Missing Data Basics
Utrecht University Winter School: Missing Data in R

Kyle M. Lang

Department of Methodology & Statistics
Utrecht University

Utrecht
University

Outline

Missing Data Descriptives

Missing Data Mechanisms

Missing Data Treatments

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
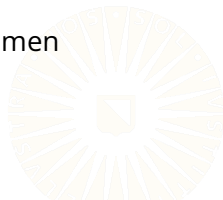
A Little Notation

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

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

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

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

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

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.
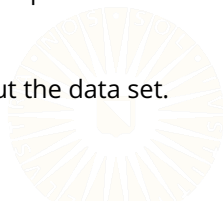
Example Missing Data Patterns

	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

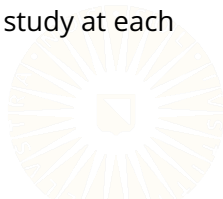|	X	Y	Z
1	x	.	z
2	x	y	z
3	x	y	z
4	x	.	z
5	x	y	z
6	x	.	z
7	.	y	z
8	x	y	z
9	x	.	.
10	x	y	.

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

Iɴfʟᴜx Cᴏᴇғғɪᴄɪᴇɴᴛ

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

- 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}
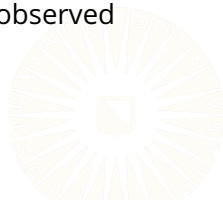
Coverage Measures

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.
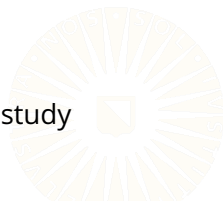
Simulate Some Toy Data

```r
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
```
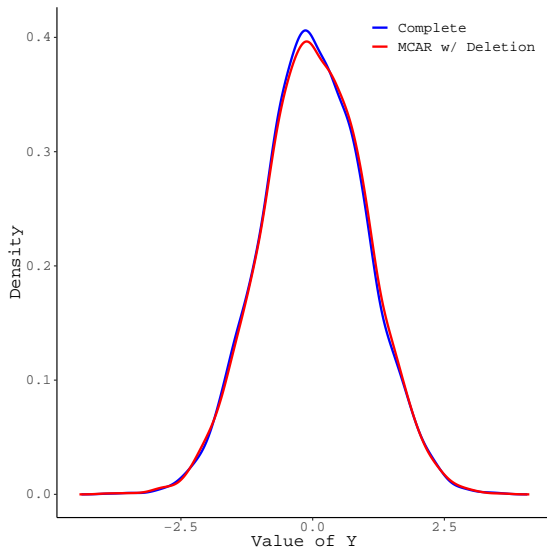
MCAR Example

```
## 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
```
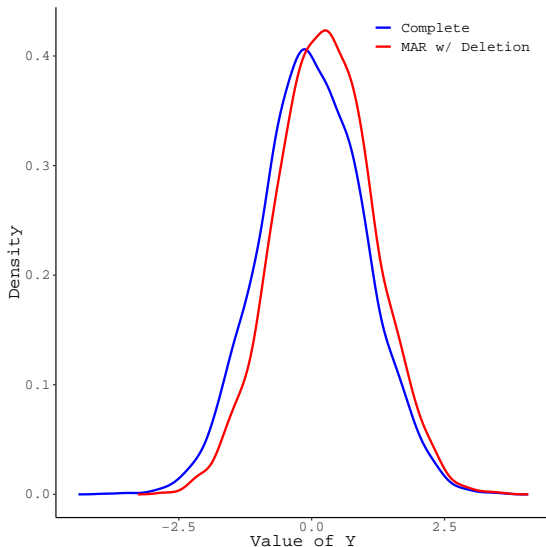
MCAR Example

