Precept 8: Missing Data

Soc 504: Advanced Social Statistics

Ian Lundberg

Princeton University

April 6, 2017

Outline

Outline

Example to motivate careful thought about missing data.


[1]

Abraham Wald

- b. 1902, Austria-Hungary
- Jewish, persecuted in WWII
- Fled to U.S. in 1938
- Namesake of the Wald test
- Statistical consultant for U.S. Navy in WWII

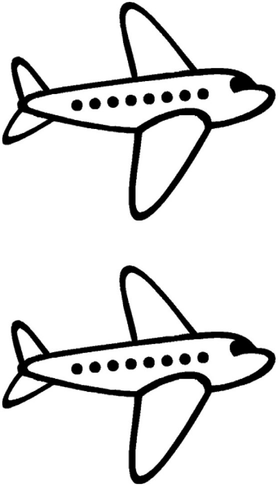
[1]PC: Wikimedia commons

Question: Where should armor be added to protect planes?

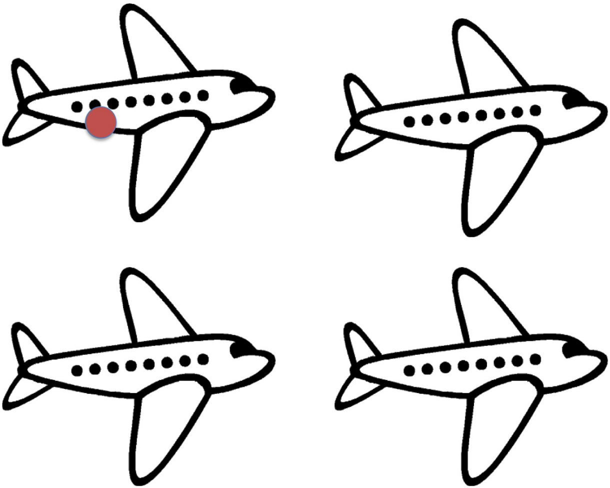
Data: Suppose we saw the following planes.[2]

[2] Story told by Mangel and Samaniego 1984 [link].
Presentation style inspired by Joe Blitzstein. See the original here [link]

