Session 6

Coping with missing data

Matt Denwood

2021-06-30

Types of missingness

MCAR: Missing completely at random

- There is absolutely no pattern to the missingness
- This is the best kind

Types of missingness

MCAR: Missing completely at random

- There is absolutely no pattern to the missingness
- This is the best kind

MAR: Missing at random

- There is a pattern to the missingness but we know what it is
- This is usually possible to deal with but needs some consideration

Types of missingness

MCAR: Missing completely at random

- There is absolutely no pattern to the missingness
- This is the best kind

MAR: Missing at random

- There is a pattern to the missingness but we know what it is
- This is usually possible to deal with but needs some consideration

MNAR: Missing not at random

- There is an unknown (or unrecorded) pattern to the missingness
- It is therefore possible that the prevalence is confounded with missingness

MCAR: Missing completely at random

Missing samples can occur for any individual with equal probability

- Missingness is not correlated with anything
- There is no possibility of being confounded with prevalence

MCAR: Missing completely at random

Missing samples can occur for any individual with equal probability

- Missingness is not correlated with anything
- There is no possibility of being confounded with prevalence

Examples

- The animal was too aggressive to facilitate a blood sample
- Somebody dropped the samples on the way to the lab

MCAR: Missing completely at random

Missing samples can occur for any individual with equal probability

- Missingness is not correlated with anything
- There is no possibility of being confounded with prevalence

Examples

- The animal was too aggressive to facilitate a blood sample
- Somebody dropped the samples on the way to the lab

Possible solutions:

- Exclude individuals with incomplete data
- Allow `template_huiwalter` to adjust the model code

MCAR: Missing completely at random

Missing samples can occur for any individual with equal probability

- Missingness is not correlated with anything
- There is no possibility of being confounded with prevalence

Examples

- The animal was too aggressive to facilitate a blood sample
- Somebody dropped the samples on the way to the lab

Possible solutions:

- Exclude individuals with incomplete data
- Allow `template_huiwalter` to adjust the model code

This is a relatively rare kind of missingness, but it does happen

MAR: Missing at random

Missing samples occur due to a known pattern

- We can (and must) assess if this is likely be correlated with prevalence

MAR: Missing at random

Missing samples occur due to a known pattern

- We can (and must) assess if this is likely be correlated with prevalence

Examples:

- Test A was not done in population 1 because of costs
- Test B was only done if Test A was positive

MAR: Missing at random

Missing samples occur due to a known pattern

- We can (and must) assess if this is likely be correlated with prevalence

Examples:

- Test A was not done in population 1 because of costs
- Test B was only done if Test A was positive

Solution depends on whether the the missigness is potentially confounded with prevalence

- No -> treat as MCAR
- Yes -> we must model the confounding

MAR: Missing at random

Missing samples occur due to a known pattern

- We can (and must) assess if this is likely be correlated with prevalence

Examples:

- Test A was not done in population 1 because of costs
- Test B was only done if Test A was positive

Solution depends on whether the the missigness is potentially confounded with prevalence

- No -> treat as MCAR
- Yes -> we must model the confounding

Very common type of missingness in practice

MNAR: Missing not at random

Missing samples occur due to an unknown pattern

- We must assume that this might be correlated with prevalence

MNAR: Missing not at random

Missing samples occur due to an unknown pattern

- We must assume that this might be correlated with prevalence

Examples:

- Test B was only done if the animal had diarrhea
- The individual patient was given a choice if they wanted Test B after knowing the result of Test A

MNAR: Missing not at random

Missing samples occur due to an unknown pattern

- We must assume that this might be correlated with prevalence

Examples:

- Test B was only done if the animal had diarrhea
- The individual patient was given a choice if they wanted Test B after knowing the result of Test A

Possible solutions:

- Exclude segments of the data that may be affected by structural missingness
- Give up and collect a better dataset

MNAR: Missing not at random

Missing samples occur due to an unknown pattern

- We must assume that this might be correlated with prevalence

Examples:

- Test B was only done if the animal had diarrhea
- The individual patient was given a choice if they wanted Test B after knowing the result of Test A

Possible solutions:

- Exclude segments of the data that may be affected by structural missingness
- Give up and collect a better dataset

