



Handling missing data in statistical programming using Multiple Imputations in SAS.

Martyna Gawłowska, Biometrics, Late-stage Development,
Respiratory and Immunology (R&I), BioPharmaceuticals R&D,
AstraZeneca, Warsaw, Poland

Marta Tomczyk, Biometrics, Late-stage Development,
Respiratory and Immunology (R&I), BioPharmaceuticals R&D,
AstraZeneca, Warsaw, Poland

Nov 2022

Why do we have missing data in clinical trials? (examples)

Patient refusal to provide data

Patient lost to follow-up

Human error

Mechanical error



Missing data mechanisms

MCAR

(Missing completely at random)

- No dependence between observed/non-observed data and existence of missing values
- There is no bias in analyses and estimation of parameters with MCAR values
- The only case where deleting records with missing values is allowed

MAR

(Missing at random)

- Probability of occurrence of empty records is related to observed, non-missing data
- Probability of missing values are not related to the true value of missing data
- There may be a bias in analyses and estimation of parameters with MAR values
- Removing these records results in the loss of information and may introduce bias

MNAR

(Missing not at random)

- Probability of occurrence of empty records is unknown
- Probability of missing values can be related to the true value of missing data
- Missingness is related to factors not collected in the dataset
- There is a risk in deleting records with missing values as it can introduce bias



Handling missing data

PREVENTION

- Proper design of the study
- Constant monitoring of raw data
- Procedures for reducing missing values described in Clinical Study Protocol

SINGLE IMPUTATION

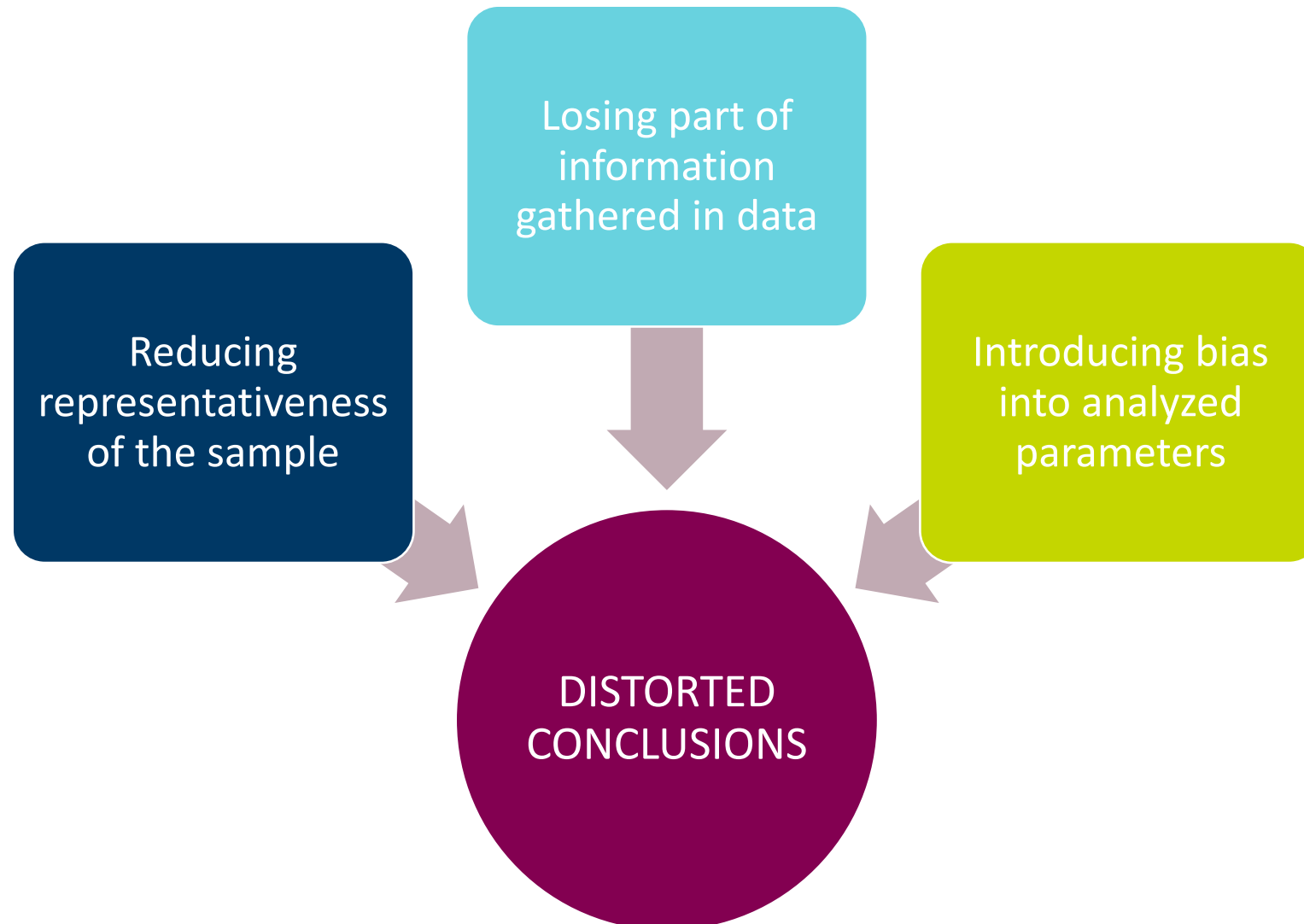
- Mean substitution
- Last observation carried forward
- Regression imputation

