
Missing Data

Missing Data Methods in ML

Multiple Imputation

PSYC 943: Fundamentals
of Multivariate Modeling
Lecture 13: November 20, 2013

Today's Lecture

- The basics of missing data:
 - Types of missing data
- How NOT to handle missing data
 - Deletion methods (both pairwise and listwise)
 - Mean-substitution
 - Single Imputation
- How maximum likelihood works with missing data
- Multiple imputation for missing data
 - How imputation works
 - How to conduct analyses with missing data using imputation

Example Data #1

- To demonstrate some of the ideas of types of missing data, let's consider a situation where you have collected two variables:
 - IQ scores
 - Job performance
- Imagine you are an employer looking to hire employees for a job where IQ is important

Complete Data
From Enders (2010)

<u>IQ</u>	<u>Performance</u>
78	9
84	13
84	10
85	8
87	7
91	7
92	9
94	9
94	11
96	7
99	7
105	10
105	11
106	15
108	10
112	10
113	12
115	14
118	16
134	12

TYPES OF MISSING DATA

Our Notational Setup

- Let's let **D** denote our data matrix, which will include dependent (**Y**) and independent (**X**) variables

$$\mathbf{D} = \{\mathbf{X}, \mathbf{Y}\}$$

- Problem:** some elements of **D** are missing

Missingness Indicator Variables

- We can construct an alternate matrix **M** consisting of indicators of missingness for each element in our data matrix **D**

$M_{ij} = 0$ if the i^{th} observation's j^{th} variable is **not** missing

$M_{ij} = 1$ if the i^{th} observation's j^{th} variable is missing

- Let \mathbf{M}_{obs} and \mathbf{M}_{mis} denote the observed and missing parts of \mathbf{M}
$$\mathbf{M} = \{\mathbf{M}_{obs}, \mathbf{M}_{mis}\}$$

Types of Missing Data

- A very rough typology of missing data puts missing observations into three categories:
 1. Missing Completely At Random (MCAR)
 2. Missing At Random (MAR)
 3. Missing Not At Random (MNAR)

Missing Completely At Random (MCAR)

- Missing data are MCAR if the events that lead to missingness are independent of:
 - The observed variables
 - and-*
 - The unobserved parameters of interest
- Examples:
 - Planned missingness in survey research
 - ♦ Some large-scale tests are sampled using booklets
 - ♦ Students receive only a few of the total number of items
 - ♦ The items not received are treated as missing – but that is completely a function of sampling and no other mechanism

A (More) Formal MCAR Definition

- Our missing data indicators, \mathbf{M} are **statistically independent** of our observed data \mathbf{D}

$$P(\mathbf{M}|\mathbf{D}) = P(\mathbf{M})$$

this comes from how independence works with pdfs

- Like saying a missing observation is due to pure randomness (i.e., flipping a coin)

Implications of MCAR

- Because the mechanism of missing is not due to anything other than chance, inclusion of MCAR in data will not bias your results
 - Can use methods based on listwise deletion, multiple imputation, or maximum likelihood
- Your effective sample size is lowered, though
 - Less power, less efficiency

<u>IQ</u>	<u>Performance</u>
78	-
84	13
84	-
85	8
87	7
91	7
92	9
94	9
94	11
96	-
99	7
105	10
105	11
106	15
108	10
112	-
113	12
115	14
118	16
134	-

MCAR Data

Missing data are dispersed randomly throughout data

Mean IQ of complete cases: 99.7

Mean IQ of incomplete cases: 100.8

Missing At Random (MAR)

- Data are MAR if the probability of missing depends **only** on some (or all) of the observed data

- \mathbf{M} is independent of \mathbf{D}_{mis}

$$P(\mathbf{M}|\mathbf{D}) = P(\mathbf{M}|\mathbf{D}_{obs})$$

<u>IQ</u>	<u>Perf</u>	<u>Indicator</u>
78	-	1
84	-	1
84	-	1
85	-	1
87	-	1
91	7	0
92	9	0
94	9	0
94	11	0
96	7	0
99	7	0
105	10	0
105	11	0
106	15	0
108	10	0
112	10	0
113	12	0
115	14	0
118	16	0
134	12	0

MAR Data

Missing data are related to other data:

Any IQ less than 90 did not have a performance variable

Mean IQ of incomplete cases: 83.6

Mean IQ of complete cases: 105.5

Implications of MAR

- If data are missing at random, biased results could occur
- Inferences based on listwise deletion will be biased and inefficient
 - Fewer data points = more error in analysis
- Inferences based on maximum likelihood will be unbiased but inefficient
- We will focus on methods for MAR data today

Missing Not At Random (MNAR)

- Data are MNAR if the probability of missing data is related to values of the variable itself

$$P(\mathbf{M}|\mathbf{D}) = P(\mathbf{M}|\mathbf{D}_{obs}, \mathbf{D}_{mis})$$

- Often called non-ignorable missingness
 - Inferences based on listwise deletion or maximum likelihood will be biased and inefficient
- Need to provide statistical model for missing data simultaneously with estimation of original model

SURVIVING MISSING DATA: A BRIEF GUIDE

Using Statistical Methods with Missing Data

- Missing data can alter your analysis results dramatically depending upon:
 1. The type of missing data
 2. The type of analysis algorithm
- The choice of an algorithm and missing data method is important in avoiding issues due to missing data

The Worst Case Scenario: MNAR

- The worst case scenario is when data are MNAR: missing not at random
 - Non-ignorable missing
- You cannot easily get out of this mess
 - Instead you have to be clairvoyant
- Analyses algorithms must incorporate models for missing data
 - And these models must also be right

The Reality

- In most empirical studies, MNAR as a condition is an afterthought
- It is impossible to know definitively if data truly are MNAR
 - So data are treated as MAR or MCAR
- Hypothesis tests do exist for MCAR
 - Although they have some issues

The Best Case Scenario: MCAR

- Under MCAR, pretty much anything you do with your data will give you the “right” (unbiased) estimates of your model parameters
- MCAR is very unlikely to occur
 - In practice, MCAR is treated as equally unlikely as MNAR

