the four methods for each percentage and overall.

BIAS

As previously mentioned, the different Bias Estimates for our data were also recorded. Table 2 displays the parameter estimate bias (while Table 3 displays the sample SD bias) for each method. In addition both tables provide 95% confidence intervals on the biases. If the confidence interval does not contain zero, the method at that percentage was considered to be biased either high or low, as shown in bold. That is, a biased high estimate will have a confidence interval with only positive bounds and vice versa. Otherwise, the method was considered to be unbiased.

Missing Percentage		**5**	**10**	**15**	**20**	**25**
List-Wise Deletion	**Mean**	0.0028	0.0031	0.0191	-0.0106	-0.0025
	CI	(-0.0044, 0.0100)	(-0.0069, 0.0132)	**(0.0070, 0.0313)**	(-0.0249, 0.0036)	(-0.0180, 0.0130)
Stochastic Regression	**Mean**	0.0176	0.0170	0.0355	0.01515	0.0166
	CI	**(0.0067, 0.0286)**	**(0.0010, 0.0330)**	**(0.0151, 0.0560)**	(-0.0092, 0.0395)	(-0.0108, 0.0441)
MLE	**Mean**	0.0124	0.0165	0.0479	0.0210	0.0309
	CI	**(0.0041, 0.0208)**	**(0.0045, 0.0286)**	**(0.0326, 0.0632)**	**(0.0026, 0.0395)**	**(0.0101, 0.0517)**
MCMC	**Mean**	0.0135	0.0140	0.0484	0.0199	0.0314
	CI	**(0.0046, 0.0224)**	**(0.0012, 0.0268)**	**(0.0321, 0.0647)**	**(0.0003, 0.0395)**	**(0.0091, 0.0537)**

Table 2. Parameter Estimate Bias and Confidence Interval for Each Method at Various Percentages of Missing Observations.

In terms of parameter estimates, biases performed in the following way. List-Wise was in general unbiased, except for at 15% missing. This is most likely an artifact of the fact that the data was missing at random, and not due the particular missing percentage. Because the data were missing at random, we essentially have a smaller sample size when using this method and our expected parameter estimate would not change. However, for the other methods they were in general biased high. MLE and MCMC were biased high for all percentages while Stochastic Regression was biased high only for 5%, 10% and 15%.

Missing Percentage		**5**	**10**	**15**	**20**	**25**
List-Wise Deletion	**Mean**	-0.1092	-0.2414	-0.3893	-0.5725	-0.7943
	CI	**(-0.1152, -0.1032)**	**(-0.2513, -0.2316)**	**(-0.4034, -0.3751)**	**(-0.5918, -0.5533)**	**(-0.8199, -0.7686)**

Stochastic Regression	**Mean**	-0.0055	-0.0092	-0.0143	-0.0057	-0.0046
	CI	**(-0.0085, -0.0025)**	**(-0.0134, -0.0050)**	**(-0.0194, -0.0093)**	(-0.0119, 0.0005)	(-0.0115, 0.0022)
MLE	**Mean**	0.0173	0.0355	0.0509	0.0789	0.1012
	CI	**(0.0144, 0.0201)**	**(0.0315, 0.0632)**	**(0.0461, 0.0556)**	**(0.0731, 0.0848)**	**(0.0948, 0.1076)**
MCMC	**Mean**	-0.0302	-0.0615	-0.0965	-0.1189	-0.1506
	CI	**(-0.0333, -0.0271)**	**(-0.0661, -0.0570)**	**(-0.1023, -0.0908)**	**(-0.1261, -0.1116)**	**(-0.1589, -0.1423)**

Table 3. Standard Deviation Bias and Confidence Interval for Each Method at Various Percentages of Missing Observations.

The last output provides a comparison of bias for each method. Figure one displays these comparisons.

