



# HANDLING MISSING VALUES WITH SAS

Proc MI / Proc MiAnalyze

# SAS MISSING VALUES

(Y, X1, X2, X3,...,Xn,)



Y (dependent)	X1	X2	X3	X4
5	2	1	Yes	1
4	3	1	Yes	2
8	7	1	No	3
.	.	.	" "	.
23	6	1	Yes	5
5	3	2	No	6
32	4	2	No	7
.	8	2	No	8
45	7	2	Yes	9
.B	56	2	No	10

# MCAR

The data are missing completely at random (MCAR) if the probability that Y is missing does not depend on X or on Y itself (Rubin 1976).

Many traditional missing data techniques are valid only if the MCAR assumption holds.

Ex: If Y is a measure of delinquency and X is years of schooling, MCAR would mean that the probability that data are missing on delinquency is unrelated to either delinquency or schooling.

# MAR

Again, this is most easily defined in the case where only a single variable  $Y$  has missing data, and another set of variables  $X$  has no missing data. We say that data on  $Y$  are missing at random if the probability that  $Y$  is missing does not depend on  $Y$ , once we control for  $X$ .

Thus, MAR allows for missingness on  $Y$  to depend on other variables that are observed. It just cannot depend on  $Y$  itself (after adjusting for the observed variables).


In essence, MAR allows missingness to depend on things that are observed, but not on things that are not observed. Clearly, if the data are missing completely at random, they are also missing at random.

It is straightforward to test whether the data are missing completely at random. For example, one could compare men and women to test whether they differ in the proportion of cases with missing data on income. Any such difference would be a violation of MCAR. However, it is impossible to test whether the data are missing at random, but not completely at random. For obvious reasons, one cannot tell whether delinquent children are more likely than nondelinquent children to have missing data on delinquency.



# MNAR

What if the data are not missing at random (NMAR)? What if, indeed, delinquent children are less likely to report their level of delinquency, even after controlling for other observed variables? If the data are truly NMAR, then the missing data mechanism must be modeled as part of the estimation process in order to produce unbiased parameter estimates. That means that, if there is missing data on  $Y$ , one must specify how the probability that  $Y$  is missing depends on  $Y$  and on other variables. This is not straightforward because there are an infinite number of different models that one could specify. Nothing in the data will indicate which of these models is correct. And, unfortunately, results could be highly sensitive to the choice of model.



Analysis can be done in SAS v9.4 if the data is MCAR or MAR. Generally, if we assume the data is MAR then maximum likelihood estimation techniques are valid\* or we can use Proc MI to get valid results in all\*\* scenarios.

\* (A) What is the procedure actually doing with missing observations? (B) The simplicity of ignoring missing data will come at the cost of (i) Power and/or (ii) asymptotic validity.

\*\* (B) Imputation Model needs to be correctly specified.

# MORE ON APPENDAGE “A”

What SAS actually doing when missing observations are present?

GEE's and Repeated/ Clustered models may not work with missing data both because of SAS Procedures aren't well equipped to handle the scenario and because mathematically it does not work



# PROCS THAT EXCLUDE MISSING VALUES

Proc Freq

Proc means

Proc Reg (excludes all missing observations with corresponding missing values in any variable)

Proc GLM\* (excludes observations with missing dependent variable observations or missing classification variable observation)

Proc Genmod (for GLMs – excludes any observation with a missing value)

Caution:

Excluding observations with missing values also ignores the possible systematic difference between the complete cases and incomplete cases, and the resulting inference might not be applicable to the population of all cases, especially with a smaller number of complete cases.

# PROCS THAT INCLUDE MISSING VALUES

Proc Freq (with by statement and/or certain table statement options)

Proc Means (with by statement)

Proc Anova (in certain nested scenarios)

Proc GLM\* (with Manova or Repeated Statemtns or Manova option in the Proc line, proc glm uses an observation if values are non-missing for all dependent variables and all variables used in independent effects)

Proc Genmod (for GEE's only– excludes missing values within clusters; By default, puts missing values within clusters at the end of the cluster and iteratively estimates the working correlation matrix with all available values)

# PROC MI

What is Proc MI?

“Instead of filling in a single value for each missing value, multiple imputation replaces each missing value with a set of plausible values that represent the uncertainty about the right value to impute (Rubin 1976, 1987). The multiply imputed data sets are then analyzed by using standard procedures for complete data and combining the results from these analyses. No matter which complete-data analysis is used, the process of combining results from different data sets is essentially the same.”