The Middle Ground: MAR

- MAR is the common compromise used in most empirical research
 - Under MAR, maximum likelihood algorithms are unbiased
- Maximum likelihood is for many methods:
 - Linear mixed models in PROC MIXED
 - Models with “latent” random effects (CFA/SEM models) in Mplus

MISSING DATA IN MAXIMUM LIKELIHOOD

Missing Data with Maximum Likelihood

- Handling missing data in maximum likelihood is much more straightforward due to the calculation of the log-likelihood function
 - Each subject contributes a portion due to their observations
- If some of the data are missing, the log-likelihood function uses a reduced form of the MVN distribution
 - Capitalizing on the property of the MVN that subsets of variables from an MVN distribution are also MVN
- The total log-likelihood is then maximized
 - Missing data just are “skipped” – they do not contribute

Each Person's Contribution to the Log-Likelihood

- For a person p , the MVN log-likelihood can be written:

$$\log L_p = -\frac{V}{2}\log(2\pi) - \frac{1}{2}\log(|\mathbf{\Sigma}_p|) - \frac{(\mathbf{y}_p - \boldsymbol{\mu}_p)^T \mathbf{\Sigma}_p^{-1} (\mathbf{y}_p - \boldsymbol{\mu}_p)}{2}$$

- From our examples with missing data, subjects could either have all of their data...so their input into $\log L_p$ uses:

$$\begin{aligned}\mathbf{y}_p &= \begin{bmatrix} y_{p,IQ} \\ y_{p,Perf} \end{bmatrix}; \\ \boldsymbol{\mu}_p = \mathbf{X}_p \boldsymbol{\beta} &= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} \mu_{IQ} \\ \mu_{Perf} \end{bmatrix}; \\ \mathbf{\Sigma}_p &= \begin{bmatrix} \sigma_{IQ}^2 & \sigma_{IQ,Perf} \\ \sigma_{IQ,Perf} & \sigma_{Perf}^2 \end{bmatrix}\end{aligned}$$

- ...or could be missing the performance variable, yielding:

$$\mathbf{y}_p = [y_{p,IQ}]; \boldsymbol{\mu}_p = \mathbf{X}_p \boldsymbol{\beta} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = [\beta_0 + \beta_1] = [\mu_{IQ}]; \mathbf{\Sigma}_p = [\sigma_{IQ}^2]$$

Evaluation of Missing Data in PROC MIXED (and pretty much all other packages)

- If the dependent variables are missing, PROC MIXED automatically skips those variables in the likelihood
 - The REPEATED statement specifies observations with the same subject ID – and uses the non-missing observations from that subject only
- If independent variables are missing, however, PROC MIXED uses listwise deletion
 - If you have missing IVs, this is a problem
 - You can sometimes phrase IVs as DVs, though
- SAS Syntax (identical to when you have complete data):

```
*EMPTY MODEL: MCAR Data;  
PROC MIXED DATA=WORK.jobstackMCAR METHOD=ML COVTEST NOPROFILE ITDETAILS IC;  
CLASS variable;  
MODEL value = variable / S;  
REPEATED / SUBJECT=ID TYPE=UN R=1,2 RCORR;  
RUN;
```

Analysis of MCAR Data with PROC MIXED

- Covariance matrices from slide #4 (MIXED is closer to complete):

MCAR Data (Pairwise Deletion)		
IQ	115.6	19.4
Performance	19.4	8.0

Complete Data		
IQ	189.6	19.5
Performance	19.5	6.8

- Estimated **R** matrix from PROC MIXED:

Covariance Parameter Estimates					
Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z
UN(1,1)	ID	189.60	59.9557	3.16	0.0008
UN(2,1)	ID	31.7352	14.0984	2.25	0.0244
UN(2,2)	ID	10.0446	4.0984	2.45	0.0071

- Output for each observation (obs #1 = missing, obs #2 = complete):

Estimated R Matrix for Subject 1			Estimated R Matrix for Subject 2		
Row	Col1		Row	Col1	Col2
1	189.60		1	189.60	31.7352
			2	31.7352	10.0446

MCAR Analysis: Estimated Fixed Effects

- Estimated mean vectors:

Variable	MCAR Data (pairwise deletion)	Complete Data
IQ	93.73	100
Performance	10.6	10.35

- Estimated fixed effects:

Solution for Fixed Effects						
Effect	variable	Estimate	Standard Error	DF	t Value	Pr > t
Intercept		10.6446	0.7623	19	13.96	<.0001
variable	IQ	89.3554	2.6244	19	34.05	<.0001
variable	Performance MCAR	0

- Means – IQ = 89.36+10.64 = 100; Performance = 10.64

Analysis of MAR Data with PROC MIXED

- Covariance matrices from slide #4 (MIXED is closer to complete):

MAR Data (Pairwise Deletion)		
IQ	130.2	19.5
Performance	19.5	7.3

Complete Data		
IQ	189.6	19.5
Performance	19.5	6.8

- Estimated **R** matrix from PROC MIXED:

Covariance Parameter Estimates					
Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr > Z
UN(1,1)	ID	189.60	59.9567	3.16	0.0008
UN(2,1)	ID	28.3696	12.6862	2.24	0.0253
UN(2,2)	ID	8.6176	3.3995	2.53	0.0056

- Output for each observation (obs #1 = missing, obs #10 = complete):

Estimated R Matrix for Subject 1	
Row	Col1
1	189.60

Estimated R Matrix for Subject 10		
Row	Col1	Col2
1	189.60	28.3696
2	28.3696	8.6176

MAR Analysis: Estimated Fixed Effects

- Estimated mean vectors:

Variable	MCAR Data (pairwise deletion)	Complete Data
IQ	105.4	100
Performance	10.7	10.35

- Estimated fixed effects:

Solution for Fixed Effects						
Effect	variable	Estimate	Standard Error	DF	t Value	Pr > t
Intercept		9.8487	0.7098	19	13.88	<.0001
variable	IQ	90.1513	2.6734	19	33.72	<.0001
variable	Performance MAR	0

- Means – $IQ = 90.15 + 9.85 = 100$; $Performance = 9.85$

Additional Issues with Missing Data and Maximum Likelihood

- Given the structure of the missing data, the standard errors of the estimated parameters may be computed differently
 - Standard errors come from $-1 \times \text{inverse information matrix}$
 - ◆ Information matrix = matrix of second derivatives = hessian
- Several versions of this matrix exist
 - Some based on what is expected under the model
 - ◆ The default in SAS – good only for MCAR data
 - Some based on what is observed from the data
 - ◆ Empirical option in SAS – works for MAR data (only for fixed effects)
- Implication: some SEs may be biased if data are MAR
 - May lead to incorrect hypothesis test results
 - Correction needed for likelihood ratio/deviance test statistics
 - ◆ Not available in SAS; available for some models in Mplus

When ML Goes Bad...