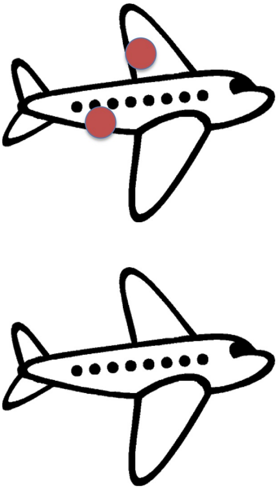
Planes that were observed



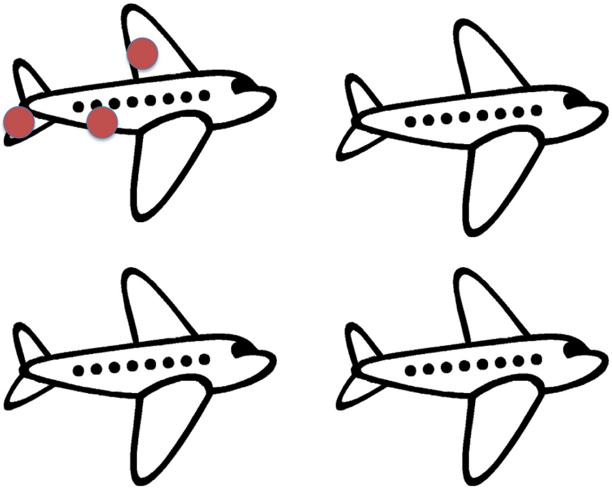
Planes that were observed



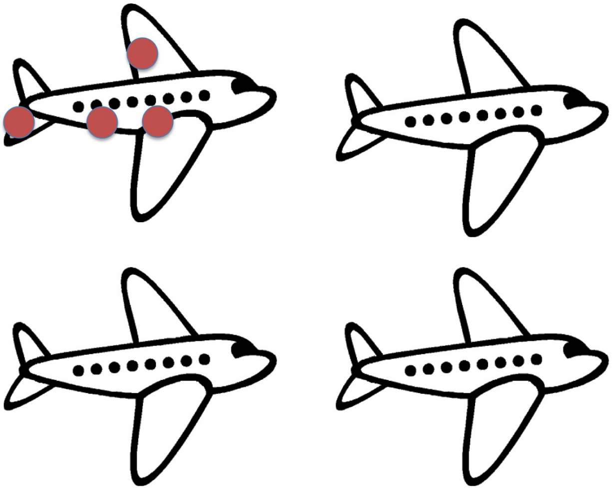
Planes that were observed



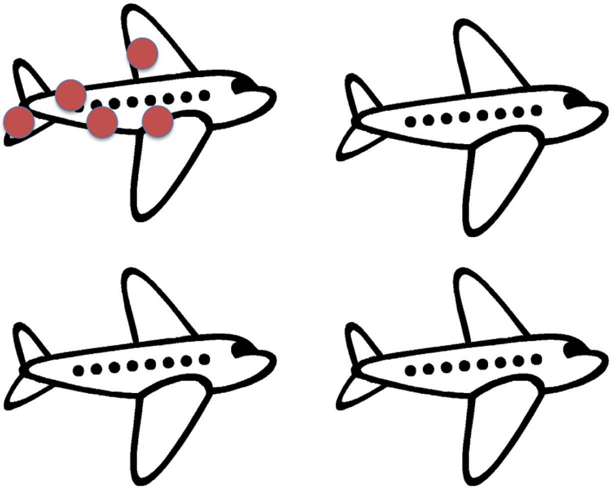
Planes that were observed



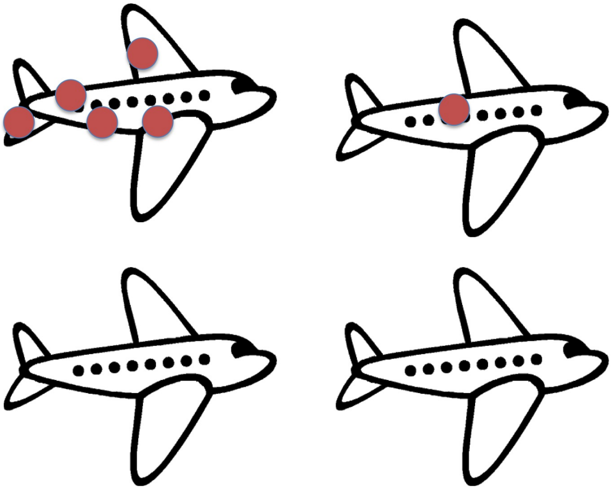
Planes that were observed



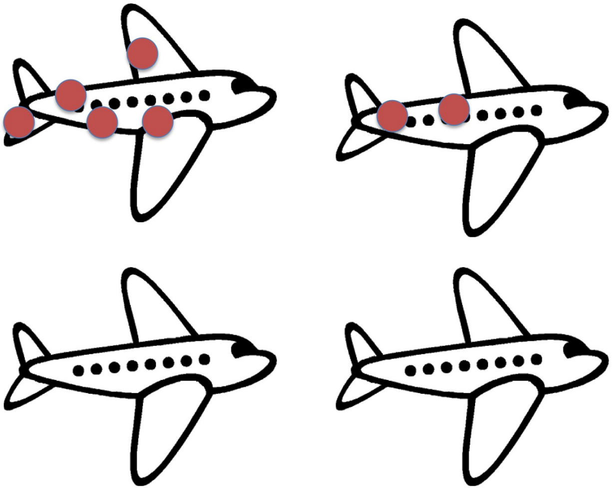
Planes that were observed



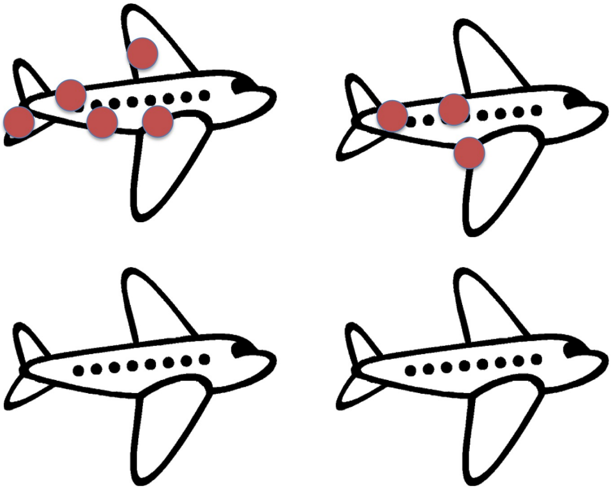
Planes that were observed



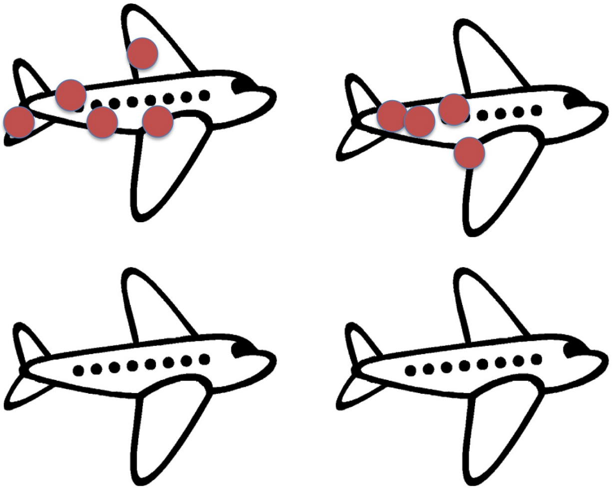
Planes that were observed



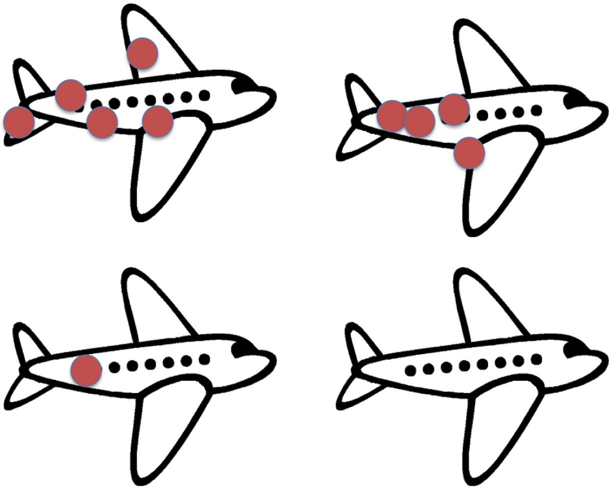
Planes that were observed



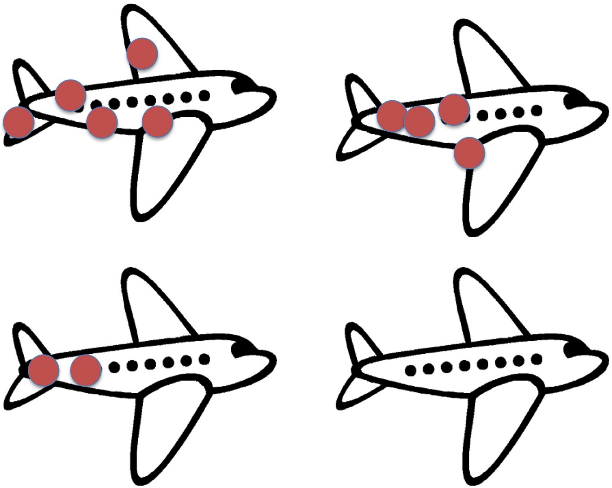
Planes that were observed



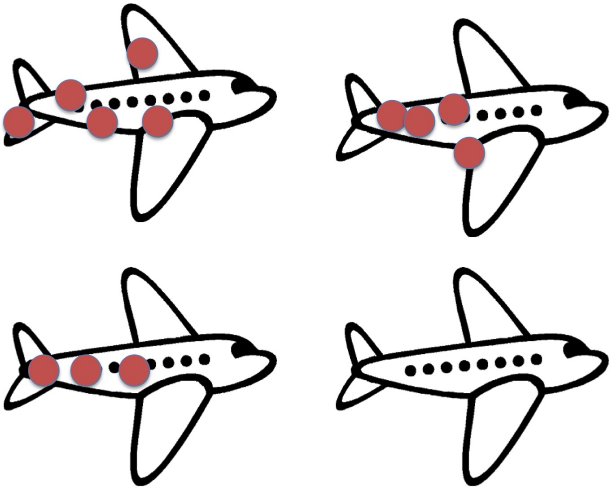
Planes that were observed



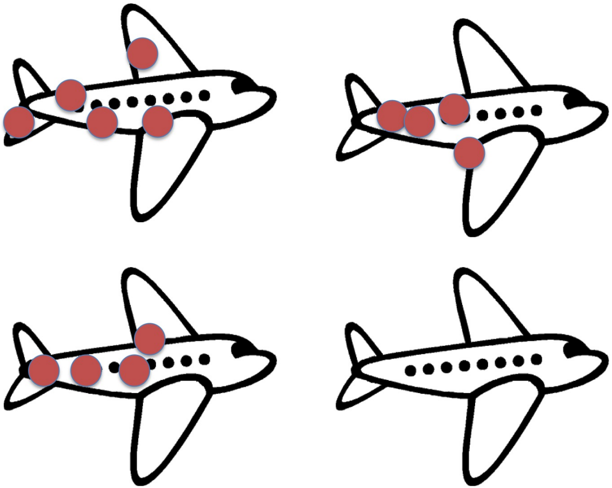
Planes that were observed



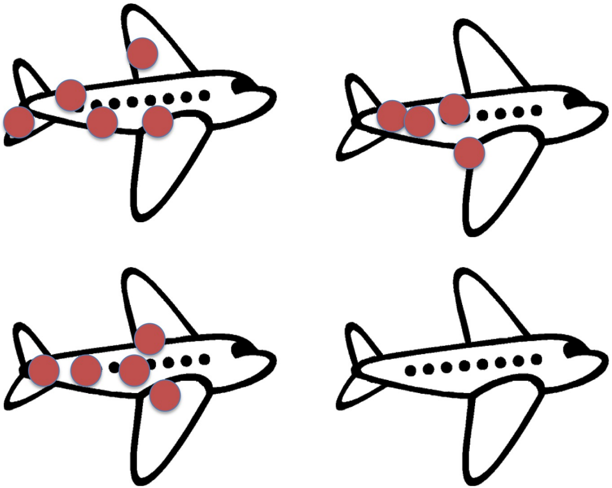
Planes that were observed



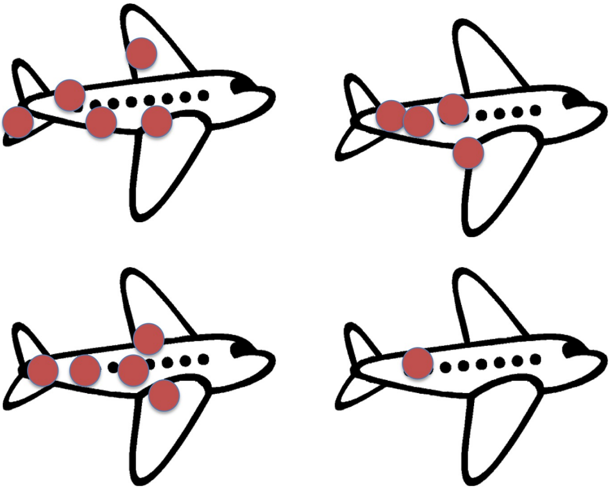
Planes that were observed



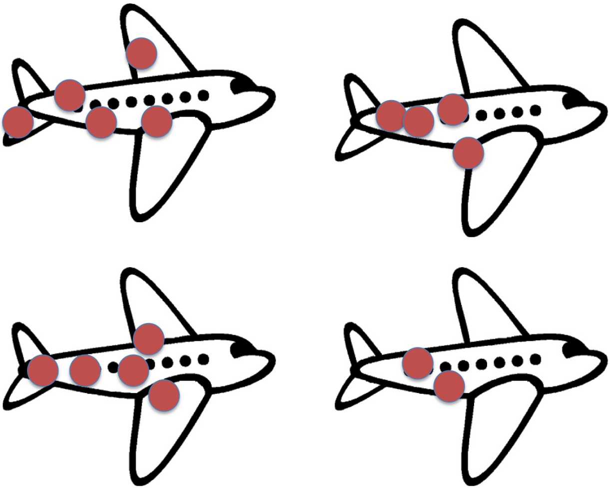
Planes that were observed



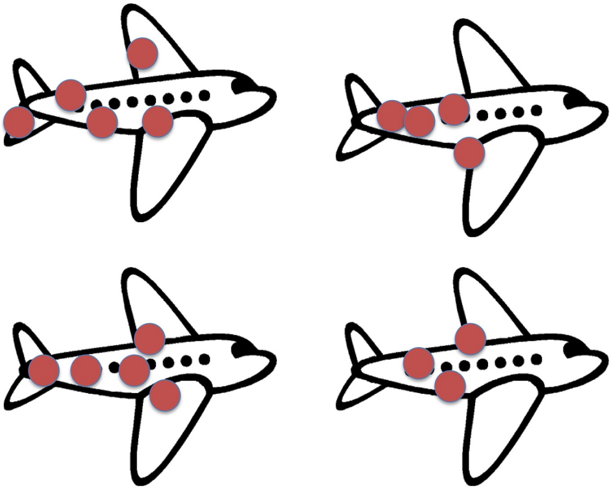
Planes that were observed



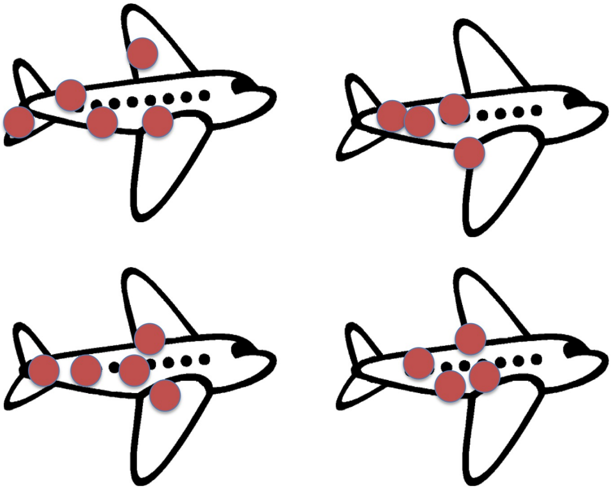
Planes that were observed



Planes that were observed

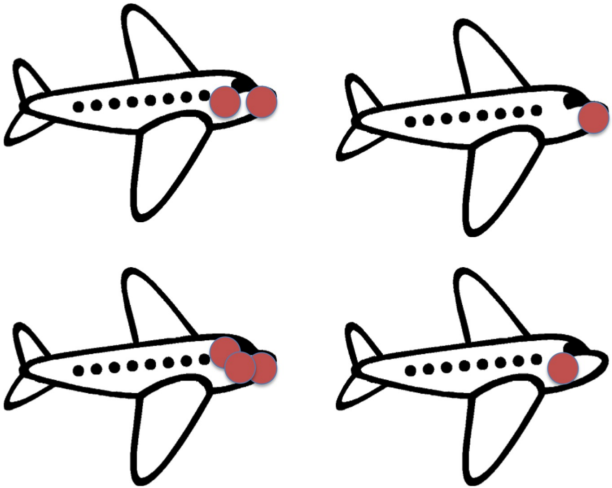


Planes that were observed



Where should we add armor?

Missing data: Planes that never returned



Now where should we add armor?

Now where should we add armor? To the nose!

Now where should we add armor? To the nose!

Results from the observed planes were misleading because data were not missing at random!

Now where should we add armor? To the nose!

Results from the observed planes were misleading because data were not missing at random!

Missing data requires careful thought.

Now where should we add armor? To the nose!

Results from the observed planes were misleading because data were not missing at random!

Missing data requires careful thought.

No algorithm solves it for you!

We will walk through the assumptions and implementation of multiple imputation.

We will walk through the assumptions and implementation of multiple imputation.

Our example will be the 2016 General Social Survey (GSS), which was released last week (March 29).

We will walk through the assumptions and implementation of multiple imputation.

Our example will be the 2016 General Social Survey (GSS), which was released last week (March 29).

The GSS measures Americans' attitudes toward lots of issues.

We will walk through the assumptions and implementation of multiple imputation.

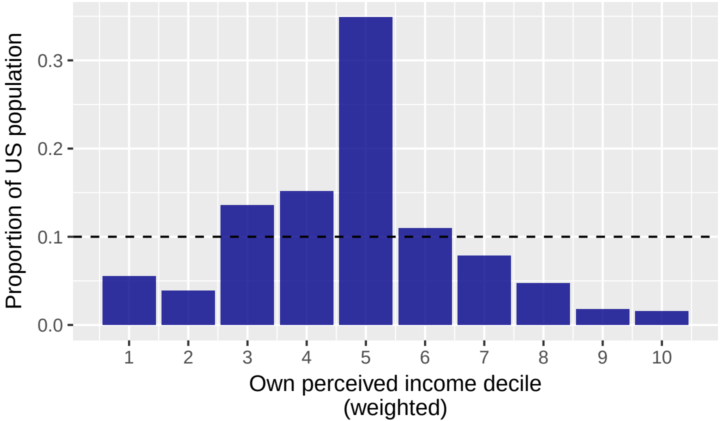
Our example will be the 2016 General Social Survey (GSS), which was released last week (March 29).