[-http://support.sas.com/documentation/cdl/en/statug/63962/HTML/default/viewer.htm#statug\\_mi\\_sect001.htm](http://support.sas.com/documentation/cdl/en/statug/63962/HTML/default/viewer.htm#statug_mi_sect001.htm)



**Go...**  
Save yourself from the Zombies

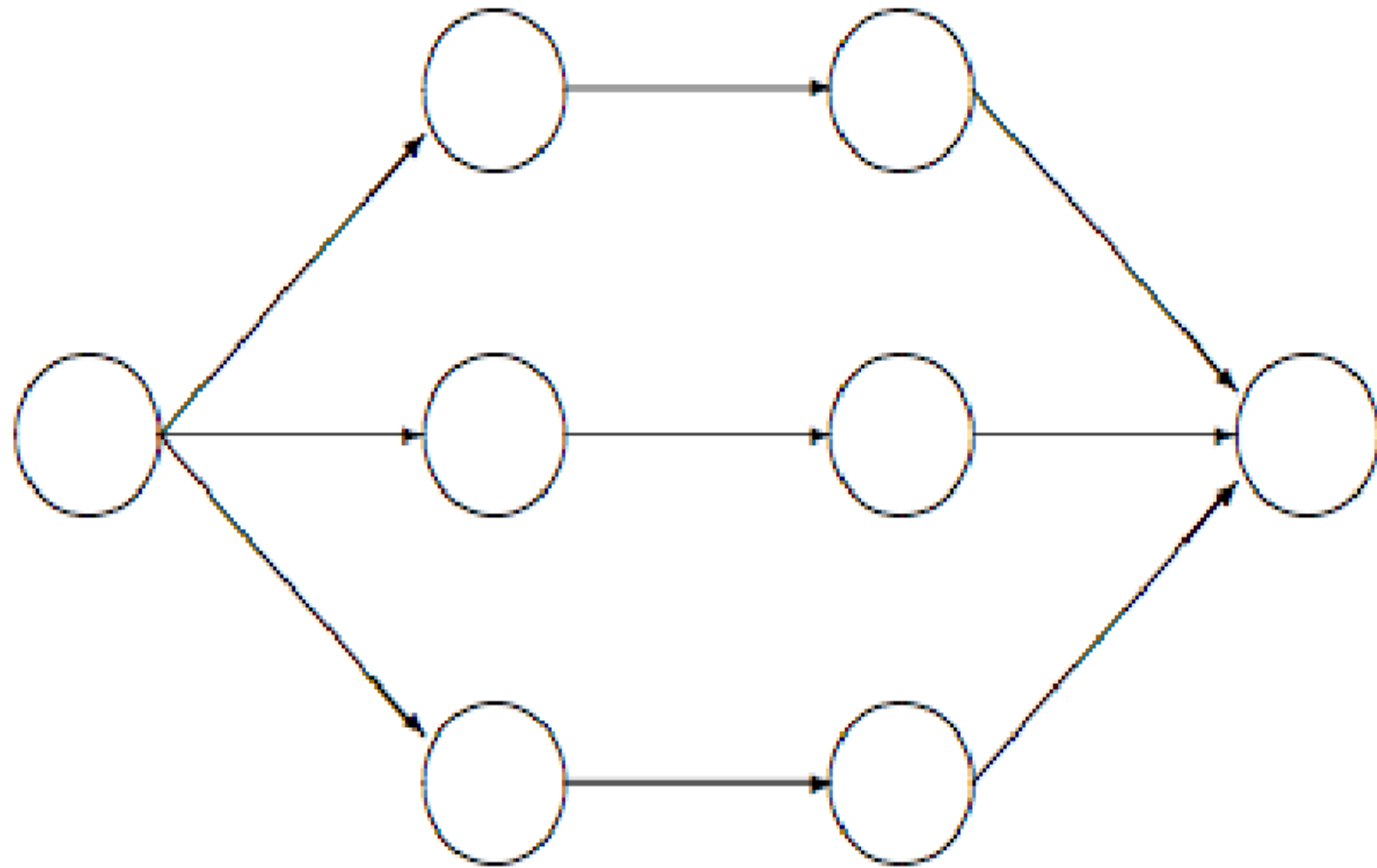
# PROC MIANALYZE

What is Proc MiAnalyze...