- For linear models with missing **dependent variable(s)** PROC MIXED and almost every other stat package works great
 - ML “skips” over the missing DVs in the likelihood function, using only the data you have observed
- For linear models with missing **independent variable(s)**, PROC MIXED and almost every other stat package uses list-wise deletion
 - Gives biased parameter estimates under MAR

Options for MAR for Linear Models with Missing Independent Variables

1. **Use ML Estimators and hope for MCAR**
2. **Rephrase IVs as DVs**
 - In SAS: hard to do, but possible for some models
 - ◆ Dummy coding, correlated random effects
 - ◆ Rely on properties of how correlations/covariances are related to linear model coefficients β
 - In Mplus: much easier...looks more like a structural equation model
 - ◆ Predicted variables then function like DVs in MIXED
3. **Impute IVs (multiple times) and then use ML Estimators**
 - Not usually a great idea...but often the only option

ANOTHER EXAMPLE DATA SET

Today's Example Data #2

- Three variables were collected from a sample of 31 men in a course at NC State
 - **Oxygen:** oxygen intake, ml per kg body weight, per minute
 - **Runtime:** time to run 1.5 miles in minutes
 - **Runpulse:** heart rate while running
- The research question: how does oxygen intake vary as a function of exertion (running time and running heart rate)
- The problem: some of the data are missing

Descriptive Statistics of Missing Data

- Descriptive statistics of our data:

The MEANS Procedure			
Variable	Mean	Std Dev	N
Oxygen	47.1161786	5.4130470	28
RunTime	10.6882143	1.3798794	28
RunPulse	171.8636364	10.1432382	22

- Patterns of missing data:

The FREQ Procedure				
MissingPattern	Frequency	Percent	Cumulative Frequency	Cumulative Percent
None Missing	21	67.74	21	67.74
Pulse Missing	4	12.90	25	80.65
Time and Pulse Missing	3	9.68	28	90.32
Oxygen Missing	1	3.23	29	93.55
Oxygen and Pulse Missing	2	6.45	31	100.00

Comparing Missing and Not Missing

OxygenMiss=Not Missing

Oxygen

The MEANS Procedure

Variable	Mean	Std Dev	N
Oxygen	47.1161786	5.4130470	28
RunTime	10.7020000	1.3943368	25
RunPulse	171.6666667	10.3505233	21

OxygenMiss=Missing

Variable	Mean	Std Dev	N
Oxygen	.	.	0
RunTime	10.5733333	1.5338296	3
RunPulse	176.0000000	.	1

RunTimeMiss=Not Missing

Running Time

The MEANS Procedure

Variable	Mean	Std Dev	N
Oxygen	46.4747200	5.0578561	25
RunTime	10.6882143	1.3798794	28
RunPulse	171.8636364	10.1432382	22

RunTimeMiss=Missing

Variable	Mean	Std Dev	N
Oxygen	52.4616667	6.3700017	3
RunTime	.	.	0
RunPulse	.	.	0

RunPulseMiss=Not Missing

Pulse Rate

The MEANS Procedure

Variable	Mean	Std Dev	N
Oxygen	46.3538095	5.4778395	21
RunTime	10.8613636	1.4576997	22
RunPulse	171.8636364	10.1432382	22

RunPulseMiss=Missing

Variable	Mean	Std Dev	N
Oxygen	49.4032857	4.8678064	7
RunTime	10.0533333	0.8612936	6
RunPulse	.	.	0

HOW NOT TO HANDLE MISSING DATA

Bad Ways to Handle Missing Data

- Dealing with missing data is important, as the mechanisms you choose can dramatically alter your results
- This point was not fully realized when the first methods for missing data were created
 - Each of the methods described in this section should **never be used**
 - Given to show perspective – and to allow you to understand what happens if you were to choose each

Deletion Methods

- Deletion methods are just that: methods that handle missing data by deleting observations
 - Listwise deletion: delete the entire observation if any values are missing
 - Pairwise deletion: delete a pair of observations if either of the values are missing
- Assumptions: Data are MCAR
- Limitations:
 - Reduction in statistical power if MCAR
 - Biased estimates if MAR or MNAR

Listwise Deletion

- Listwise deletion discards **all** of the data from an observation if one or more variables are missing
- Most frequently used in statistical software packages that are not optimizing a likelihood function (need ML)
- In linear models:
 - SAS GLM list-wise deletes cases where **IVs** or **DVs** are missing

Listwise Deletion Example

- If you wanted to predict Oxygen from Running Time and Pulse Rate you could:

- Start with one variable (running time):

Dependent Variable: Oxygen					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	442.6707707	442.6707707	59.44	<.0001
Error	23	171.2950243	7.4476098		
Corrected Total	24	613.9657950			

- Then add the other (running time + pulse rate):

Dependent Variable: Oxygen					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	449.4733700	224.7366850	26.85	<.0001
Error	18	150.6611373	8.3700632		
Corrected Total	20	600.1345072			

- The nested-model comparison test cannot be formed
 - Degrees of freedom error changes as missing values are omitted