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
```
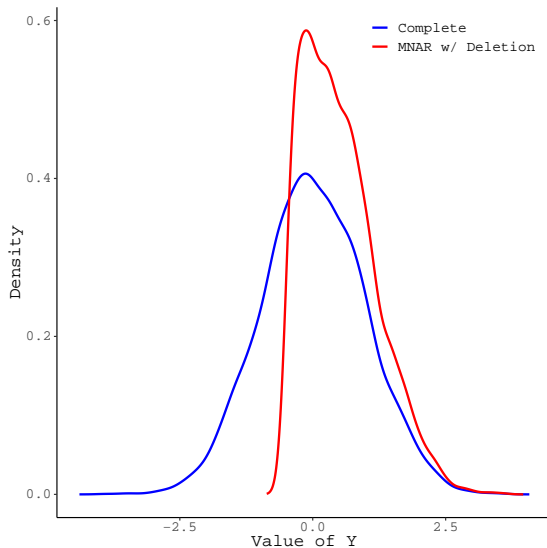
MAR Example

MNAR Example

```r
## 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?

```r
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.
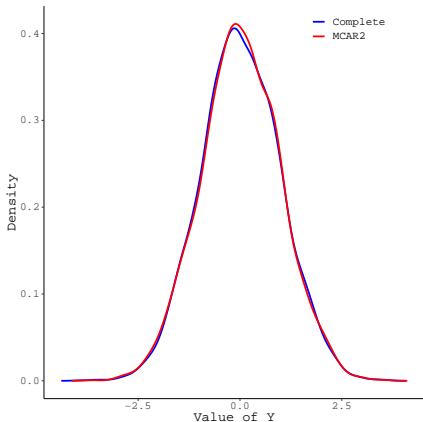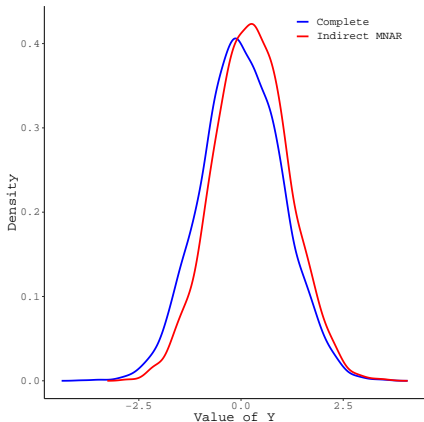
Testing the Missing Data Mechanism

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.
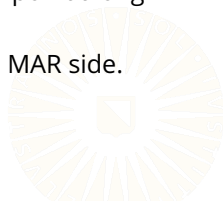
Testing the Missing Data Mechanism

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
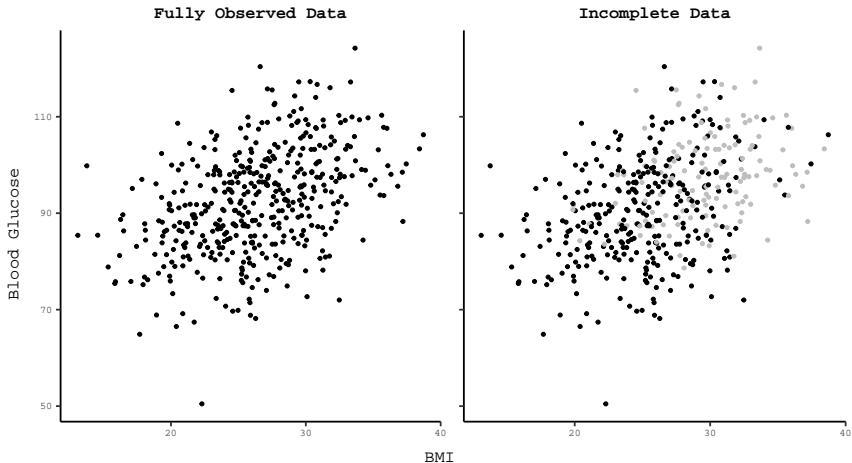 - Can cause computational issues
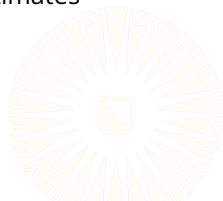
Example

```
## Read in some example data:
diabetes1 <- diabetes2 <-
    readRDS(paste0(dataDir, "diabetes_norm.rds"))

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

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

Example

Example

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

          bmi       glu        bp
bmi 1.0000000 0.4211879 0.3656630
glu 0.4211879 1.0000000 0.4028993
bp  0.3656630 0.4028993 1.0000000