“Multiple imputation does not attempt to estimate each missing value through simulated values, but rather to represent a random sample of the missing values. This process results in valid statistical inferences that properly reflect the uncertainty due to missing values; for example, valid confidence intervals for parameters.”

“For data sets with monotone missing patterns, the variables with missing values can be imputed sequentially with covariates constructed from their corresponding sets of preceding variables. To impute missing values for a continuous variable, you can use a regression method (Rubin 1987, pp. 166–167), a predictive mean matching method (Heitjan and Little 1991; Schenker and Taylor 1996), or a propensity score method (Rubin 1987, pp. 124, 158; Lavori, Dawson, and Shera 1995). To impute missing values for a classification variable, you can use a logistic regression method when the classification variable has a binary or ordinal response, or use a discriminant function method when the classification variable has a binary or nominal response.”

“For data sets with arbitrary missing patterns, you can use either of the following methods to impute missing values: a Markov chain Monte Carlo (MCMC) method (Schafer 1997) that assumes multivariate normality, or a fully conditional specification (FCS) method (Brand 1999; van Buuren 2007) that assumes the existence of a joint distribution for all variables.”



Incomplete data

Imputed data

Analysis results

Pooled results

# WHY ARE THERE TWO PROC(EDURES) ?

Because missing data occurs in many different settings...

The primary analysis question can be many possible procedures built into SAS

-> SUBSTANTIVE model

sub·stan·tive

*adjective*

1.

having a firm basis in reality and therefore important, meaningful, or considerable.

"there is no substantive evidence for the efficacy of these drugs"

2.

having a separate and independent existence.

# DEFINING THE IMPUTATION MODEL

This is the “hardest” part. Most of the analysis time may be spent at this step. Here are 2 things to consider:

(1) What is random part in your model?

(2) If the analysis has more than one *variable*, then what is the relationship between the random variable with missing data and the other variables.

Ex: Least-squares regression

What happens if you add more variables?

What happens if you have less variables in the imputation model than the substantive model?

PROC MI <options> ; ←Required  
BY variables ; ←Optional  
CLASS variables ; ←Optional  
EM <options> ; ←Optional  
FCS <options> ; ←Optional  
FREQ variable ; ←Optional  
MCMC <options> ; ←Optional  
MONOTONE <options> ; ←Optional  
TRANSFORM transform ( variables </ options>) <...transform ( variables </ options>) > ; ←Optional  
VAR variables ; ←Optional  
Run;





# PROC STATEMENT OPTIONS

NIMPU=number specifies the number of imputations. The default is NIMPU=5.

OUT=SAS-data-set creates an output SAS data set in which to put the imputation results. The data set includes an identification variable, `–IMPUTATION–`, to identify the imputation number. For each imputation, the data set contains all variables in the input data set, with missing values being replaced by the imputed values.

SEED=number specifies a positive integer that is used to start the pseudorandom number generator. The default is a value generated from reading the time of day from the computer's clock. However, in order to be able to duplicate the result under identical situations, you must control the value of the seed explicitly rather than rely on the clock reading. If the default value is used, the seed information is displayed so that the results can be reproduced by specifying this seed with the SEED= option. You need to specify exactly the same seed number in the future to reproduce the same results.

# MORE PROC STATEMENT OPTIONS

MAXIMUM=numbers MINIMUM=numbers specifies the maximum/minimum values for imputed variables. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a VAR statement, and the specified numbers correspond to variables in the VAR statement. A missing value indicates no restriction on maximum/minimum for the corresponding variable.

ROUND=numbers specifies the units to round variables in the imputation. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a VAR statement, and the specified numbers correspond to variables in the VAR statement. The following statements generate imputed values rounded to the desired units:

```
proc mi data=Fitness1 seed=37851 out=miout2
```

```
round=.001 .01 1;
```

```
multinormal method=regression;
```

```
var Oxygen RunTime RunPulse;
```

```
run;
```

NBITER=number specifies the number of burn-in iterations before the first imputation in each chain. The default is NBITER=50. NITER=number specifies the number of iterations between imputations in a single chain. The default is NITER=30.



