Missing Data in SEM

Introduction to SEM with Lavaan

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Outline

What are Missing Data?

Missing data are empty cells in a dataset where there should be observed values.

- The missing cells correspond to true population values, but we haven’t observed those values.

What are Missing Data?

Missing data are empty cells in a dataset where there should be observed values.

- The missing cells correspond to true population values, but we haven't observed those values.

Not every empty cell is a missing datum.

- Quality-of-life ratings for dead patients in a mortality study
- Firm profitability after the company goes out of business
- Self-reported severity of menstrual cramping for men
- Empty blocks of data following “gateway” items

A Little Notation

$$
\begin{aligned}
Y &:= \text{An } N \times P \text{ Matrix of Arbitrary Data} \\
Y_{mis} &:= \text{The } \textit{missing} \text{ part of } Y \\
Y_{obs} &:= \text{The } \textit{observed} \text{ part of } Y \\
R &:= \text{An } N \times P \text{ response matrix} \\
M &:= \text{An } N \times P \text{ missingness matrix}
\end{aligned}
$$

The R and M matrices are complementary.

- $r_{np} = 1$ means y_{np} is observed; $m_{np} = 1$ means y_{np} is missing.
- $r_{np} = 0$ means y_{np} is missing; $m_{np} = 0$ means y_{np} is observed.
- M_p is the *missingness* of Y_p.

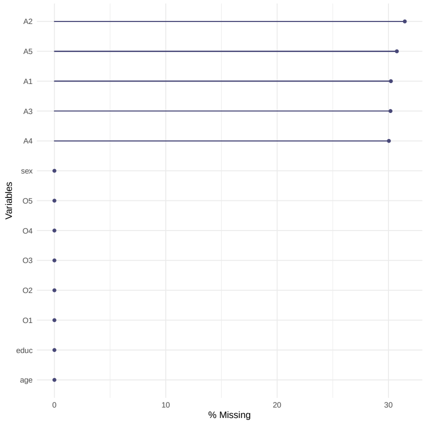
Example

```r
## Load some useful packages:
library(dplyr)
library(naniar)
library(ggmice)

## Read in some data:
bfi <- readRDS("../data/bfi_datasets.rds")$incomplete %>%
    select(-matches("N\\d|C\\d|E\\d|male"))

## Compute the variablewise proportions of missing data:
bfi %>% is.na() %>% colMeans() %>% round(2)

  A1   A2   A3   A4   A5   O1   O2   O3   O4   O5  age  sex educ
0.30 0.31 0.30 0.30 0.31 0.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00
```

Example

Visualize the percentages missing via **naniar**`::gg_miss_var()`.

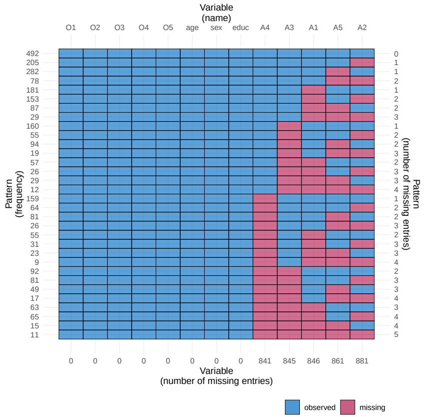
```
gg_miss_var(bfi, show_pct = TRUE)
```



Example

Visualize the missing data patterns via **ggmice**`::plot_pattern()`.

```
plot_pattern(bfi)
```



Example

In **lavaan**, we can directly fit a model with incomplete data.

```r
library(lavaan)

## Specify the measurement model:
mod <- "
agree =~ A1 + A2 + A3 + A4 + A5
open  =~ O1 + O2 + O3 + O4 + O5
"

## Estimate the model:
out <- cfa(mod, data = bfi, std.lv = TRUE)
```