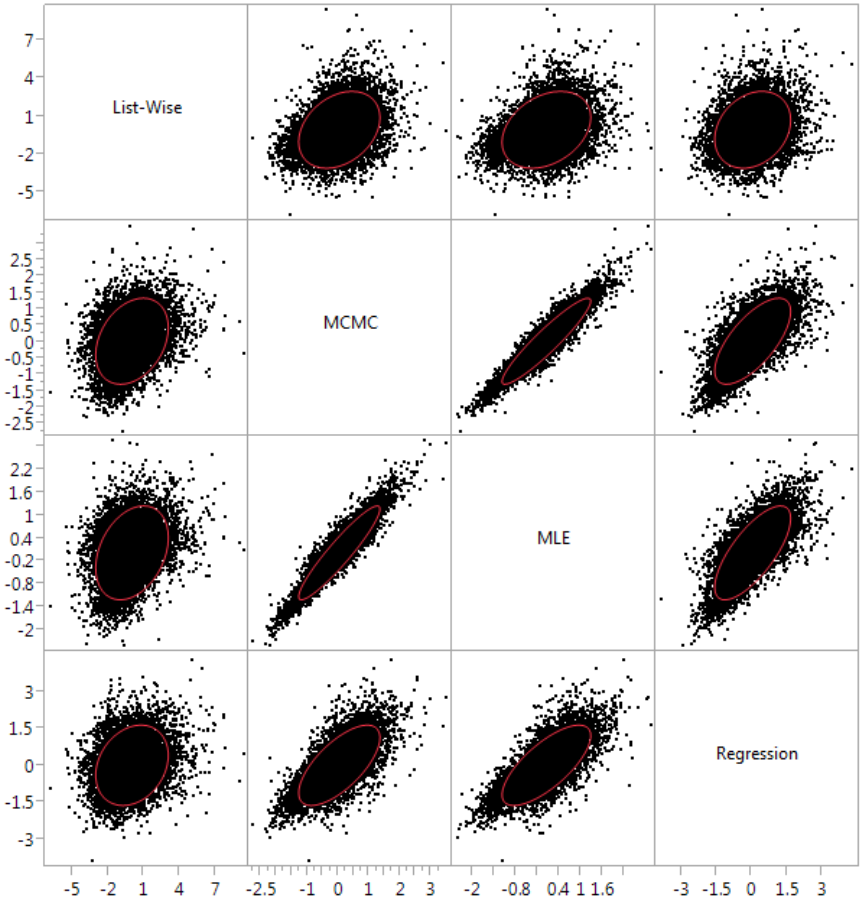

Figure 1. Scatterplot Matrix of the Parameter Estimate of each Method.

CONCLUSION

As seen above, the MSE for MLE and MCMC were consistently the smallest out of the four methods. For each increase in missing percentage, list-wise deletion's MSE increased on average by 0.728. Regression increased on average by 0.157, MLE by 0.095, and MCMC by 0.109. Based on this, it appears MLE and MCMC deal with higher missing percentages better than the other methods.

For the parameter estimates, List-Wise Deletion was unbiased the most. However this result cannot be expected across all data sets. The unbiasedness of the results is due to the fact that the simulation data was designed to be missing at random. However, the slightest relationship between observations and their completeness will skew the parameter estimates. The next best method based on the simulation is stochastic regression. If your data is linear with respect to both predictor variables and response variables, stochastic regression imputation will be the most precise method. However, unless this is known a priori, it is safer to use MLE and MCMC as they are more robust methods.

For the standard deviation (SD) estimates, the List-Wise Deletion underestimated the standard deviation consistently and severely. Stochastic Regression underestimated for smaller percentages of missingness, but for 20% and 25%, the SD estimate was on point. MLE overestimated consistently for all missing rates and MCMC underestimated consistently. However, none of them are as drastic as the list-wise deletion method.

The scatterplot matrix reveals a couple of interesting things. MCMC and MLE imputation methods appear to be very similar to each other. Regression also appears to be equally related to MLE and MCMC. However list-wise deletion is nothing like the other methods and is fairly inconsistent.

For this analysis to be truly comprehensive, the same method should be used on various datasets with different attributes. In addition, as an improvement on the methods used here, a greater population of observations could be created that could be sampled from and then each sample would have random recordings removed. Plans for future analysis include exploring datasets with categorical responses and using different models to see how each method estimates.

REFERENCES

Mistler, Stephen. 2013. "A SAS® Macro for Applying Multiple Imputation to Multilevel Data". *SAS Global Forum 2013 Proceedings*. Paper 438-2013.

Multiple Imputation in SAS, Part 1". UCLA: Statistical Computing Seminars. January 15th, 2015. Available at http://www.ats.ucla.edu/stat/sas/seminars/missing_data/part1.htm

"Multiple Imputation in SAS, Part 2". UCLA: Statistical Computing Seminars. January 15th, 2015. Available at http://www.ats.ucla.edu/stat/sas/seminars/missing_data/part2.htm

Yuan, Yang. 2000. "Multiple Imputation for Missing Data: Concepts and New Development". *SAS Users Group International 25 Proceedings*. P267-25

ACKNOWLEDGMENTS

I would like to thank Rebecca Ottesen for motivating and encouraging me to write and submit this paper. Thank you to Dr. Andrew Schaffner for inspiring my work and guiding me with the design. Thank you to Dr. Louise Hadden for helping with editing and the general submission process.

RECOMMENDED READING

- *SAS® Knowledge Base Focus Areas Multiple Imputation for Missing Data http://support.sas.com/rnd/app/stat/new/dami.html*
- *SAS® Documentation: The MI Procedure*
- *SAS® Documentation: the MIANALYZE Procedure*

CONTACT INFORMATION

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

Christopher Yim
yimcunechris@gmail.com

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.

Other brand and product names are trademarks of their respective companies.