```r
options(width = 60)

set.seed(235711)

library(knitr)
library(ggplot2)
library(mice)
library(mvtnorm)
library(xtable)
library(pROC)
library(dplyr)
library(magrittr)
library(naniar)
library(ggpubr)
library(lme4)
library(lavaan)

dataDir <- "data/"

source("code/supportFunctions.R")
source("code/sim_missing/code/simMissingness.R")
```

Missing Data Basics

Utrecht University Winter School: Missing Data in R

Utrecht University

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.

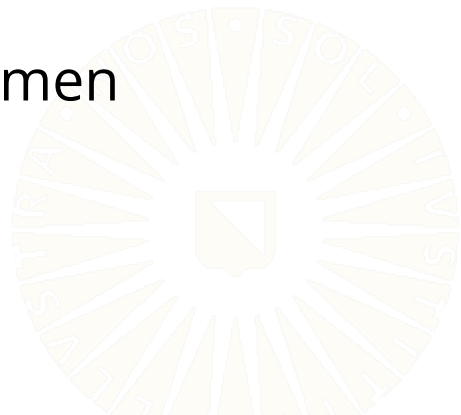
What are Missing Data?

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

$$Y := \text{An } N \times P \text{ Matrix of Arbitrary Data}$$

$$Y_{mis} := \text{The } \textit{missing} \text{ part of } Y$$

$$Y_{obs} := \text{The } \textit{observed} \text{ part of } Y$$

$$R := \text{An } N \times P \text{ response matrix}$$

$$M := \text{An } N \times P \text{ missingness matrix}$$

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.

Missing Data Descriptives

Missing Data Pattern

Missing data (or response) patterns represent unique combinations of observed and missing items.

- P items $\Rightarrow 2^P$ possible patterns.

	X	Y
1	x	y
2	x	.
3	.	y
4	.	.

Patterns for $P = 2$

	X	Y	Z
1	x	y	z
2	x	y	.
3	x	.	z
4	.	y	z
5	x	.	.
6	.	.	z
7	.	y	.
8	.	.	.

Patterns for $P = 3$

Missing Data Pattern

The concept of a “missing data pattern” can also be used to classify the spatial arrangement of missing cells on a data set.

- Univariate
  - Missing data occur on only one variable
- Monotone
  - The proportion of complete elements, in both rows and columns, decreases when traversing the data set.
  - The observed cells can be arranged into a “staircase” pattern.
- Arbitrary
  - Missing values are “randomly” scattered throughout the data set.

Example Missing Data Patterns

	X	Y	Z
1	x	y	z
2	x	y	z
3	x	y	z
4	x	y	z
5	x	y	z
6	x	.	z
7	x	.	z
8	x	.	z
9	x	.	z
10	x	.	z

Univariate Pattern

	X	Y	Z
1	x	y	z
2	x	y	z
3	x	y	z
4	x	y	.
5	x	y	.
6	x	y	.
7	x	.	.
8	x	.	.
9	x	.	.
10	.	.	.

Monotone Pattern

	X	Y	Z
1	x	.	z
2	x	y	z
3	x	y	z
4	x	.	z
5	x	y	z
6	x	.	z
7	.	y	z
8	x	y	z
9	x	.	.
10	x	y	.

Arbitrary Pattern

Nonresponse Rates

Proportion Missing

- The proportion of cells containing missing data
- Good early screening measure
- Should be computed for each variable, not for the entire dataset

Attrition Rate

- The proportion of participants that drop-out of a study at each measurement occasion

Nonresponse Rates

Proportion of Complete Cases

- The proportion of observations with no missing data
- Often reported but nearly useless quantity

Fraction of Missing Information

- Associated with an estimated parameter, not with an incomplete variable
- Like an R^2 for the missing data
- Most important diagnostic value for missing data problems
- Can only be computed after treating the missing data

Coverage Measures

Covariance Coverage

$$CC_{jk} = N^{-1} \sum_{n=1}^{N} r_{nj} r_{nk}$$

- The proportion of cases available to estimate a given pairwise relationship (e.g., a covariance between two variables)
- Very important to have adequate coverage of the parameters you want to estimate

Coverage Measures

Inbound Statistic

$$I_{jk} = \frac{\sum_{n=1}^{N}(1 - r_{nj})r_{nk}}{\sum_{n=1}^{N}(1 - r_{nj})}$$

- The proportion of missing cases in Y_j for which Y_k is observed

Outbound Statistic

$$O_{jk} = \frac{\sum_{n=1}^{N} r_{nj}(1 - r_{nk})}{\sum_{n=1}^{N} r_{nj}}$$

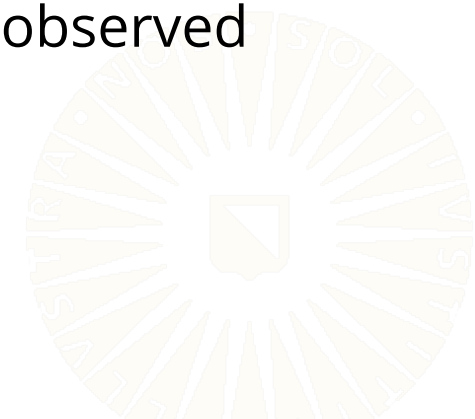
- The proportion of observed cases in Y_j for which Y_k is missing

Coverage Measures

Influx Coefficient

$$I_j = \frac{\sum_{k=1}^{P} \sum_{n=1}^{N} (1 - r_{nj}) r_{nk}}{\sum_{k=1}^{P} \sum_{n=1}^{N} r_{nk}}$$

- The proportion of observed cells in Y that exists in cases for which Y_j is missing
- How well the missing values in Y_j connect to the observed values in Y_{-j}



Coverage Measures

Outflux Coefficient

$$O_j = \frac{\sum_{k=1}^{P} \sum_{n=1}^{N} r_{nj}(1 - r_{nk})}{\sum_{k=1}^{P} \sum_{n=1}^{N} (1 - r_{nk})}$$

- The proportion of missing cells in Y that exists in cases for which Y_j is observed
- How well the observed values in Y_j connect to the missing values in Y_{-j}

Missing Data Mechanisms

Missing Data Mechanisms

Missing Completely at Random (MCAR)

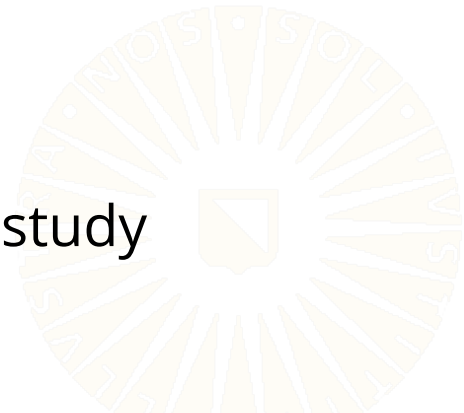
- $P(R|Y_{mis}, Y_{obs}) = P(R)$
- Missingness is unrelated to any study variables.

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.



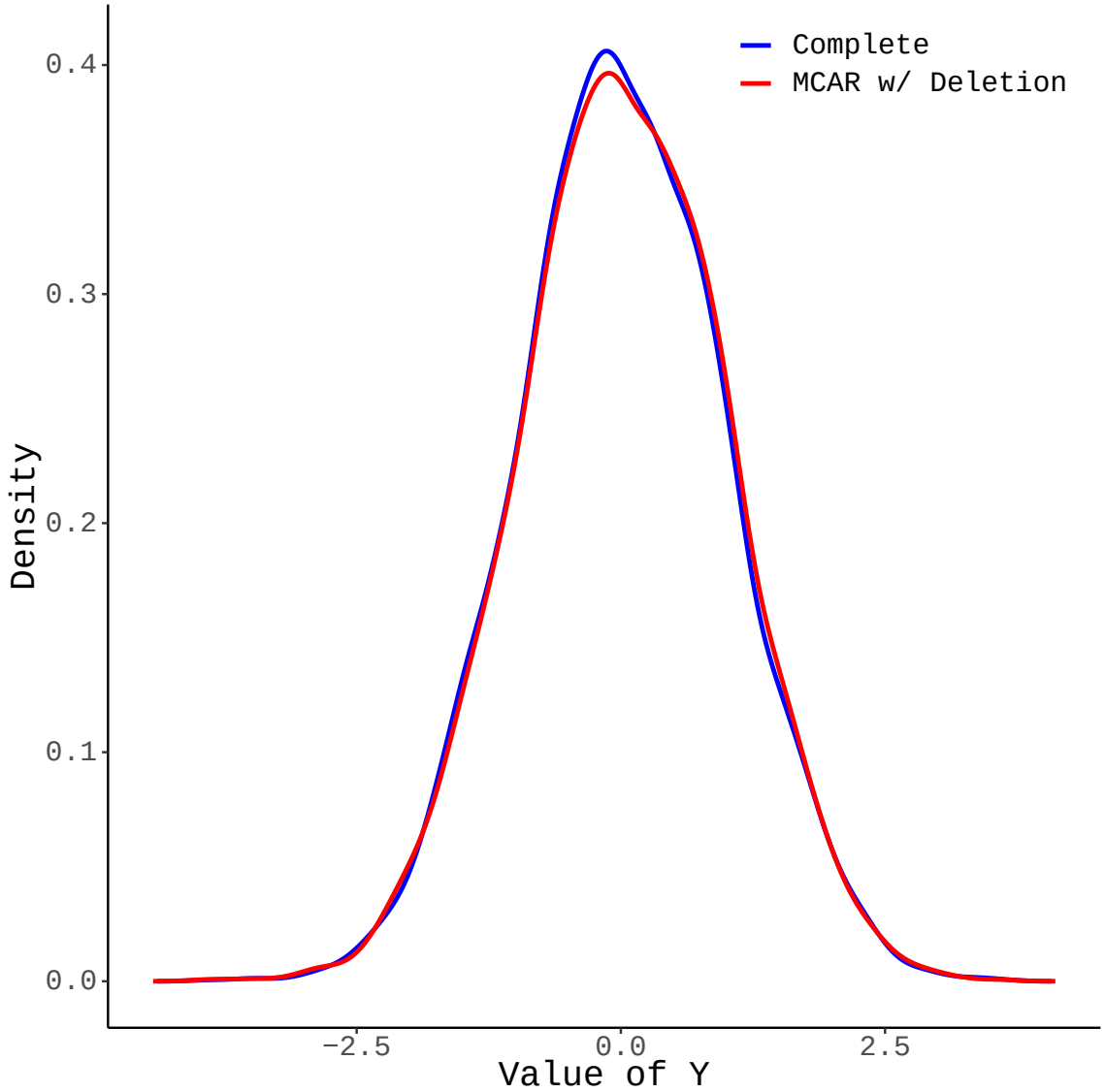
Simulate Some Toy Data

```
library(mvtnorm)
library(dplyr)
library(magrittr)

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)
dat0 <- rmvnorm(nObs, c(0, 0, 0), sigma) %>% data.frame()
colnames(dat0) <- c("x", "y", "z")

dat0 %$% cor(y, x)

[1] 0.5001822
```

MCAR Example

```
## Simulate MCAR Missingness:
m <- sample(1:nObs, size = pm * nObs)

## Impose MCAR missing on Y:
mcarData          <- dat0
mcarData[m, "y"] <- NA

## Check the correlation between X & Y:
mcarData %$% cor(y, x, use = "pairwise")

[1] 0.5197437
```