Pairwise Deletion

- Pairwise deletion discards a pair of observations if either one is missing
 - Different from listwise: uses more data (rest of data not thrown out)
- Assumes: MCAR
- Limitations:
 - Reduction in statistical power if MCAR
 - Biased estimates if MAR or MNAR
- Can be an issue when forming covariance/correlation matrices
 - May make them non-invertible, problem if used as input into statistical procedures

Pairwise Deletion Example

- Covariance Matrix from PROC CORR (see the different DF):

```

3 Variables:      Oxygen      RunTime      RunPulse

              Variances and Covariances
Covariance / Row Var  Variance / Col Var  Variance / DF

              Oxygen              RunTime              RunPulse

Oxygen
29.3010776      -5.9882853      -19.5021167
29.3010776      25.5819081      30.0067254
29.3010776      1.9441750      107.1333333
              27              24              20

RunTime
-5.9882853      1.9040671      3.6559091
1.9441750      1.9040671      2.1248885
25.5819081      1.9040671      102.8852814
              24              27              21

RunPulse
-19.5021167      3.6559091      102.8852814
107.1333333      102.8852814      102.8852814
30.0067254      2.1248885      102.8852814
              20              21              21

```

Single Imputation Methods

- **Single imputation** methods replace missing data with some type of value
 - **Single**: one value used
 - **Imputation**: replace missing data with value
- Upside: can use entire data set if missing values are replaced
- Downside: biased parameter estimates and standard errors (even if missing is MCAR)
 - Type-I error issues
- Still: never use these techniques

Unconditional Mean Imputation

- Unconditional mean imputation replaces the missing values of a variable with its estimated mean
 - Unconditional = mean value without any input from other variables
- Example: missing Oxygen = 47.1; missing RunTime = 10.7;
missing RunPulse = 171.9

Before Single Imputation:

The MEANS Procedure

Variable	Mean	Std Dev	N
Oxygen	47.1161786	5.4130470	28
RunTime	10.6882143	1.3798794	28
RunPulse	171.8636364	10.1432382	22

After Single Imputation:

The MEANS Procedure

Variable	Mean	Std Dev	N
Oxygen	47.1146129	5.1352696	31
RunTime	10.6893548	1.3090733	31
RunPulse	171.8741935	8.4864585	31

- Notice: uniformly smaller standard deviations

Conditional Mean Imputation (Regression)

- Conditional mean imputation uses regression analyses to impute missing values
 - The missing values are imputed using the predicted values in each regression (conditional means)
- For our data we would form regressions for each outcome using the other variables
 - $\text{OXYGEN} = \beta_{01} + \beta_{11} * \text{RUNTIME} + \beta_{21} * \text{PULSE}$
 - $\text{RUNTIME} = \beta_{02} + \beta_{12} * \text{OXYGEN} + \beta_{22} * \text{PULSE}$
 - $\text{PULSE} = \beta_{03} + \beta_{13} * \text{OXYGEN} + \beta_{23} * \text{RUNTIME}$
- More accurate than unconditional mean imputation
 - But still provides biased parameters and SEs

Stochastic Conditional Mean Imputation

- Stochastic conditional mean imputation adds a random component to the imputation
 - Representing the error term in each regression equation
 - Assumes MAR rather than MCAR
- Again, uses regression analyses to impute data:
 - $\text{OXYGEN} = \beta_{01} + \beta_{11} * \text{RUNTIME} + \beta_{21} * \text{PULSE} + \text{Error}$
 - $\text{RUNTIME} = \beta_{02} + \beta_{12} * \text{OXYGEN} + \beta_{22} * \text{PULSE} + \text{Error}$
 - $\text{PULSE} = \beta_{03} + \beta_{13} * \text{OXYGEN} + \beta_{23} * \text{RUNTIME} + \text{Error}$
- **Error** is random: drawn from a normal distribution
 - Zero mean and variance equal to residual variance σ_e^2 for respective regression

Imputation by Proximity: Hot Deck Matching

- Hot deck matching uses real data – from other observations as its basis for imputing
- Observations are “matched” using similar scores on variables in the data set
 - Imputed values come directly from matched observations
- Upside: Helps to preserve univariate distributions; gives data in an appropriate range
- Downside: biased estimates (especially of regression coefficients), too-small standard errors

Scale Imputation by Averaging

- In psychometric tests, a common method of imputation has been to use a scale average rather than total score
 - Can re-scale to total score by taking # items * average score
- Problem: treating missing items this way is like using person mean
 - Reduces standard errors
 - Makes calculation of reliability biased

Longitudinal Imputation: Last Observation Carried Forward

- A commonly used imputation method in longitudinal data has been to treat observations that dropped out by carrying forward the last observation
 - More common in medical studies and clinical trials
- Assumes scores do not change after dropout – bad idea
 - Thought to be conservative
- Can exaggerate group differences
 - Limits standard errors that help detect group differences

Why Single Imputation Is Bad Science

- Overall, the methods described in this section are not useful for handling missing data
- If you use them you will likely get a statistical answer that is an artifact
 - Actual estimates you interpret (parameter estimates) will be biased (in either direction)
 - Standard errors will be too small
 - ♦ Leads to Type-I Errors
- Putting this together: you will likely end up making conclusions about your data that are wrong

WHAT TO DO WHEN ML WON'T GO: MULTIPLE IMPUTATION

Multiple Imputation

- Rather than using single imputation, a better method is to use multiple imputation
 - The multiply imputed values will end up adding variability to analyses – helping with biased parameter and SE estimates
- Multiple imputation is a mechanism by which you “fill in” your missing data with “plausible” values
 - End up with multiple data sets – need to run multiple analyses
 - Missing data are predicted using a statistical model using the observed data (the MAR assumption) for each observation
- MI is possible due to statistical assumptions
 - The most often used assumption is that the observed data are multivariate normal
 - We will focus on this today – and expand upon it on Friday

Multiple Imputation Steps

1. The missing data are filled in a number of times (say, m times) to generate m complete data sets
2. The m complete data sets are analyzed using standard statistical analyses
3. The results from the m complete data sets are combined to produce inferential results

