# Missing Data; Missing Data Methods in ML; Multiple Imputation via Predictive Mean Matching

EPSY 905: Multivariate Analysis

Spring 2016

Lecture #11: April 13, 2016

# 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)

| IQ | Performance |
|---|---|
| 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 $\mathbf{D}$ 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 $\boldsymbol{M}_{obs}$ and $\boldsymbol{M}_{mis}$ denote the observed and missing parts of $\boldsymbol{M}$

$$\boldsymbol{M} = \{\boldsymbol{M}_{obs}, \boldsymbol{M}_{mis}\}$$

# 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, $\boldsymbol{M}$ are **statistically independent** of our observed data $\boldsymbol{D}$

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

**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

# MCAR Data

| IQ | Performance |
|---|---|
| 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 | - |

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

- $\boldsymbol{M}$ is independent of $\boldsymbol{D}_{mis}$

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

# MAR Data

| IQ | Perf | Indicator |
|---|---|---|
| 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 |

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

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

- 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, gls, lavaan
  - Models with "latent" random effects (CFA/SEM models) in Mplus, lavaan

# 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:

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

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

$$\boldsymbol{y}_p = \begin{bmatrix} y_{p,IQ} \\ y_{p,Perf} \end{bmatrix};$$

$$\boldsymbol{\mu}_p = \mathbf{X}_p\boldsymbol{\beta} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix} = \begin{bmatrix} \beta_0 + \beta_1 \\ \beta_0 \end{bmatrix} = \begin{bmatrix} \mu_{IQ} \\ \mu_{Perf} \end{bmatrix};$$

$$\boldsymbol{\Sigma}_p = \begin{bmatrix} \sigma_{IQ}^2 & \sigma_{IQ,Perf} \\ \sigma_{IQ,Perf} & \sigma_{Perf}^2 \end{bmatrix}$$

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

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

# 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

| Row | Col1 |
|---|---|
| 1 | 189.60 |

Estimated R Matrix for Subject 2

| Row | Col1 | Col2 |
|---|---|---|
| 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 $\mathbf{R}$ 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*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 $\beta$
   - 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

# 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

# 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

# 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
  - OXYGEN = $\beta_{01} + \beta_{11}$*RUNTIME $+ \beta_{21}$*PULSE
  - RUNTIME = $\beta_{02} + \beta_{12}$*OXYGEN $+ \beta_{22}$*PULSE
  - PULSE = $\beta_{03} + \beta_{13}$*OXYGEN $+ \beta_{23}$*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:
  - OXYGEN = $\beta_{01}$ + $\beta_{11}$*RUNTIME + $\beta_{21}$*PULSE + **Error**
  - RUNTIME = $\beta_{02}$ + $\beta_{12}$*OXYGEN + $\beta_{22}$*PULSE + **Error**
  - PULSE = $\beta_{03}$ + $\beta_{13}$*OXYGEN + $\beta_{23}$*RUNTIME + **Error**

- **Error** is random: drawn from a normal distribution
  - Zero mean and variance equal to residual variance $\sigma_e^2$ 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

# 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 R

- R has two well-known packages for multiple imputation: MICE and Amelia
  - MICE: Multiple Imputation by Chained Equations
  - Amelia: MI via MCMC assuming Multivariate Normal

- The MICE package has a number of methods available for multiple imputation
  - We restrict our focus today on those that are called "predictive mean matching" as these are very commonly used in research

# IMPUTATION PHASE

# Multiple Imputation via Chained Equations

- The key to MI via chained equations is creating a set of **univariate** regression models
  - One for each variable with missing data
- The process starts by randomly imputing values for all missing observations
- For each variable with missing data, in sequence, a regression is run whereby a predicted mean is created for all observations with missing data
  - The predicted mean comes from applying the sampled values of the posterior distribution of the regression weights (so sampling error doesn't get forgotten)
- Each missing observation is then imputed via one of a host of methods (up next)
- This process continues for all variables with missing data
  - Each time through, newly imputed values aid in the estimation of the betas
- At the end of the process, a number of "imputed" (complete) data sets now exist

# Example of MI via CE and PMM

- To demonstrate multiple imputation with chained equations using predictive mean matching we will consider three variables from our oft-used math performance data
  - Performance (PERF), Sex (MALE), and College Credit Hours (CC)

- Amount of missing data in sample:

```
> length(which(is.na(data02$male)))/dim(data02)[1]
[1] 0
> #proportion missing perf variable:
> length(which(is.na(data02$perf)))/dim(data02)[1]
[1] 0.1714286
> #proportion missing cc variable:
> length(which(is.na(data02$cc)))/dim(data02)[1]
[1] 0.1057143
> #proportion missing both CC and PERF:
> length(which(is.na(data02$cc) & is.na(data02$perf)))/dim(data02)[1
[1] 0.02285714
>
```