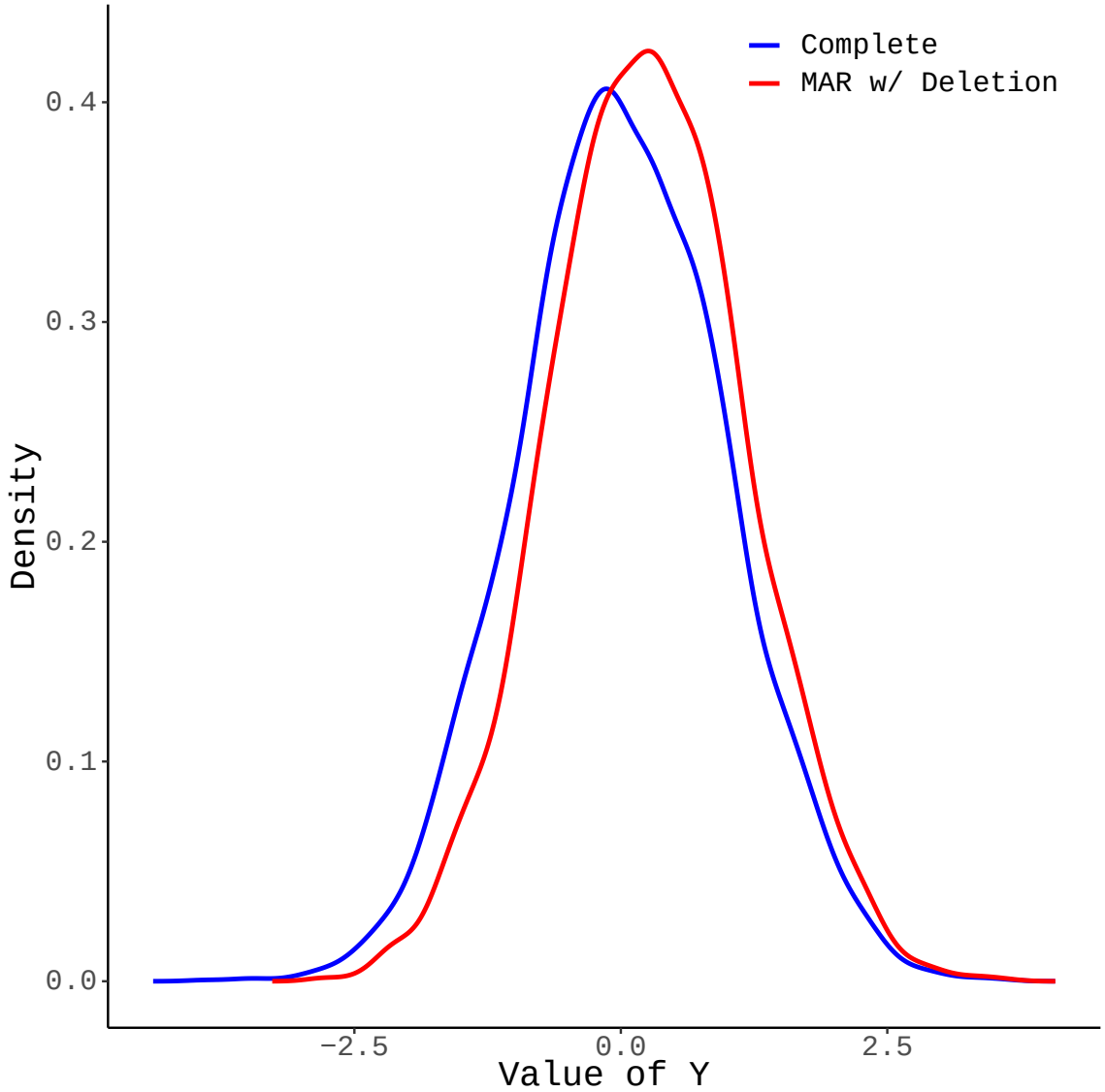
MCAR Example



MAR Example

```
## Simulate MAR Missingness:
m <- with(dat0, x < quantile(x, probs = pm))

## Impose MAR missing on Y:
marData          <- dat0
marData[m, "y"] <- NA

## Check the correlation between X & Y:
marData %$% cor(y, x, use = "pairwise")

[1] 0.3825876
```

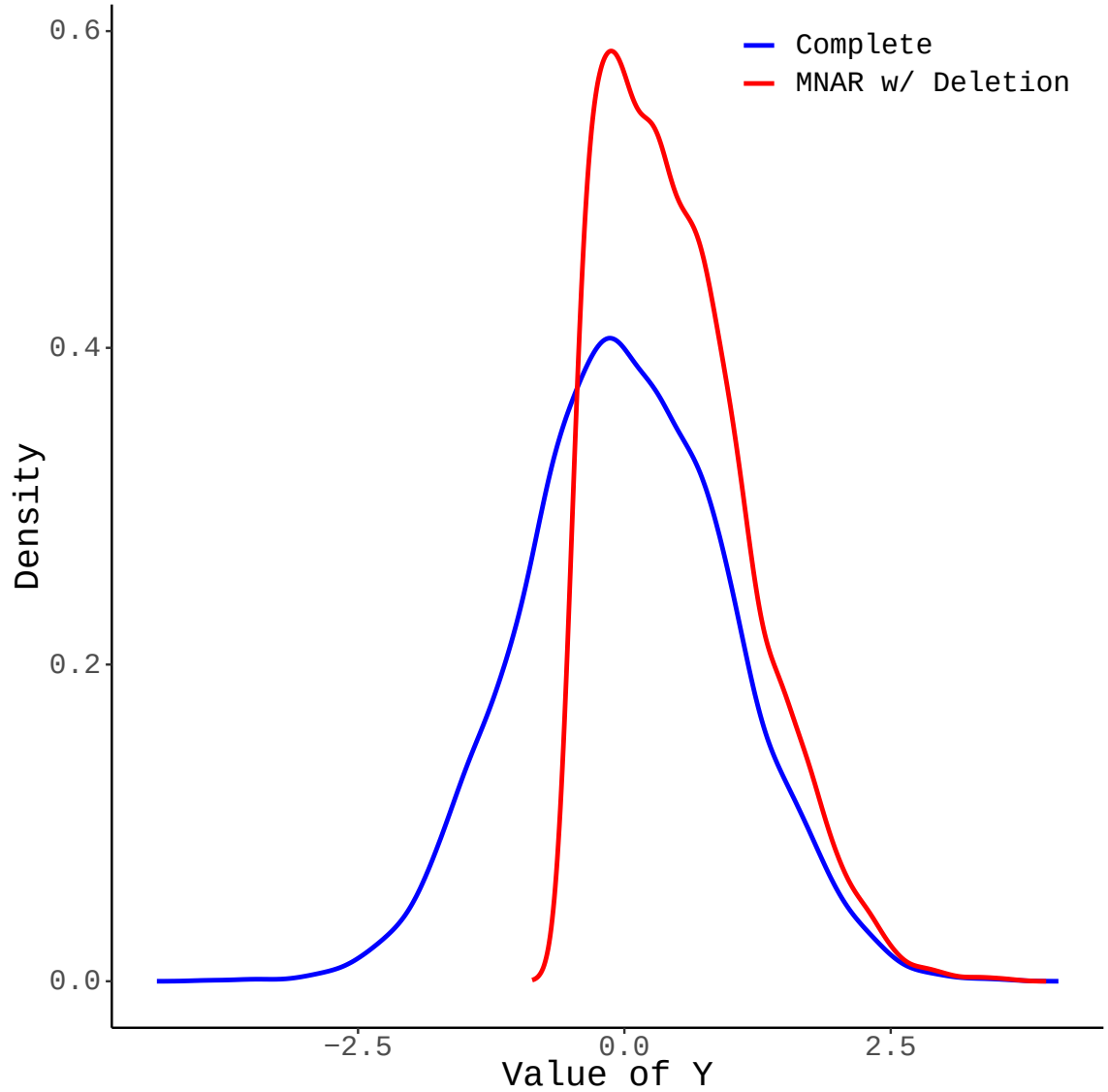
MAR Example


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

MNAR Example

```
## Simulate MNAR Missingness:
m <- with(dat0, y < quantile(y, probs = pm))

## Impose MNAR missing on Y:
mnarData         <- dat0
mnarData[m, "y"] <- NA

## Check the correlation between X & Y:
mnarData %$% cor(y, x, use = "pairwise")

[1] 0.3901487
```

MNAR Example



Crucial Nuance

In our previous MAR example, ignoring the predictor of missingness actually produces *Indirect MNAR*.

Crucial Nuance

In our previous MAR example, ignoring the predictor of missingness actually produces *Indirect MNAR*.

QUESTION: What happens if we ignore the predictor of missingness, but that predictor is independent of our study variables?

Crucial Nuance

In our previous MAR example, ignoring the predictor of missingness actually produces *Indirect MNAR*.

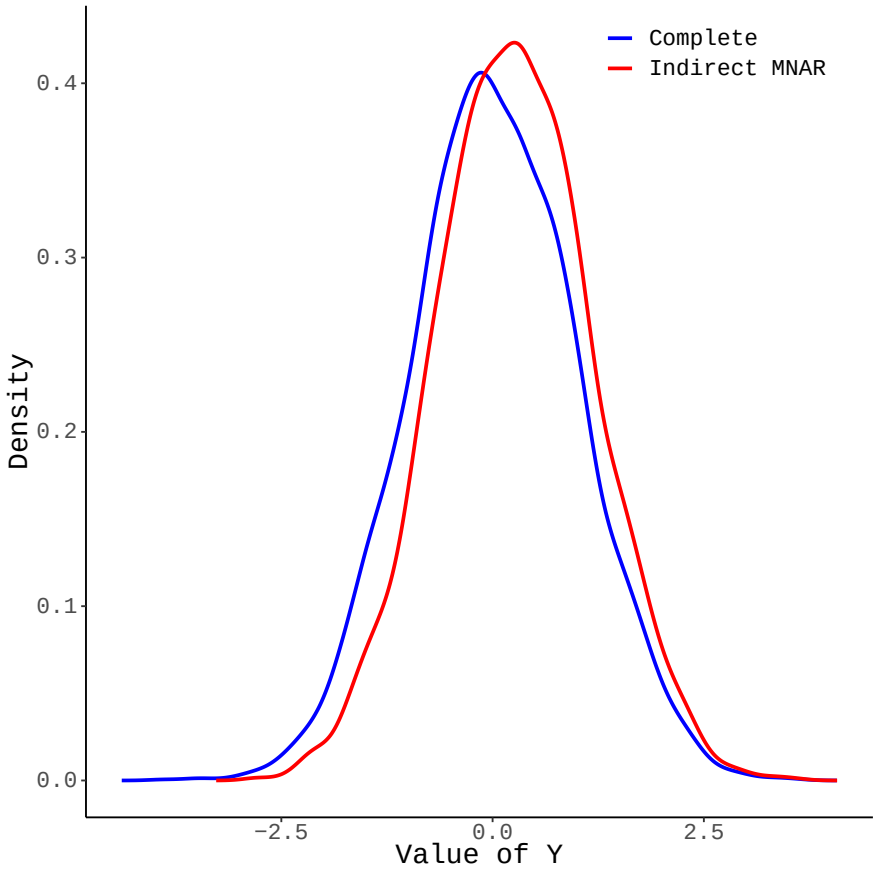
QUESTION: What happens if we ignore the predictor of missingness, but that predictor is independent of our study variables?

