Biostatistics II Exam

2025-10-27

Question 1

The Missingness in Cannabis Use does not depend on the outcome or the covariates in the Figure 1A. There is no path going from $Miss_{cu}$ to *Outcome*, *Exposure* and the *covariates*, hence we can say that $Miss_{cu}$ is d-separated from them and is independent of them. So, the probability of missingness does not depend on any variable in the Figure 1A, therefore, missingness in cannabis use is missing completely at random (MCAR).

Question 2

By definition, Missing at Random (MAR) is when the missingness in Cannabis Use can be explained by associations with the observed data, specifically fully observed data of covariates or outcome or both. Which is what we observe in Figure 1B, 1F and 1D respectively. In terms of d-separation, in the Figure 1B, there are multiple open paths from $Miss_{cu}$ to *Outcome* :

1. $$Miss_{cu} \leftarrow Maternal\ Substance\ Use \rightarrow Outcome$$
2. $$Miss_{cu} \leftarrow Maternal\ Substance\ Use \rightarrow Cannabis\ Use \rightarrow Outcome$$
3. $$Miss_{cu} \leftarrow Maternal\ Substance\ Use \rightarrow Cannabis\ Use \leftarrow Sex \rightarrow Outcome$$

and can all be blocked when *Maternal Substance Use* is adjusted for, where it is possible as *Maternal Substance Use* is fully observed. Then, $Miss_{cu}$ and *Outcome* are independent conditioned on *Maternal Substance Use*. Hence, as missingness can be explained with observed data, this is MAR.

Secondly in the Figure 1D, there is an additional open path compared to Figure 1B:

$$Miss_{cu} \leftarrow Outcome$$

which means that *Outcome* is causing the missingness in Cannabis Use and not the other way around. This path can be blocked by conditioning on *Outcome*, which is fully observed as well. Hence, the Missingness in Cannabis Use and *Outcome* are d-separated conditioned on *Maternal Substance Use* and *Outcome*. Therefore, missingness is MAR.

Lastly, in the Figure 1F, the only open path existing is:

$$Miss_{cu} \leftarrow Outcome$$

Hence, the Missingness in Cannabis Use and *Outcome* are d-separated conditioned on *Outcome* and is MAR.

Question 3

By definition, Missingness Not At Random (MNAR) is when missingness cannot be explained by the observed data. For example, when Missingness in Cannabis Use is explained by the Cannabis Use itself, we can say that the missingness is not at random. In terms of d-separation, firstly in Figure 1C, there are multiple open paths from $Miss_{cu}$ to $Outcome$:

1. $$Miss_{cu} \leftarrow Cannabis\ Use \rightarrow Outcome$$
2. $$Miss_{cu} \leftarrow Cannabis\ Use \leftarrow Sex \rightarrow Outcome$$
3. $$Miss_{cu} \leftarrow Maternal\ Substance\ Use \rightarrow Outcome$$
4. $$Miss_{cu} \leftarrow Maternal\ Substance\ Use \rightarrow Cannabis\ Use \rightarrow Outcome$$
5. $$Miss_{cu} \leftarrow Maternal\ Substance\ Use \rightarrow Cannabis\ Use \leftarrow Sex \rightarrow Outcome$$

Here, the first two paths can only be blocked by conditioning on $CannabisUse$, which is not fully observed. Hence, $Miss_{cu}$ and $Outcome$ cannot be d-separated and are not independent of each other. Therefore, is MNAR.

Similarly in Figure 1E, the open path of

$$Miss_{cu} \leftarrow Outcome$$

is added to the previous list of Figure 1C, which can be blocked by adjusting for $Outcome$. Still, the $Miss_{cu}$ and $Outcome$ cannot be d-separated, even after adjusting for all observed variables and is MNAR.

Question 4

The linear regression model is:

$\mathbf{Outcome} \sim \mathbf{MaternalSubstanceUse} + \mathbf{Sex} + \mathbf{CannabisUse}$

$\Rightarrow \mathbf{Y} = \beta_{\mathbf{0}} + \beta_{\mathbf{1}} \times \mathbf{M} + \beta_{\mathbf{2}} \times \mathbf{S} + \beta_{\mathbf{3}} \times \mathbf{X}$

Here, in both Figure 1C and 1D, we know that the exposure:

$\mathbf{CannabisUse}(\mathbf{X}) \sim \mathbf{MaternalSubstanceUse}(\mathbf{M}) + \mathbf{Sex}(\mathbf{S})$.

Although,

in Figure 1C : $\mathbf{MissingnessInCannabisUse}(\mathbf{R}) \sim \mathbf{MaternalSubstanceUse}(\mathbf{M}) + \mathbf{CannabisUse}(\mathbf{X})$,

in Figure 1D : $\mathbf{R} \sim \mathbf{MaternalSubstanceUse}(\mathbf{M}) + \mathbf{Outcome}(\mathbf{Y})$.