Distributions: The Key to Multiple Imputation

- The key idea behind multiple imputation is that each missing value has a **distribution** of likely values
 - The distribution reflects the uncertainty about what the variable may have been
- Multiple imputation can be accomplished using variables outside an analysis
 - All contribute to multivariate normal distribution
 - Harder to justify why un-important variables omitted
- Single imputation, by any method, disregards the uncertainty in each missing data point
 - Results from singly imputed data sets may be biased or have higher Type-I errors

Multiple Imputation in SAS

- SAS has a pair of procedures for multiple imputation:
 - PROC MI: generates multiple complete data sets
 - PROC MIANALYZE: analyzes the results of statistical analyses with imputed data sets
- Most frequent assumption SAS uses is that data are multivariate normal
- Not MVN? Mplus provides imputation options

IMPUTATION PHASE

SAS PROC MI

- PROC MI uses a variety of methods depending on the type of missing data present
 - Monotone missing pattern: ordered missingness – if you order your variables sequentially, only the tail end of the variables collected is missing
 - ◆ Multiple methods exist for imputation
 - Arbitrary missing pattern: missing data follow no pattern
 - ◆ Most typical in data
 - ◆ Markov Chain Monte Carlo assuming MVN is used

Multivariate Normal Data

- The MVN distribution has several nice properties
- In SAS PROC MI, multiple imputation of arbitrary missing data takes advantage of the MVN properties
- Imagine we have N observations of V variables from a MVN:

$$\mathbf{Y}_{(N \times V)} \sim N_V(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

- The property we will use is the conditional distribution of MVN variables
 - We will examine the conditional distribution of missing data given the data we have observed

Conditional Distributions of MVN Variables

- The conditional distribution of sets of variables from a MVN is also MVN
 - Used as the data-generating distribution in PROC MI
- If we were interested in the distribution of the first q variables, we partition three matrices:
 - The data: $[Y_{(N \times q)} \quad X_{(N \times V-q)}]$
 - The mean vector: $\begin{bmatrix} \mu_{Y:(q \times 1)} \\ \mu_{X:(V-q \times 1)} \end{bmatrix}$
 - The covariance matrix: $\begin{bmatrix} \Sigma_{YY:(q \times q)} & \Sigma_{YX:(q \times V-q)} \\ \Sigma_{XY:(V-q \times q)} & \Sigma_{XX:(V-q \times V-q)} \end{bmatrix}$

Conditional Distributions of MVN Variables

- The conditional distribution of Y given the values of $X = x$ is then:

$$Y|X \sim N_q(\mu^*, \Sigma^*)$$

Where (using our partitioned matrices):

$$\mu^* = \mu_Y + \Sigma_{YX}\Sigma_{XX}^{-1}(x' - \mu_X)$$

And:

$$\Sigma^* = \Sigma_{YY} - \Sigma_{YX}\Sigma_{XX}^{-1}\Sigma_{XY}$$

Example from our Data

- From estimates with missing data:

$$\bar{\mathbf{y}} = \begin{bmatrix} 47.1 \\ 10.7 \\ 171.9 \end{bmatrix}; \mathbf{S} = \begin{bmatrix} 29.3 & -6.0 & -19.5 \\ -6.0 & 1.9 & 3.7 \\ -19.5 & 3.7 & 102.9 \end{bmatrix}$$

- For observation #4 (missing oxygen): $\mathbf{x} = [11.96 \quad 176]$
 - We wish to impute the first observation (oxygen) conditional on the values of runtime and pulse

- Assuming MVN, we get the following sub-matrices:

$$\begin{aligned} \bar{\mathbf{x}}_Y &= [47.1]; \bar{\mathbf{x}}_X = \begin{bmatrix} 10.7 \\ 171.9 \end{bmatrix} \\ \mathbf{S}_{YY} &= [29.3]; \mathbf{S}_{YX} = [-6.0 \quad -19.5]; \\ \mathbf{S}_{XY} &= \begin{bmatrix} -6.0 \\ -19.5 \end{bmatrix}; \mathbf{S}_{XX} = \begin{bmatrix} 1.9 & 3.7 \\ 3.7 & 102.9 \end{bmatrix}; \mathbf{S}_{XX}^{-1} = \begin{bmatrix} .56 & -.02 \\ -.02 & .01 \end{bmatrix} \end{aligned}$$

Imputation Distribution

- The imputed value for Oxygen for observation #4 is drawn from a $N_1(43.0, 9.8)$:

$$\begin{aligned}\text{Mean: } \bar{\mathbf{y}}^* &= \bar{\mathbf{x}}_Y + \mathbf{S}_{YX} \mathbf{S}_{XX}^{-1} (\mathbf{x}' - \bar{\mathbf{x}}_X) = [47.1] + \\ & [-6.0 \quad -19.5] \begin{bmatrix} .56 & -.02 \\ -.02 & .01 \end{bmatrix} \left(\begin{bmatrix} 11.96 \\ 176 \end{bmatrix} - \begin{bmatrix} 10.7 \\ 171.9 \end{bmatrix} \right) \\ &= 43.0\end{aligned}$$

Variance:

$$\begin{aligned}\mathbf{S}^* &= \mathbf{S}_{YY} - \mathbf{S}_{YX} \mathbf{S}_{XX}^{-1} \mathbf{S}_{XY} \\ &= [29.3] - [-6.0 \quad -19.5] \begin{bmatrix} .56 & -.02 \\ -.02 & .01 \end{bmatrix} \begin{bmatrix} -6.0 \\ -19.5 \end{bmatrix} \\ &= 9.8\end{aligned}$$

Using the MVN for Missing Data

- If we consider our missing data to be Y , we can then use the result from the last slide to generate imputed (plausible) values for our missing data
- Data generated from a MVN distribution is fairly common and “easy” to do computationally
- However....