MULTIPLE IMPUTATION

- PROC MI procedure



What are the consequences of not handling missing values properly?



Input Data Structure

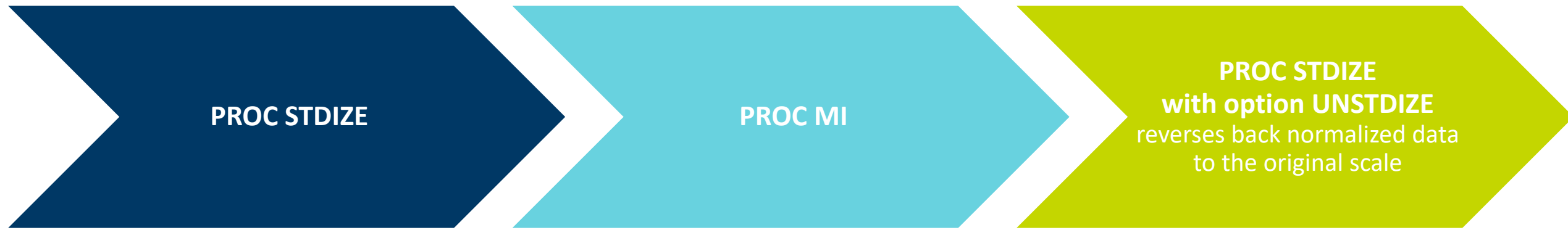
Dummy data generated to resemble data from a completed study

- *1400 missing values (~10%)
- *14 visits
- *1000 subjects and 2 subgroups:
 - Treatment (N=500),
 - Placebo (N=500)

USUBJID	TRT	TEST NAME	Week 2	Week 4	...	Week 44
1	Treatment	TEST1	1.3	MISSING VALUE	...	1.5
2	Placebo	TEST1	MISSING VALUE	2.7	...	0.7
...	...	...	...	...	...	...



Steps in data imputation



Normalization is an important step in multiple imputation.