The GSS measures Americans' attitudes toward lots of issues.

List of files (we use 2016): `http://gss.norc.org/get-the-data/spss`
Link directly to data download

The GSS captures Americans' attitudes about lots of things.
Sometimes our collective beliefs are nonsensical.

The GSS captures Americans' attitudes about lots of things.
Sometimes our collective beliefs are nonsensical.



The GSS also has information on parents' characteristics for mobility research.

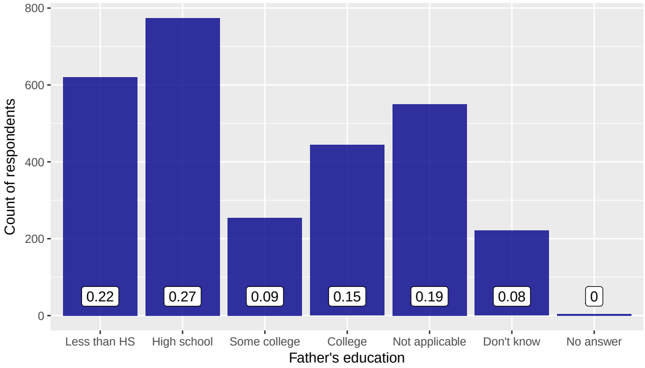
The GSS also has information on parents' characteristics for mobility research.

For instance, `paeduc` captures father's education in years.

The GSS also has information on parents' characteristics for mobility research.

For instance, `paeduc` captures father's education in years.

But it's sometimes missing. We need to know why!



Code for prior slide

```
gss %>%
  mutate(paeduc = factor(ifelse(paeduc < 12, 1,
                                ifelse(paeduc == 12, 2,
                                       ifelse(paeduc < 16, 3,
                                              ifelse(paeduc >= 16 & paeduc <= 20, 4,
                                                     paeduc)))),
                         labels = c("Less than HS","High school",
                                    "Some college","College",
                                    "Not applicable","Don't know","No answer"))) %>%
  group_by(paeduc) %>%
  summarize(num = n()) %>%
  ggplot(aes(x = paeduc, y = num)) +
  geom_bar(stat = "identity", fill = "darkblue", alpha = .8) +
  geom_label(aes(y = 50, label = round(num / nrow(gss),2))) +
  xlab("Father's education") + ylab("Count of respondents") +
  ggsave("figs/PaEduc.pdf",
         height = 4, width = 7)
         height = 3, width = 5)
```

Why is father's education missing?

Check the codebook (p. 176) [link]

<u>IF NOT LIVING WITH OWN FATHER,</u> ASK
PAOCC16 to PAIND16, PAEDUC, AND PADEG
IN TERMS OF STEPFATHER OR OTHER MALE SPECIFIED ABOVE.
IF NO STEPFATHER OR OTHER MALE, SKIP
PAOCC16 to PAIND16, PAEDUC, AND PADEG.

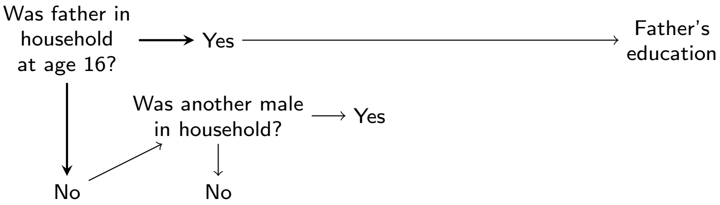
These are the "Not applicable" cases.
You should always make sure you know what your variables are!

Questionnaire logic, graphically

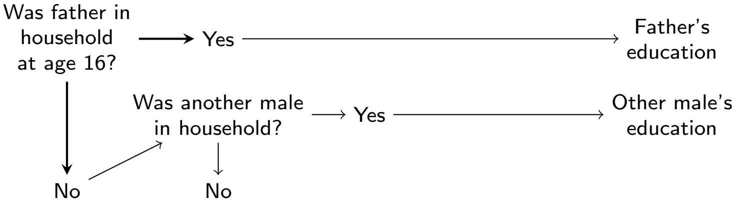
$$\begin{tikzcd}
\text{Was father in household at age 16?} \arrow[r] \arrow[d] & \text{Yes} \\
\text{No} &
\end{tikzcd}$$

Questionnaire logic, graphically

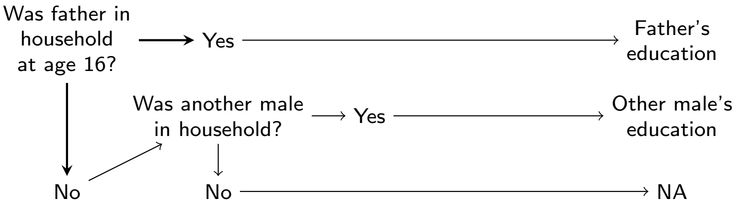
Questionnaire logic, graphically



Questionnaire logic, graphically



Questionnaire logic, graphically



What to do? Two options

1. Fill in with theoretically meaningful values.

What to do? Two options

1. Fill in with theoretically meaningful values.
   WARNING: This changes what the measure captures.

What to do? Two options

1. Fill in with theoretically meaningful values.
   WARNING: This changes what the measure captures.
2. Multiply impute

What to do? Option 1: Manual Filling

Was father in household at age 16? → Yes

↓

No

GSS questionnaire logic

What to do? Option 1: Manual Filling

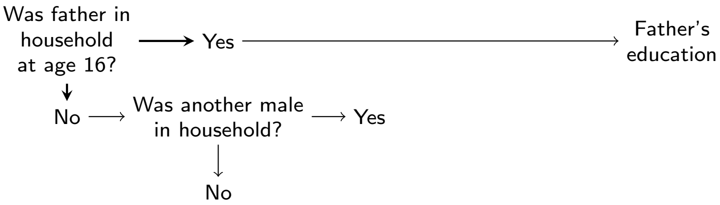
Was father in household at age 16? $\longrightarrow$ Yes $\longrightarrow$ Father's education

$\downarrow$

No

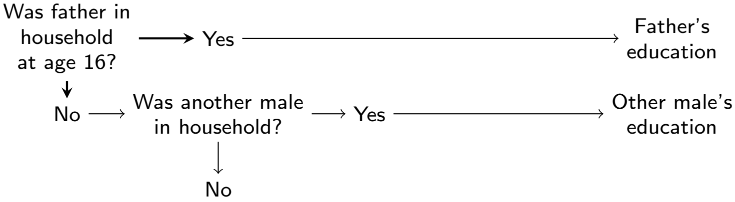
GSS questionnaire logic

What to do? Option 1: Manual Filling



GSS questionnaire logic

What to do? Option 1: Manual Filling



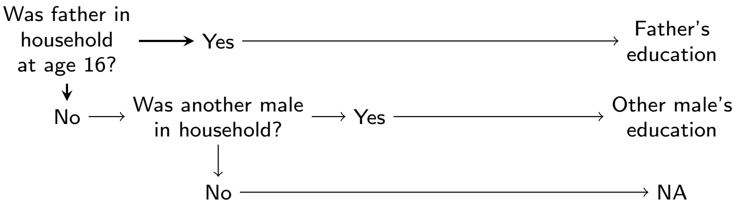
GSS questionnaire logic

What to do? Option 1: Manual Filling

$$
\begin{tikzcd}
\text{Was father in household at age 16?} \arrow[r] \arrow[d] & \text{Yes} \arrow[rr] & & \text{Father's education} \\
\text{No} \arrow[r] & \text{Was another male in household?} \arrow[r] \arrow[d] & \text{Yes} \arrow[r] & \text{Other male's education} \\
& \text{No} \arrow[rr] & & \text{NA}
\end{tikzcd}
$$

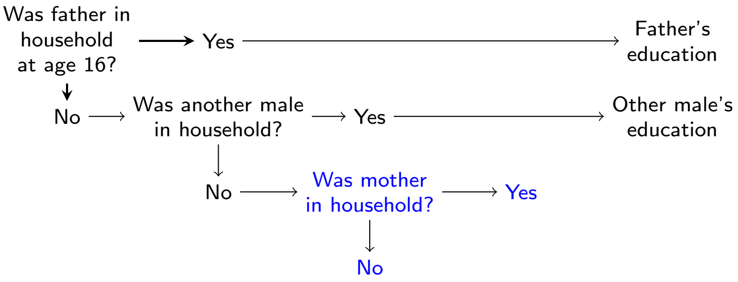
GSS questionnaire logic

What to do? Option 1: Manual Filling



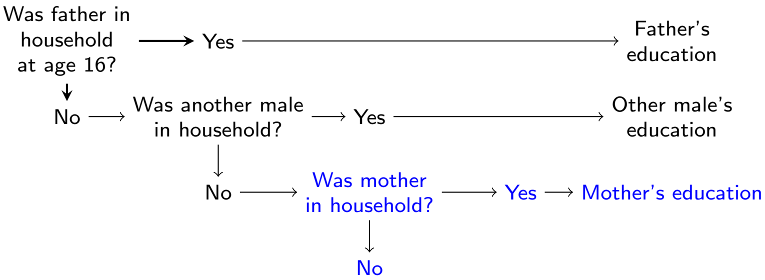
GSS questionnaire logic could be extended to mother's education.

What to do? Option 1: Manual Filling



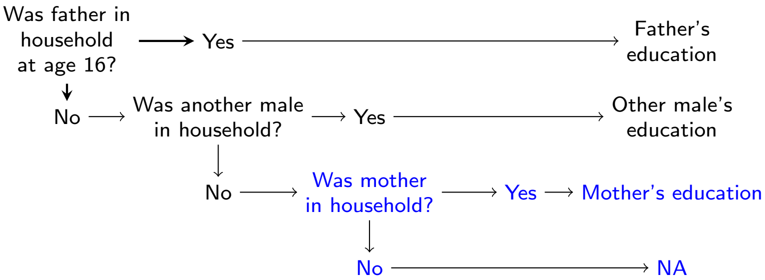
GSS questionnaire logic could be extended to mother's education.

What to do? Option 1: Manual Filling



GSS questionnaire logic could be extended to mother's education.

What to do? Option 1: Manual Filling



GSS questionnaire logic could be extended to mother's education.

It's often possible to fill in missing values manually as above.

It's often possible to fill in missing values manually as above.

But be WARNED - this often changes the meaning of the predictor.

It's often possible to fill in missing values manually as above.

But be WARNED - this often changes the meaning of the predictor.

In this example, it became a fuzzy measure of family background.

It's often possible to fill in missing values manually as above.

But be WARNED - this often changes the meaning of the predictor.

In this example, it became a fuzzy measure of family background.

What if you really wanted a measure of the education of the father or other male in the household at age 16?

What to do? Option 2: Multiple Imputation

What to do? Option 2: Multiple Imputation

All respondents do have a father, even if that father wasn't around at age 16.