The Problem: True μ and Σ are Unknown

- Problem: the true mean vector and covariance matrix for our data is unknown
 - We only have sample estimates
 - ◆ Sample estimates have sampling error
 - The mean vector has a MVN distribution
 - The sample covariance matrix has a (scaled) Wishart distribution
 - Missing data complicate the situation by providing even fewer observations to estimate either parameter
- The example from before used one estimate (but that is unlikely to be correct)
 - It used pairwise deletion

The PROC MI Solution

- PROC MI: use MCMC to estimate data and parameters simultaneously:

Step 0: Create starting value estimates for μ and Σ : ($\mu_{t-1=0}, \Sigma_{t-1=0}$)

Iterate t times through:

Step 1: Using μ_{t-1}, Σ_{t-1} generate the missing data from the conditional MVN (conditional on the observed values for each case)

Step 2: Using the imputed and observed data, draw a new μ_t, Σ_t from the MVN and Wishart distributions, respectively

The Process of Imputation

- The iterations take “a while” to reach a steady state – stable values for the distribution of μ_t, Σ_t
 - The burn in period
- After this period, you can take sets of imputed data to be used in your multiple analyses
 - The sets should be taken with “enough” iterations in between so as to not be highly correlated
 - ◆ The thinning interval

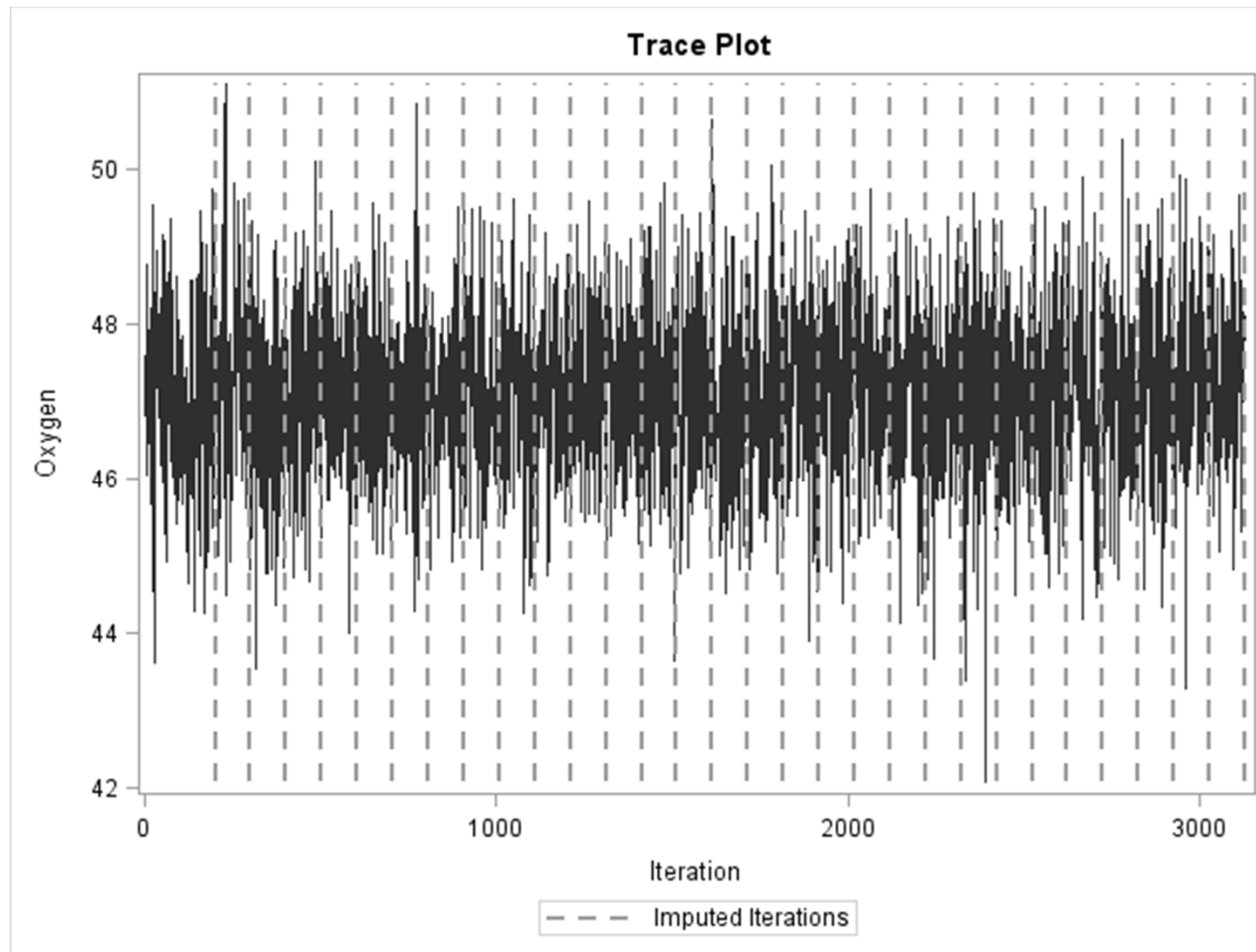
Using PROC MI

- PROC MI Syntax:

```
*IMPUTATION PHASE;;  
*USING PROC MI TO IMPUTE DATA;;  
PROC MI DATA=WORK.fitmiss OUT=WORK.fitimpute NIMPUTE=30 SEED=10292012;  
MCMC CHAIN=SINGLE DISPLAYINIT INITIAL=EM(ITPRINT) PLOTS=ALL  
      OUTITER=WORK.outiter OUTEST=WORK.outest;  
VAR oxygen runtime runpulse;  
RUN;
```