# FREQ STATEMENT

## **FREQ variable ;**

If one variable in your input data set represents the frequency of occurrence of other values in the observation, specify the variable name in a FREQ statement. PROC MI then treats the data set as if each observation appears  $n$  times, where  $n$  is the value of the FREQ variable for the observation. If the value of the FREQ variable is less than one, the observation is not used in the analysis. Only the integer portion of the value is used. The total number of observations is considered to be equal to the sum of the FREQ variable when PROC MI calculates significance probabilities.

# EM (EXPECTATION-MAXIMIZATION)

The EM statement uses the EM algorithm to compute the maximum likelihood estimate (MLE) of the data with missing values, assuming a multivariate normal distribution for the data.

**CONVERGE=*p***  
**XCONV=*p***

sets the convergence criterion. The value must be between 0 and 1. The iterations are considered to have converged when the change in the parameter estimates between iteration steps is less than *p* for each parameter—that is, for each of the means and covariances. For each parameter, the change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change. By default, CONVERGE=1E-4.

**INITIAL=CC | AC | AC(R=*r*)**

sets the initial estimates for the EM algorithm. The INITIAL=CC option uses the means and covariances from complete cases; the INITIAL=AC option uses the means and standard deviations from available cases, and the correlations are set to zero; and the INITIAL=AC(R=*r*) option uses the means and standard deviations from available cases with correlation *r*, where *r* is the number of variables to be analyzed. The default is INITIAL=AC.

**ITPRINT**

prints the iteration history in the EM algorithm.

**MAXITER=*number***

specifies the maximum number of iterations used in the EM algorithm. The default is MAXITER=200.

**OUT=SAS-*data-set***

creates an output SAS data set that contains results from the EM algorithm. The data set contains all variables in the input data set, with missing values being replaced by the expected values from the EM algorithm. See the section [Output Data Sets](#) for a description of this data set.

**OUTEM=SAS-*data-set***

creates an output SAS data set of TYPE=COV that contains the MLE of the parameter vector *beta*. These estimates are computed with the EM algorithm. See the section [Output Data Sets](#) for a description of this output data set.

# MONOTONE DATA

For data sets with monotone missing patterns, the variables with missing values can be imputed sequentially with covariates constructed from their corresponding sets of preceding variables. To impute missing values for a continuous variable, you can use a regression method (Rubin 1987, pp. 166–167), a predictive mean matching method (Heitjan and Little 1991; Schenker and Taylor 1996), or a propensity score method (Rubin 1987, pp. 124, 158; Lavori, Dawson, and Shera 1995). To impute missing values for a classification variable, you can use a logistic regression method when the classification variable has a binary or ordinal response, or use a discriminant function method when the classification variable has a binary or nominal response.

# SAS MONOTONE CHOICES

Option	Description
<u>DISCRIM</u>	Specifies the discriminant function method
<u>LOGISTIC</u>	Specifies the logistic regression method
<u>PROPENSITY</u>	Specifies the propensity scores method
<u>REG</u>	Specifies the regression method
<u>REGPMM</u>	Specifies the predictive mean matching method

# TRANSFORM OPTION

BOXCOX

EXP

LOG

LOGIT

POWER

(specify for all of the above methods)  $C=number$

(specify for POWER or BOXCOX methods)  $LAMBDA=number$

**EX:** `transform log(y1) power(y2/c=1 lambda=.5);`



# MCMC OPTIONS

Option	Description
<b>Data Sets</b>	
<u>INEST=</u>	Inputs parameter estimates for imputations
<u>OUTEST=</u>	Outputs parameter estimates used in imputations
<u>OUTITER=</u>	Outputs parameter estimates used in iterations
<b>Imputation Details</b>	
<u>IMPUTE=</u>	Specifies monotone or full imputation
<u>CHAIN=</u>	Specifies single or multiple chain
<u>NBITER=</u>	Specifies the number of burn-in iterations for each chain
<u>NITER=</u>	Specifies the number of iterations between imputations in a chain
<u>INITIAL=</u>	Specifies initial parameter estimates for MCMC
<u>PRIOR=</u>	Specifies the prior parameter information
<u>START=</u>	Specifies starting parameters
<b>ODS Output Graphics</b>	
<u>PLOTS=TRACE</u>	Displays trace plots
<u>PLOTS=ACF</u>	Displays autocorrelation plots
<b>Traditional Graphics</b>	
<u>TIMEPLOT</u>	Displays trace plots
<u>ACFPLOT</u>	Displays autocorrelation plots
<u>GOUT=</u>	Specifies the graphics catalog name for saving graphics output
<b>Printed Output</b>	
<u>WLF</u>	Displays the worst linear function
<u>DISPLAYINIT</u>	Displays initial parameter values for MCMC

# FCS

## Imputation Methods

DISCRIM

Specifies the discriminant function method

LOGISTIC

Specifies the logistic regression method

REG

Specifies the regression method

REGPMM

Specifies the predictive mean matching method

The FCS statement specifies a multivariate imputation by fully conditional specification methods. If you specify an FCS statement, you must also specify a VAR statement.