PROC MI

Selected statements

PROC MI options

Data= ,
Out= ,
Nimpute= ,
Seed=

BY statement

PROC MI procedure

TRANSFORM statement

VAR statement



Sample code

```
proc stdize data = data_input out=missing_norm  
    outstat=param;  
    var Week: ;  
run;
```

```
proc mi data=missing_norm out=missing_imputed  
    nimpute=31 seed=2138654;  
    by trtn;  
    EM; FCS; MCMC;  
    var Week: ;  
run;
```

Number of imputations

Used default values of:

- EM (Expectation- Maximization)
- FCS (Fully conditional specification)
- MCMC (Markov Chain Monte Carlo)

```
proc stdize data = missing_imputed out=final  
    UNSTDIZE method=in(param);  
    var Week: ;  
run;
```



Comparison of observed and imputed data – Week 44

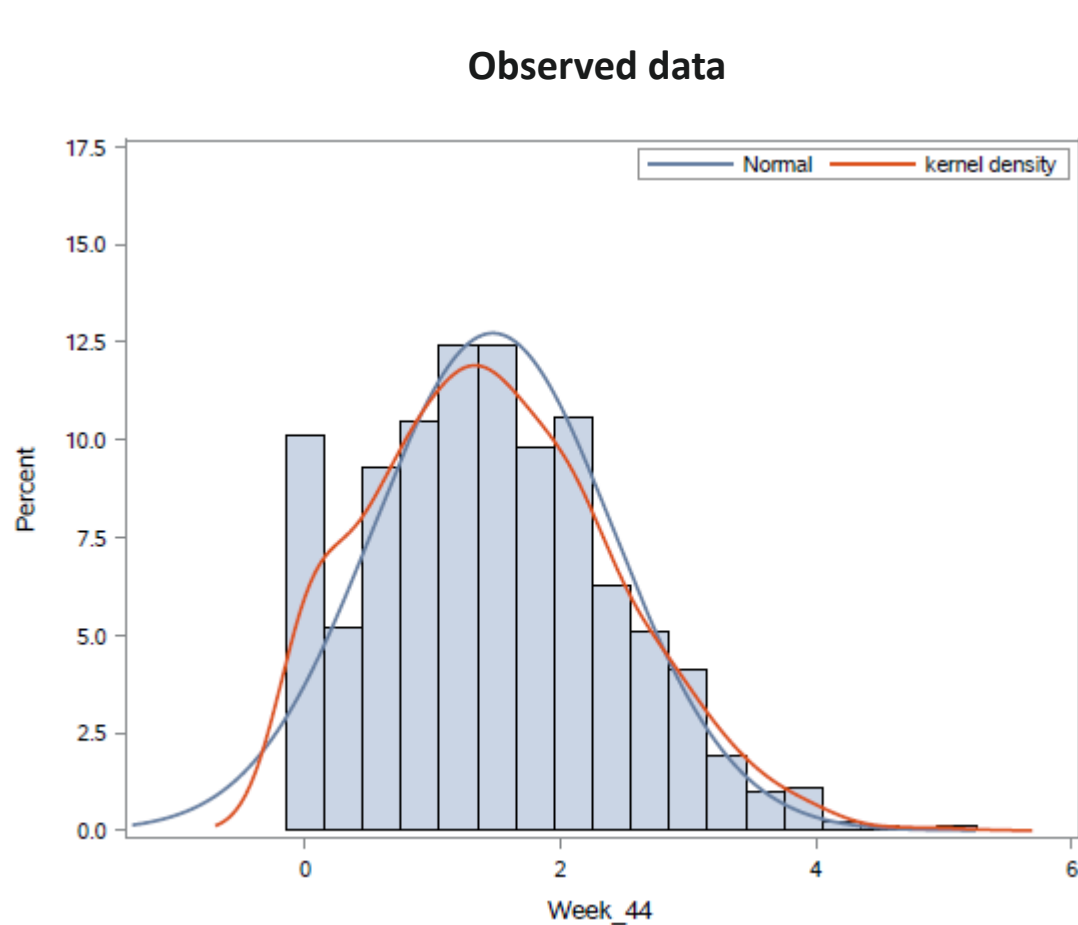


Figure1. Histogram of observed, non-imputed data (Week 44)

