Missing Data Methods (Part I): Multiple Imputation

Multivariate Methods in Education
ERSH 8350
Lecture #5 – September 14, 2011
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

• Multiple imputation for missing data
  ➢ Better, but not quite the best (wait until ML next week)

• How imputation works
  ➢ How to conduct analyses with missing data using imputation
TYPES OF MISSING DATA
Our Notational Setup

- Let’s let \( D \) denote our data matrix, which will include dependent (\( Y \)) and independent (\( X \)) variables
  \[ D = \{X, Y\} \]

- **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 $M_{obs}$ and $M_{mis}$ denote the observed and missing parts of $M$

$$M = \{M_{obs}, M_{mis}\}$$
Example Data

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

Complete Data
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
  - 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, $M$ are statistically independent of our observed data $D$

\[ P(M|D) = P(M) \]

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

**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.

- \( M \) is independent of \( D_{mis} \)

\[
P(M|D) = P(M|D_{obs})
\]
### 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

<table>
<thead>
<tr>
<th>IQ</th>
<th>Perf</th>
<th>Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>78</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>84</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>84</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>85</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>87</td>
<td>-</td>
<td>1</td>
</tr>
<tr>
<td>91</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>92</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>94</td>
<td>9</td>
<td>0</td>
</tr>
<tr>
<td>94</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>96</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>99</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>105</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>105</td>
<td>11</td>
<td>0</td>
</tr>
<tr>
<td>106</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>108</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>112</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>113</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>115</td>
<td>14</td>
<td>0</td>
</tr>
<tr>
<td>118</td>
<td>16</td>
<td>0</td>
</tr>
<tr>
<td>134</td>
<td>12</td>
<td>0</td>
</tr>
</tbody>
</table>
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
  \[ P(M|D) = P(M|D_{obs}, D_{mis}) \]

• 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 models in PROC MIXED
  - CFA/SEM models in Mplus
When ML Goes Bad...

- For linear models with missing **dependent variable(s)**
  PROC MIXED 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 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
   - In Mplus: much easier...looks more like a SEM
     - 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
TODAY’S EXAMPLE DATA
Today’s Example Data

• 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:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Std Dev</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxygen</td>
<td>47.1161786</td>
<td>5.4130470</td>
<td>28</td>
</tr>
<tr>
<td>RunTime</td>
<td>10.6882143</td>
<td>1.3798794</td>
<td>28</td>
</tr>
<tr>
<td>RunPulse</td>
<td>171.8636364</td>
<td>10.1432382</td>
<td>22</td>
</tr>
</tbody>
</table>

- Patterns of missing data:

<table>
<thead>
<tr>
<th>MissingPattern</th>
<th>Frequency</th>
<th>Percent</th>
<th>Cumulative Frequency</th>
<th>Cumulative Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>None Missing</td>
<td>21</td>
<td>67.74</td>
<td>21</td>
<td>67.74</td>
</tr>
<tr>
<td>Pulse Missing</td>
<td>4</td>
<td>12.90</td>
<td>25</td>
<td>80.65</td>
</tr>
<tr>
<td>Time and Pulse Missing</td>
<td>3</td>
<td>9.68</td>
<td>28</td>
<td>90.32</td>
</tr>
<tr>
<td>Oxygen Missing</td>
<td>1</td>
<td>3.23</td>
<td>29</td>
<td>93.55</td>
</tr>
<tr>
<td>Oxygen and Pulse Missing</td>
<td>2</td>
<td>6.45</td>
<td>31</td>
<td>100.00</td>
</tr>
</tbody>
</table>
# Comparing Missing and Not Missing

### Oxygen

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Std Dev</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxygen</td>
<td>47.1161786</td>
<td>5.4130470</td>
<td>28</td>
</tr>
<tr>
<td>RunTime</td>
<td>10.7020000</td>
<td>1.3943368</td>
<td>25</td>
</tr>
<tr>
<td>RunPulse</td>
<td>171.6666667</td>
<td>10.3505233</td>
<td>21</td>
</tr>
</tbody>
</table>