What to do? Option 2: Multiple Imputation

All respondents do have a father, even if that father wasn't around at age 16.

We could multiply impute the missing values of father's education, using mother's education as a predictor.

What to do? Option 2: Multiple Imputation

All respondents do have a father, even if that father wasn't around at age 16.

We could multiply impute the missing values of father's education, using mother's education as a predictor.

In this case, the predictor truly is father's education.

Missingness Assumptions (adapted from lecture)

Missingness Assumptions (adapted from lecture)

1. **MCAR**: Missing Completely At Random (naive)

Missingness Assumptions (adapted from lecture)

1. **MCAR**: Missing Completely At Random (naive)

$$P(M|X) = P(M)$$

Missingness Assumptions (adapted from lecture)

1. **MCAR**: Missing Completely At Random (naive)

$$P(M|X) = P(M)$$

Missingness (M) is unrelated to father's education (X)

Missingness Assumptions (adapted from lecture)

1. **MCAR**: Missing Completely At Random (naive)

$$P(M|X) = P(M)$$

   Missingness (M) is unrelated to father's education (X)

2. **MAR**: Missing At Random (empirical)

Missingness Assumptions (adapted from lecture)

1. **MCAR**: Missing Completely At Random (naive)

$$P(M|X) = P(M)$$

   Missingness (M) is unrelated to father's education (X)

2. **MAR**: Missing At Random (empirical)

$$P(M|X, Z) = P(M|Z)$$

Missingness Assumptions (adapted from lecture)

1. **MCAR**: Missing Completely At Random (naive)

$$P(M|X) = P(M)$$

   Missingness (M) is unrelated to father's education (X)

2. **MAR**: Missing At Random (empirical)

$$P(M|X,Z) = P(M|Z)$$

   Missingness is not a function of the missing variable ($X =$ Father's education), conditional on measured variables ($Z =$ Mother's education)
   e.g., Children with lesser-educated mothers are more likely to have missing fathers

Missingness Assumptions (adapted from lecture)

1. **MCAR**: Missing Completely At Random (naive)

$$P(M|X) = P(M)$$

   Missingness (M) is unrelated to father's education (X)

2. **MAR**: Missing At Random (empirical)

$$P(M|X,Z) = P(M|Z)$$

   Missingness is not a function of the missing variable ($X =$ Father's education), conditional on measured variables ($Z =$ Mother's education)
   e.g., Children with lesser-educated mothers are more likely to have missing fathers

3. **NI**: Non-ignorable (fatalistic)
   $P(M|X)$ doesn't simplify

Missingness Assumptions (adapted from lecture)

1. **MCAR**: Missing Completely At Random (naive)

$$P(M|X) = P(M)$$

   Missingness (M) is unrelated to father's education (X)

2. **MAR**: Missing At Random (empirical)

$$P(M|X,Z) = P(M|Z)$$

   Missingness is not a function of the missing variable (X = Father's education), conditional on measured variables (Z = Mother's education)
   e.g., Children with lesser-educated mothers are more likely to have missing fathers

3. **NI**: Non-ignorable (fatalistic)
   $P(M|X)$ doesn't simplify
   e.g., within cells of mother's education, missingness is still related to father's education

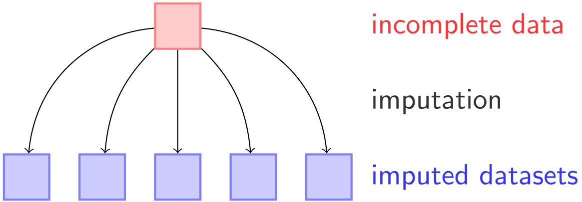
   Adding variables to predict father's education can change NI to MAR

The Multiple Imputation Scheme (from lecture)

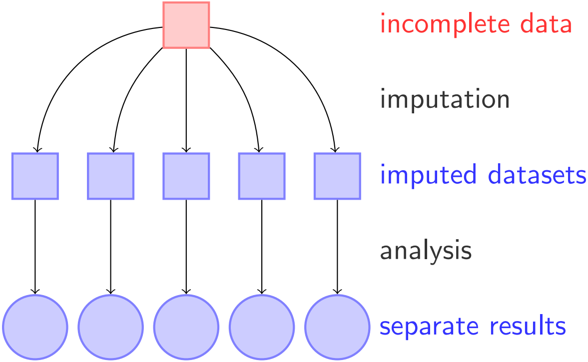
The Multiple Imputation Scheme (from lecture)

incomplete data

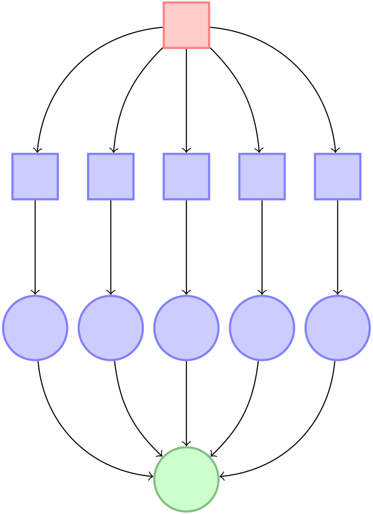
The Multiple Imputation Scheme (from lecture)



The Multiple Imputation Scheme (from lecture)



The Multiple Imputation Scheme (from lecture)



incomplete data

imputation

imputed datasets

analysis

separate results

combination

final results

Multiple Imputation (from lecture)

Multiple Imputation (from lecture)

REGRESSION

To preserve the relationships in the data.

Multiple Imputation (from lecture)

REGRESSION

To preserve the relationships in the data.

SIMULATION

To reflect the uncertainty of our imputation.

Imputing father's education

We wanted to impute missing values of father's education.

Imputing father's education

We wanted to impute missing values of father's education.

Is it dubious that father's education is missing at random?

Imputing father's education

We wanted to impute missing values of father's education.

Is it dubious that father's education is missing at random? YES

Imputing father's education

We wanted to impute missing values of father's education.

Is it dubious that father's education is missing at random? YES

We proceed cautiously anyway, realizing MAR is a heroic assumption

Imputing father's education

We wanted to impute missing values of father's education.

Is it dubious that father's education is missing at random? YES

We proceed cautiously anyway, realizing MAR is a heroic assumption (heroic in a bad way)

Choosing variables

We want to impute father's education with other variables that might be associated with it.

Choosing variables

We want to impute father's education with other variables that might be associated with it.

- Mother's education
- Respondent's education
- Respondent's perceived financial standing
- Respondent's perceived income decile
- Respondent's age
- Respondent's race

Choosing variables

We want to impute father's education with other variables that might be associated with it.

- Mother's education
- Respondent's education
- Respondent's perceived financial standing
- Respondent's perceived income decile
- Respondent's age
- Respondent's race

We'd have to argue that, net of all these, father's education is missing at random.

Choosing variables

```
toImpute <- gss %>%
  transmute(id = id,
            paeduc = ifelse(paeduc > 20, NA, paeduc),
            finrela = ifelse(finrela == "DK" | finrela == "NA",
                             NA, finrela),
            maeduc = ifelse(maeduc > 20, NA, maeduc),
            educ = ifelse(educ > 20, NA, educ),
            rank = ifelse(rank > 10, NA, rank),
            age, race)
```

Choosing variables

```
> summary(toImpute)
       id              paeduc           finrela          maeduc
 Min.    :   1.0   Min.    : 0.0   Min.    :2.000   Min.    : 0.00
 1st Qu.: 717.5   1st Qu.:10.0   1st Qu.:3.000   1st Qu.:11.00
 Median :1434.0   Median :12.0   Median :4.000   Median :12.00
 Mean   :1434.0   Mean   :11.8   Mean   :3.861   Mean   :11.86
 3rd Qu.:2150.5   3rd Qu.:14.0   3rd Qu.:4.000   3rd Qu.:14.00
 Max.   :2867.0   Max.   :20.0   Max.   :6.000   Max.   :20.00
                  NA's   :775    NA's   :28      NA's   :286
      educ             rank            age             race
 Min.   : 0.00   Min.   : 1.0   Min.   :18.00   IAP  :   0
 1st Qu.:12.00   1st Qu.: 4.0   1st Qu.:34.00   WHITE:2100
 Median :13.00   Median : 5.0   Median :50.00   BLACK: 490
 Mean   :13.74   Mean   : 4.8   Mean   :49.33   OTHER: 277
 3rd Qu.:16.00   3rd Qu.: 6.0   3rd Qu.:62.00
 Max.   :20.00   Max.   :10.0   Max.   :99.00
 NA's   :9       NA's   :79
```

Implementation in Amelia

Run Amelia

```
library(Amelia)
```

Run Amelia

```
library(Amelia)
filled <- amelia(toImpute,
                 noms = "race",
                 idvars = "id")
```

Run Amelia

```
> summary(filled)
```

Run Amelia

```
> summary(filled)

Amelia output with 5 imputed datasets.
Return code:  1
Message:  Normal EM convergence.

Chain Lengths:
--------------
Imputation 1:  5
Imputation 2:  8
Imputation 3:  6
Imputation 4:  7
Imputation 5:  5
```

Run Amelia

```
Rows after Listwise Deletion:  1893
Rows after Imputation:  2867
Patterns of missingness in the data:  20

Fraction Missing for original variables:
-----------------------------------------

         Fraction Missing
id            0.000000000
paeduc        0.270317405
finrela       0.009766306
maeduc        0.099755842
educ          0.003139170
rank          0.027554935
age           0.000000000
race          0.000000000
```

Patterns of missingness

Amelia told us there were 20 patterns of missignness. What were they?

Patterns of missingness

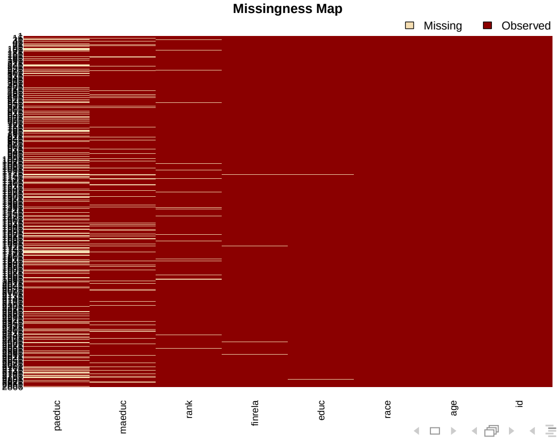
Amelia told us there were 20 patterns of missignness. What were they?

```
missmap(filled)
```

Patterns of missingness

Amelia told us there were 20 patterns of missignness. What were they?

```
missmap(filled)
```



Overimputing

Overimputing is a check that the imputation model we've set up is doing roughly what we think it's doing.

Overimputing

Overimputing is a check that the imputation model we've set up is doing roughly what we think it's doing.