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

          bmi       glu        bp
bmi 1.0000000 0.3262834 0.2776195
glu 0.3262834 1.0000000 0.3537169
bp  0.2776195 0.3537169 1.0000000
```

Example

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

          bmi       glu        bp
bmi 1.0000000 0.4211879 0.3656630
glu 0.4211879 1.0000000 0.4028993
bp  0.3656630 0.4028993 1.0000000

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

          bmi       glu        bp
bmi 1.0000000 0.3262834 0.2776195
glu 0.3262834 1.0000000 0.3537169
bp  0.2776195 0.3537169 1.0000000
```

Example

```
mean(diabetes1$glu)

[1] 92.34938

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

[1] 89.9979

var(diabetes1$glu)

[1] 117.29

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

[1] 115.4501
```

Example

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

s1$r.squared

[1] 0.2489047

s2$r.squared

[1] 0.1814826
```

Example

```
s1$coef

             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 50.6762960 3.55510857 14.254500 3.689251e-38
bmi          0.7369749 0.10360177  7.113535 4.628540e-12
bp           0.2333118 0.03608963  6.464788 2.704883e-10

s2$coef

             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 52.4139024 4.61043871 11.368528 3.224459e-25
bmi          0.6267222 0.13652617  4.590491 6.464076e-06
bp           0.2368355 0.04472047  5.295907 2.266520e-07
```
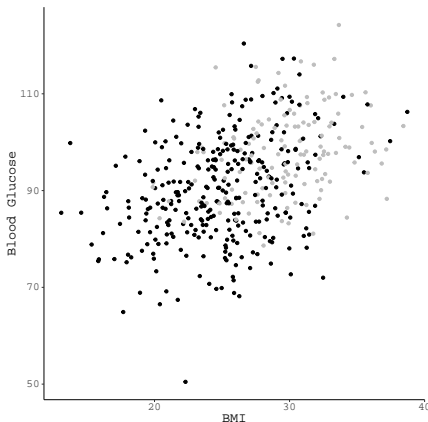
Bad Methods (These almost never work)
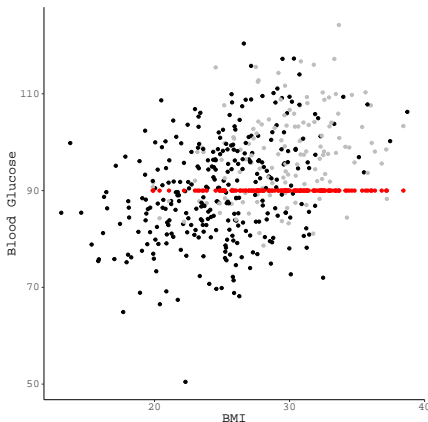
(Unconditional) Mean Substitution

- 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)
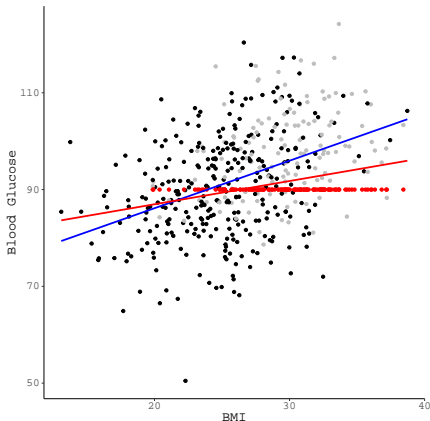
(Unconditional) Mean Substitution

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

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

          bmi       glu        bp
bmi 1.0000000 0.4211879 0.3656630
glu 0.4211879 1.0000000 0.4028993
bp  0.3656630 0.4028993 1.0000000

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

          bmi       glu        bp
bmi 1.0000000 0.2486881 0.2115973
glu 0.2486881 1.0000000 0.3537169
bp  0.2115973 0.3537169 1.0000000
```

Example

```
mean(diabetes1$glu)

[1] 92.34938

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

[1] 89.9979

var(diabetes1$glu)

[1] 117.29

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

[1] 80.63184
```