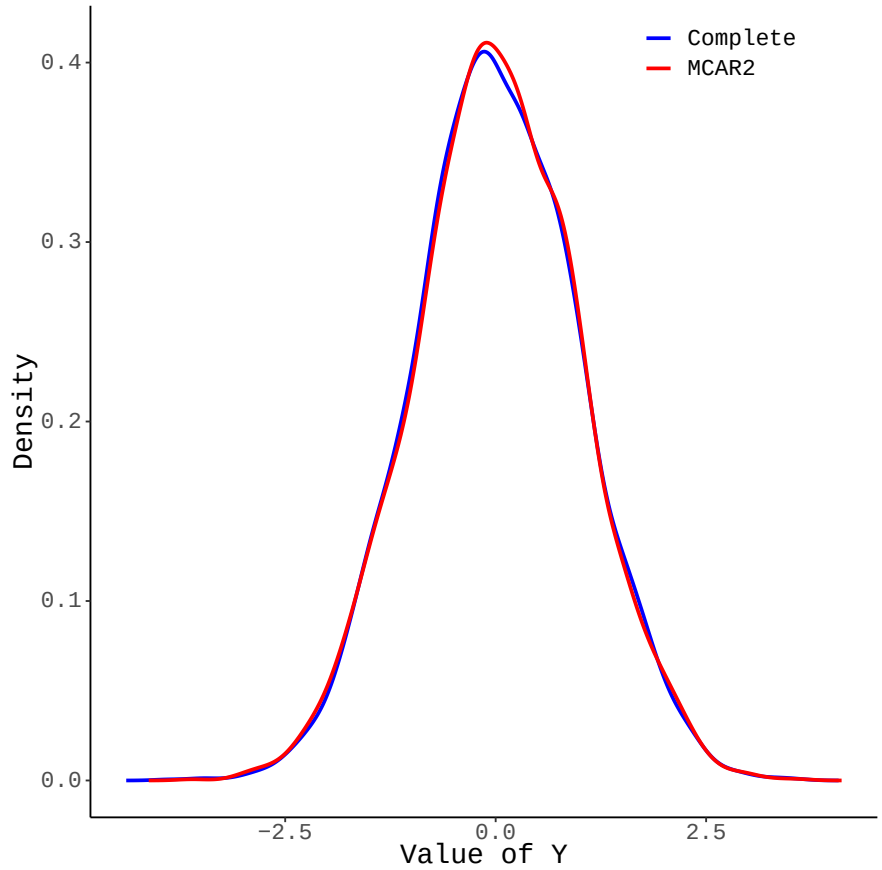
```
m <- with(dat0, z < quantile(z, probs = pm))

mcarData2          <- dat0
mcarData2[m, "y"] <- NA

mcarData2 %$% cor(y, x, use = "pairwise")

[1] 0.5119953
```

ANSWER: We get back to MCAR :)

Crucial Nuance

The missing data mechanisms are not simply characteristics of an incomplete dataset; we also need to account for the analysis.

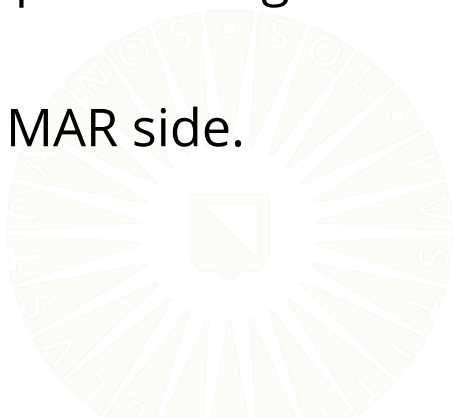




Testing the Missing Data Mechanism

We cannot fully test the MAR or MNAR assumptions.

- To do so would require knowing the values of the missing data.
- We can find observed predictors of missingness.
  - Use classification algorithms to predict missingness from Y_{obs}.
  - We can never know that we have discovered all MAR predictors.
- In practice, MAR and MNAR live on the ends of a continuum.
  - Our missing data problem exists at some unknown point along this continuum.
  - We can do a lot to nudge our problem towards the MAR side.



Testing the Missing Data Mechanism

We can (partially) test the MCAR assumption.

- With MCAR, the missing data and the observed data should have the same distribution.
- We can test for MCAR by testing the distributions of *auxiliary variables*, $\mathbf{Z}$.
  - Use a t-test to compare the subset of Z_p that corresponds to Y_{mis} to the subset corresponding to Y_{obs}.
  - The Little (1988) MCAR test is a multivariate version of this.

These procedures actually test if the data are *observed* completely at random.

Missing Data Treatments

Bad Methods (These almost never work)

Listwise Deletion (Complete Case Analysis)

- Use only complete observations for the analysis
  - Very wasteful (can throw out lots of useful data)
  - Loss of statistical power

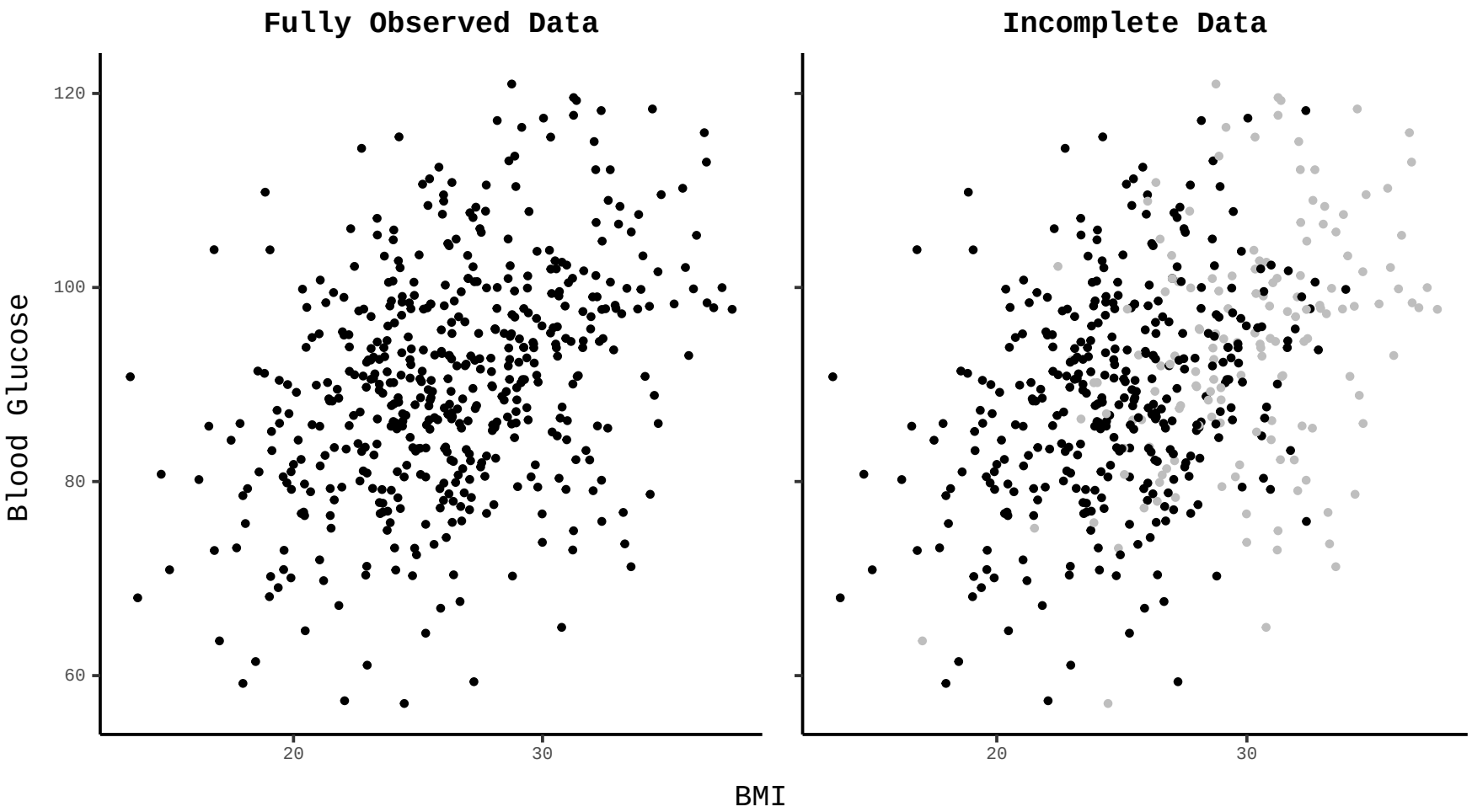
Pairwise Deletion (Available Case Analysis)

- Use only complete pairs of observations for analysis
  - Different samples sizes for different parameter estimates
  - Can cause computational issues

Example

```
## Read in some data:
tmp <- readRDS(paste0(dataDir, "diabetes.rds")) %>%
    select(bmi, bp, glu, age)
diabetes1 <- diabetes2 <-
    rmvnorm(500, colMeans(tmp), cov(tmp)) %>% data.frame()

## Simulated missingness based on 'bmi':
m <- simLinearMissingness(data  = diabetes2,
                          pm    = 0.30,
                          preds = "bmi",
                          auc   = 0.85)$r

## Impose missing on 'glu' according to the missingess above:
diabetes2[m, "glu"] <- NA
```

Example


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

Example

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

          bmi       glu       bp
bmi 1.0000000 0.3854220 0.3885409
glu 0.3854220 1.0000000 0.3843339
bp  0.3885409 0.3843339 1.0000000

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

          bmi       glu       bp
bmi 1.0000000 0.3204328 0.3174496
glu 0.3204328 1.0000000 0.3610424
bp  0.3174496 0.3610424 1.0000000
```

Example

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

          bmi       glu       bp
bmi 1.0000000 0.3854220 0.3885409
glu 0.3854220 1.0000000 0.3843339
bp  0.3885409 0.3843339 1.0000000

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

          bmi       glu       bp
bmi 1.0000000 0.3204328 0.3885409
glu 0.3204328 1.0000000 0.3610424
bp  0.3885409 0.3610424 1.0000000
```

Example

```
mean(diabetes1$glu)

[1] 90.37821

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

[1] 88.84524

var(diabetes1$glu)

[1] 133.8721

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

[1] 120.0214
```

Example

```
s1 <- lm(glu ~ bmi + bp + age, data = diabetes1) %>% summary()
s2 <- lm(glu ~ bmi + bp + age, data = diabetes2) %>% summary()

s1$r.squared

[1] 0.2291948

s2$r.squared

[1] 0.1925953
```