First, we simulate data based on the knowledge we know from the above mentioned causal diagrams. Since, there is missingness in exposure, we need to factor in the marginal distribution of $CannibasUse$ into the likelihood. Then, we use the `frm_em()` function to carry out the actual estimation:

```
set.seed(1)

generate_data <- function(n, dag) {
  beta0 <- 0
  beta1 <- 1
  beta2 <- 1
  beta3 <- 1
```

```r
  pX <- function(M, S)
    plogis(M + S)

  M <- rbinom(n, 1, 0.5)
  S <- rbinom(n, 1, 0.5)
  X <- rbinom(n, 1, pX(0.5*M, 0.5*S))
  Y <- rnorm(n, mean = beta0 + beta1*M + beta2*S + beta3*X, sd=1)

  # chose the parameters in a way to show bias evidently
  if (dag == "fig1D") {
    # scale the continuous value Y so logit doesn't explode
    R <- rbinom(n, 1, pX(0.5*M, 3*scale(Y)[,1]))
  } else if(dag == "fig1C") {
    R <- rbinom(n, 1, pX(0.5* M, 3*X))
  }
  Xobs <- ifelse(R==1, X, NA_integer_)
  dat <- data.frame(Y=Y, X=Xobs, M=M, S=S, R=R, X_true=X)
  return(dat)
}

# Maximum Likelihood Estimation with Missing data
dep <- list(model="linreg", formula=Y ~ X + M + S)
ind <- list(X = list(model="logistic", formula=X ~ M + S))


fit_frm_em_once <- function(n, dag) {
  dat <- generate_data(n, dag)
  sink(tempfile()) # to suppress the progress bar output
  fit <- frm_em(dat=dat, dep=dep, ind=ind, verbose=FALSE)
  sink()
  cf  <- coef(fit)
  beta_X  <- unname(cf["Y ON X"])
  se_X  <- unname(fit$se["Y ON X"])
  beta_M  <- unname(cf["Y ON M"])
  se_M  <- unname(fit$se["Y ON M"])
  beta_S  <- unname(cf["Y ON S"])
  se_S  <- unname(fit$se["Y ON S"])

  c(beta_X=beta_X, se_X=se_X, beta_M=beta_M, se_M=se_M, beta_S=beta_S, se_S=se_S)
}

eval_ML_grid <- function(n_vec, reps, dag){
  beta_true <- 1
  out <- lapply(n_vec, function(n){
    M <- replicate(reps, fit_frm_em_once(n, dag))
    beta_X <- M["beta_X",]; se_X <- M["se_X",]
    beta_M <- M["beta_M",]; se_M <- M["se_M",]
    beta_S <- M["beta_S",]; se_S <- M["se_S",]
    coverage_X <- mean((beta_X - 1.96*se_X) <= beta_true & (beta_X + 1.96*se_X) >= beta_true)
    coverage_M <- mean((beta_M - 1.96*se_M) <= beta_true & (beta_M + 1.96*se_M) >= beta_true)
    coverage_S <- mean((beta_S - 1.96*se_S) <= beta_true & (beta_S + 1.96*se_S) >= beta_true)
    data.frame(
      n=n,
```

```
      mean_X = mean(beta_X), bias_X = mean(beta_X)-beta_true, coverage_X = coverage_X,
      mean_M = mean(beta_M), bias_M = mean(beta_M)-beta_true, coverage_M = coverage_M,
      mean_S = mean(beta_S), bias_S = mean(beta_S)-beta_true, coverage_S = coverage_S
    )
  })
  rbindlist(out)
}
```

Now, we compare the ML estimates with the true value $\beta_{\mathbf{X}} = \mathbf{1}$.

```
# should be unbiased for Fig 1D
# repeating for increasing values of n towards infinity to discover the asymptotic behavior accurately
n_vec <- c(1000, 5000, 10000, 50000)
reps  <- 100

res_1D  <- eval_ML_grid(n_vec, reps, "fig1D")
print(res_1D)
```

```
##        n    mean_X        bias_X coverage_X    mean_M       bias_M coverage_M
##    <num>     <num>         <num>      <num>     <num>        <num>      <num>
## 1:  1000 0.9986669 -0.0013331487       1.00 1.0060119  0.006011905       0.96
## 2:  5000 0.9925474 -0.0074526433       0.92 0.9979522 -0.002047811       0.99
## 3: 10000 0.9957489 -0.0042510871       0.98 1.0044077  0.004407706       0.93
## 4: 50000 1.0003786  0.0003785598       0.93 0.9989651 -0.001034925       0.98
##      mean_S       bias_S coverage_S
##       <num>        <num>      <num>
## 1: 1.000181 0.0001812538       0.94
## 2: 1.005911 0.0059112280       0.89
## 3: 1.002693 0.0026930806       0.98
## 4: 1.000387 0.0003866465       0.96
```

As n increases, we see that the mean of ML estimates converges towards the true value 1, the biases converges towards 0 and the coverage for $\beta_X, \beta_M, \beta_S$ is above 95%. Hence, we can conclude that the maximum likelihood estimation of the regression coefficients is asymptotically unbiased for Figure 1D.