Example

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

s1$r.squared

[1] 0.2489047

s2$r.squared

[1] 0.1567534
```

Example

```
s1$coef

            Estimate Std. Error  t value     Pr(>|t|)
(Intercept) 50.6762960 3.55510857 14.254500 3.689251e-38
bmi          0.7369749 0.10360177  7.113535 4.628540e-12
bp           0.2333118 0.03608963  6.464788 2.704883e-10

s2$coef

            Estimate Std. Error  t value     Pr(>|t|)
(Intercept) 56.434068 3.73857631 15.095069 9.056479e-42
bmi          0.351768 0.08667589  4.058429 5.848672e-05
bp           0.261839 0.03725021  7.029194 7.987962e-12
```
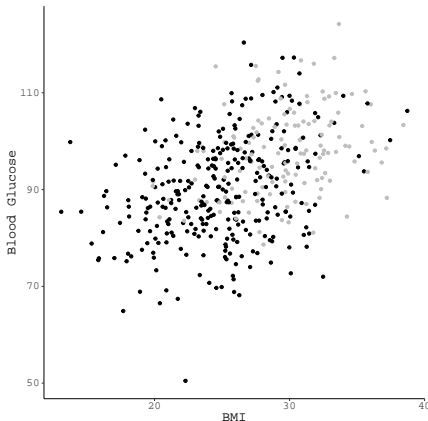
Bad Methods (These almost never work)

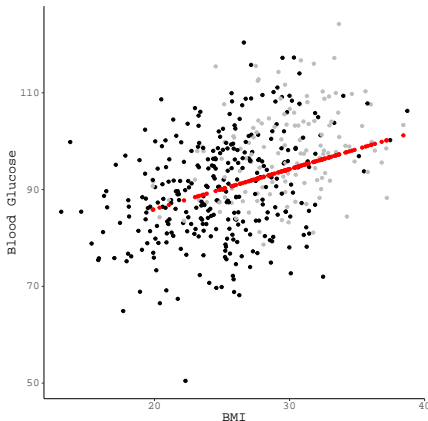Deterministic Regression
Imputation
(Conditional Mean Substitution)

- 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
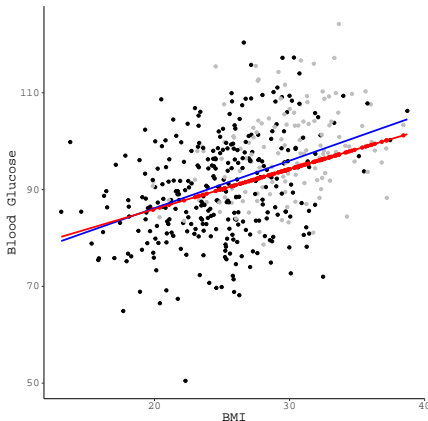Imputation
(Conditional Mean Substitution)

- 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   = 20,
                printFlag = FALSE,
                method  = "norm.predict") %>%
    complete(1)
```

Example

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

          bmi       glu        bp
bmi 1.0000000 0.4211879 0.3656630
glu 0.4211879 1.0000000 0.4028993
bp  0.3656630 0.4028993 1.0000000

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

          bmi       glu        bp
bmi 1.0000000 0.4520298 0.4189667
glu 0.4520298 1.0000000 0.4524985
bp  0.4189667 0.4524985 1.0000000
```

Example

```
mean(diabetes1$glu)

[1] 92.34938

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

[1] 92.03593

var(diabetes1$glu)

[1] 117.29

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

[1] 95.84904
```

Example

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

s1$r.squared

[1] 0.2489047

s2$r.squared

[1] 0.2882985
```

Example

```
s1$coef

             Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 50.6762960 3.55510857 14.254500 3.689251e-38
bmi          0.7369749 0.10360177  7.113535 4.628540e-12
bp           0.2333118 0.03608963  6.464788 2.704883e-10

s2$coef

             Estimate Std. Error  t value    Pr(>|t|)
(Intercept) 49.3228188 3.33953283 14.769377 2.311090e-40
bmi          0.6708372 0.09344954  7.178604 3.027927e-12
bp           0.2630784 0.03655493  7.196795 2.687794e-12
```
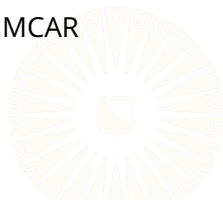
Bad Methods (These almost never work)