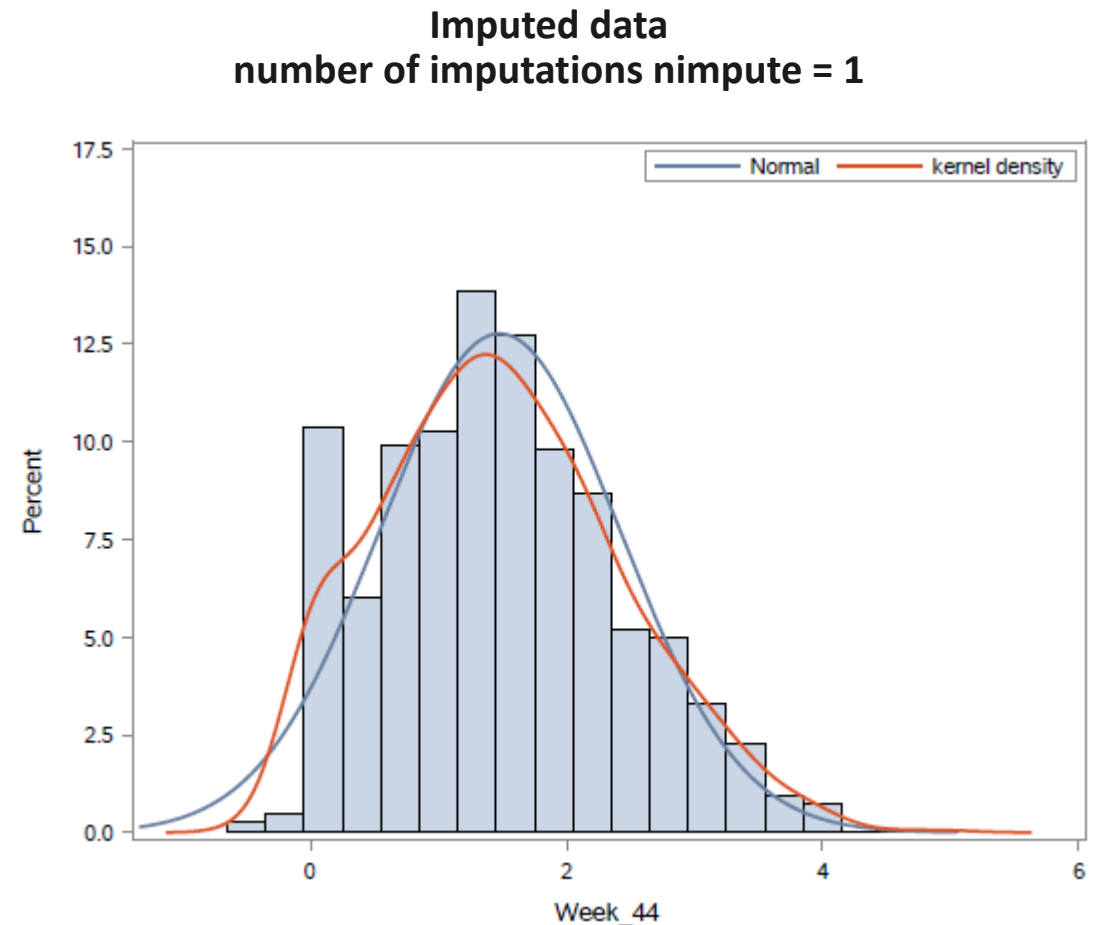


Figure2. Histogram of imputed data for nimpute=1 (Week 44)