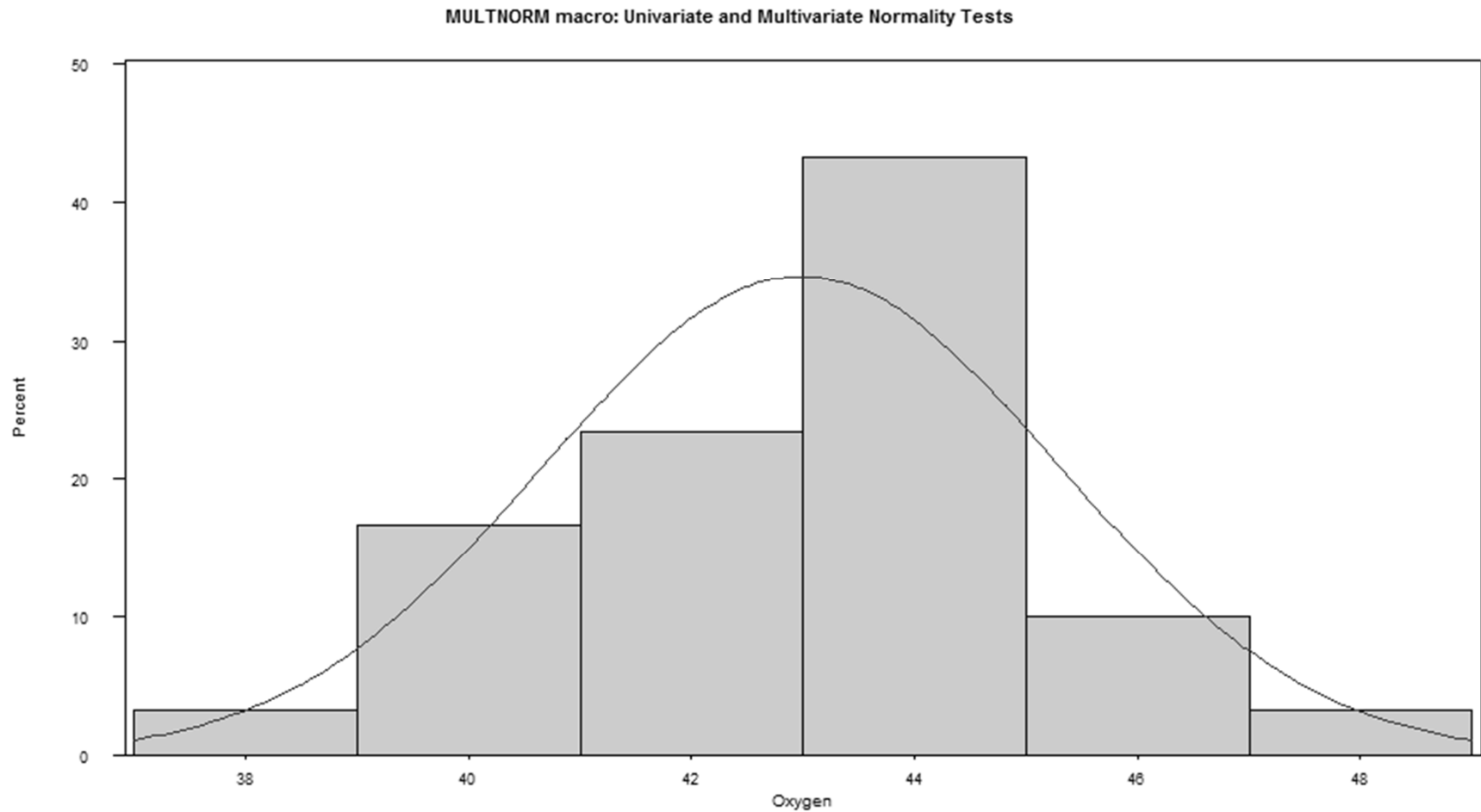
- More often than not, the output of MI does not have much useful information
 - Must assume convergence of mean vector and covariance matrix – but limited statistics to check convergence
- Of interest is the new data set (WORK.fitimpute)
 - Here it contains 30 imputations for each missing variable
 - ◆ Need to run the regression 30 times – Analysis and Pooling Phase

MCMC Trace Plots – Use for Checking Convergence



Inspecting Imputed Values

- To demonstrate the imputed values, look at the histogram of the 30 values for observation 4:



Resulting Data Sets

- The new data sets are all stacked on top of each other
- Analyses now must add a line that says BY so each new data set has its own analysis

	Imputation Number	Oxygen	RunTime	RunPulse	MissingPattern
1	1	44.609	11.37	178	0
2	1	54.297	8.65	156	0
3	1	49.874	9.22	177.95495543	1
4	1	42.352679775	11.95	176	4
5	1	39.442	13.08	174	0
6	1	50.541	9.4485552222	177.07803459	3
7	1	44.754	11.12	176	0
8	1	51.855	10.33	166	0
9	1	40.836	10.95	168	0
10	1	46.774	10.25	166.09209796	1
11	1	39.407	12.63	174	0
12	1	45.441	9.63	164	0
13	1	45.118	11.08	179.26160327	1
14	1	45.79	10.47	186	0
15	1	48.673	9.4	186	0
16	1	47.467	10.5	170	0
17	1	45.313	10.07	185	0
18	1	59.571	7.704924026	172.89225626	3
19	1	44.811	11.63	176	0
20	1	44.2901159	10.85	177.48176509	5
21	1	60.055	8.63	170	0
22	1	37.388	14.03	186	0
23	1	47.273	9.7465552053	162.99703576	3
24	1	49.156	8.95	180	0
25	1	46.672	10	183.28206872	1
26	1	50.388	10.08	168	0
27	1	46.08	11.17	156	0
28	1	55.656883116	8.92	149.36487646	5
29	1	39.203	12.88	168	0
30	1	50.545	9.93	148	0
31	1	47.92	11.5	170	0
32	2	44.609	11.37	178	0
33	2	54.297	8.65	156	0

MULTIPLE IMPUTATION: ANALYSIS PHASE

Up Next: Multiple Analyses

- Once you run PROC MI, the next step is to use each of the imputed data sets in its own analysis
 - Called the analysis phase
 - For our example, that would be 30 times
- The multiple analyses are then compiled and processed into a single result
 - Yielding the answers to your analysis questions (estimates, SEs, and P-values)
- GOOD NEWS: SAS will automate all of this for you

Analysis Phase

- Analysis Phase: run the analysis on all imputed data sets

```
*ANALYSIS PHASE;;  
PROC MIXED DATA=WORK.fitimpute METHOD=ML COVTEST NOPROFILE ITDETAILS IC ASYCOV;  
BY _IMPUTATION_;  
MODEL oxygen = runtime runpulse / SOLUTION COVB;  
ODS OUTPUT SolutionF=WORK.FixedEffects CovB=WORK.CovMatrices;  
RUN;
```