It does not verify the key identification assumption: missing at random.

Overimputing

Overimputing is a check that the imputation model we've set up is doing roughly what we think it's doing.

It does not verify the key identification assumption: missing at random.

Idea:

Overimputing

Overimputing is a check that the imputation model we've set up is doing roughly what we think it's doing.

It does not verify the key identification assumption: missing at random.

Idea:

- Knock out some values

Overimputing

Overimputing is a check that the imputation model we've set up is doing roughly what we think it's doing.

It does not verify the key identification assumption: missing at random.

Idea:

- Knock out some values
- Fill in as though they were missing

Overimputing

Overimputing is a check that the imputation model we've set up is doing roughly what we think it's doing.

It does not verify the key identification assumption: missing at random.

Idea:

- Knock out some values
- Fill in as though they were missing
- Compare our imputations to the truth

Overimputing

Overimputing is a check that the imputation model we've set up is doing roughly what we think it's doing.

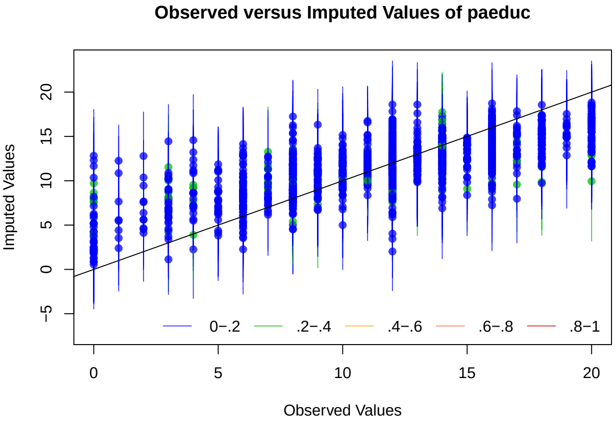
It does not verify the key identification assumption: missing at random.

Idea:

- Knock out some values
- Fill in as though they were missing
- Compare our imputations to the truth
- We want the truth to generally fall in the range of imputed values.

Overimputing

```
overimpute(filled, var = "paeduc")
```



Checking convergence

EM can sometimes end up in weird places.

Checking convergence

EM can sometimes end up in weird places.

We want to know our results converge the same place regardless of the starting values.

Checking convergence

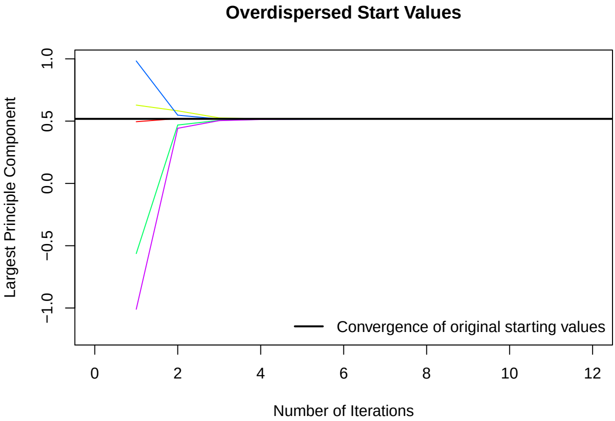
EM can sometimes end up in weird places.

We want to know our results converge the same place regardless of the starting values.

Amelia's `disperse()` command shows us that the first principle component (a unidimensional summary of the data) converges to the same value regardless of a few randomly chosen starting points.

Checking convergence

```
disperse(filled, dims = 1, m = 5)
```



Amelia objects

Your Amelia object holds lots of things, including 5 versions of the data.

Amelia objects

Your Amelia object holds lots of things, including 5 versions of the data.

```
> head(filled$imputations[[1]])
```

Amelia objects

Your Amelia object holds lots of things, including 5 versions of the data.

```
> head(filled$imputations[[1]])
  id   paeduc finrela maeduc educ rank age  race
1  1 18.00000       6     13   16    1  47 WHITE
2  2  8.00000       4     12   12    5  61 WHITE
3  3 12.00000       3      8   16    4  72 WHITE
4  4 15.36367       5     12   12    3  43 WHITE
5  5 16.00000       5     12   18    3  55 WHITE
6  6 11.00000       4     12   14    5  53 WHITE
```

`transform`: Operating on an Amelia object

What if we now want the respondent's education to be coded as college or not?

`transform`: Operating on an Amelia object

What if we now want the respondent's education to be coded as college or not? `transform` operates on all imputations at once.

```
filled <- transform(filled,
                    college = educ >= 16)
```

transform: Operating on an Amelia object

What if we now want the respondent's education to be coded as college or not? transform operates on all imputations at once.

```
filled <- transform(filled,
                    college = educ >= 16)
head(filled$imputations[[1]])
```

`transform`: Operating on an Amelia object

What if we now want the respondent's education to be coded as college or not? `transform` operates on all imputations at once.

```
filled <- transform(filled,
                    college = educ >= 16)
head(filled$imputations[[1]])
```

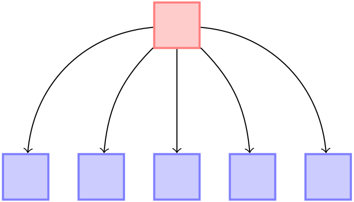
```
  id   paeduc finrela maeduc educ rank age  race college
1  1 18.00000       6     13   16    1  47 WHITE    TRUE
2  2  8.00000       4     12   12    5  61 WHITE   FALSE
3  3 12.00000       3      8   16    4  72 WHITE    TRUE
4  4 15.36367       5     12   12    3  43 WHITE   FALSE
5  5 16.00000       5     12   18    3  55 WHITE    TRUE
6  6 11.00000       4     12   14    5  53 WHITE   FALSE
```

The Multiple Imputation Scheme (again)

The Multiple Imputation Scheme (again)

incomplete data

The Multiple Imputation Scheme (again)

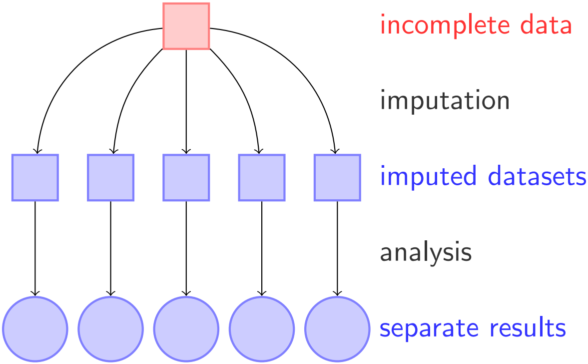


incomplete data

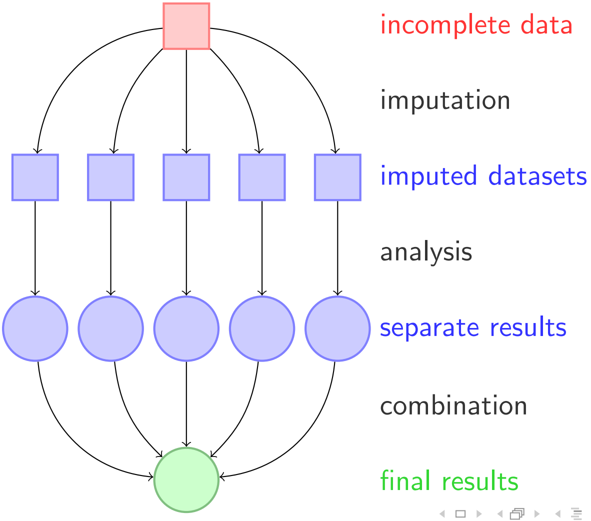
imputation

imputed datasets

The Multiple Imputation Scheme (again)



The Multiple Imputation Scheme (again)



Modeling with imputed data

We will model respondent's college completion as a function of race and father's years of schooling.

Modeling with imputed data

We will model respondent's college completion as a function of race and father's years of schooling.

We could just fit using single imputation.

Modeling with imputed data

We will model respondent's college completion as a function of race and father's years of schooling.

We could just fit using single imputation.

```
fit <- glm(college ~ paeduc + race + paeduc:race,
           family = binomial(link = "logit"),
           data = filled$imputations[[1]])
summary(fit)
```

Modeling with imputed data

We will model respondent's college completion as a function of race and father's years of schooling.

We could just fit using single imputation.

```
fit <- glm(college ~ paeduc + race + paeduc:race,
           family = binomial(link = "logit"),
           data = filled$imputations[[1]])
summary(fit)
```

But this would understate our uncertainty.

Fitting on all imputations

We can use `lapply()` to

- apply a function to all the imputations and
- return a list of model fits.

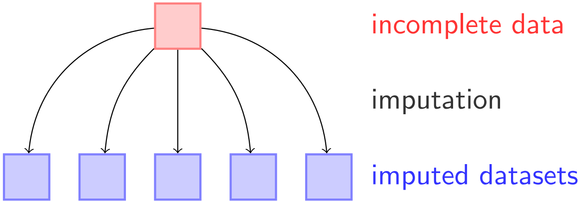
```
fits <- lapply(filled$imputations, function(data)
  fit <- glm(college ~ paeduc + race + paeduc:race,
             family = binomial(link = "logit"),
             data = data)
  return(fit)
)
```

The Multiple Imputation Scheme (again)

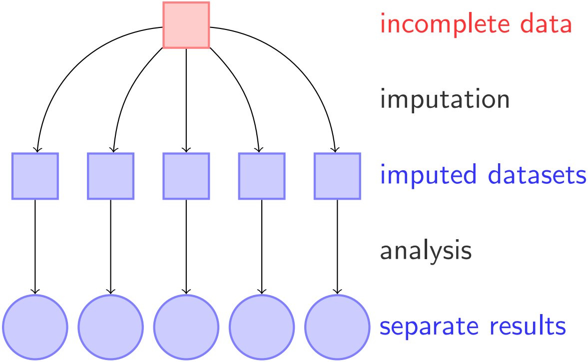
The Multiple Imputation Scheme (again)

incomplete data

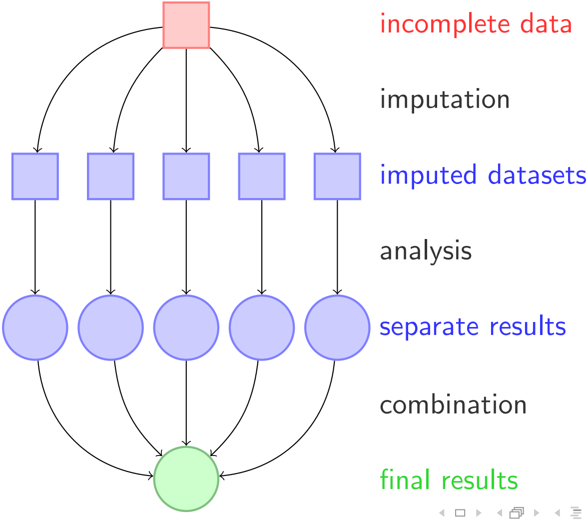
The Multiple Imputation Scheme (again)



The Multiple Imputation Scheme (again)



The Multiple Imputation Scheme (again)



Combining models

We can combine models by

- Rubin's rules
- Simulation

Approach 1: Rubin's rules

Rubin's rules give analytic formulas for the combined estimates.