For each imputed variable, if no covariates are specified, then all other variables in the VAR statement are used as the covariates. That is, each continuous variable is used as a regressor effect, and each classification variable is used as a main effect. For the discriminant function method, only the continuous variables can be used as covariate effects.

With an FCS statement, the variables are imputed sequentially in the order specified in the ORDER= option. For a continuous variable, you can use a regression method or a regression predicted mean matching method to impute missing values. For a nominal classification variable, you can use a discriminant function method to impute missing values without using the ordering of the class levels. For an ordinal classification variable, you can use a logistic regression method to impute missing values by using the ordering of the class levels. For a binary classification variable, either a discriminant function method or a logistic regression method can be used. By default, a regression method is used for a continuous variable, and a discriminant function method is used for a classification variable.

# DATA VISUALIZATION

Make `nimpute=0`

Note: Use the substantive model , not the imputation model

# EX: MONOTONE DATA

```
*-----Data on Physical Fitness-----*
| These measurements were made on men involved in a      |
| physical fitness course at N.C. State University.      |
| Only selected variables of                             |
| Oxygen (intake rate, ml per kg body weight per minute), |
| Runtime (time to run 1.5 miles in minutes),            |
| RunPulse (heart rate while running) are used.         |
|                                                        |
| Certain values were changed to missing for the analysis |
*-----*

data Fitness1;
  input Oxygen RunTime RunPulse @@;
  datalines;
44.609 11.37 178      45.313 10.07 185
54.297  8.65 156      59.571  .      .
49.874  9.22  .       44.811 11.63 176
45.681 11.95 176      49.091 10.85  .
39.442 13.08 174      60.055  8.63 170
50.541  .      .       37.388 14.03 186
44.754 11.12 176      47.273  .      .
51.855 10.33 166      49.156  8.95 180
40.836 10.95 168      46.672 10.00  .
46.774 10.25  .       50.388 10.08 168
39.407 12.63 174      46.080 11.17 156
45.441  9.63 164      54.625  8.92 146
45.118 11.08  .       39.203 12.88 168
45.790 10.47 186      50.545  9.93 148
48.673  9.40 186      47.920 11.50 170
47.467 10.50 170
;

```

The following statements are used just to show the monotone missingness of the output data set *outex7*:

```
proc mi data=outex7 nimpute=0;
  var Oxygen RunTime RunPulse;
run;
```

The "Missing Data Patterns" table in [Output 54.7.3](#) displays a monotone missing data pattern.

#### Output 54.7.3 Monotone Missing Data Patterns

##### The MI Procedure

Missing Data Patterns								
Group	Oxygen	RunTime	RunPulse	Freq	Percent	Group Means		
						Oxygen	RunTime	RunPulse
1	X	X	X	110	70.97	46.152428	10.861364	171.863636
2	X	X	.	30	19.35	47.796038	10.053333	.
3	X	.	.	15	9.68	52.461667	.	.

```
proc print data=miout1 (obs=10) ;  
run;
```



Obs	_Imputation_	Oxygen	RunTime	Run Pulse
1	1	44.609	11.3700	178.000
2	1	45.313	10.0700	185.000
3	1	54.297	8.6500	156.000
4	1	59.571	9.8709	170.676
5	1	49.874	9.2200	137.623
6	1	44.811	11.6300	176.000
7	1	45.681	11.9500	176.000
8	1	49.091	10.8500	121.146
9	1	39.442	13.0800	174.000
10	1	60.055	8.6300	170.000

# NOW WHAT?

Use your substantive model (and the corresponding proc ) to analyze the multiply imputed data set. Make Use of the statement:

```
By _imputation_ ;
```

Or in Proc Surveyreg and proc surveyreg:

```
Domain _imputation_ ;
```

Simple Example :

```
proc means data=outimputedex1 ;
```

```
by _imputation_ ;
```

```
var weight ;
```

```
run ;
```