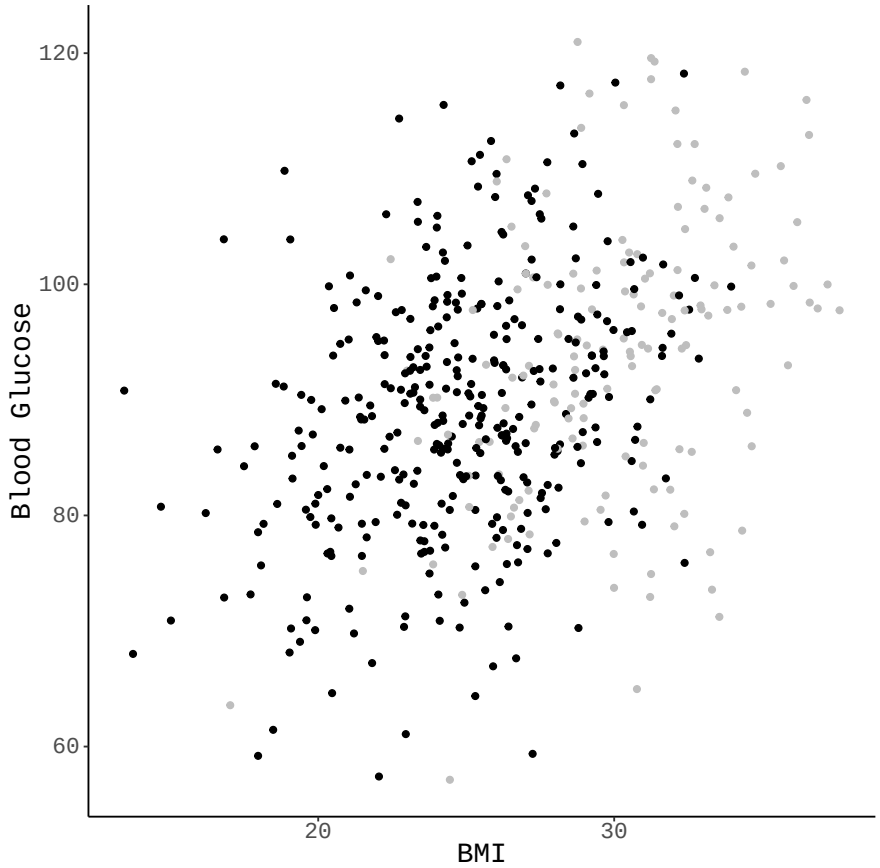
Example

```
s1$coef

              Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 48.7487495 3.50863686 13.893928 2.646709e-37
bmi          0.7048819 0.11361568  6.204090 1.161375e-09
bp           0.1880635 0.03657436  5.141948 3.920295e-07
age          0.1120532 0.03510684  3.191777 1.503693e-03

s2$coef

              Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 49.9610607 4.41348938 11.320082 1.739952e-25
bmi          0.6610981 0.15319065  4.315526 2.080933e-05
bp           0.1926684 0.04250596  4.532738 8.032382e-06
age          0.1023814 0.04020184  2.546685 1.130773e-02
```

Bad Methods (These almost never work)

(Unconditional) Mean Substitution

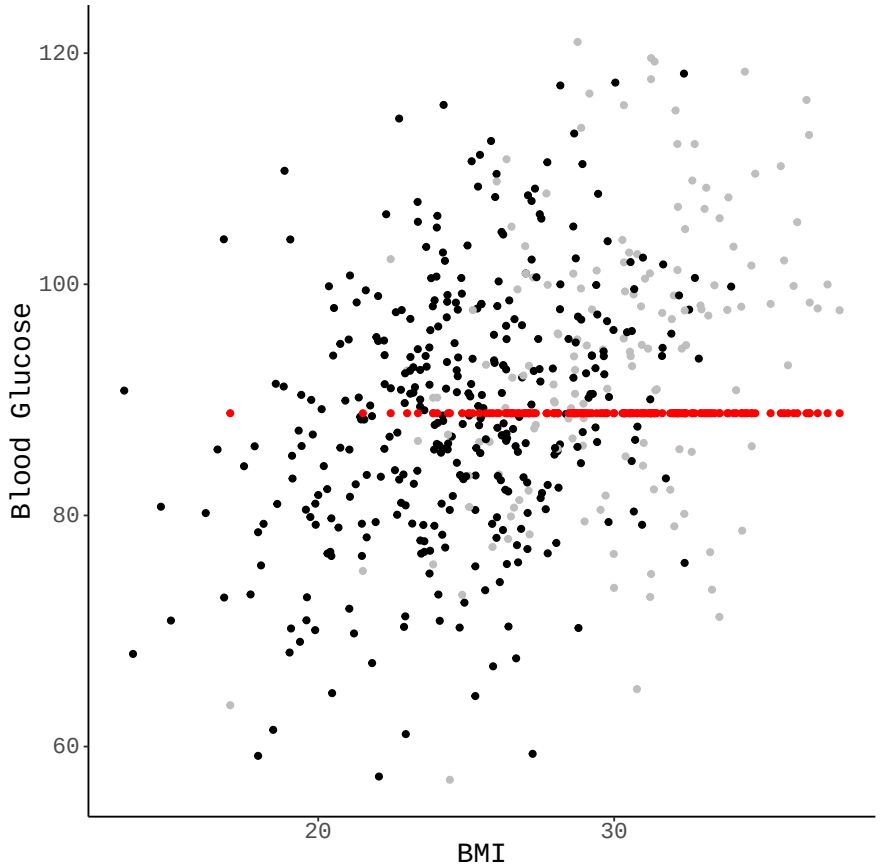
- Replace Y_{mis} with $\bar{Y}_{obs}$
  - Negatively biases regression slopes and correlations
  - Attenuates measures of linear association



Bad Methods (These almost never work)

(Unconditional) Mean Substitution

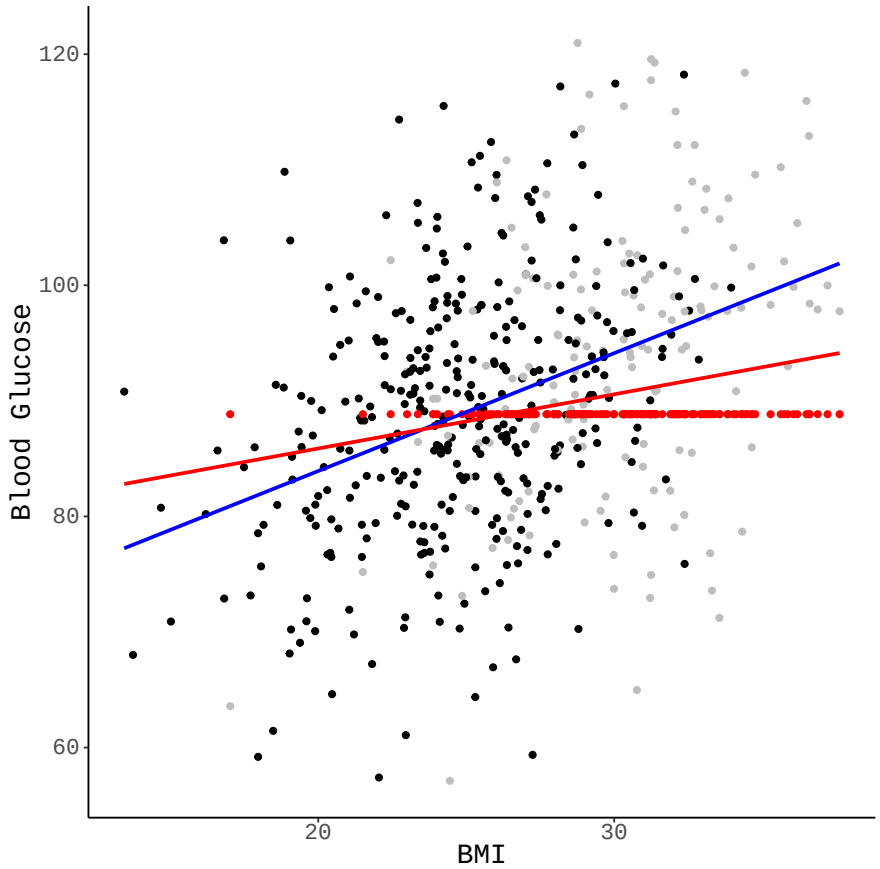
- Replace Y_{mis} with $\bar{Y}_{obs}$
  - Negatively biases regression slopes and correlations
  - Attenuates measures of linear association



Bad Methods (These almost never work)

(Unconditional) Mean Substitution

- Replace Y_{mis} with $\bar{Y}_{obs}$
  - Negatively biases regression slopes and correlations
  - Attenuates measures of linear association



Example

```
imputed <- diabetes2
imputed[m, "glu"] <- mean(imputed$glu, na.rm = TRUE)

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

          bmi       glu       bp
bmi 1.0000000 0.3854220 0.3885409
glu 0.3854220 1.0000000 0.3843339
bp  0.3885409 0.3843339 1.0000000

imputed %>% select(bmi, glu, bp) %>% cor()

          bmi       glu       bp
bmi 1.0000000 0.2237473 0.3885409
glu 0.2237473 1.0000000 0.2941464
bp  0.3885409 0.2941464 1.0000000
```

Example

```
mean(diabetes1$glu)

[1] 90.37821

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

[1] 88.84524

var(diabetes1$glu)

[1] 133.8721

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

[1] 83.94286
```

Example

```
s1 <- lm(glu ~ bmi + bp + age, data = diabetes1) %>% summary()
s2 <- lm(glu ~ bmi + bp + age, data = imputed) %>% summary()

s1$r.squared

[1] 0.2291948

s2$r.squared

[1] 0.1123001
```

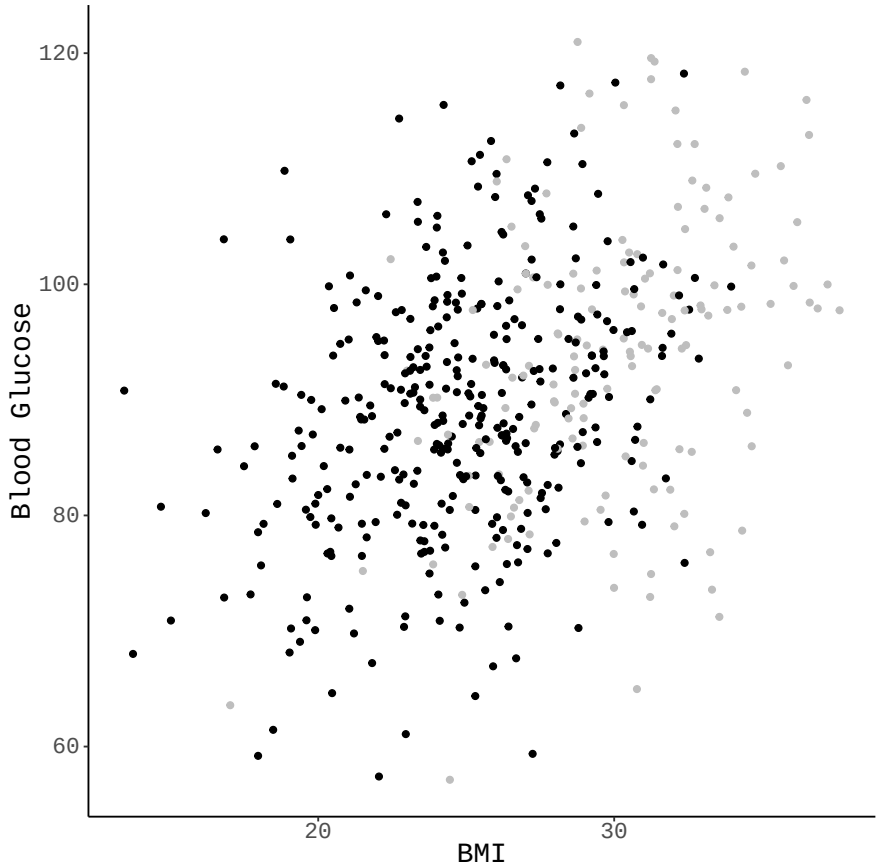
Example

```
s1$coef

              Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 48.7487495 3.50863686 13.893928 2.646709e-37
bmi          0.7048819 0.11361568  6.204090 1.161375e-09
bp           0.1880635 0.03657436  5.141948 3.920295e-07
age          0.1120532 0.03510684  3.191777 1.503693e-03

s2$coef

               Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 66.18746060 2.98157675 22.198812 2.501941e-76
bmi          0.24912441 0.09654857  2.580301 1.015781e-02
bp           0.13226112 0.03108024  4.255473 2.494880e-05
age          0.07616808 0.02983316  2.553135 1.097403e-02
```