Approach 1: Rubin's rules

Rubin's rules give analytic formulas for the combined estimates.

But, they are less automatic for quantities of interest beyond coefficients,
and they rely on normality assumptions that may not hold. But they are easy!

Approach 1: Rubin's rules

```
> library(mitools)
> MIcombine(fits)
```

Approach 1: Rubin's rules

```
> library(mitools)
> MIcombine(fits)
Multiple imputation results:
      MIcombine.default(fits)
                      results         se
(Intercept)       -3.23887997 0.20511552
paeduc             0.21203001 0.01597134
raceBLACK          0.26445325 0.47785166
raceOTHER         -0.33563350 0.60387117
paeduc:raceBLACK  -0.06866543 0.03697217
paeduc:raceOTHER   0.02652747 0.04888252
```

Approach 2: Simulation

Simulation is:

Approach 2: Simulation

Simulation is:

- More flexible

Approach 2: Simulation

Simulation is:

- More flexible
- A simple extension of what we've done

Approach 2: Simulation

For each imputed dataset

Approach 2: Simulation

For each imputed dataset

- Fit the model

Approach 2: Simulation

For each imputed dataset

- Fit the model
- Simulate quantities of interest

Approach 2: Simulation

For each imputed dataset

- Fit the model
- Simulate quantities of interest

Combine the simulations across models,

Approach 2: Simulation

For each imputed dataset

- Fit the model
- Simulate quantities of interest

Combine the simulations across models,

and you have the combined results. No formulas required!

Approach 2: Simulation for coefficients

```
library(mvtnorm)
list.of.sims <- lapply(fits, function(fit)
  sim.coefs <- rmvnorm(n = 1000,
                       mean = coef(fit),
                       sigma = vcov(fit))
  return(sim.coefs)
)
```

Approach 2: Simulation for coefficients

```
library(mvtnorm)
list.of.sims <- lapply(fits, function(fit)
  sim.coefs <- rmvnorm(n = 1000,
                       mean = coef(fit),
                       sigma = vcov(fit))
  return(sim.coefs)
)
dim(list.of.sims[[1]])
```

Approach 2: Simulation for coefficients

```
library(mvtnorm)
list.of.sims <- lapply(fits, function(fit)
  sim.coefs <- rmvnorm(n = 1000,
                       mean = coef(fit),
                       sigma = vcov(fit))
  return(sim.coefs)
)
dim(list.of.sims[[1]])
sims <- do.call(rbind, list.of.sims)
```

Approach 2: Simulation for coefficients

```
library(mvtnorm)
list.of.sims <- lapply(fits, function(fit)
  sim.coefs <- rmvnorm(n = 1000,
                       mean = coef(fit),
                       sigma = vcov(fit))
  return(sim.coefs)
)
dim(list.of.sims[[1]])
sims <- do.call(rbind, list.of.sims)
cbind(apply(sims, 2, mean),
      apply(sims, 2, sd))
```

Approach 2: Simulation for coefficients

	Rubin's rules		Simulation	
	Coefficient	SE	Coefficient	SE
(Intercept)	-3.24	0.21	-3.24	0.20
paeduc	0.21	0.02	0.21	0.02
raceBLACK	0.26	0.48	0.26	0.48
raceOTHER	-0.34	0.60	-0.35	0.58
paeduc:raceBLACK	-0.07	0.04	-0.07	0.04
paeduc:raceOTHER	0.03	0.05	0.03	0.05

Approach 2: Simulation for QOIs

Same approach works for quantities of interest!

We will examine the probability of college completion by race and father's education.

Approach 2: Simulation for for QOIs: Set x

```
x <- rbind(
  ## White, grades 10-16
  White.10 = c(1,10,0,0,0,0),
  White.11 = c(1,11,0,0,0,0),
  White.12 = c(1,12,0,0,0,0),
  White.13 = c(1,13,0,0,0,0),
  White.14 = c(1,14,0,0,0,0),
  White.15 = c(1,15,0,0,0,0),
  White.16 = c(1,16,0,0,0,0),

  ## Black, grades 10-16
  Black.10 = c(1,10,1,0,10,0),
  Black.11 = c(1,11,1,0,11,0),
  Black.12 = c(1,12,1,0,12,0),
  Black.13 = c(1,13,1,0,13,0),
  Black.14 = c(1,14,1,0,14,0),
  Black.15 = c(1,15,1,0,15,0),
  Black.16 = c(1,16,1,0,16,0)
)
```

Approach 2: Simulate in each imputation

```
list.of.sims <- lapply(fits, function(fit)
  sim.coef <- rmvnorm(n = 1000,
                      mean = coef(fit),
                      sigma = vcov(fit))
```

Approach 2: Simulate in each imputation

```
list.of.sims <- lapply(fits, function(fit)
  sim.coef <- rmvnorm(n = 1000,
                      mean = coef(fit),
                      sigma = vcov(fit))
  linear.predictor <- x %*% t(sim.coef)
```

Approach 2: Simulate in each imputation

```
list.of.sims <- lapply(fits, function(fit)
  sim.coef <- rmvnorm(n = 1000,
                      mean = coef(fit),
                      sigma = vcov(fit))
  linear.predictor <- x %*% t(sim.coef)
  pred.p <- plogis(linear.predictor)
```

Approach 2: Simulate in each imputation

```
list.of.sims <- lapply(fits, function(fit)
  sim.coef <- rmvnorm(n = 1000,
                      mean = coef(fit),
                      sigma = vcov(fit))
  linear.predictor <- x %*% t(sim.coef)
  pred.p <- plogis(linear.predictor)
  rownames(pred.p) <- rownames(x)
```

Approach 2: Simulate in each imputation

```
list.of.sims <- lapply(fits, function(fit)
  sim.coef <- rmvnorm(n = 1000,
                      mean = coef(fit),
                      sigma = vcov(fit))
  linear.predictor <- x %*% t(sim.coef)
  pred.p <- plogis(linear.predictor)
  rownames(pred.p) <- rownames(x)
  return(pred.p)
)
```

Approach 2: Combine across imputations

Note: `do.call(function, list)` does the `function` to all elements of the `list`.

Approach 2: Combine across imputations

Note: `do.call(function, list)` does the `function` to all elements of the `list`.

```
sims <- do.call(cbind, list.of.sims)
```

Approach 2: Combine across imputations

Note: `do.call(function, list)` does the `function` to all elements of the `list`.

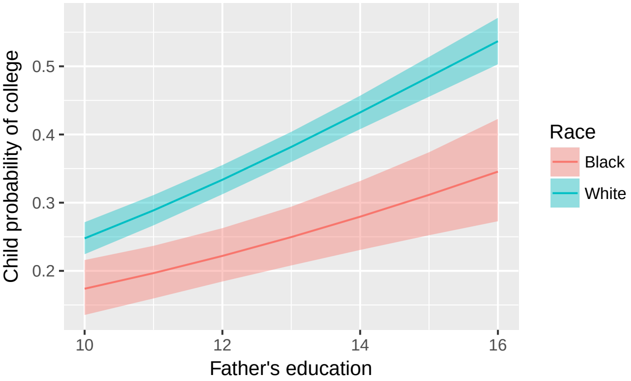
```
sims <- do.call(cbind, list.of.sims)
```

Here we column bind them all into one matrix.

Approach 2: Plot results

```
t(sims) %>%
  melt(id = NULL) %>%
  separate(Var2, into = c("Race","Education")) %>%
  group_by(Race, Education) %>%
  summarize(Estimate = mean(value),
            min = quantile(value, .025),
            max = quantile(value, .975)) %>%
  group_by() %>%
  mutate(Education = as.numeric(Education)) %>%
  ggplot(aes(x = Education, y = Estimate,
             ymin = min, ymax = max,
             fill = Race)) +
  geom_line(aes(color = Race)) +
  geom_ribbon(alpha = .4) +
  ylab("Child probability of college") +
  xlab("Father's education")
```

Approach 2: Simulation for QOIs



Another source of missingness: Ballots

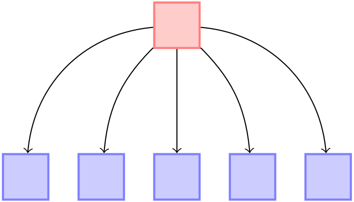
To RStudio!

The Multiple Imputation Scheme (last time I will show)

The Multiple Imputation Scheme (last time I will show)

incomplete data

The Multiple Imputation Scheme (last time I will show)

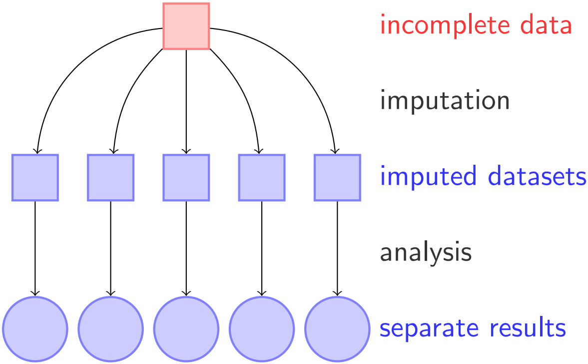


incomplete data

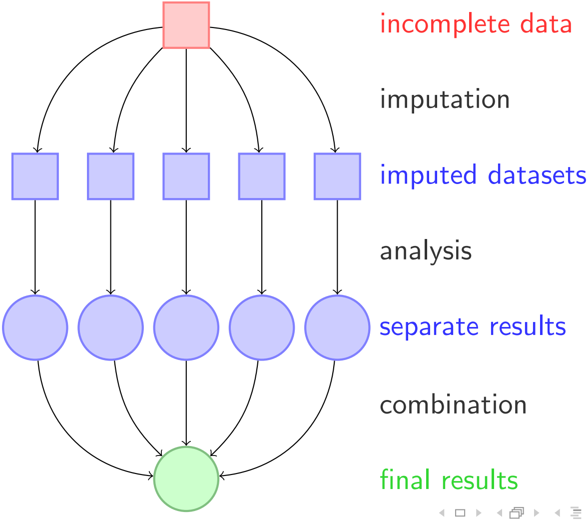
imputation

imputed datasets

The Multiple Imputation Scheme (last time I will show)



The Multiple Imputation Scheme (last time I will show)



Should we transform variables?

Quoted from Amelia documentation, p. 16:

As it turns out, much evidence in the literature (discussed in King et al. 2001) indicates that the multivariate normal model used in Amelia usually works well for the imputation stage even when discrete or non- normal variables are included and when the analysis stage involves these limited dependent variable models.