### Running Time

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Std Dev</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxygen</td>
<td>46.4747200</td>
<td>5.0578561</td>
<td>25</td>
</tr>
<tr>
<td>RunTime</td>
<td>10.6882143</td>
<td>1.3798794</td>
<td>28</td>
</tr>
<tr>
<td>RunPulse</td>
<td>171.8636364</td>
<td>10.1432382</td>
<td>22</td>
</tr>
</tbody>
</table>

### Pulse Rate

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Std Dev</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxygen</td>
<td>46.3538095</td>
<td>5.4778395</td>
<td>21</td>
</tr>
<tr>
<td>RunTime</td>
<td>10.8613636</td>
<td>1.4576997</td>
<td>22</td>
</tr>
<tr>
<td>RunPulse</td>
<td>171.8636364</td>
<td>10.1432382</td>
<td>22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Std Dev</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxygen</td>
<td>49.4032857</td>
<td>4.8678064</td>
<td>7</td>
</tr>
<tr>
<td>RunTime</td>
<td>10.0533333</td>
<td>0.8612936</td>
<td>6</td>
</tr>
<tr>
<td>RunPulse</td>
<td>.</td>
<td>.</td>
<td>0</td>
</tr>
</tbody>
</table>
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):
    
    | 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):
    
    | 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-invertable, 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