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
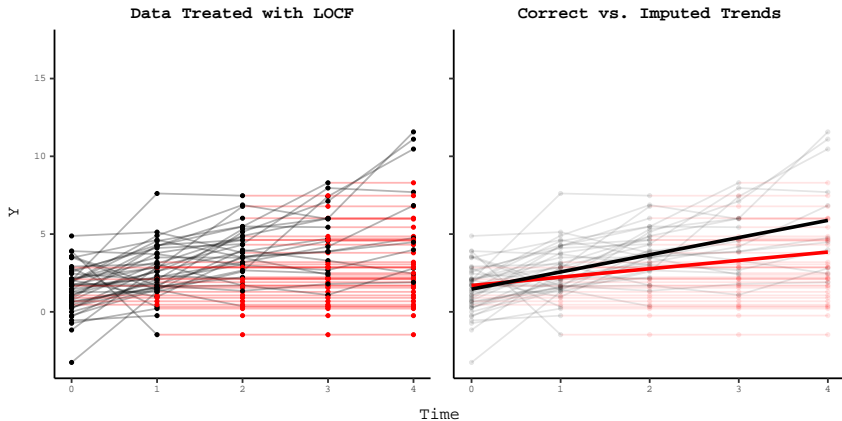
Last Observation Carried Forward (LOCF)
- Replace post-dropout values with the most recent observed value
  - Assume that dropouts would maintain their last known values
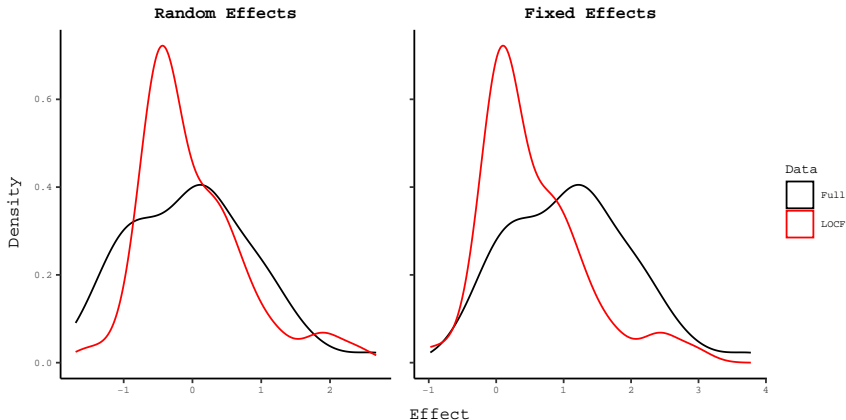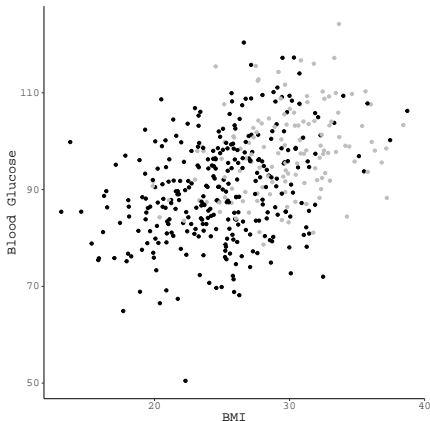 - Attenuates estimates of growth/development