Mixture of exponentials

$$X_{0i} \sim \text{Exponential}(\lambda_0)$$

Mixture of exponentials

$$X_{0i} \sim \text{Exponential}(\lambda_0)$$
$$X_{1i} \sim \text{Exponential}(\lambda_1)$$

Mixture of exponentials

$$
\begin{aligned}
X_{0i} &\sim \text{Exponential}(\lambda_0) \\
X_{1i} &\sim \text{Exponential}(\lambda_1) \\
Z_i &\sim \text{Bernoulli}(p)
\end{aligned}
$$

Mixture of exponentials

$$
\begin{aligned}
X_{0i} &\sim \text{Exponential}(\lambda_0) \\
X_{1i} &\sim \text{Exponential}(\lambda_1) \\
Z_i &\sim \text{Bernoulli}(p) \\
Y_i &\equiv (1 - Z_i)X_{0i} + Z_i X_{1i}
\end{aligned}
$$

Mixture of exponentials

$$
\begin{aligned}
X_{0i} &\sim \text{Exponential}(\lambda_0) \\
X_{1i} &\sim \text{Exponential}(\lambda_1) \\
Z_i &\sim \text{Bernoulli}(p) \\
Y_i &\equiv (1 - Z_i)X_{0i} + Z_i X_{1i}
\end{aligned}
$$

Simulate the data

```
set.seed(08544)
```

Simulate the data

```
set.seed(08544)
x0 <- rexp(100, rate = 0.5)
```

Simulate the data

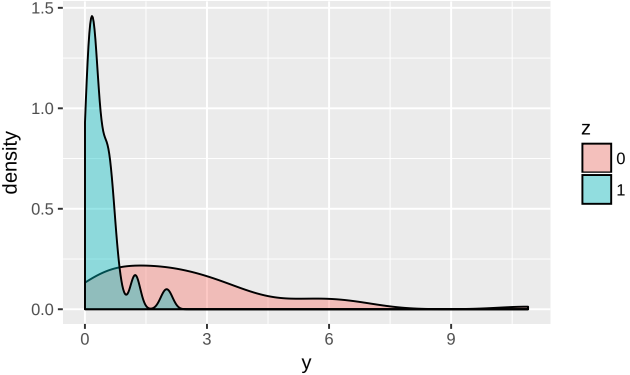
```
set.seed(08544)
x0 <- rexp(100, rate = 0.5)
x1 <- rexp(100, rate = 2)
```

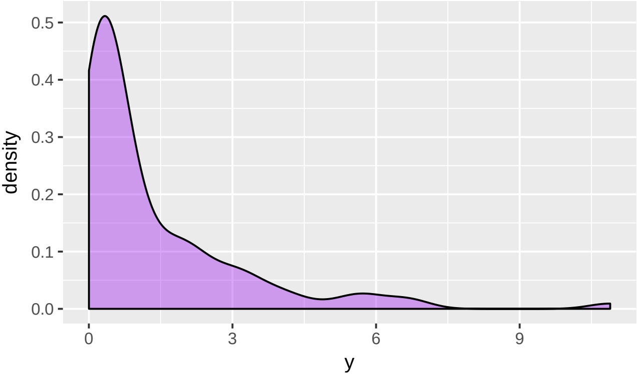
Simulate the data

```
set.seed(08544)
x0 <- rexp(100, rate = 0.5)
x1 <- rexp(100, rate = 2)
z <- rbinom(100, size = 1, prob = .6)
```

Simulate the data

```
set.seed(08544)
x0 <- rexp(100, rate = 0.5)
x1 <- rexp(100, rate = 2)
z <- rbinom(100, size = 1, prob = .6)
y <- (1 - z)*x0 + z*x1
```


1.5
1.0
0.5
0.0
density
0
3
6
9
y
z
0
1


density
0.5
0.4
0.3
0.2
0.1
0.0
0
3
6
9
y

E-step

Find the expected value of the latent variable Z_i, given the parameters $\{p^t, \lambda_0^t, \lambda_1^t\}$ and the data Y_i.

E-step

Find the expected value of the latent variable Z_i, given the parameters $\{p^t, \lambda_0^t, \lambda_1^t\}$ and the data Y_i.

We sometimes call these the responsibilities.

E-step

Find the expected value of the latent variable Z_i, given the parameters $\{p^t, \lambda_0^t, \lambda_1^t\}$ and the data Y_i.

We sometimes call these the responsibilities.

$$E(Z_i \mid p^t, \lambda_0^t, \lambda_1^t, Y_i) =$$

E-step

Find the expected value of the latent variable Z_i, given the parameters $\{p^t, \lambda_0^t, \lambda_1^t\}$ and the data Y_i.

We sometimes call these the responsibilities.

$$E(Z_i \mid p^t, \lambda_0^t, \lambda_1^t, Y_i) = P(Z_i = 1 \mid p^t, \lambda_0^t, \lambda_1^t, Y_i)$$

E-step

Find the expected value of the latent variable Z_i, given the parameters $\{p^t, \lambda_0^t, \lambda_1^t\}$ and the data Y_i.

We sometimes call these the responsibilities.

$$
\begin{aligned}
E(Z_i \mid p^t, \lambda_0^t, \lambda_1^t, Y_i) &= P(Z_i = 1 \mid p^t, \lambda_0^t, \lambda_1^t, Y_i) \\
&= \frac{P(Y_i \mid Z_i = 1) P(Z_i = 1)}{P(Y_i)}
\end{aligned}
$$

E-step

Find the expected value of the latent variable Z_i, given the parameters $\{p^t, \lambda_0^t, \lambda_1^t\}$ and the data Y_i.

We sometimes call these the responsibilities.

$$\begin{aligned}
E(Z_i \mid p^t, \lambda_0^t, \lambda_1^t, Y_i) &= P(Z_i = 1 \mid p^t, \lambda_0^t, \lambda_1^t, Y_i) \\
&= \frac{P(Y_i \mid Z_i = 1)P(Z_i = 1)}{P(Y_i)} \\
&= \frac{P(Y_i \mid Z_i = 1)P(Z_i = 1)}{P(Y_i \mid Z_i = 1)P(Z_i = 1) + P(Y_i \mid Z_i = 0)P(Z_i = 0)}
\end{aligned}$$

E-step

Find the expected value of the latent variable Z_i, given the parameters $\{p^t, \lambda_0^t, \lambda_1^t\}$ and the data Y_i.

We sometimes call these the responsibilities.

$$\begin{aligned}
E(Z_i \mid p^t, \lambda_0^t, \lambda_1^t, Y_i) &= P(Z_i = 1 \mid p^t, \lambda_0^t, \lambda_1^t, Y_i) \\
&= \frac{P(Y_i \mid Z_i = 1)P(Z_i = 1)}{P(Y_i)} \\
&= \frac{P(Y_i \mid Z_i = 1)P(Z_i = 1)}{P(Y_i \mid Z_i = 1)P(Z_i = 1) + P(Y_i \mid Z_i = 0)P(Z_i = 0)} \\
&= \frac{\lambda_1 e^{-y_i\lambda_1} p}{\lambda_1 e^{-y_i\lambda_1} p + \lambda_0 e^{-y_i\lambda_0}(1-p)}
\end{aligned}$$

Note: Conditioning on the parameters is not written explicitly after the first step to simplify the presentation. But all quantities throughout are conditional on p^t, λ_0^t, and $\lambda_1^t\}$. Likewise, P refers to both probability and probability densities for simplicity.

E-step

```
e.step <- function(p, lambda0, lambda1, y) {
  e.z <- lambda1 * exp(-y * lambda1) * p /
    lambda1 * exp(-y * lambda1) * p +
    lambda0 * exp(-y * lambda0) * (1 - p)
  return(e.z)
}
```

M-step

Find updated MLE estimates of $\{p^t, \lambda_0^t, \lambda_1^t\}$ using the data z^t created in the E-step.

M-step

Find updated MLE estimates of $\{p^t, \lambda_0^t, \lambda_1^t\}$ using the data z^t created in the E-step.

First, write the complete data log likelihood, which includes both observed and latent variables.

M-step

Find updated MLE estimates of $\{p^t, \lambda_0^t, \lambda_1^t\}$ using the data z^t created in the E-step.

First, write the complete data log likelihood, which includes both observed and latent variables.

$$L(p^t, \lambda_0^t, \lambda_1^t \mid y, z) = f(y, z \mid p^t, \lambda_0^t, \lambda_1^t)$$

M-step

Find updated MLE estimates of $\{p^t, \lambda_0^t, \lambda_1^t\}$ using the data z^t created in the E-step.

First, write the complete data log likelihood, which includes both observed and latent variables.

$$
\begin{aligned}
L(p^t, \lambda_0^t, \lambda_1^t \mid y, z) &= f(y, z \mid p^t, \lambda_0^t, \lambda_1^t) \\
&= f(y \mid z, p^t, \lambda_0^t, \lambda_1^t) f(z)
\end{aligned}
$$

M-step

Find updated MLE estimates of $\{p^t, \lambda_0^t, \lambda_1^t\}$ using the data z^t created in the E-step.

First, write the complete data log likelihood, which includes both observed and latent variables.

$$
\begin{aligned}
L(p^t, \lambda_0^t, \lambda_1^t \mid y, z) &= f(y, z \mid p^t, \lambda_0^t, \lambda_1^t) \\
&= f(y \mid z, p^t, \lambda_0^t, \lambda_1^t) f(z) \\
&= \prod_{i=1}^{n} (\lambda_1 e^{-y_i \lambda_1})^{z_i} (\lambda_0 e^{-y_i \lambda_0})^{1-z_i} p^{z_i} (1-p)^{1-z_i}
\end{aligned}
$$

M-step

Find updated MLE estimates of $\{p^t, \lambda_0^t, \lambda_1^t\}$ using the data z^t created in the E-step.

First, write the complete data log likelihood, which includes both observed and latent variables.

$$
\begin{aligned}
L(p^t, \lambda_0^t, \lambda_1^t \mid y, z) &= f(y, z \mid p^t, \lambda_0^t, \lambda_1^t) \\
&= f(y \mid z, p^t, \lambda_0^t, \lambda_1^t) f(z) \\
&= \prod_{i=1}^{n} (\lambda_1 e^{-y_i \lambda_1})^{z_i} (\lambda_0 e^{-y_i \lambda_0})^{1-z_i} p^{z_i} (1-p)^{1-z_i} \\
\ell(p^t, \lambda_0^t, \lambda_1^t \mid y, z) &= \sum_{i=1}^{n} \Big(z_i (\log \lambda_1 - y_i \lambda_1)
\end{aligned}
$$

M-step

Find updated MLE estimates of $\{p^t, \lambda_0^t, \lambda_1^t\}$ using the data z^t created in the E-step.