Bad Methods (These almost never work)

Deterministic Regression Imputation
(Conditional Mean Substitution)

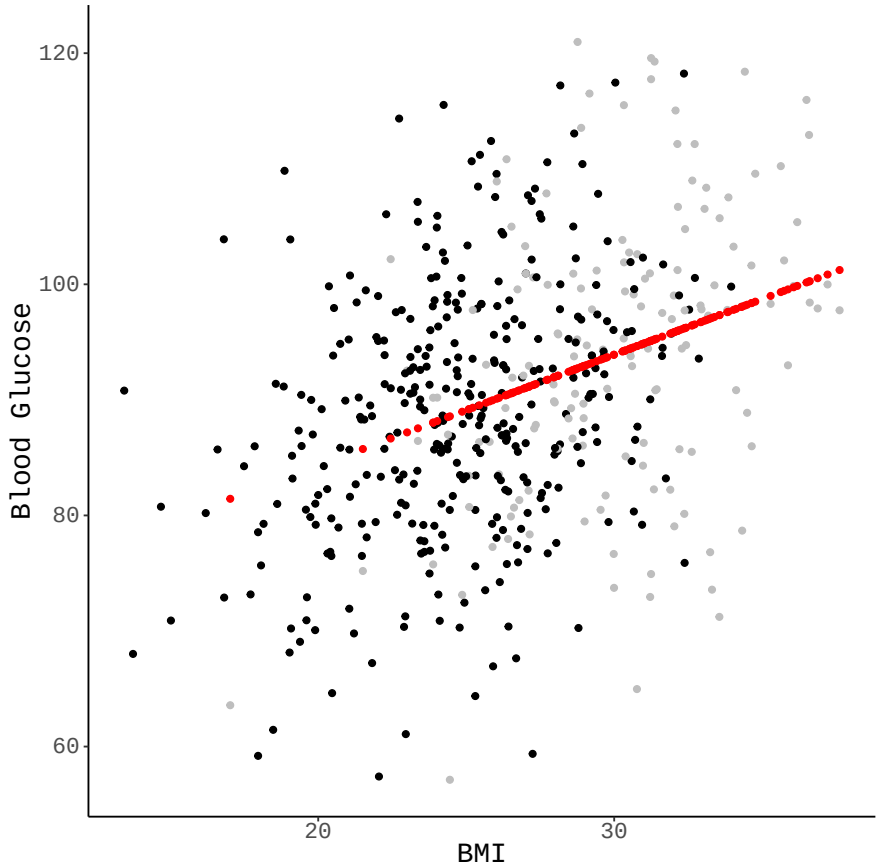
- Replace Y_{mis} with $\widehat{Y}_{mis}$ from some regression equation
  - Positively biases regression slopes and correlations
  - Inflates measures of linear association



Bad Methods (These almost never work)

Deterministic Regression Imputation
(Conditional Mean Substitution)

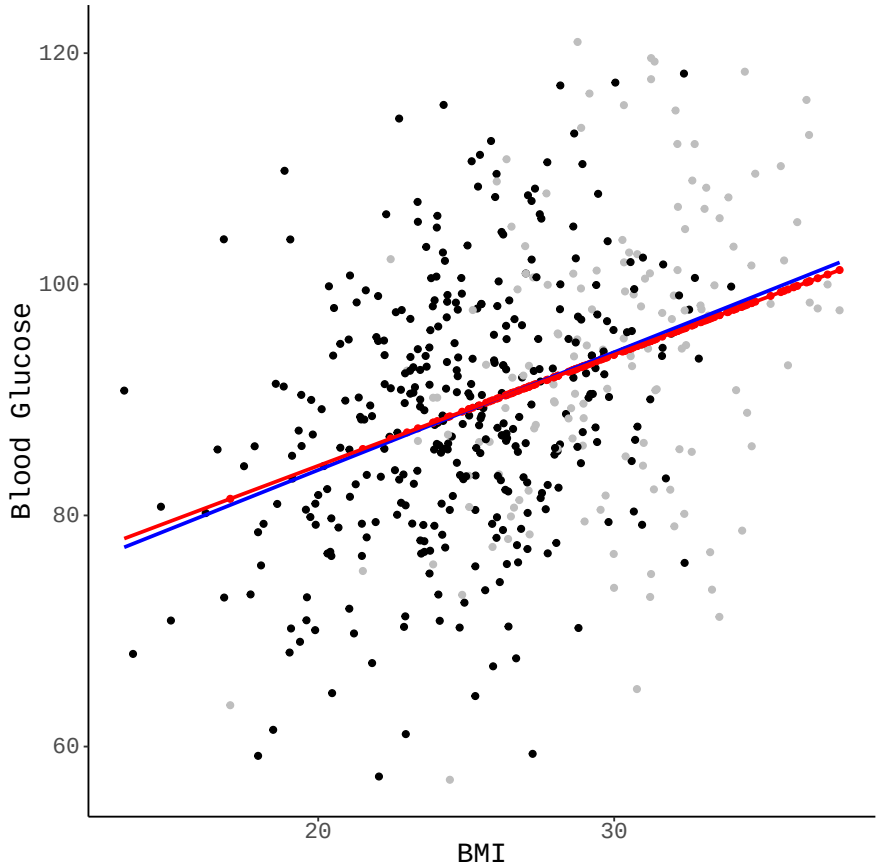
- Replace Y_{mis} with $\widehat{Y}_{mis}$ from some regression equation
  - Positively biases regression slopes and correlations
  - Inflates measures of linear association



Bad Methods (These almost never work)

Deterministic Regression Imputation
(Conditional Mean Substitution)

- Replace Y_{mis} with $\widehat{Y}_{mis}$ from some regression equation
  - Positively biases regression slopes and correlations
  - Inflates measures of linear association



Example

```
imputed <- mice(data       = diabetes2,
                m          = 1,
                maxit      = 1,
                printFlag = FALSE,
                method     = "norm.predict") %>%
    complete(1)
```

Example

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

          bmi       glu       bp
bmi 1.0000000 0.3854220 0.3885409
glu 0.3854220 1.0000000 0.3843339
bp  0.3885409 0.3843339 1.0000000

imputed %>% select(bmi, glu, bp) %>% cor()

          bmi       glu       bp
bmi 1.0000000 0.4340619 0.3885409
glu 0.4340619 1.0000000 0.4467209
bp  0.3885409 0.4467209 1.0000000
```

Example

```
mean(diabetes1$glu)

[1] 90.37821

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

[1] 90.40121

var(diabetes1$glu)

[1] 133.8721

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

[1] 96.52101
```

Example

```
s1 <- lm(glu ~ bmi + bp + age, data = diabetes1) %>% summary()
s2 <- lm(glu ~ bmi + bp + age, data = imputed) %>% summary()

s1$r.squared

[1] 0.2291948

s2$r.squared

[1] 0.2978124
```

Example

```
s1$coef

              Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 48.7487495 3.50863686 13.893928 2.646709e-37
bmi          0.7048819 0.11361568  6.204090 1.161375e-09
bp           0.1880635 0.03657436  5.141948 3.920295e-07
age          0.1120532 0.03510684  3.191777 1.503693e-03