LOCF
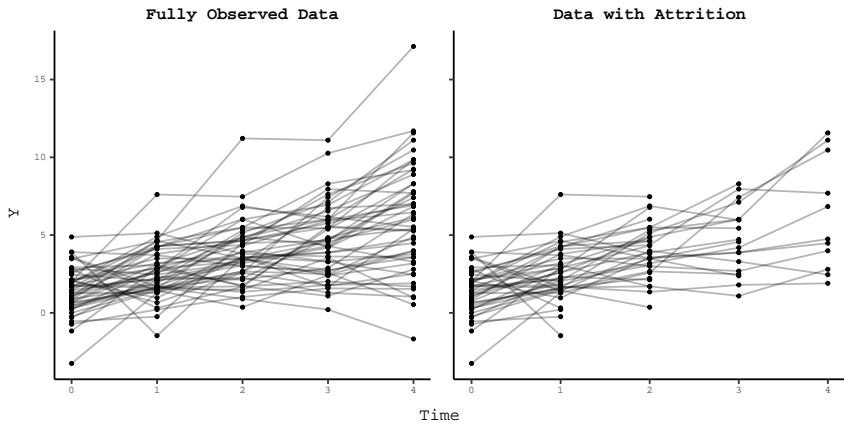
LOCF

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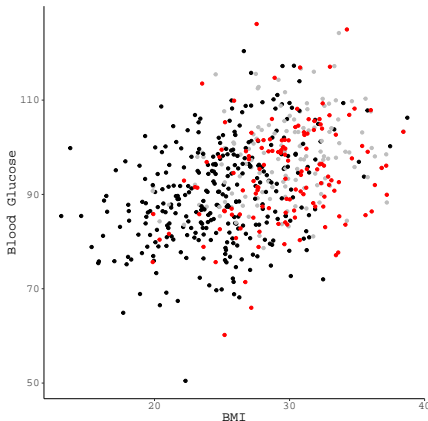
- Fill Y_{mis} with $\widehat{Y}_{mis}$ plus some
  random noise.
  - Produces unbiased parameter
    estimates and predictions
  - Computationally efficient
  - Attenuates standard errors
  - Makes CIs and prediction
    intervals too narrow

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
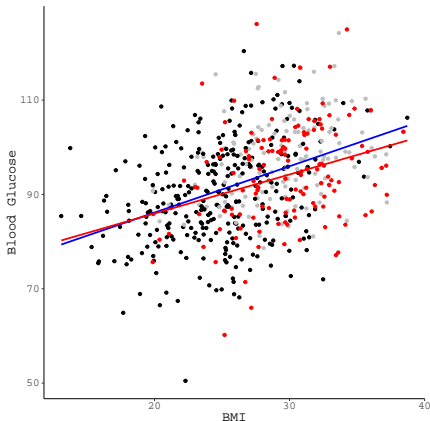Imputation

- 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    = 20,
                printFlag = FALSE,
                method   = "norm.nob") %>%
    complete(1)
```

Example

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

          bmi       glu        bp
bmi 1.0000000 0.4211879 0.3656630
glu 0.4211879 1.0000000 0.4028993
bp  0.3656630 0.4028993 1.0000000

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

          bmi       glu        bp
bmi 1.0000000 0.3676541 0.3459863
glu 0.3676541 1.0000000 0.3986198
bp  0.3459863 0.3986198 1.0000000
```

Example

```
mean(diabetes1$glu)

[1] 92.34938

mean(imputed$glu)

[1] 91.98755

var(diabetes1$glu)

[1] 117.29

var(imputed$glu)

[1] 123.7001
```

Example

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

s1$r.squared

[1] 0.2489047

s2$r.squared

[1] 0.2188541
```

Example

```
s1$coef

             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 50.6762960 3.55510857 14.254500 3.689251e-38
bmi          0.7369749 0.10360177  7.113535 4.628540e-12
bp           0.2333118 0.03608963  6.464788 2.704883e-10

s2$coef

             Estimate Std. Error   t value     Pr(>|t|)
(Intercept) 51.2876572 3.71778033 13.795236 3.223422e-36
bmi          0.6248014 0.10763621  5.804751 1.237000e-08
bp           0.2552101 0.03721424  6.857862 2.383498e-11
```

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] 0.5040896
```
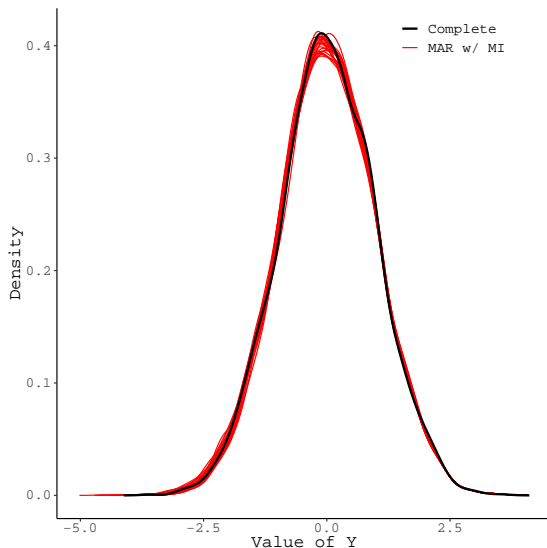
The MI-based parameter estimate looks good.