- We will impute for PERF and for CC

# Regressions for PERF and CC

- To make the chained equations work, we regression models for each variable with missing data:

$$PERF_p = \beta_0^P + \beta_M^P M_p + \beta_{CC}^P CC_p + \beta_{M*CC}^P M_p CC_p + e_p^P$$

$$CC_p = \beta_0^C + \beta_M^C M_p + \beta_P^C P_p + \beta_{M*P}^C M_p P_p + e_p^C$$

- Each variable's prediction model can be made up of anything (we'll just use all other variables plus the interaction)

- We'll continue this process in our R example

# MI via Predictive Mean Matching

- In the imputation phase, missing values are imputed using matching on the predicted mean (from R)

**Details**

Imputation of `y` by predictive mean matching, based on Rubin (1987, p. 168, formulas a and b). The procedure is as follows:

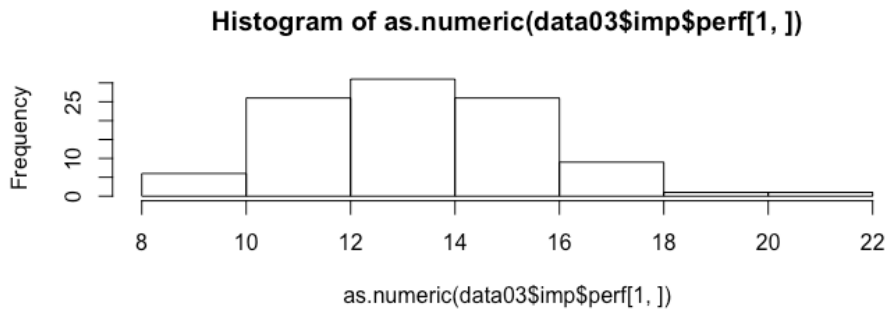
1. Estimate beta and sigma by linear regression
2. Draw beta* and sigma* from the proper posterior
3. Compute predicted values for `yobs` `beta` and `ymis` `beta*`
4. For each `ymis`, find `donors` observations with closest predicted values, randomly sample one of these, and take its observed value in `y` as the imputation.
5. Ties are broken by making a random draw among ties. Note: The matching is done on predicted `y`, NOT on observed `y`.

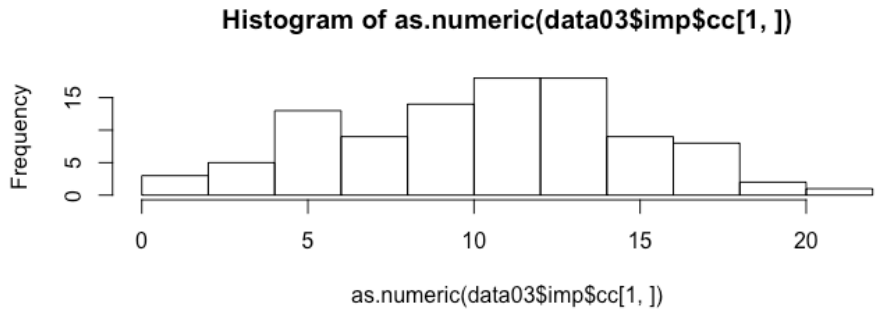
# Running MICE

- You don't have to use my clunky R code to make an imputation run…you can use MICE:

```
#-----------------------------------------------------------------------
#running MICE for data imputation (100 samples from 20 iterations):
data03 = mice(data=data01, m = 100, maxit = 20, diagnostics = TRUE)
```

- Here is some MICE imputation distributions:



# MULTIPLE IMPUTATION: ANALYSIS PHASE

# Up Next: Multiple Analyses

- Once you run MICE, 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 100 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: MICE will automate all of this for you

# Analysis Phase

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

```
#using MICE functions to estimate a linear regression predicting PERF from CC, MALE and CC*MALE
model01 = with(data03, lm(perf ~ male + cc + male*cc))
```

- Syntax runs for each data set (with() )

- Model01 now has all 100 models contained within it

# 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 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):

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

- 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 R: MICE's pool() Function

- MICE has a function that will pool all results (results for using certain classes of models, that is)

```
#using MICE functions to pool estimates
model01ests = pool(model01)
summary(model01ests)
```

- Behold, the results:

```
> summary(model01ests)
                  est         se          t       df   Pr(>|t|)       lo 95      hi 95 nmis        fmi     lambda
(Intercept) 12.69000305 0.40908837 31.0201998 309.0277 0.00000000 11.88505205 13.4949540   NA 0.08818622 0.08230407
male2       -0.27710360 0.69044015 -0.4013434 280.8424 0.68847252 -1.63619835  1.0819911   NA 0.14350366 0.13742582
cc           0.09626833 0.03321765  2.8981076 300.2802 0.00403064  0.03089946  0.1616372   37 0.10582882 0.09989301
male2:cc     0.08233524 0.06120595  1.3452164 273.2318 0.17967077 -0.03815994  0.2028304   NA 0.15795204 0.15181090
```

# 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

# 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