s2$coef

              Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 49.9610607 2.84353479 17.570054 4.384784e-54
bmi          0.6610981 0.09207853  7.179720 2.568843e-12
bp           0.1926684 0.02964127  6.500004 1.960043e-10
age          0.1023814 0.02845194  3.598399 3.523183e-04
```

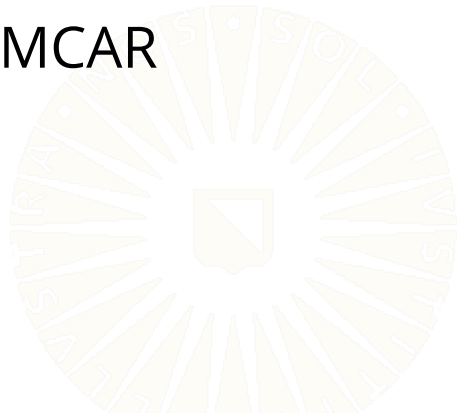
Bad Methods (These almost never work)

General Issues with Deletion-Based Methods

- Biased parameter estimates unless data are MCAR
- Generalizability issues

General Issues with Simple Single Imputation Methods

- Biased parameter estimates even when data are MCAR
- Attenuates variability in any treated variables



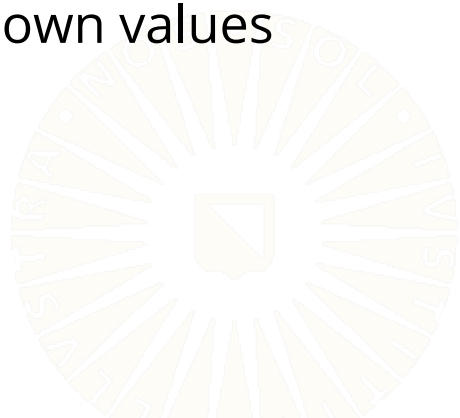
Bad Methods (These almost never work)

Averaging Available Items (Person-Mean Imputation)

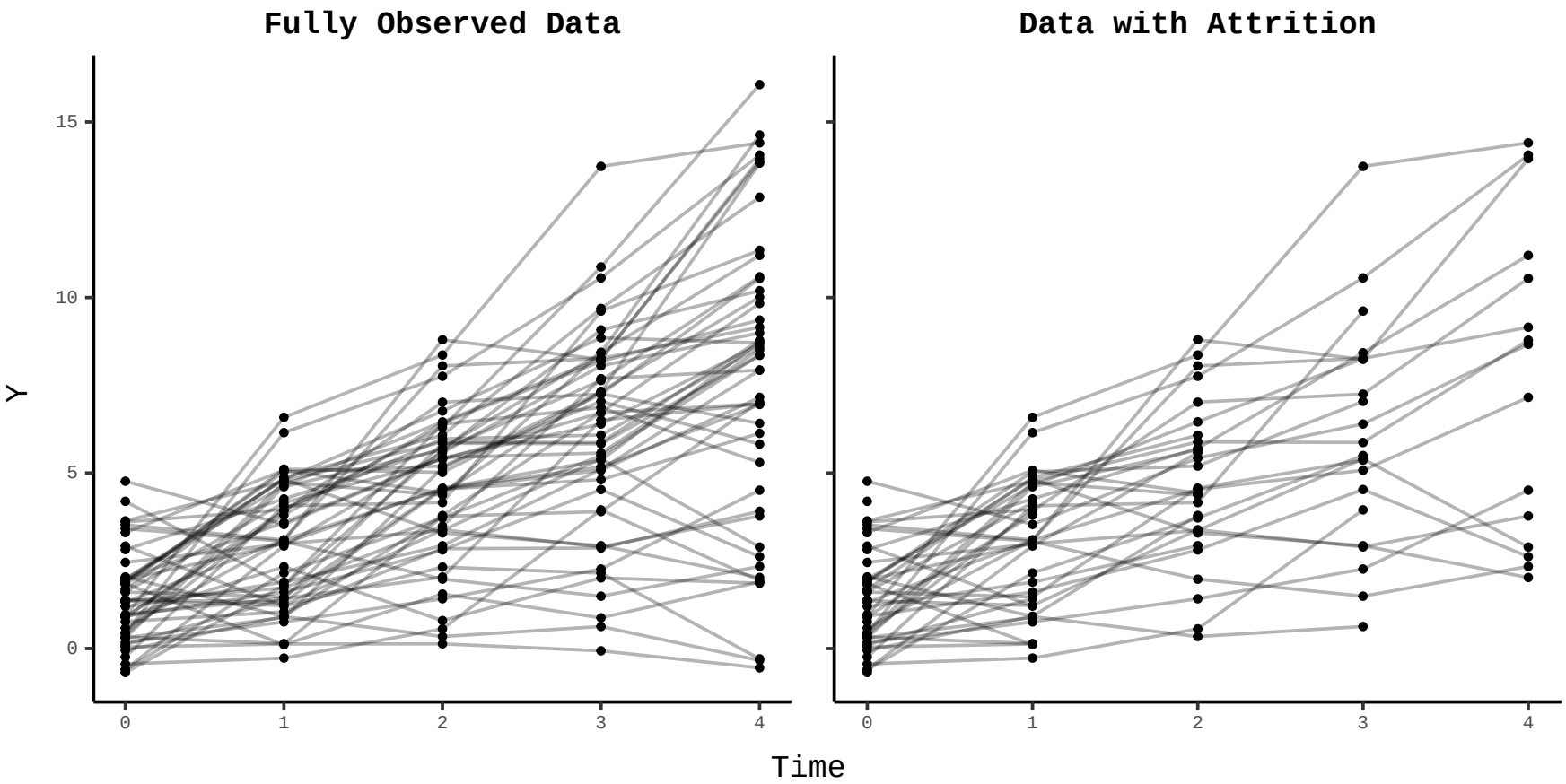
- Compute aggregate scores using only available values
  - Missing data must be MCAR
  - Each item must contributes equally to the aggregate score

Last Observation Carried Forward (LOCF)

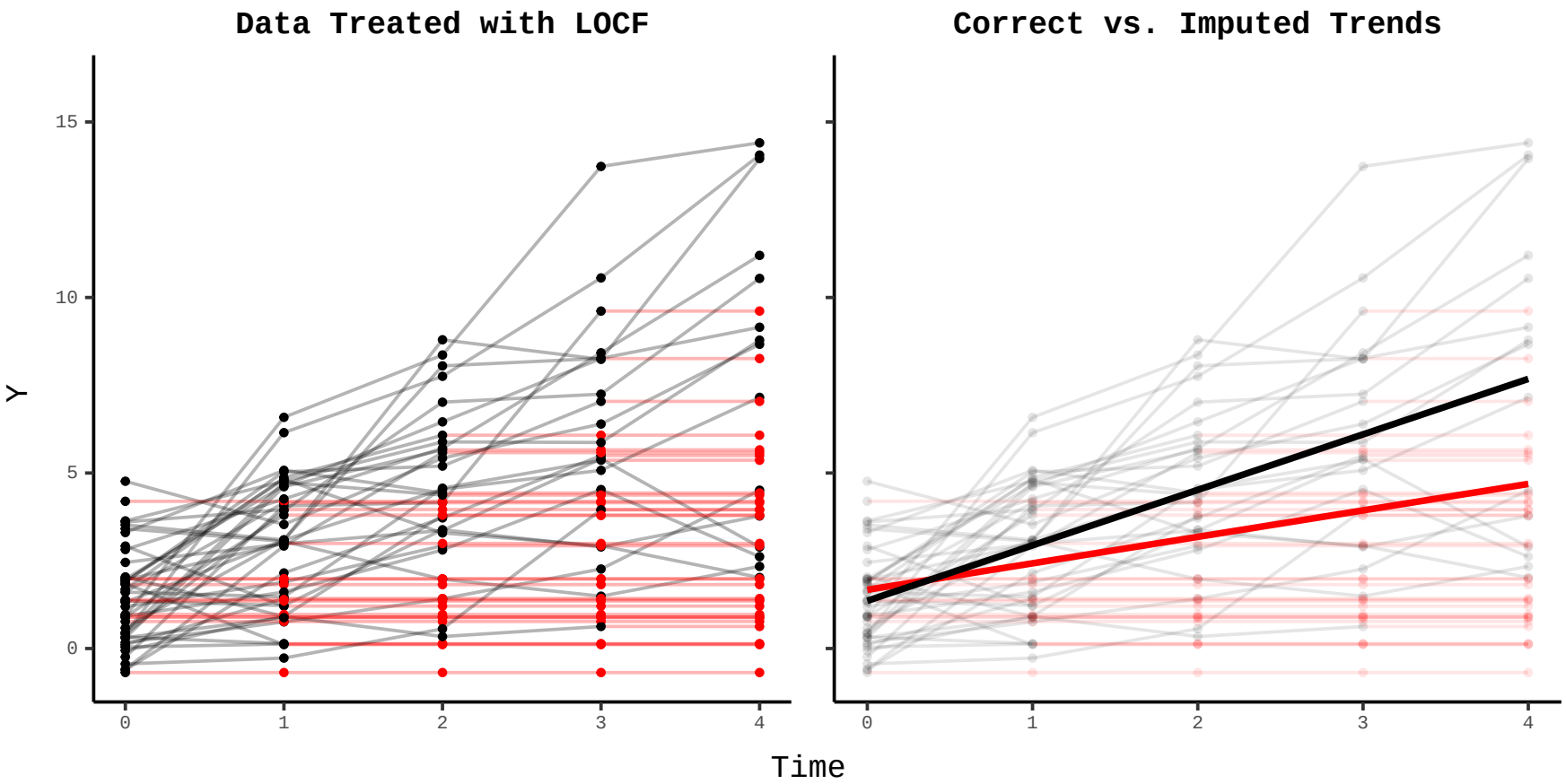
- Replace post-dropout values with the most recent observed value
  - Assume that dropouts would maintain their last known values
  - Attenuates estimates of growth/development



LOCF

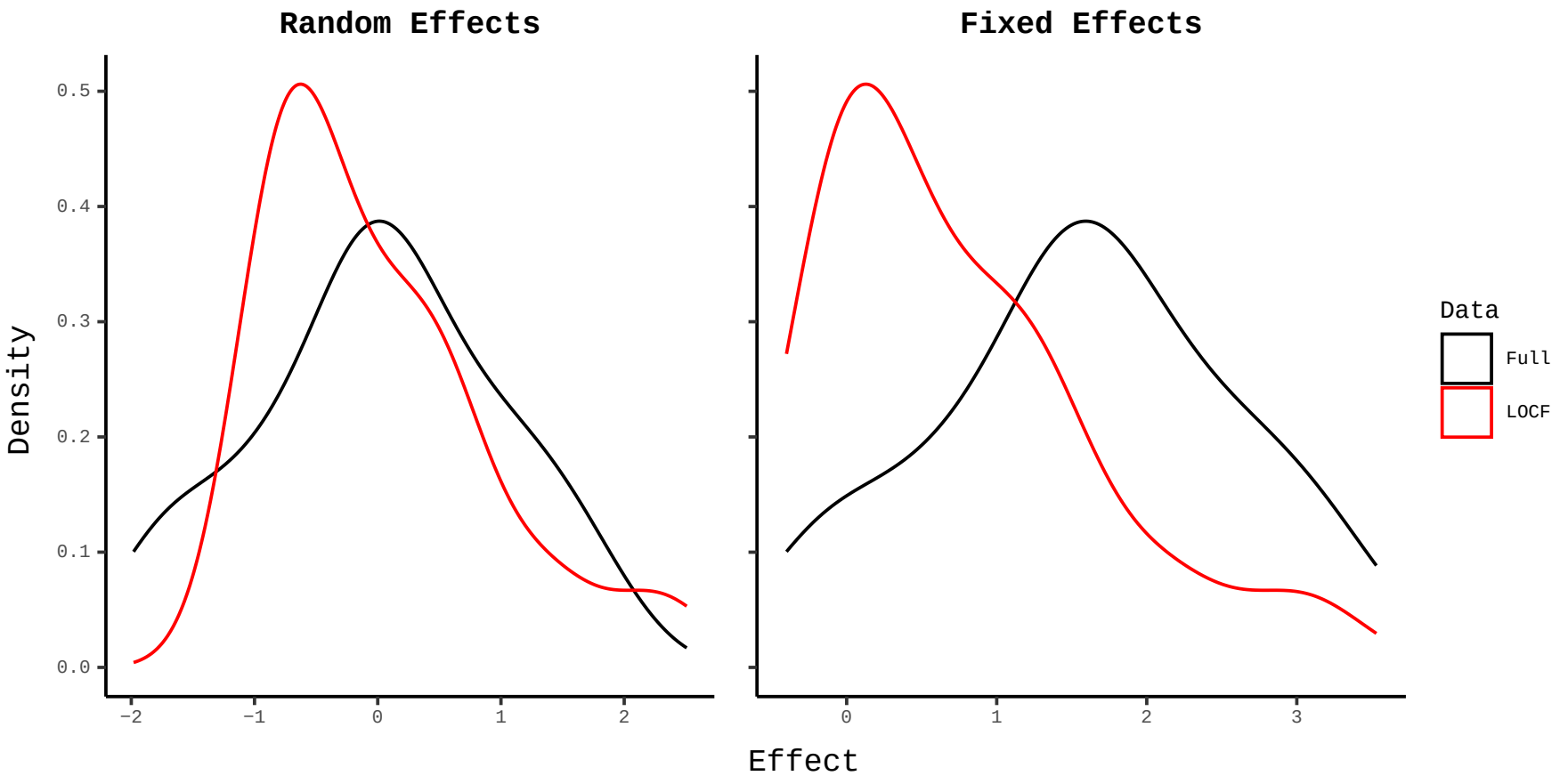

Fully Observed Data
Data with Attrition
Y
Time
0
5
10
15
0
1
2
3
4

LOCF


Data Treated with LOCF
Correct vs. Imputed Trends
Y
Time
15
10
5
0
0
1
2
3
4

Example

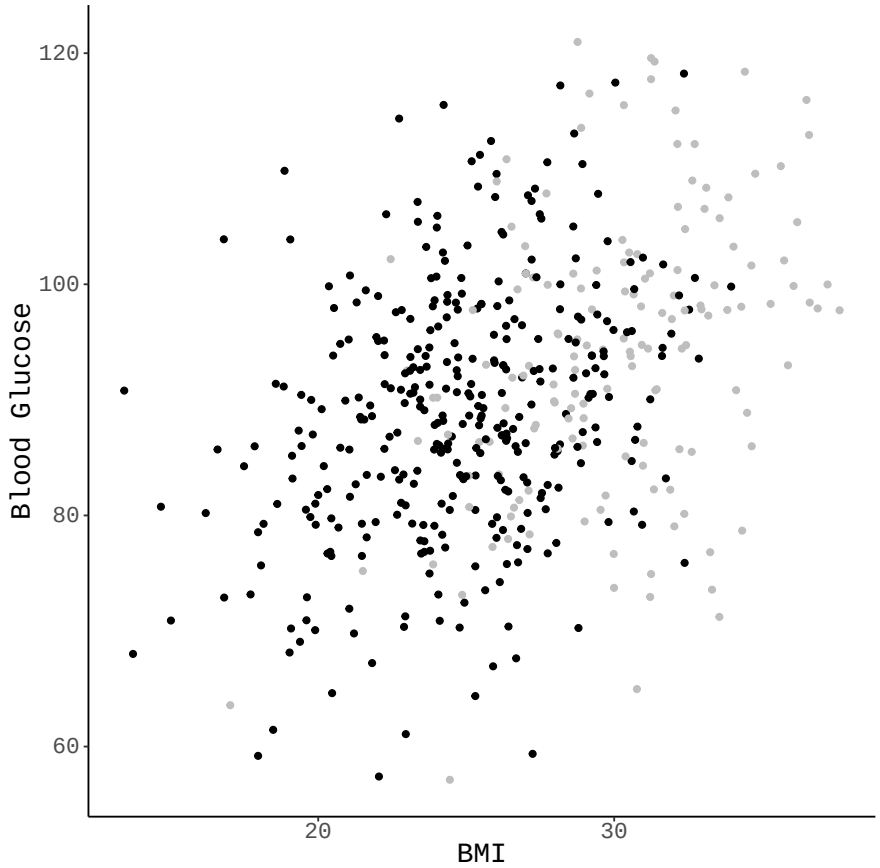
```
## Fit some multilevel regression models
fit1 <- lmer(y ~ t + (t | id), data = dat1) # Full data
fit2 <- lmer(y ~ t + (t | id), data = dat3) # LOCF data
```



OK Methods (These sometimes work)

Stochastic Regression Imputation

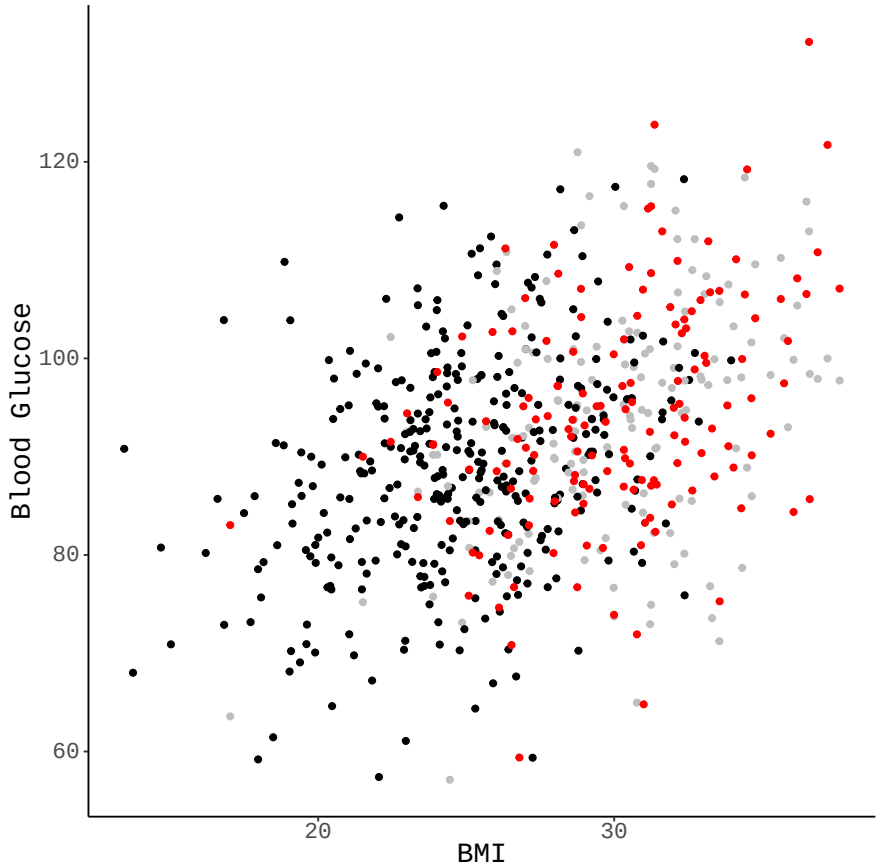
- Fill Y_{mis} with $\widehat{Y}_{mis}$ plus some random noise.
  - Produces unbiased parameter estimates and predictions
  - Computationally efficient
  - Attenuates standard errors
  - Makes CIs and prediction intervals too narrow



OK Methods (These sometimes work)

Stochastic Regression Imputation

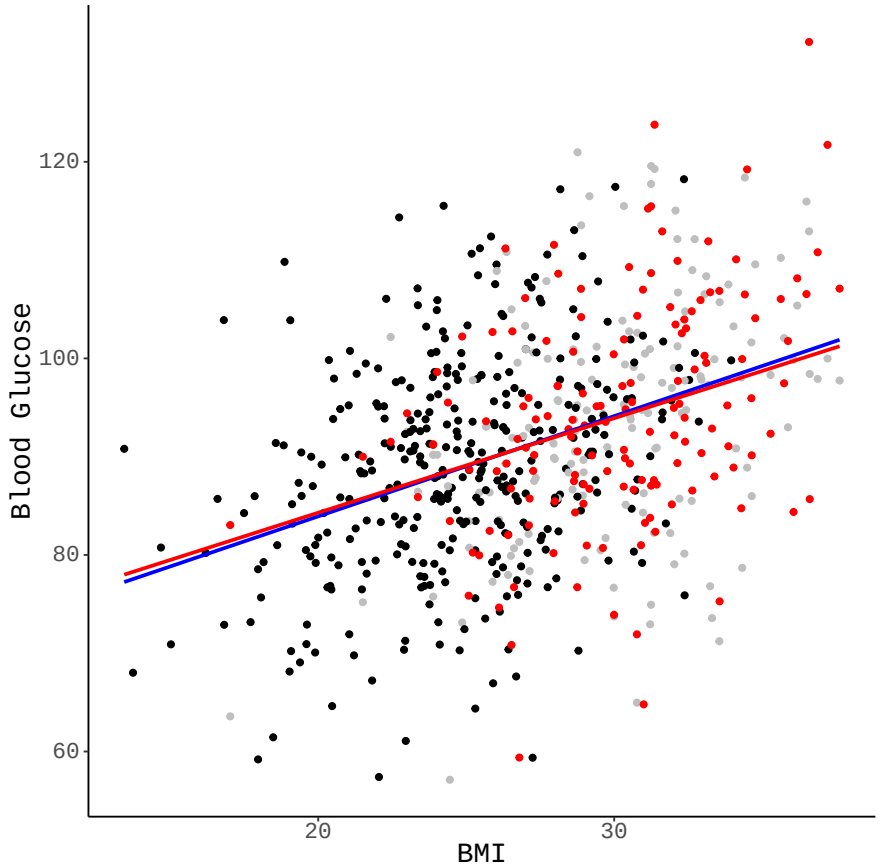
- Fill Y_{mis} with $\widehat{Y}_{mis}$ plus some random noise.
  - Produces unbiased parameter estimates and predictions
  - Computationally efficient
  - Attenuates standard errors
  - Makes CIs and prediction intervals too narrow



OK Methods (These sometimes work)

Stochastic Regression Imputation

- Fill Y_{mis} with $\widehat{Y}_{mis}$ plus some random noise.
  - Produces unbiased parameter estimates and predictions
  - Computationally efficient
  - Attenuates standard errors
  - Makes CIs and prediction intervals too narrow



Example