First, write the complete data log likelihood, which includes both observed and latent variables.

$$
\begin{aligned}
L(p^t, \lambda_0^t, \lambda_1^t \mid y, z) &= f(y, z \mid p^t, \lambda_0^t, \lambda_1^t) \\
&= f(y \mid z, p^t, \lambda_0^t, \lambda_1^t) f(z) \\
&= \prod_{i=1}^{n} (\lambda_1 e^{-y_i \lambda_1})^{z_i} (\lambda_0 e^{-y_i \lambda_0})^{1-z_i} p^{z_i} (1-p)^{1-z_i} \\
\ell(p^t, \lambda_0^t, \lambda_1^t \mid y, z) &= \sum_{i=1}^{n} \Big(z_i (\log \lambda_1 - y_i \lambda_1) \\
&\qquad + (1 - z_i)(\log \lambda_0 - y_i \lambda_0)
\end{aligned}
$$

M-step

Find updated MLE estimates of $\{p^t, \lambda_0^t, \lambda_1^t\}$ using the data z^t created in the E-step.

First, write the complete data log likelihood, which includes both observed and latent variables.

$$
\begin{aligned}
L(p^t, \lambda_0^t, \lambda_1^t \mid y, z) &= f(y, z \mid p^t, \lambda_0^t, \lambda_1^t) \\
&= f(y \mid z, p^t, \lambda_0^t, \lambda_1^t) f(z) \\
&= \prod_{i=1}^{n} (\lambda_1 e^{-y_i \lambda_1})^{z_i} (\lambda_0 e^{-y_i \lambda_0})^{1-z_i} p^{z_i} (1-p)^{1-z_i} \\
\ell(p^t, \lambda_0^t, \lambda_1^t \mid y, z) &= \sum_{i=1}^{n} \Big(z_i (\log \lambda_1 - y_i \lambda_1) \\
&\quad + (1 - z_i)(\log \lambda_0 - y_i \lambda_0) \\
&\quad + z_i \log p_i + (1 - z_i) \log(1 - p_i) \Big)
\end{aligned}
$$

M-step

Find updated MLE estimates of $\{p^t, \lambda_0^t, \lambda_1^t\}$ using the data z^t created in the E-step.

First, write the complete data log likelihood, which includes both observed and latent variables.

$$
\begin{aligned}
L(p^t, \lambda_0^t, \lambda_1^t \mid y, z) &= f(y, z \mid p^t, \lambda_0^t, \lambda_1^t) \\
&= f(y \mid z, p^t, \lambda_0^t, \lambda_1^t) f(z) \\
&= \prod_{i=1}^{n} (\lambda_1 e^{-y_i \lambda_1})^{z_i} (\lambda_0 e^{-y_i \lambda_0})^{1-z_i} p^{z_i} (1-p)^{1-z_i} \\
\ell(p^t, \lambda_0^t, \lambda_1^t \mid y, z) &= \sum_{i=1}^{n} \Big(z_i (\log \lambda_1 - y_i \lambda_1) \\
&\qquad + (1 - z_i)(\log \lambda_0 - y_i \lambda_0) \\
&\qquad + z_i \log p_i + (1 - z_i) \log(1 - p_i) \Big)
\end{aligned}
$$

M-step

```
comp.data.log.lik <- function(par,z,y) {
```

M-step

```
comp.data.log.lik <- function(par,z,y) {
  p <- plogis(par[1])
  lambda0 <- exp(par[2])
  lambda1 <- exp(par[3])
```

M-step

```
comp.data.log.lik <- function(par,z,y) {
  p <- plogis(par[1])
  lambda0 <- exp(par[2])
  lambda1 <- exp(par[3])
  log.lik <- sum(z*(log(lambda1) - y*lambda1) +
                 (1 - z)*(log(lambda0) - y*lambda0) +
                 z*log(p) + (1 - z)*log(1 - p))
  return(log.lik)
}
```

M-step

Write a function to maximize that log likelihood

```
m.step <- function(z,y) {
```

M-step

Write a function to maximize that log likelihood

```
m.step <- function(z,y) {
  opt.out <- optim(
    par = c(0,0,0),
    z = z,
    y = y,
    fn = comp.data.log.lik,
    method = "BFGS",
    control = list(fnscale = -1)
  )
```

M-step

Write a function to maximize that log likelihood

```
m.step <- function(z,y) {
  opt.out <- optim(
    par = c(0,0,0),
    z = z,
    y = y,
    fn = comp.data.log.lik,
    method = "BFGS",
    control = list(fnscale = -1)
  )
  p <- plogis(opt.out$par[1])
  lambda0 <- exp(opt.out$par[2])
  lambda1 <- exp(opt.out$par[3])
  return(list(p = p, lambda0 = lambda0,
              lambda1 = lambda1))
}
```

Put E and M together!

Initialize the matrix to store parameters

```
par.estimates <- matrix(nrow = 11, ncol = 3)
colnames(par.estimates) <- c("p.t","lambda0.t","lambda1.t")
```

Put E and M together!

Initialize the matrix to store parameters

```
par.estimates <- matrix(nrow = 11, ncol = 3)
colnames(par.estimates) <- c("p.t","lambda0.t","lambda1.t")
```

Choose starting values

```
p.t <- 0.5
lambda0.t <- 1
lambda1.t <- 1
set.seed(12345)
z.t <- rbinom(n = length(y),
              size = 1,
              prob = .5)
```

Put E and M together!

Initialize the matrix to store parameters

```
par.estimates <- matrix(nrow = 11, ncol = 3)
colnames(par.estimates) <- c("p.t","lambda0.t","lambda1.t")
```

Choose starting values

```
p.t <- 0.5
lambda0.t <- 1
lambda1.t <- 1
set.seed(12345)
z.t <- rbinom(n = length(y),
              size = 1,
              prob = .5)
```

Store our starting parameters in the matrix

```
par.estimates[1,] <- c(p.t, lambda0.t, lambda1.t)
```

Put E and M together!

Iterate

```
for (i in 2:11) {
```

Put E and M together!

Iterate

```
for (i in 2:11) {
  z.t <- e.step(p = p.t,
                lambda0 = lambda0.t,
                lambda1 = lambda1.t,
                y = y)
```

Put E and M together!

Iterate

```
for (i in 2:11) {
  z.t <- e.step(p = p.t,
                lambda0 = lambda0.t,
                lambda1 = lambda1.t,
                y = y)

  m.out <- m.step(z = z.t, y = y)
```

Put E and M together!

Iterate

```
for (i in 2:11) {
  z.t <- e.step(p = p.t,
                lambda0 = lambda0.t,
                lambda1 = lambda1.t,
                y = y)

  m.out <- m.step(z = z.t, y = y)

  p.t <- m.out$p
  lambda0.t <- m.out$lambda0
  lambda1.t <- m.out$lambda1
```

Put E and M together!

Iterate

```
for (i in 2:11) {
  z.t <- e.step(p = p.t,
                lambda0 = lambda0.t,
                lambda1 = lambda1.t,
                y = y)

  m.out <- m.step(z = z.t, y = y)

  p.t <- m.out$p
  lambda0.t <- m.out$lambda0
  lambda1.t <- m.out$lambda1

  par.estimates[i,] <- c(p.t, lambda0.t, lambda1.t)
}
```

EM convergence

Iteration	p^t	λ_0^t	λ_1^t
0	0.5000	1.0000	1.0000

EM convergence

Iteration	p^t	λ_0^t	λ_1^t
0	0.5000	1.0000	1.0000
1	0.3482	0.5409	2.7712

EM convergence

Iteration	p^t	λ_0^t	λ_1^t
0	0.5000	1.0000	1.0000
1	0.3482	0.5409	2.7712
2	0.2474	0.6271	1.8944

EM convergence

Iteration	p^t	λ_0^t	λ_1^t
0	0.5000	1.0000	1.0000
1	0.3482	0.5409	2.7712
2	0.2474	0.6271	1.8944
3	0.3010	0.5934	1.9726

EM convergence

Iteration	p^t	λ_0^t	λ_1^t
0	0.5000	1.0000	1.0000
1	0.3482	0.5409	2.7712
2	0.2474	0.6271	1.8944
3	0.3010	0.5934	1.9726
4	0.2779	0.6076	1.9548

EM convergence

Iteration	p^t	λ_0^t	λ_1^t
0	0.5000	1.0000	1.0000
1	0.3482	0.5409	2.7712
2	0.2474	0.6271	1.8944
3	0.3010	0.5934	1.9726
4	0.2779	0.6076	1.9548
5	0.2874	0.6019	1.9603

EM convergence

Iteration	p^t	λ_0^t	λ_1^t
0	0.5000	1.0000	1.0000
1	0.3482	0.5409	2.7712
2	0.2474	0.6271	1.8944
3	0.3010	0.5934	1.9726
4	0.2779	0.6076	1.9548
5	0.2874	0.6019	1.9603
6	0.2835	0.6042	1.9580

EM convergence

Iteration	p^t	λ_0^t	λ_1^t
0	0.5000	1.0000	1.0000
1	0.3482	0.5409	2.7712
2	0.2474	0.6271	1.8944
3	0.3010	0.5934	1.9726
4	0.2779	0.6076	1.9548
5	0.2874	0.6019	1.9603
6	0.2835	0.6042	1.9580
7	0.2851	0.6033	1.9589

EM convergence

Iteration	p^t	λ_0^t	λ_1^t
0	0.5000	1.0000	1.0000
1	0.3482	0.5409	2.7712
2	0.2474	0.6271	1.8944
3	0.3010	0.5934	1.9726
4	0.2779	0.6076	1.9548
5	0.2874	0.6019	1.9603
6	0.2835	0.6042	1.9580
7	0.2851	0.6033	1.9589
8	0.2844	0.6037	1.9585

EM convergence

Iteration	p^t	λ_0^t	λ_1^t
0	0.5000	1.0000	1.0000
1	0.3482	0.5409	2.7712
2	0.2474	0.6271	1.8944
3	0.3010	0.5934	1.9726
4	0.2779	0.6076	1.9548
5	0.2874	0.6019	1.9603
6	0.2835	0.6042	1.9580
7	0.2851	0.6033	1.9589
8	0.2844	0.6037	1.9585
9	0.2847	0.6035	1.9587

EM convergence

Iteration	p^t	λ_0^t	λ_1^t
0	0.5000	1.0000	1.0000
1	0.3482	0.5409	2.7712
2	0.2474	0.6271	1.8944
3	0.3010	0.5934	1.9726
4	0.2779	0.6076	1.9548
5	0.2874	0.6019	1.9603
6	0.2835	0.6042	1.9580
7	0.2851	0.6033	1.9589
8	0.2844	0.6037	1.9585
9	0.2847	0.6035	1.9587
10	0.2846	0.6036	1.9586

Next week: Causal inference

Next week: Causal inference

Questions?