A common type of missingness in secondary data

Missingness and template Hui-Walter

We can simulate MCAR data as follows:

```
set.seed(2021-06-30)
# Parameter values to simulate:
N <- 1000
sensitivity <- c(0.8, 0.9, 0.95)
specificity <- c(0.95, 0.99, 0.95)

Populations <- 2
prevalence <- c(0.25,0.5)

data <- tibble(Population = sample(seq_len(Populations), N,
↪  replace=TRUE)) %>%
  mutate(Status = rbinom(N, 1, prevalence[Population])) %>%
  mutate(Test1 = rbinom(N, 1, sensitivity[1]*Status +
  ↪  (1-specificity[1])*(1-Status))) %>%
  mutate(Test2 = rbinom(N, 1, sensitivity[2]*Status +
  ↪  (1-specificity[2])*(1-Status))) %>%
  mutate(Test3 = rbinom(N, 1, sensitivity[3]*Status +
  ↪  (1-specificity[3])*(1-Status))) %>%
  select(-Status)
```

Now introduce missingness in all 3 tests:

```
missingness <- c(0.1, 0.2, 0.3)
data <- data %>%
  mutate(Test1 = case_when(
    rbinom(n(), 1, missingness[1]) == 1L ~ NA_integer_,
    TRUE ~ Test1
  )) %>%
  mutate(Test2 = case_when(
    rbinom(n(), 1, missingness[2]) == 1L ~ NA_integer_,
    TRUE ~ Test2
  )) %>%
  mutate(Test3 = case_when(
    rbinom(n(), 1, missingness[3]) == 1L ~ NA_integer_,
    TRUE ~ Test3
  ))
```

```r
data %>% count(Missing1 = is.na(Test1), Missing2 = is.na(Test2),
↪  Missing3 = is.na(Test3))
## # A tibble: 8 x 4
##   Missing1 Missing2 Missing3     n
##   <lgl>    <lgl>    <lgl>    <int>
## 1 FALSE    FALSE    FALSE      513
## 2 FALSE    FALSE    TRUE       210
## 3 FALSE    TRUE     FALSE      126
## 4 FALSE    TRUE     TRUE        56
## 5 TRUE     FALSE    FALSE       54
## 6 TRUE     FALSE    TRUE        20
## 7 TRUE     TRUE     FALSE       14
## 8 TRUE     TRUE     TRUE         7
```

We can simply feed this data to `template_huiwalter`:

```
template_huiwalter(data, outfile="huiwalter_MAR.txt")
## The model and data have been written to huiwalter_MAR.txt in the
↪  current working directory
## You should check and alter priors before running the model
```

What does that look like. . . ?

```
model{

    ## Observation layer:

    # Complete observations (N=513):
    for(p in 1:Populations){
        Tally_RRR[1:8,p] ~ dmulti(prob_RRR[1:8,p], N_RRR[p])

        prob_RRR[1:8,p] <- se_prob[1:8,p] + sp_prob[1:8,p]
    }
```

```
# Partial observations (Test1: Recorded, Test2: Missing, Test3:
↪  Missing; N=56):
for(p in 1:Populations){
    Tally_RMM[1:2,p] ~ dmulti(prob_RMM[1:2,p], N_RMM[p])

    prob_RMM[1:2,p] <-  se_prob[c(1,2),p] + sp_prob[c(1,2),p] +
                        se_prob[c(3,4),p] + sp_prob[c(3,4),p] +
                        se_prob[c(5,6),p] + sp_prob[c(5,6),p] +
                        se_prob[c(7,8),p] + sp_prob[c(7,8),p]
}

# Partial observations (Test1: Recorded, Test2: Recorded, Test3:
↪  Missing; N=210):
for(p in 1:Populations){
    Tally_RRM[1:4,p] ~ dmulti(prob_RRM[1:4,p], N_RRM[p])

    prob_RRM[1:4,p] <-  se_prob[c(1,2,3,4),p] +
    ↪  sp_prob[c(1,2,3,4),p] +
```

```
}

# Partial observations (Test1: Missing, Test2: Recorded, Test3:
↪  Recorded; N=54):
for(p in 1:Populations){
    Tally_MRR[1:4,p] ~ dmulti(prob_MRR[1:4,p], N_MRR[p])

    prob_MRR[1:4,p] <-  se_prob[c(1,3,5,7),p] +
    ↪  sp_prob[c(1,3,5,7),p] +
                        se_prob[c(2,4,6,8),p] +
                        ↪  sp_prob[c(2,4,6,8),p]
}

# Partial observations (Test1: Missing, Test2: Recorded, Test3:
↪  Missing; N=20):
for(p in 1:Populations){
    Tally_MRM[1:2,p] ~ dmulti(prob_MRM[1:2,p], N_MRM[p])

    prob_MRM[1:2,p] <-  se_prob[c(1,3),p] + sp_prob[c(1,3),p] +
                        se_prob[c(2,4),p] + sp_prob[c(2,4),p] +
                        se_prob[c(5,7),p] + sp_prob[c(5,7),p] +
                        se_prob[c(6,8),p] + sp_prob[c(6,8),p]
}
```

```
# Partial observations (Test1: Missing, Test2: Missing, Test3:
↪  Recorded; N=14):
for(p in 1:Populations){
    Tally_MMR[1:2,p] ~ dmulti(prob_MMR[1:2,p], N_MMR[p])

    prob_MMR[1:2,p] <-  se_prob[c(1,5),p] + sp_prob[c(1,5),p] +
                        se_prob[c(2,6),p] + sp_prob[c(2,6),p] +
                        se_prob[c(3,7),p] + sp_prob[c(3,7),p] +
                        se_prob[c(4,8),p] + sp_prob[c(4,8),p]
}
```

```
# Partial observations (Test1: Missing, Test2: Missing, Test3:
↪  Recorded; N=14):
for(p in 1:Populations){
    Tally_MMR[1:2,p] ~ dmulti(prob_MMR[1:2,p], N_MMR[p])

    prob_MMR[1:2,p] <-  se_prob[c(1,5),p] + sp_prob[c(1,5),p] +
                        se_prob[c(2,6),p] + sp_prob[c(2,6),p] +
                        se_prob[c(3,7),p] + sp_prob[c(3,7),p] +
                        se_prob[c(4,8),p] + sp_prob[c(4,8),p]
}
```