```
imputed <- mice(data       = diabetes2,
                m          = 1,
                maxit      = 1,
                printFlag = FALSE,
                method     = "norm.nob") %>%
    complete(1)
```

Example

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

          bmi       glu       bp
bmi 1.0000000 0.3854220 0.3885409
glu 0.3854220 1.0000000 0.3843339
bp  0.3885409 0.3843339 1.0000000

imputed %>% select(bmi, glu, bp) %>% cor()

          bmi       glu       bp
bmi 1.0000000 0.3971192 0.3885409
glu 0.3971192 1.0000000 0.3905143
bp  0.3885409 0.3905143 1.0000000
```

Example

```
mean(diabetes1$glu)

[1] 90.37821

mean(imputed$glu)

[1] 90.4662

var(diabetes1$glu)

[1] 133.8721

var(imputed$glu)

[1] 129.1035
```

Example

```
s1 <- lm(glu ~ bmi + bp + age, data = diabetes1) %>% summary()
s2 <- lm(glu ~ bmi + bp + age, data = imputed) %>% summary()

s1$r.squared

[1] 0.2291948

s2$r.squared

[1] 0.2353581
```

Example

```
s1$coef

              Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 48.7487495 3.50863686 13.893928 2.646709e-37
bmi          0.7048819 0.11361568  6.204090 1.161375e-09
bp           0.1880635 0.03657436  5.141948 3.920295e-07
age          0.1120532 0.03510684  3.191777 1.503693e-03

s2$coef

               Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 48.84777739 3.43177719 14.233959 8.522463e-39
bmi          0.72466078 0.11112683  6.521024 1.722970e-10
bp           0.19092709 0.03577317  5.337159 1.439093e-07
age          0.09554028 0.03433779  2.782365 5.602110e-03
```

OK Methods (These sometimes work)

Nonresponse Weighting

- Weight the observed cases to correct for nonresponse bias
  - Popular in survey research and official statistics
  - Only worth considering with *Unit Nonresponse*
  - Doesn't make any sense with *Item Nonresponse*

Good Methods (These almost always work)

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 (These almost always work)