<table>
<thead>
<tr>
<th></th>
<th>Oxygen</th>
<th>RunTime</th>
<th>RunPulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxygen</td>
<td>29.3010776</td>
<td>-5.9882853</td>
<td>-19.5021167</td>
</tr>
<tr>
<td></td>
<td>29.3010776</td>
<td>25.5819081</td>
<td>30.0067254</td>
</tr>
<tr>
<td></td>
<td>29.3010776</td>
<td>1.9441750</td>
<td>107.1333333</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>24</td>
<td>20</td>
</tr>
<tr>
<td>RunTime</td>
<td>-5.9882853</td>
<td>1.9040671</td>
<td>3.6559091</td>
</tr>
<tr>
<td></td>
<td>1.9441750</td>
<td>1.9040671</td>
<td>2.1248885</td>
</tr>
<tr>
<td></td>
<td>25.5819081</td>
<td>1.9040671</td>
<td>102.8852814</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>27</td>
<td>21</td>
</tr>
<tr>
<td>RunPulse</td>
<td>-19.5021167</td>
<td>3.6559091</td>
<td>102.8852814</td>
</tr>
<tr>
<td></td>
<td>107.1333333</td>
<td>102.8852814</td>
<td>102.8852814</td>
</tr>
<tr>
<td></td>
<td>30.0067254</td>
<td>2.1248885</td>
<td>102.8852814</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>21</td>
<td>21</td>
</tr>
</tbody>
</table>
```
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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Std Dev</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxygen</td>
<td>47.1161786</td>
<td>5.4130470</td>
<td>28</td>
</tr>
<tr>
<td>RunTime</td>
<td>10.6882143</td>
<td>1.3798794</td>
<td>28</td>
</tr>
<tr>
<td>RunPulse</td>
<td>171.8636364</td>
<td>10.1432382</td>
<td>22</td>
</tr>
</tbody>
</table>

Before Single Imputation:

After Single Imputation:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>Std Dev</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxygen</td>
<td>47.1146129</td>
<td>5.1352696</td>
<td>31</td>
</tr>
<tr>
<td>RunTime</td>
<td>10.6893548</td>
<td>1.3090733</td>
<td>31</td>
</tr>
<tr>
<td>RunPulse</td>
<td>171.8741935</td>
<td>8.4864585</td>
<td>31</td>
</tr>
</tbody>
</table>

• 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:
  ➢ \[ \text{OXYGEN} = \beta_{01} + \beta_{11} \times \text{RUNTIME} + \beta_{21} \times \text{PULSE} + \text{Error} \]
  ➢ \[ \text{RUNTIME} = \beta_{02} + \beta_{12} \times \text{OXYGEN} + \beta_{22} \times \text{PULSE} + \text{Error} \]
  ➢ \[ \text{PULSE} = \beta_{03} + \beta_{13} \times \text{OXYGEN} + \beta_{23} \times \text{RUNTIME} + \text{Error} \]

• **Error** is random: drawn from a normal distribution
  ➢ Zero mean and variance equal to residual variance \( \sigma^2_e \) 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
A BETTER WAY:
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
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
  - Better option: use maximum likelihood (stay tuned)
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 $p$ variables from a MVN:

$$X_{(N \times p)} \sim N_p(\mu, \Sigma)$$

• 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:\[ \begin{bmatrix} X_1: (N \times q) & X_2: (N \times p-q) \end{bmatrix} \]
  ➢ The mean vector: \[ \begin{bmatrix} \mu_1: (q \times 1) \\ \mu_2: (p-q \times 1) \end{bmatrix} \]
  ➢ The covariance matrix: \[ \begin{bmatrix} \Sigma_{11}: (q \times q) & \Sigma_{12}: (q \times p-q) \\ \Sigma_{21}: (p-q \times q) & \Sigma_{22}: (p-q \times p-q) \end{bmatrix} \]
Conditional Distributions of MVN Variables

- The conditional distribution of $X_1$ given the values of $X_2 = x_2$ is then:

$$X_1 | X_2 \sim N_q(\mu^*, \Sigma^*)$$

Where (using our partitioned matrices):

$$\mu^* = \mu_1 + \Sigma_{12} \Sigma_{22}^{-1} (x'_2 - \mu_2)$$

And:

$$\Sigma^* = \Sigma_{11} - \Sigma_{12} \Sigma_{22}^{-1} \Sigma_{21}$$
Example from our Data

1. From estimates with missing data:

\[
\bar{x} = \begin{bmatrix} 47.1 \\ 10.7 \\ 171.9 \end{bmatrix};
S = \begin{bmatrix} 29.3 & -6.0 & -19.5 \\ -6.0 & 1.9 & 3.7 \\ -19.5 & 3.7 & 102.9 \end{bmatrix}
\]

2. For observation #4 (missing oxygen): \(x_2 = [11.96, 176]\)
   - We wish to impute the first observation (oxygen) conditional on the values of runtime and pulse.

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

\[
\bar{x}_1 = [47.1];
\bar{x}_2 = \begin{bmatrix} 10.7 \\ 171.9 \end{bmatrix}
\]

\[
S_{11} = [29.3];
S_{12} = [-6.0, -19.5];
S_{21} = [-6.0, -19.5];
S_{22} = \begin{bmatrix} 1.9 & 3.7 \\ 3.7 & 102.9 \end{bmatrix};
S_{22}^{-1} = \begin{bmatrix} 0.56 & -0.02 \\ -0.02 & 0.01 \end{bmatrix}
\]
Imputation Distribution

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

\[
\bar{x}^* = \bar{x}_1 + s_{12}s_{22}^{-1}(x_2' - \bar{x}_2) = [47.1] + [-6.0 \quad -19.5] \begin{bmatrix} .56 & -0.02 \\ -0.02 & 0.01 \end{bmatrix} \begin{bmatrix} 11.96 \\ 176 \end{bmatrix} - \begin{bmatrix} 10.7 \\ 171.9 \end{bmatrix} = 43.0
\]

Variance:

\[
s^* = s_{11} - s_{12}s_{22}^{-1}s_{21} = [29.3] - [-6.0 \quad -19.5] \begin{bmatrix} .56 & -0.02 \\ -0.02 & 0.01 \end{bmatrix} [-19.5] = 9.8
\]
Using the MVN for Missing Data

• If we consider our missing data to be $X_1$, 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 $\mu$ and $\Sigma$ 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 $\mu$ and $\Sigma$:

\[(\mu_{t-1}=0, \Sigma_{t-1}=0)\]

Iterate $t$ times through:

Step 1: Using $\mu_{t-1}, \Sigma_{t-1}$ 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 $\mu_t, \Sigma_t$ 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 $\mu_t, \Sigma_t$
  - Called a 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
    - Called a thinning interval
Using PROC MI

- PROC MI Syntax:

```sql
*USING PROC MI TO IMPUTE DATA:;
PROC MI DATA=WORK.fitmiss OUT=WORK.fitimpute NIMPUTE=30;
MCMC CHAIN=MULTITPLE DISPLAYINIT INITIAL=EM(ITPRINT);
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 (fit impute)
  - Here it contains 30 imputations for each missing variable
    - Need to run the regression 30 times – Analysis and Pooling Phase
Inspecting Imputed Values

- To demonstrate the imputed values, look at the histogram of the 30 values for observation 4:
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
  - Here we use PROC GLM

```
*ANALYSIS PHASE:;
PROC GLM DATA=WORK.fitimpute;
BY _IMPUTATION_
MODEL oxygen = runtime runpulse / INVERSE;
ODS OUTPUT ParameterEstimates = WORK.regparms
   InvXPX = WORK.glmxpxi;
RUN;
```

- Syntax runs for each data set (BY _IMPUTATION_)
- Saves from each:
  - Parameter estimates (to make parameter estimates)
  - $(X^TX)^{-1}$ matrix (to make standard errors)
    - $\text{Var}(\beta) = \sigma^2 (X^TX)^{-1}$ in general linear models
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 straightforward
  ➢ 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 \( \theta \) 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

```sas
*POOLING PHASE:;
PROC MIANALYZE PARMS=WORK.regparms XPXI=WORK.glmxpxi EDF=28;
MODELEFFECTS intercept runtime runpulse;
RUN;
```

- The parameter data set, the \((X^T X)^{-1}\) dataset, and the number of error degrees of freedom are all input

- The MODELEFFECTS line combs through the input data and conducts the pooling
## PROC MIANALYZE OUTPUT

### Variances:
See Next Slides

#### Variance Information

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Between</th>
<th>Within</th>
<th>Total</th>
<th>DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>intercept</td>
<td>17.134847</td>
<td>71.359492</td>
<td>89.065501</td>
<td>20.403</td>
</tr>
<tr>
<td>runtime</td>
<td>0.034190</td>
<td>0.124585</td>
<td>0.159915</td>
<td>19.729</td>
</tr>
<tr>
<td>runpulse</td>
<td>0.000646</td>
<td>0.002502</td>
<td>0.003169</td>
<td>20.044</td>
</tr>
</tbody>
</table>

#### Parameter Estimates – With Hypothesis Test P-Values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Estimate</th>
<th>Std Error</th>
<th>95% Confidence Limits</th>
<th>DF</th>
</tr>
</thead>
<tbody>
<tr>
<td>intercept</td>
<td>91.962975</td>
<td>9.437452</td>
<td>72.30168 - 111.6243</td>
<td>20.403</td>
</tr>
<tr>
<td>runtime</td>
<td>-3.051541</td>
<td>0.399894</td>
<td>-3.88644 - 2.2166</td>
<td>19.729</td>
</tr>
<tr>
<td>runpulse</td>
<td>-0.073695</td>
<td>0.056295</td>
<td>-0.19111 - 0.0437</td>
<td>20.044</td>
</tr>
</tbody>
</table>

| Parameter | Minimum | Maximum | Parameter | Theta0 | t for H0: Parameter=Theta0 | Pr > |t| |
|-----------|---------|---------|-----------|--------|---------------------------|------|---|
| intercept | 83.981397 | 102.046398 | intercept | 0 | 9.74 | <.0001 |
| runtime   | -3.437656 | -2.509580 | runtime   | 0 | -7.63 | <.0001 |
| runpulse  | -0.129071 | -0.031825 | runpulse  | 0 | -1.31 | 0.2053 |
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 = 0.20; runtime = .22; runpulse = .21
  - ~20% 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 = 0.25; runtime = .28; runpulse = .27
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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$m = 1$</th>
<th>$m = 10$</th>
<th>$m = 30$</th>
<th>$m = 100$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>7.04</td>
<td>10.48</td>
<td>9.27</td>
<td>10.03</td>
</tr>
<tr>
<td>RunTime</td>
<td>0.38</td>
<td>0.42</td>
<td>0.39</td>
<td>0.41</td>
</tr>
<tr>
<td>RunPulse</td>
<td>0.05</td>
<td>0.06</td>
<td>0.05</td>
<td>0.06</td>
</tr>
</tbody>
</table>
CONCLUDING REMARKS
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