Comparison of observed and imputed data – Week 44

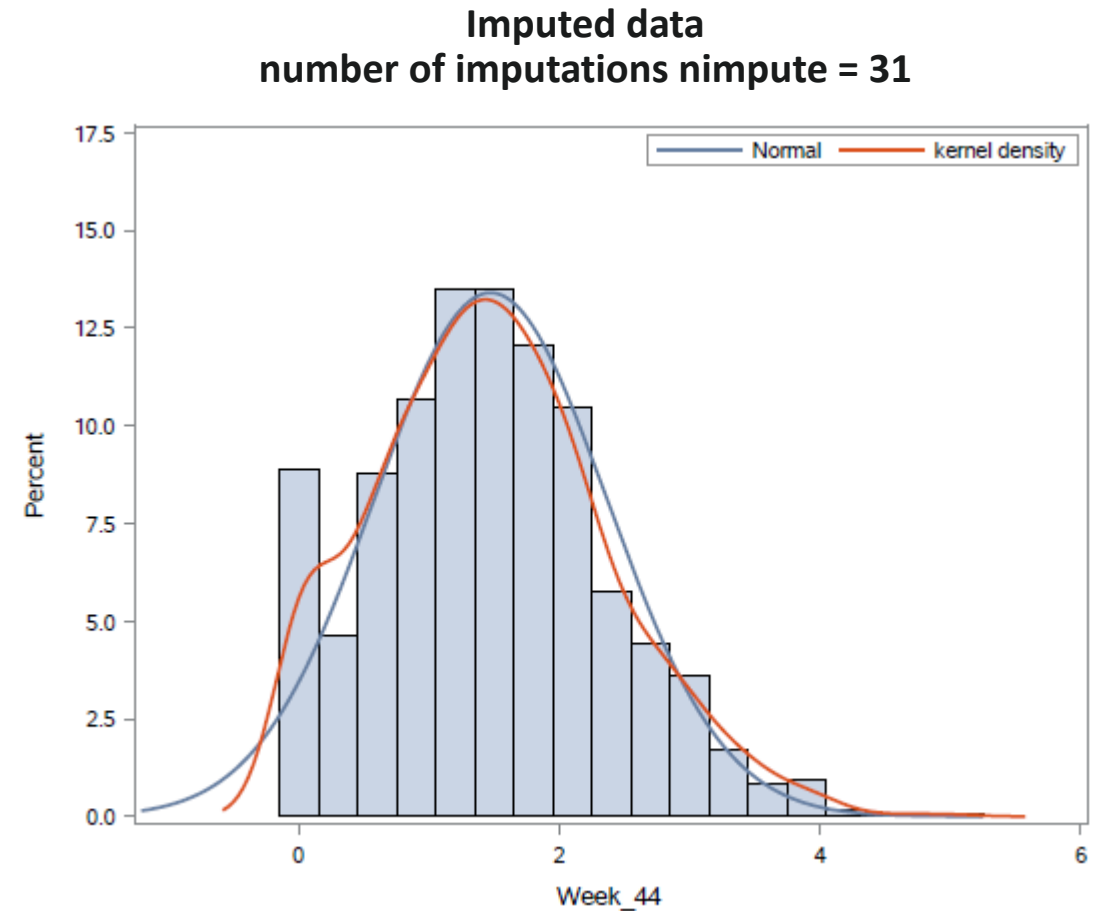
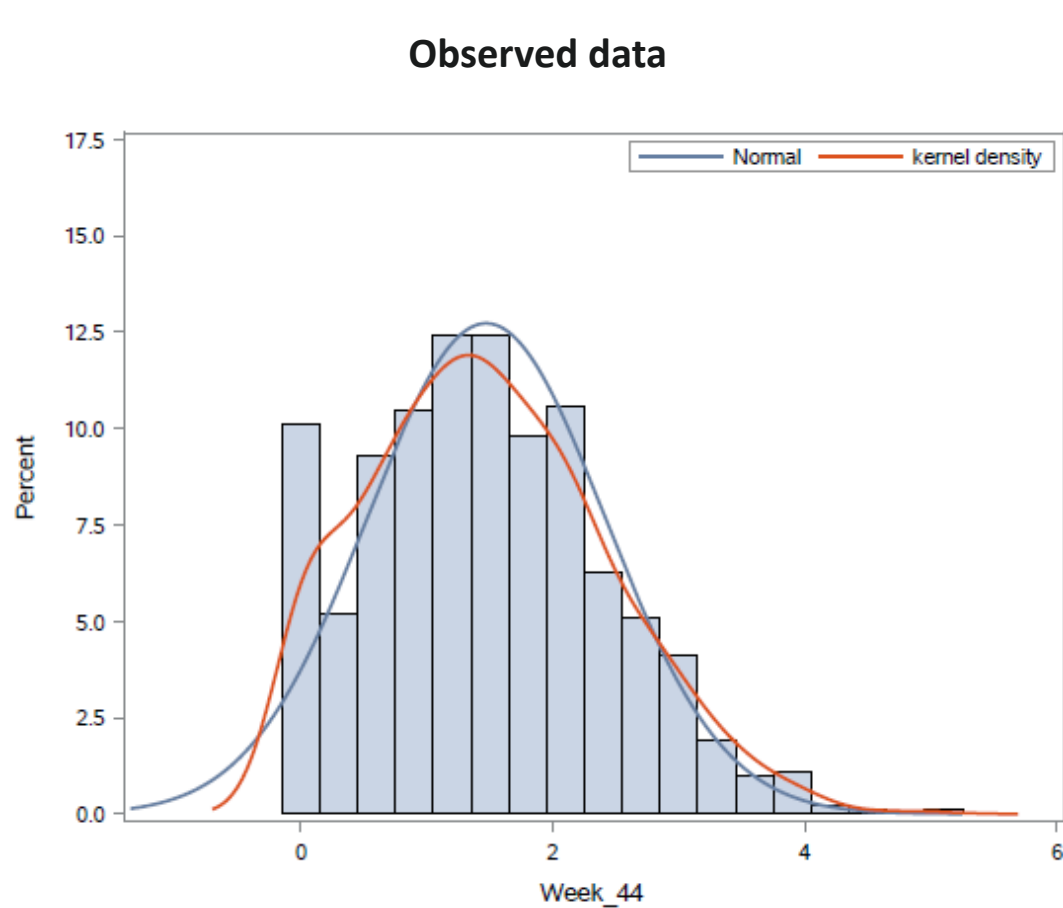