- Syntax runs for each data set (BY _IMPUTATION_)
- The ODS OUTPUT line saves information needed in the pooling phase:
 - Parameter estimates (to make parameter estimates)
 - ♦ SolutionF=WORK.fixedeffects
 - Asymptotic covariance matrix of the fixed effects $(\mathbf{X}^T \mathbf{V}^{-1} \mathbf{X})^{-1}$
 - ♦ CovB=WORK.CovMatrices

Saving Information from Other SAS PROCs

- Because of the various number of PROC types SAS implements, there are a variety of difference commands you must use if you are using Multiple Imputation in SAS
- The SAS User's Group document by Yuan posted on our website outlines the varying ways to do so
 - Although, some will not work without a reference to the SAS 9.3 manual

MULTIPLE IMPUTATION: POOLING PHASE

Pooling Parameters from Analyses of Imputed Data Sets

- In the pooling phase, the results are pooled and reported
- For parameter estimates, the pooling is straight forward
 - The estimated parameter is the average parameter value across all imputed data sets
 - ♦ For our example the average intercept, slope for runtime, and slope for runpulse are taken over the 30 imputed data sets and analyses
- For standard errors, pooling is more complicated
 - Have to worry about sources of variation:
 - ♦ Variation from sampling error that would have been present had the data not been missing
 - ♦ Variation from sampling error resulting from missing data

Pooling Standard Errors Across Imputation Analyses

- Standard error information comes from two sources of variation from imputation analyses (for m imputations)

- Within Imputation Variation:

$$V_W = \frac{1}{m} \sum_{i=1}^m SE_i^2$$

- Between Imputation Variation (here θ is an estimated parameter from an imputation analysis):

$$V_B = \frac{1}{m-1} \sum_{i=1}^m (\hat{\theta}_i - \bar{\theta})^2$$

- Then, the total sampling variance is: $V_T = V_W + V_B + \frac{V_B}{M}$
- The subsequent (imputation pooled) SE is $SE = \sqrt{V_T}$

Pooling Phase in SAS: PROC MIANALYZE

- SAS PROC MIANALYZE conducts the pooling phase of imputations: no calculations are needed

```
*POOLING PHASE;;  
PROC MIANALYZE PARMS=WORK.fixedeffects CovB(EFFECTVAR=ROWCOL)=Work.CovMatrices EDF=28;  
MODELEFFECTS Intercept RunTime RunPulse;  
RUN;
```

- The parameter data set, the asymptotic covariance matrix dataset, and the number of error degrees of freedom are all input
- The MODELEFFECTS line combs through the input data and conducts the pooling
- NOTE: different PROC lines have different input values. SEE:
http://support.sas.com/documentation/cdl/en/statug/63962/HTML/default/viewer.htm#mianalyze_toc.htm

PROC MIANALYZE OUTPUT

Variance Information				
Parameter	-----Variance-----			DF
	Between	Within	Total	
Intercept	24.841980	67.878685	93.548731	18.112
RunTime	0.039703	0.117858	0.158884	18.599
RunPulse	0.000808	0.002397	0.003233	18.594

**Variances:
See Next Slides**

Variance Information			
Parameter	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	0.378175	0.278142	0.990814
RunTime	0.348097	0.261601	0.991355
RunPulse	0.348394	0.261768	0.991350

Parameter Estimates – With Hypothesis Test P-Values

Parameter Estimates					
Parameter	Estimate	Std Error	95% Confidence Limits		DF
Intercept	92.129564	9.672059	71.81834	112.4408	18.112
RunTime	-3.055738	0.398603	-3.89124	-2.2202	18.599
RunPulse	-0.074091	0.056855	-0.19327	0.0451	18.594

Parameter Estimates					
Parameter	Minimum	Maximum	Theta0	t for H0: Parameter=Theta0	Pr > t
Intercept	83.042973	101.702192	0	9.53	<.0001
RunTime	-3.409403	-2.709447	0	-7.67	<.0001
RunPulse	-0.132395	-0.003353	0	-1.30	0.2084

Additional Pooling Information

- The decomposition of imputation variance leads to two helpful diagnostic measures about the imputation:

- Fraction of Missing Information: $FMI = \frac{V_B + \frac{V_B}{m}}{V_T}$
 - Measure of influence of missing data on sampling variance
 - Example: intercept = 0.28; runtime = .26; runpulse = .26
 - ~27% of parameters variance attributable to missing data

- Relative Increase in Variance: $RIV = \frac{V_B + \frac{V_B}{m}}{V_W} = \frac{FMI}{1 - FMI}$
 - Another measure of influence of missing data on sampling variance
 - Example: intercept = 0.38; runtime = .35; runpulse = .35

ISSUES WITH IMPUTATION

Common Issues that can Hinder Imputation

- MCMC Convergence
 - Need “stable” mean vector/covariance matrix
- Non-normal data: counts, skewed distributions, categorical (ordinal or nominal) variables
 - Mplus is a good option
 - Some claim it doesn’t matter as much with many imputations
- Preservation of model effects
 - Imputation can strip out effects in data
 - ◆ Interactions are most difficult – form as auxiliary variable
- Imputation of multilevel data
 - Differing covariance matrices

Number of Imputations

- The number of imputations (m from the previous slides) is important: bigger is better
 - Basically, run as many as you can (100s)
- Take a look at the SEs for our parameters as I varied the number of imputations:

Parameter	$m = 1$	$m = 10$	$m = 30$	$m = 100$
Intercept	8.722	9.442	9.672	9.558
RunTime	0.366	0.386	0.399	0.389
RunPulse	0.053	0.053	0.057	0.056

WRAPPING UP

Wrapping Up

- Missing data are common in statistical analyses
- They are frequently neglected
 - MNAR: hard to model missing data and observed data simultaneously
 - MCAR: doesn't often happen
 - MAR: most missing imputation assumes MVN
- More often than not, ML is the best choice
 - Software is getting better at handling missing data
 - We will discuss how ML works next week