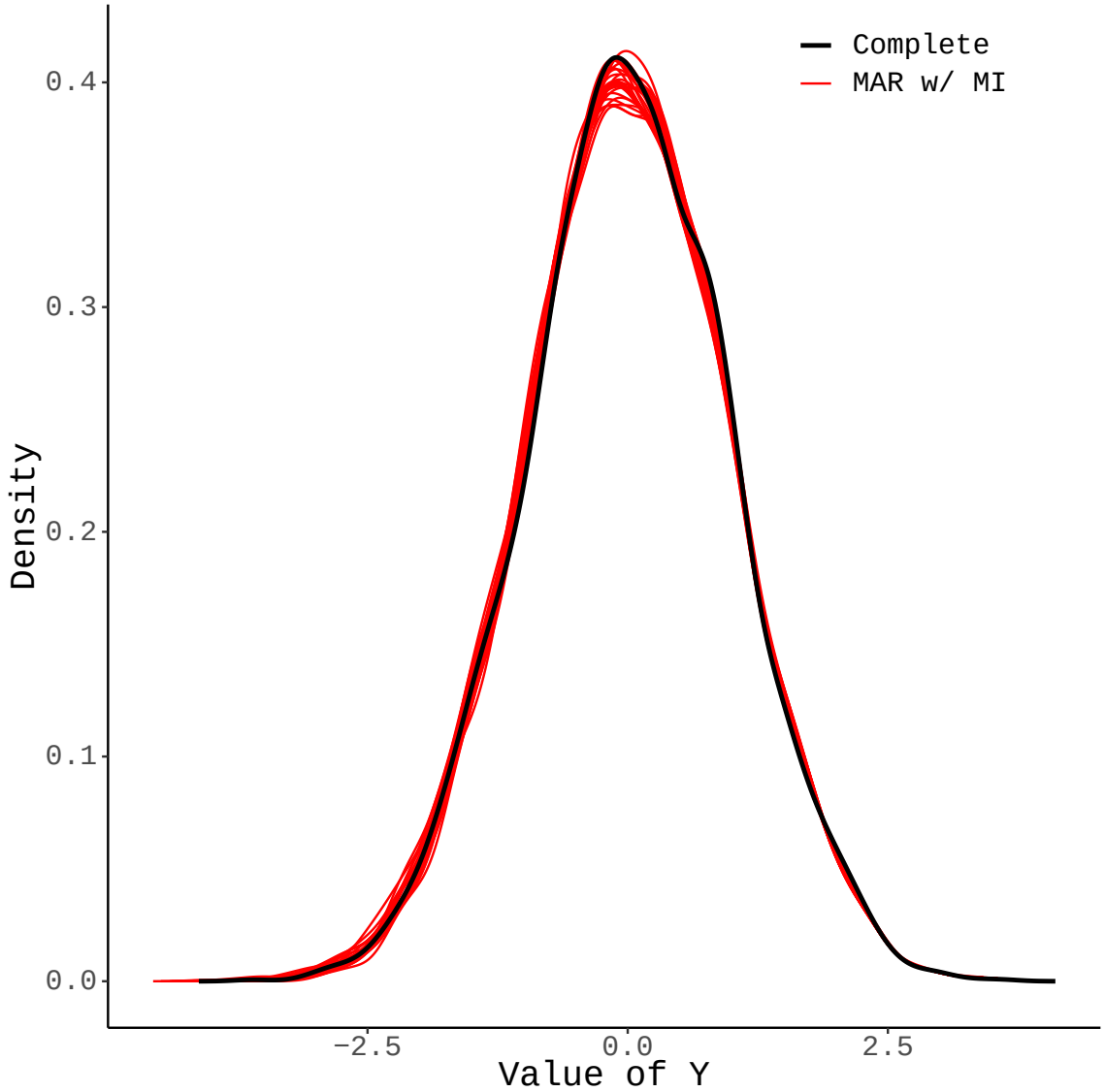
What happens when we apply MI to our previous MAR example?

```
## Estimate imputation model:
miceOut <- mice(data       = marData,
                m          = 25,
                maxit      = 1,
                method     = "norm",
                printFlag = FALSE)

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

[1] 0.5071492
```

The MI-based parameter estimate looks good.

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

Good Methods (These almost always work)



Good Methods (These *almost* always work)

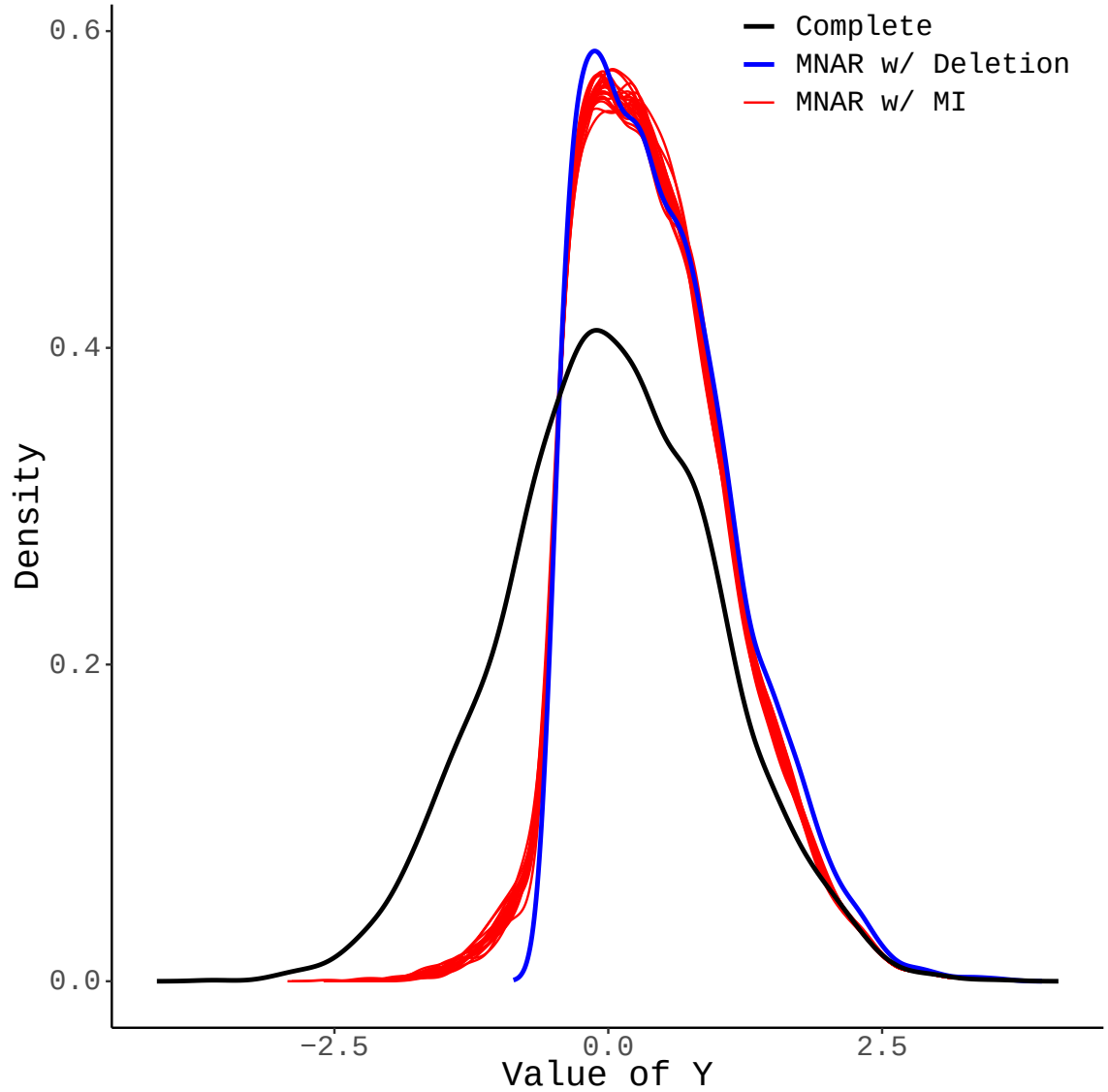
What about applying MI to our MNAR example?

```
## Estimate imputation model:
miceOut <- mice(data       = mnarData,
                m          = 25,
                maxit      = 1,
                method     = "norm",
                printFlag = FALSE)

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

[1] 0.4177734
```

The MI-based parameter estimate is still biased.

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

Good Methods (These *almost* always work)



Example

```
miceOut <- mice(data       = diabetes2,
                m          = 25,
                maxit      = 1,
                printFlag = FALSE,
                method     = "norm")
```

Example

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

          bmi       glu       bp
bmi 1.0000000 0.3854220 0.3885409
glu 0.3854220 1.0000000 0.3843339
bp  0.3885409 0.3843339 1.0000000

pooledCorMat(miceOut, c("bmi", "glu", "bp"))

          bmi       glu       bp
bmi 1.0000000 0.3829271 0.3885409
glu 0.3829271 1.0000000 0.3872346
bp  0.3885409 0.3872346 1.0000000
```

Example

```
mean(diabetes1$glu)

[1] 90.37821

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

[1] 90.43268

var(diabetes1$glu)

[1] 133.8721

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

[1] 126.8673
```

Example

```
fit1 <- lm(glu ~ bmi + bp + age, data = diabetes1)
fit2 <- with(miceOut, lm(glu ~ bmi + bp + age))

summary(fit1)$r.squared

[1] 0.2291948

pool.r.squared(fit2)

          est    lo 95     hi 95       fmi
R^2 0.2297497 0.149106 0.3167648 0.4221505
```

Example

```
summary(fit1)$coef

              Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 48.7487495 3.50863686 13.893928 2.646709e-37
bmi          0.7048819 0.11361568  6.204090 1.161375e-09
bp           0.1880635 0.03657436  5.141948 3.920295e-07
age          0.1120532 0.03510684  3.191777 1.503693e-03

pool(fit2) %>% summary() %>% select(-df)

         term   estimate  std.error statistic      p.value
1 (Intercept) 49.8940856 4.45211017 11.206840 0.000000e+00
2         bmi  0.6777821 0.15863934  4.272471 6.374475e-05
3          bp  0.1885484 0.04227740  4.459793 1.562717e-05
4         age  0.1033259 0.03906305  2.645105 8.811910e-03
```

Good Methods (These almost always work)

Bayesian Modeling

- Treat missing values as just another parameter to be estimated
  - Models can be directly estimated in the presence of missing data
    - Essentially, runs MI behind-the-scenes during model estimation
  - The predictors of nonresponse must be included in the model, somehow
  - Computationally expensive

Good Methods (These almost always work)

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)

Example

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

mean(diabetes1$glu)

[1] 90.37821

fit$mean["glu"]

     glu
90.42414
```

Example

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

          bmi       glu         bp
bmi 19.12113  19.50017  24.19964
glu 19.50017 133.87211  63.33870
bp  24.19964  63.33870 202.87609

fit$cov

    bmi     glu     bp
bmi  19.083
glu  18.605 125.492
bp   24.151  62.687 202.470
```

Example

```
mod <- "glu ~ 1 + bmi + bp + age"
fit <- sem(mod, data = diabetes2, missing = "fiml")

summary(fit1)$r.squared

[1] 0.2291948

inspect(fit, "r2")

  glu
0.229
```

Example

```
summary(fit1)$coef %>% round(3)

            Estimate Std. Error t value Pr(>|t|)
(Intercept)   48.749      3.509  13.894    0.000
bmi            0.705      0.114   6.204    0.000
bp             0.188      0.037   5.142    0.000
age            0.112      0.035   3.192    0.002

parameterEstimates(fit, ci = FALSE)[1:4, ]

  lhs op rhs    est    se      z pvalue
1 glu ~1     49.961 4.388 11.385   0.00
2 glu  ~ bmi  0.661 0.152  4.340   0.00
3 glu  ~  bp  0.193 0.042  4.559   0.00
4 glu  ~ age  0.102 0.040  2.561   0.01
```

References

Little, R. J. A. (1988). A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, *83*(404), 1198–1202. doi: 10.1080/01621459.1988.10478722