Figure3. Histogram of observed, non-imputed data (Week 44)

Figure4. Histogram of imputed data for nimpute=31 (Week 44)



Analysis of imputed data for selected weeks - summary table

MEAN					
DATA	No of imputations	WEEK 2	WEEK 16	WEEK 32	WEEK 44
Original	N/A	2,001	1,422	1,353	1,305
Imputed	nimpute=1	1,984	1,406	1,378	1,318
Imputed	nimpute=10	2,004	1,416	1,354	1,302
Imputed	nimpute=31	2,004	1,414	1,349	1,300
Imputed	nimpute=61	2,003	1,414	1,350	1,299
Imputed	nimpute=91	2,002	1,414	1,348	1,298
STD					
Original	N/A	0,971	0,910	0,913	0,901
Imputed	nimpute=1	0,993	0,914	0,928	0,893
Imputed	nimpute=10	0,940	0,882	0,877	0,857
Imputed	nimpute=31	0,937	0,880	0,873	0,852
Imputed	nimpute=61	0,937	0,880	0,872	0,853
Imputed	nimpute=91	0,937	0,880	0,872	0,853



Performance – PROC MI run times

No of imputations	10% missing data		20% missing data	
	Proc MI run time [s]	Proc MI run time per 1 imputation [s]	Proc MI run time [s]	Proc MI run time per 1 imputation [s]
nimpute=1	0,183	0,183	0,383	0,383
nimpute=10	0,930	0,093	1,888	0,189
nimpute=31	5,151	0,166	10,309	0,333
nimpute=61	7,650	0,125	15,645	0,256
nimpute=91	10,217	0,112	20,348	0,224
	Average time per 1 imputation [s]:	0,136	Average time per 1 imputation [s]:	0,277

Table 2. Comparison of PROC MI total run times and run times per 1 imputation for input datasets with 10% and 20% of missing data, for a different number of imputations nimpute: 1, 10, 31, 61 and 91.