Example

The model will estimate, just fine...

```
lavaan 0.6-11 ended normally after 21 iterations

Latent Variables:
                   Estimate  Std.Err  z-value  P(>|z|)
  agree =~
    A1                0.561    0.071    7.866    0.000
    A2               -0.725    0.056  -13.055    0.000
    A3               -0.901    0.063  -14.383    0.000
    A4               -0.599    0.074   -8.103    0.000
    A5               -0.781    0.059  -13.264    0.000
  open =~
    O1                0.490    0.055    8.983    0.000
    O2               -0.844    0.079  -10.649    0.000
    O3                0.786    0.058   13.574    0.000
    O4                0.311    0.055    5.699    0.000
    O5               -0.874    0.073  -11.987    0.000
```

Example

```
Covariances:
                   Estimate  Std.Err  z-value  P(>|z|)
  agree ~~
    open             -0.254    0.060   -4.204    0.000

Variances:
                   Estimate  Std.Err  z-value  P(>|z|)
   .A1                1.662    0.114   14.593    0.000
   .A2                0.781    0.067   11.650    0.000
   .A3                0.865    0.086   10.050    0.000
   .A4                1.776    0.122   14.517    0.000
   .A5                0.863    0.075   11.430    0.000
   .O1                0.880    0.064   13.823    0.000
   .O2                1.704    0.133   12.776    0.000
   .O3                0.658    0.072    9.150    0.000
   .O4                0.976    0.065   15.020    0.000
   .O5                1.288    0.112   11.460    0.000
```

Example

But not everything is as it seems.

```
  Estimator                                         ML
  Optimization method                           NLMINB
  Number of model parameters                        21

                                                  Used       Total
  Number of observations                           492        2800

Model Test User Model:

  Test statistic                                79.928
  Degrees of freedom                                34
  P-value (Chi-square)                           0.000
```

Default Approach

Like most software packages, **lavaan** will default to *complete case analysis* when asked to analyze incomplete data.

- In the absence of user input, this is a sensible option.
- That doesn’t mean you should actually use deletion to treat the missing data in your analysis.

Default Approach

Like most software packages, **lavaan** will default to *complete case analysis* when asked to analyze incomplete data.

- In the absence of user input, this is a sensible option.
- That doesn’t mean you should actually use deletion to treat the missing data in your analysis.

Complete case analysis has two major problems.

1. Throws out useful information (potentially a lot of information)
2. Probably biases parameter estimates.

To understand the second point, we need to discuss *missing data mechanisms*.

Missing Data Mechanisms

Missing Data Mechanisms

Missing Completely at Random (MCAR)

- $P(R|Y_{mis}, Y_{obs}) = P(R)$
- Missingness is unrelated to any study variables.

Missing at Random (MAR)

- $P(R|Y_{mis}, Y_{obs}) = P(R|Y_{obs})$
- Missingness is related to only the *observed* parts of study variables.

Missing not at Random (MNAR)

- $P(R|Y_{mis}, Y_{obs}) \neq P(R|Y_{obs})$
- Missingness is related to the *unobserved* parts of study variables.

Simulate Some Toy Data

```r
nObs <- 5000 # Sample Size
pm   <- 0.3  # Proportion Missing

sigma <- matrix(c(1.0, 0.5, 0.3,
                  0.5, 1.0, 0.0,
                  0.3, 0.0, 1.0),
                ncol = 3)
tmp <- rmvnorm(nObs, c(0, 0, 0), sigma)

x0 <- tmp[ , 1]
y0 <- tmp[ , 2]
z0 <- tmp[ , 3]

cor(y0, x0) # Check correlation between X and Y
```

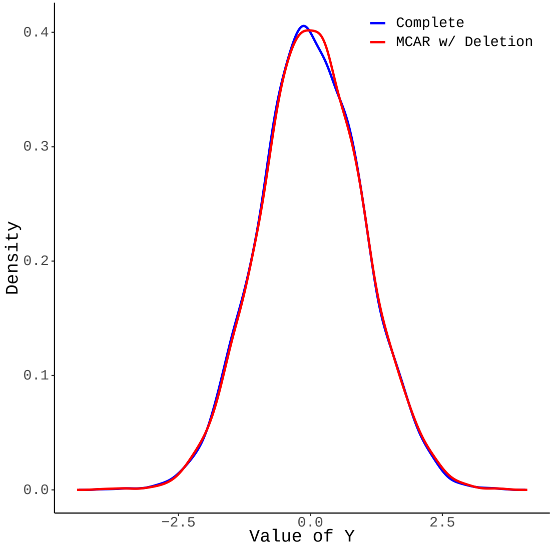
```
[1] 0.4997145
```

MCAR Example

```r
## Simulate MCAR Missingness:
mVec <- sample(1 : length(y0), size = pm * length(y0))

yMcar        <- y0
yMcar[mVec] <- NA

cor(yMcar, x0, use = "pairwise") # Look at correlation
```

```
[1] 0.5195767
```