# MONOTONE EX CONTINUED

```
proc reg data=outmi outest=outreg covout noprint;  
model Oxygen= RunTime RunPulse;  
→ by _Imputation_;  
run;  
proc print data=outreg(obs=8);  
Var _Imputation_ _Type_ _Name_ Intercept RunTime RunPulse;  
title 'Parameter Estimates from Imputed Data Sets';  
run;
```



Parameter Estimates from Imputed Data Sets						
Obs	_Imputation_	_TYPE_	_NAME_	Intercept	RunTime	RunPulse
1	1	PARMS		97.2874	-2.98892	-0.10684
2	1	COV	Intercept	55.7516	-0.73348	-0.27870
3	1	COV	RunTime	-0.7335	0.15167	-0.00509
4	1	COV	RunPulse	-0.2787	-0.00509	0.00194
5	2	PARMS		90.9324	-2.93338	-0.07391
6	2	COV	Intercept	37.5576	-0.25970	-0.20442
7	2	COV	RunTime	-0.2597	0.13978	-0.00722
8	2	COV	RunPulse	-0.2044	-0.00722	0.00166

Figure 10.1. Parameter Estimates



```
proc mianalyze data=outreg;  
  var Intercept RunTime RunPulse;  
run;
```

The MIANALYZE Procedure

Model Information

Data Set                    WORK.OUTREG  
Number of Imputations     5

Multiple Imputation Variance Information

Parameter	-----Variance-----			DF
	Between	Within	Total	
Intercept	74.179857	57.287519	146.303348	10.805
RunTime	0.034202	0.142151	0.183193	79.694
RunPulse	0.001533	0.002304	0.004144	20.292

Multiple Imputation Variance Information

Parameter	Relative Increase in Variance	Fraction Missing Information
Intercept	1.553843	0.665161
RunTime	0.288719	0.242803
RunPulse	0.798522	0.491731

Figure 10.2. Model Information and Variance Information Tables

# PROC MI ANALYZE

PROC MIANALYZE <options> ; ←Required

BY variables ; ←Optional

CLASS variables ; ←Optional

MODELEFFECTS effects ; ←Required

<label:> TEST <, ..., <> > </> ; ←Optional

STDERR variables ; ←Optional

Run;



# MOST IMPORTANT

Tasks	Options
<b>Specify input data sets</b> a COV, CORR, or EST type data set parameter estimates and covariance matrices parameter estimates and $(X'X)^{-1}$ matrices	DATA= PARMS=, COVB= PARMS=, XPXI=
<b>Specify statistical analysis</b> parameters under the null hypothesis level for the confidence interval complete-data degrees of freedom multivariate inferences	THETA0= ALPHA= EDF= MULT

# INPUT DATASET

```
Proc Mianalyze Data=outset1;  
    var Intercept X1 T12;  
Run;
```

The appropriate combination depends on the SAS procedure you used to create the parameter estimates and associated covariance matrix.

→ For instance, if you used PROC REG to create an OUTEST= data set containing the parameter estimates and covariance matrix, you would use the DATA= option to read the OUTEST= data set.

→ Each input data set contains the variable `_Imputation_` to identify the imputation by number.

→ If you do not specify an input data set with the DATA=, COVB=, or XPXI= option, then the most recently created SAS data set is used as an input DATA= data set.

# PROCEDURES THAT WORK WITH PROC MI

All procedures work with Proc MI -- the trick is to create an output data set using the SAS ods output system with the correct procedure options so that the Analysis results can be summarized by Proc MIANALYZE

# SOME GOOD ADVICE

(1) “The inclusion of auxiliary variables can improve a multiple imputation model. However, inclusion of too many variables leads to downward bias of regression coefficients and decreases precision. When the correlations are low, inclusion of auxiliary variables is not useful.” (Hardy, Herke, Leohart)

(2) Make your imputation Model at least as complicated as your substantive model.

→ point 1 +2 seem to suggest using the same regression imputation model as your substantive model (when data is MAR)

(3) Increase burn-in and Number of iterations. Defaults settings (200? And 100?) are sometimes not large enough for convergence.

(4) Nimpote default is 5. This is probably good, but using nimpote=10 is a good choice.

# BIBLIOGRAPHY

<http://support.sas.com/documentation/>