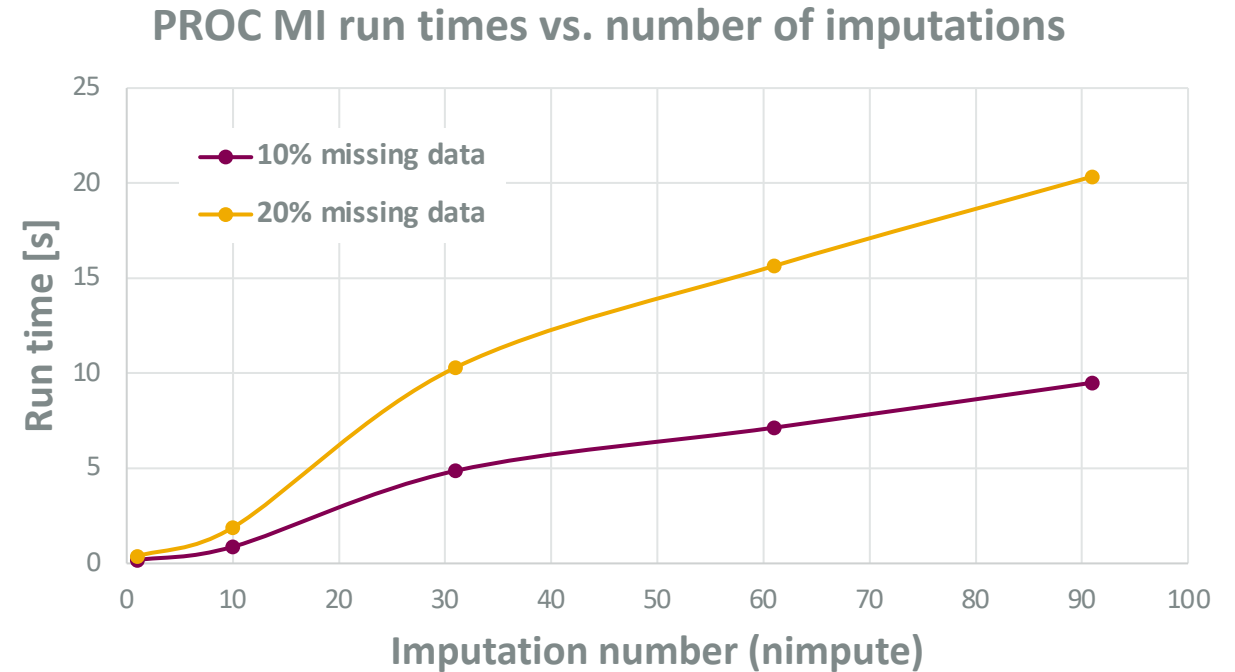


Figure 5. PROC MI total run times for input datasets with 10% and 20% of missing data, for a different number of imputations nimpute: 1, 10, 31, 61 and 91.



Conclusions

1. Multiple imputation is an effective method of dealing with missing values;
2. High degree of customization available in SAS PROC MI enables adjusting multiple imputation process to specific mechanism of missing data;
3. From performance perspective it may be recommended to define nimpute based on data rather than assume upfront:
 - From experience, nimpute=100 is often used, however, there may be no need to use as many as 100 imputations.
Our example shows satisfactory results with only 31 imputations.
 - The built-in PROC MI option nimpute=PCTMISSING uses the percentage of incomplete cases as the number of imputations. In this example, nimpute=PCTMISSING proposed 50 imputations;



Conclusions

4. Generally, the higher number of imputations, the more distribution of imputed dataset resembles the original distribution. However, reducing to acceptable minimum number of imputations will greatly reduce the time needed for further analyses such as for example PROC MIXED;
5. The performance of the PROC MI was affected by both, number of imputations and changes in the percentage of missing data. For 10% missing values, the average run time per 1 imputation was approx. 0.14s and for 20% of missing values it was 0.28s.

However, run times are still within seconds which shows very good PROC MI performance.

From the performance perspective, PROC MI is not a crucial factor affecting the whole data analyses process.