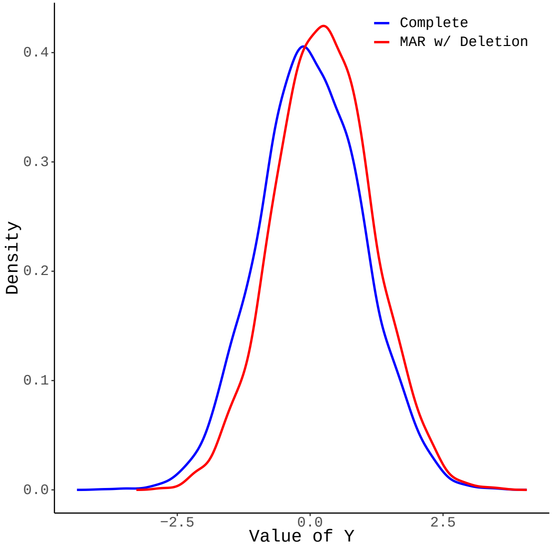
MCAR Example



MAR Example

```
## Simulate MAR Missingness:
mVec <- x0 < quantile(x0, probs = pm)
mean(mVec)

[1] 0.3

yMar       <- y0
yMar[mVec] <- NA

cor(yMar, x0, use = "pairwise") # Not looking so good :(

[1] 0.3822143
```

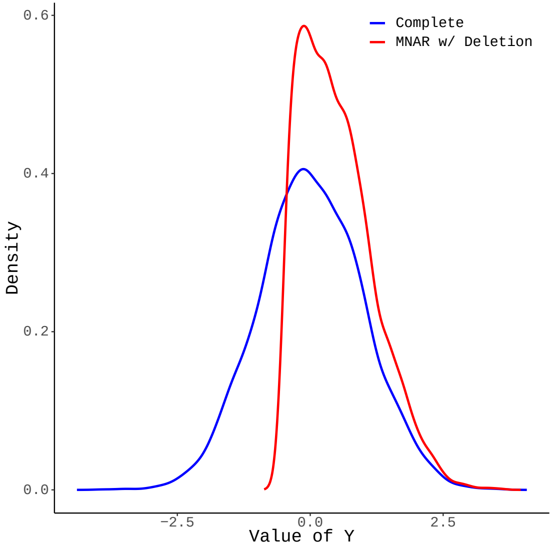
MAR Example


Complete
MAR w/ Deletion
Density
0.0
0.1
0.2
0.3
0.4
−2.5
0.0
2.5
Value of Y

MNAR Example

```
## Simulate MNAR Missingness:
mVec <- y0 < quantile(y0, probs = pm)
mean(mVec)

[1] 0.3

yMnar       <- y0
yMnar[mVec] <- NA

cor(yMnar, x0, use = "pairwise") # Hmm...looks pretty bad.

[1] 0.3902962
```