- 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] 0.4166246
```
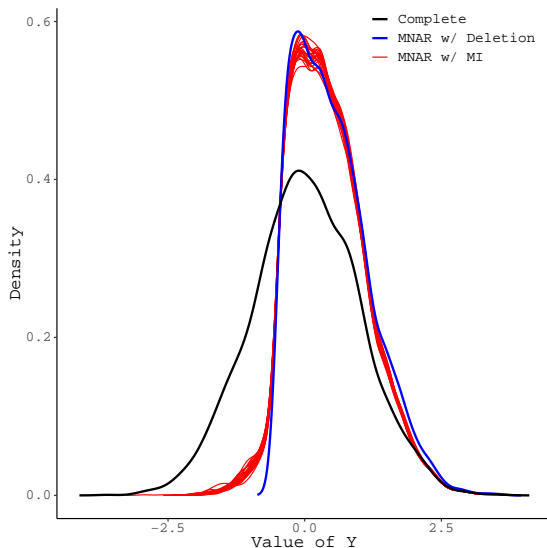
The MI-based parameter estimate is still biased.

- 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   = 20,
                printFlag = FALSE,
                method  = "norm")
```

Example

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

          bmi       glu        bp
bmi 1.0000000 0.4211879 0.3656630
glu 0.4211879 1.0000000 0.4028993
bp  0.3656630 0.4028993 1.0000000

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

          bmi       glu        bp
bmi 1.0000000 0.3962675 0.3696741
glu 0.3962675 1.0000000 0.3996248
bp  0.3696741 0.3996248 1.0000000
```

Example

```
mean(diabetes1$glu)

[1] 92.34938

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

[1] 92.06123

var(diabetes1$glu)

[1] 117.29

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

[1] 125.3791
```

Example

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

summary(fit1)$r.squared

[1] 0.2489047

pool.r.squared(fit2)

          est     lo 95     hi 95      fmi
R^2 0.2324348 0.1485566  0.322997 0.3927639
```

Example

```
summary(fit1)$coef

              Estimate Std. Error  t value     Pr(>|t|)
(Intercept) 50.6762960 3.55510857 14.254500 3.689251e-38
bmi          0.7369749 0.10360177  7.113535 4.628540e-12
bp           0.2333118 0.03608963  6.464788 2.704883e-10

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

          term    estimate  std.error statistic      p.value
1 (Intercept) 50.7094660 4.67447330 10.848167 0.000000e+00
2         bmi  0.6938806 0.12956378  5.355514 3.329401e-07
3          bp  0.2420252 0.04645062  5.210375 8.817078e-07
```

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)
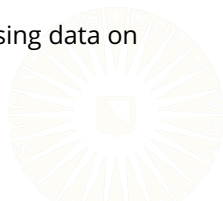
Example

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

mean(diabetes1$glu)

[1] 92.34938

fit$mean["glu"]

     glu
91.17804
```

Example

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

         bmi       glu        bp
bmi 21.58209  21.19108  22.65475
glu 21.19108 117.28999  58.19137
bp  22.65475  58.19137 177.85378

fit$cov

    bmi     glu      bp
bmi 21.533
glu 17.818 117.569
bp  18.250  51.554 169.383
```

Example

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

summary(fit1)$r.squared

[1] 0.2489047

inspect(fit, "r2")

  glu
0.181
```

Example

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

           Estimate Std. Error t value Pr(>|t|)
(Intercept)  50.676      3.555  14.255        0
bmi           0.737      0.104   7.114        0
bp            0.233      0.036   6.465        0

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

  lhs op rhs    est    se      z pvalue
1 glu ~1     52.414 4.588 11.424      0
2 glu ~ bmi   0.627 0.136  4.613      0
3 glu ~  bp   0.237 0.045  5.322      0
4 glu ~~ glu 94.192 7.578 12.430      0
```

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