NB: `MMM` combinations have been removed!

```
## Observation probabilities:

for(p in 1:Populations){

    # Probability of observing Test1- Test2- Test3- from a true
    ↪  positive::
    se_prob[1,p] <- prev[p] * ((1-se[1])*(1-se[2])*(1-se[3])
    ↪  +covse12 +covse13 +covse23)
    # Probability of observing Test1- Test2- Test3- from a true
    ↪  negative::
    sp_prob[1,p] <- (1-prev[p]) * (sp[1]*sp[2]*sp[3] +covsp12
    ↪  +covsp13 +covsp23)

    # Probability of observing Test1+ Test2- Test3- from a true
    ↪  positive::
    se_prob[2,p] <- prev[p] * (se[1]*(1-se[2])*(1-se[3]) -covse12
    ↪  -covse13 +covse23)
    # Probability of observing Test1+ Test2- Test3- from a true
    ↪  negative::
    sp_prob[2,p] <- (1-prev[p]) * ((1-sp[1])*sp[2]*sp[3] -covsp12
    ↪  -covsp13 +covsp23)

    # Probability of observing Test1- Test2+ Test3- from a true
    ↪  positive::
    se_prob[3,p] <- prev[p] * ((1-se[1])*se[2]*(1-se[3]) -covse12
    ↪  +covse13 -covse23)
    # Probability of observing Test1- Test2+ Test3- from a true
```

```
## Data:
data{
"Populations" <- 2
"N_RRR" <- c(233, 280)
"Tally_RRR" <- structure(c(148, 8, 1, 2, 9, 4, 11, 50, 133, 3, 4, 5, 8,
↪  9, 20, 98), .Dim = c(8, 2))
"N_RMR" <- c(65, 61)
"Tally_RMR" <- structure(c(51, 3, 1, 10, 29, 2, 11, 19), .Dim = c(4, 2))
"N_RMM" <- c(22, 34)
"Tally_RMM" <- structure(c(16, 6, 20, 14), .Dim = c(2, 2))
"N_RRM" <- c(100, 110)
"Tally_RRM" <- structure(c(74, 5, 2, 19, 58, 10, 5, 37), .Dim = c(4, 2))
"N_MRR" <- c(27, 27)
"Tally_MRR" <- structure(c(18, 1, 4, 4, 15, 2, 1, 9), .Dim = c(4, 2))
"N_MRM" <- c(10, 10)
"Tally_MRM" <- structure(c(7, 3, 4, 6), .Dim = c(2, 2))
"N_MMR" <- c(6, 8)
"Tally_MMR" <- structure(c(4, 2, 2, 6), .Dim = c(2, 2))
}
```

What about other types of missing?

MAR:

- As for MCAR
- As long as the randomness structure is not confounded with prevalence!

What about other types of missing?

MAR:

- As for MCAR
- As long as the randomness structure is not confounded with prevalence!

MNAR:

- Solution depends entirely on the problem
- And sometimes there is no solution...

What about other types of missing?

MAR:

- As for MCAR
- As long as the randomness structure is not confounded with prevalence!

MNAR:

- Solution depends entirely on the problem
- And sometimes there is no solution...

But remember: bigger datasets are not always better datasets...