MNAR Example



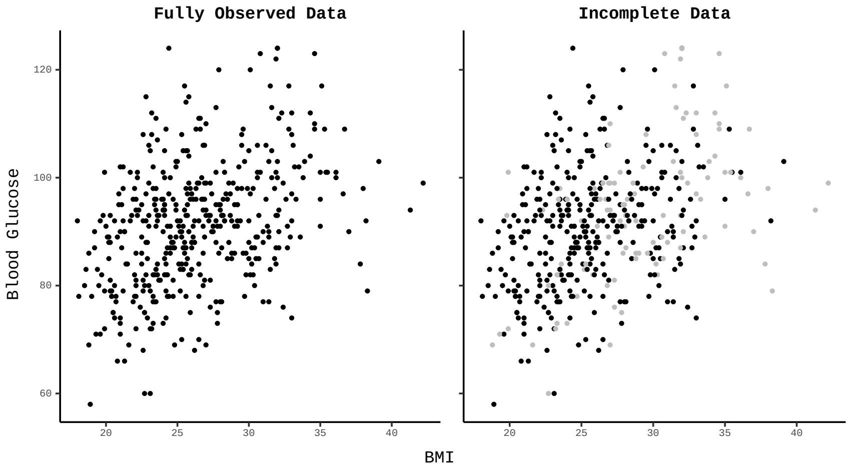
Effects of Deletion

As we saw in the preceding plots, excluding incomplete cases usually alters the variables' distributions.

- The statistics upon which we base our analyses generally summarize these distributions.
- Problems with the distributions show up as bias in the results of our anlayses.

```
diabetes1 <- diabetes2 <- readRDS("../data/diabetes.rds")

mVec <- simLogisticMissingness0(data     = diabetes1,
                                pm       = 0.3,
                                preds    = c("bmi", "bp"),
                                stdData = TRUE)$r
diabetes2[mVec, "glu"] <- NA
```

Example


Fully Observed Data
Incomplete Data
Blood Glucose
BMI
120
100
80
60
20
25
30
35
40

Example

```
diabetes1 %>% select(bmi, glu, bp) %>% cor()

          bmi     glu        bp
bmi 1.0000000 0.38868 0.3954109
glu 0.3886800 1.00000 0.3904300
bp  0.3954109 0.39043 1.0000000

diabetes2 %>% select(bmi, glu, bp) %>% cor(use = "complete")

          bmi       glu        bp
bmi 1.0000000 0.2566595 0.2052338
glu 0.2566595 1.0000000 0.3011547
bp  0.2052338 0.3011547 1.0000000
```

Example

```
mean(diabetes1$glu)

[1] 91.26018

mean(diabetes2$glu, na.rm = TRUE)

[1] 88.86424

var(diabetes1$glu)

[1] 132.1657

var(diabetes2$glu, na.rm = TRUE)

[1] 115.8254
```

Good Methods

Multiple Imputation (MI)

- Replace the missing values with M plausible estimates
  - Essentially, a repeated application of stochastic regression imputation (with a particular type of regression model)
  - Produces unbiased parameter estimates and predictions
  - Produces “correct” standard errors, CIs, and prediction intervals
  - Very, very flexible
  - Computationally expensive

Good Methods

Full Information Maximum Likelihood (FIML)

- Adjust the objective function to only consider the observed parts of the data
    - Models are directly estimated in the presence of missing data
    - The predictors of nonresponse must be included in the model, somehow
    - Unless you write your own optimization program, FIML is only available for certain types of models
    - In linear regression models, FIML cannot treat missing data on predictors (if the predictors are taken as fixed)

Good Methods

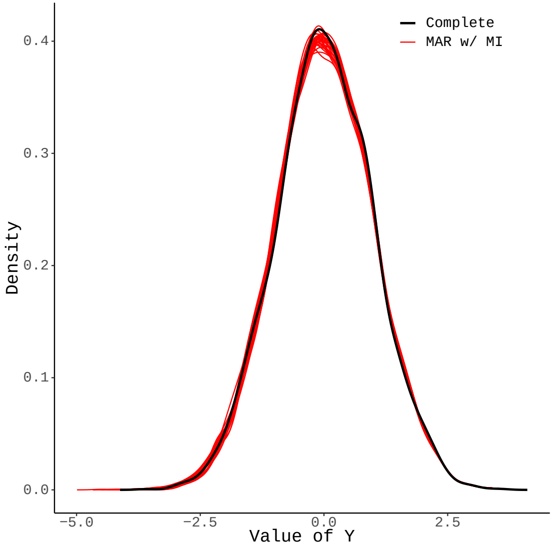
What happens when we apply MI to our previous MAR example?

```
## Estimate imputation model:
miceOut <- mice(data      = data.frame(y = yMar, x = x0),
                m         = 25,
                maxit     = 1,
                method    = "norm",
                printFlag = FALSE)

## Estimate and pool M correlations:
with(miceOut, cor(y, x))$analyses %>% unlist() %>% mean()

[1] 0.504661
```

The MI-based parameter estimate looks good.

- MI produces unbiased estimates of the parameter when data are MAR.

Good Methods



Good Methods

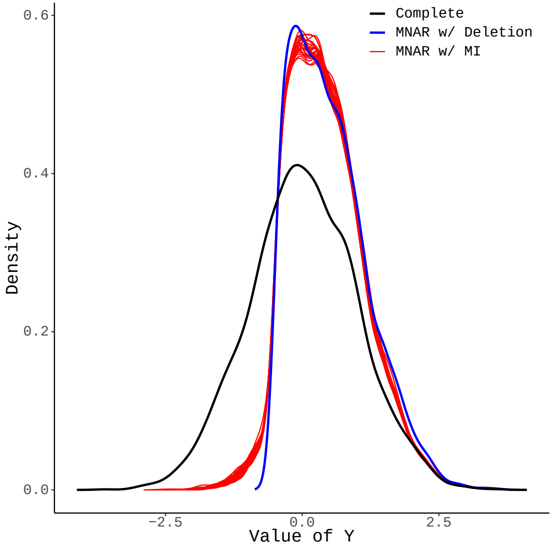
What about applying MI to our MNAR example?

```r
## Estimate imputation model:
miceOut <- mice(data      = data.frame(y = yMnar, x = x0),
                m         = 25,
                maxit     = 1,
                method    = "norm",
                printFlag = FALSE)

## Estimate and pool M correlations:
with(miceOut, cor(y, x))$analyses %>% unlist() %>% mean()

[1] 0.4116519
```

The MI-based parameter estimate is still biased.

- MI cannot correct bias in parameter estimates when data are MNAR.

Good Methods



MI Example

```r
## The mice package does MI:
library(mice)

## Multiply impute the missing data:
miceOut <- mice(data      = diabetes2,
                m         = 25,
                maxit     = 1,
                printFlag = FALSE,
                method    = "norm")
```

MI Example

```
## Complete data:
diabetes1 %>% select(bmi, glu, bp) %>% cor()

          bmi     glu        bp
bmi 1.0000000 0.38868 0.3954109
glu 0.3886800 1.00000 0.3904300
bp  0.3954109 0.39043 1.0000000

## MI:
pooledCorMat(miceOut, c("bmi", "glu", "bp"))

          bmi       glu        bp
bmi 1.0000000 0.3135162 0.3954109
glu 0.3135162 1.0000000 0.3563903
bp  0.3954109 0.3563903 1.0000000
```

MI Example

```
mean(diabetes1$glu)

[1] 91.26018

with(miceOut, mean(glu))$analyses %>% unlist() %>% mean()

[1] 90.61747

var(diabetes1$glu)

[1] 132.1657

with(miceOut, var(glu))$analyses %>% unlist() %>% mean()

[1] 123.3748
```

FIML Example

```
fit <- diabetes2 %>%
    select(bmi, glu, bp) %>%
    lavCor(missing = "fiml", output = "sampstat")

## Complete data:
diabetes1 %>% summarize(mean = mean(glu), var = var(glu))

      mean      var
1 91.26018 132.1657

## FIML:
fit %$% c(mean = mean[["glu"]], var = cov["glu", "glu"])

     mean       var
 90.82487 125.27146
```

FIML Example

```
diabetes1 %>% select(bmi, glu, bp) %>% cor() %>% round(3)

      bmi   glu    bp
bmi 1.000 0.389 0.395
glu 0.389 1.000 0.390
bp  0.395 0.390 1.000

fit$cov %>% cov2cor()

    bmi   glu   bp
bmi 1.000
glu 0.357 1.000
bp  0.395 0.386 1.000

diabetes2 %>% select(bmi, glu, bp) %>% cor(use = "complete") %>% round(3

      bmi   glu    bp
bmi 1.000 0.257 0.205
glu 0.257 1.000 0.301
bp  0.205 0.301 1.000
```

References