Making your data missing

What happens if we eliminate:

- One population at a time (where we have >2)?
- One test at a time (where we have >2)?

Making your data missing

What happens if we eliminate:

- One population at a time (where we have >2)?
- One test at a time (where we have >2)?

Do the results change?

Making your data missing

What happens if we eliminate:

- One population at a time (where we have >2)?
- One test at a time (where we have >2)?

Do the results change?

If we have >2 populations *and* >2 tests then we can eliminate one combination at a time!

Making your data missing

What happens if we eliminate:

- One population at a time (where we have >2)?
- One test at a time (where we have >2)?

Do the results change?

If we have >2 populations *and* >2 tests then we can eliminate one combination at a time!

This is a very useful form of cross-validation

How can we do this?

```
all_combinations <- data %>%
  pivot_longer(-Population, names_to = "Test", values_to = "Result") %>%
  filter(!is.na(Result)) %>%
  count(Population, Test) %>%
  print()
## # A tibble: 6 x 3
##   Population Test      n
##        <int> <chr> <int>
## 1          1 Test1   420
## 2          1 Test2   370
## 3          1 Test3   331
## 4          2 Test1   485
## 5          2 Test2   427
## 6          2 Test3   376
all_results <- vector('list', length=nrow(all_combinations))
all_summary <- vector('list', length=nrow(all_combinations))
```

```r
crossval_data <- data %>%
  mutate(Test1 = case_when(
    Population == 1 ~ NA_integer_,
    TRUE ~ Test1
  ))

template_huiwalter(crossval_data, "model_m11.txt")
all_results[[1]] <- run.jags("model_m11.txt")
## Loading required namespace: rjags
all_summary[[1]] <- summary(all_results[[1]], vars="^s") %>%
  as.data.frame() %>%
  rownames_to_column("Parameter") %>%
  mutate(Model = "M11") %>%
  select(Model, Parameter, Median, Lower95, Upper95)
```

```
all_summary[[1]]
##   Model Parameter    Median   Lower95   Upper95
## 1   M11     se[1] 0.8200204 0.7607160 0.8760699
## 2   M11     se[2] 0.8936436 0.8390739 0.9442273
## 3   M11     se[3] 0.9457673 0.9063256 0.9792820
## 4   M11     sp[1] 0.9711451 0.9386485 0.9969071
## 5   M11     sp[2] 0.9851258 0.9673249 0.9999803
## 6   M11     sp[3] 0.9503531 0.9150855 0.9811673
```

```r
crossval_data <- data %>%
  mutate(Test2 = case_when(
    Population == 1 ~ NA_integer_,
    TRUE ~ Test2
  ))

template_huiwalter(crossval_data, "model_m12.txt")
all_results[[2]] <- run.jags("model_m12.txt")
all_summary[[2]] <- summary(all_results[[2]], vars="^s") %>%
  as.data.frame() %>%
  rownames_to_column("Parameter") %>%
  mutate(Model = "M11") %>%
  select(Model, Parameter, Median, Lower95, Upper95)
```

Are there any substantial disagreements:

```
bind_rows(all_summary) %>%
  arrange(Parameter, Model)
```

```
##    Model Parameter    Median   Lower95   Upper95
## 1    M11     se[1] 0.8200204 0.7607160 0.8760699
## 2    M11     se[1] 0.8267084 0.7691586 0.8801953
## 3    M11     se[2] 0.8936436 0.8390739 0.9442273
## 4    M11     se[2] 0.8948449 0.8408723 0.9456082
## 5    M11     se[3] 0.9457673 0.9063256 0.9792820
## 6    M11     se[3] 0.9382296 0.8916897 0.9779413
## 7    M11     sp[1] 0.9711451 0.9386485 0.9969071
## 8    M11     sp[1] 0.9654473 0.9395500 0.9908151
## 9    M11     sp[2] 0.9851258 0.9673249 0.9999803
## 10   M11     sp[2] 0.9777336 0.9485060 0.9999996
## 11   M11     sp[3] 0.9503531 0.9150855 0.9811673
## 12   M11     sp[3] 0.9530065 0.9173638 0.9859744
```

Practical session 6

Points to consider

1. How does MCAR data impact your results?
2. What about if you have data using confirmatory tests?
3. How can we use cross-validation as a method of checking assumptions?

Summary

- Observations that are MCAR are trivial to deal with using JAGS
- We can also treat MAR observations as if they are MCAR as long as the reason for missingness does not confound with expected prevalence, or we allow prevalence to differ between groups where the structural missingness differs
- MNAR is bad news
- Deliberately making observations missing is a good way to assess model assumptions