Next for Figure 1C, we run ML estimation:

```
# should be biased for Fig 1C
res_1C  <- eval_ML_grid(n_vec, reps, "fig1C")
print(res_1C)
```

```
##        n   mean_X     bias_X coverage_X   mean_M     bias_M coverage_M   mean_S
##    <num>    <num>      <num>      <num>    <num>      <num>      <num>    <num>
## 1:  1000 1.018998 0.01899751       0.94 1.037731 0.03773132       0.92 1.017965
## 2:  5000 1.017253 0.01725283       0.95 1.042217 0.04221682       0.72 1.011160
## 3: 10000 1.016842 0.01684236       0.89 1.046801 0.04680146       0.34 1.008715
## 4: 50000 1.017755 0.01775527       0.65 1.047371 0.04737123       0.00 1.010984
##         bias_S coverage_S
##          <num>      <num>
## 1: 0.017965001       0.94
## 2: 0.011160213       0.90
## 3: 0.008714647       0.95
## 4: 0.010984011       0.70
```

Here, as n increases, the mean of the ML estimates don't converge to the true value, their biases don't converge towards 0 and the coverage for $\beta_X, \beta_M, \beta_S$ goes well below 95%. Hence, we can say that the ML estimation of these regression coefficients is biased for Figure 1C where the missingness of the exposure

depends on the exposure itself.

Question 5

Now, for the same linear regression model, let us check whether the complete case analysis estimates are asymptotically unbiased or biased for the causal diagram in Figure 1C and 1D.

In Complete Case Analysis (CCA), we only consider the data that is observed for all variables in the main analysis. To calculate the CCA estimates of the regression coefficients, we simply fit the `lm` model.

```
fit_lm_once <- function(n, dag) {
  dat <- generate_data(n, dag)
  fit <- lm(formula=Y ~ X + M + S, data=dat)
  cf  <- coef(fit)
  beta_X  <- unname(cf["X"])
  beta_M  <- unname(cf["M"])
  beta_S  <- unname(cf["S"])

  c(beta_X=beta_X, beta_M=beta_M, beta_S=beta_S)
}

eval_CCA_grid <- function(n_vec, reps, dag){
  beta_true <- 1
  out <- lapply(n_vec, function(n){
    M <- replicate(reps, fit_lm_once(n, dag))
    beta_X <- M["beta_X",]
    beta_M <- M["beta_M",]
    beta_S <- M["beta_S",]
    data.frame(
      n=n,
      mean_X = mean(beta_X), bias_X = mean(beta_X)-beta_true,
      mean_M = mean(beta_M), bias_M = mean(beta_M)-beta_true,
      mean_S = mean(beta_S), bias_S = mean(beta_S)-beta_true
    )
  })
  rbindlist(out)
}
```

Now, we compare the CCA estimates with the true value $\beta_{\mathbf{X}} = \mathbf{1}$.

```
# should be unbiased for Fig 1C
# repeating for increasing values of n towards infinity to discover the asymptotic behavior accurately
n_vec <- c(1000, 5000, 10000, 50000)
reps  <- 100

res_1C  <- eval_CCA_grid(n_vec, reps, "fig1C")
print(res_1C)
```

```
##        n    mean_X        bias_X    mean_M        bias_M    mean_S
##    <num>     <num>         <num>     <num>         <num>     <num>
## 1:  1000 1.0033806  0.0033806460 1.0135501  0.0135501104 0.9924906
## 2:  5000 1.0063460  0.0063459812 0.9981872 -0.0018127691 1.0017105
## 3: 10000 0.9983305 -0.0016694988 0.9976824 -0.0023175923 0.9992814
## 4: 50000 1.0005474  0.0005474277 0.9997930 -0.0002069994 0.9999567
##           bias_S
##            <num>
```

```
## 1: -7.509376e-03
## 2:  1.710527e-03
## 3: -7.185517e-04
## 4: -4.330296e-05
```

As n increases, we see that the mean of CCA estimates converges towards the true value 1 and their biases converges towards 0. Hence, we can conclude that the complete case analysis estimation of the regression coefficients is asymptotically unbiased for Figure 1C.

Next for Figure 1D, we run CCA estimation:

```
# should be biased for Fig 1D
res_1D  <- eval_CCA_grid(n_vec, reps, "fig1D")
print(res_1D)
```

```
##        n    mean_X     bias_X    mean_M     bias_M    mean_S     bias_S
##    <num>     <num>      <num>     <num>      <num>     <num>      <num>
## 1:  1000 0.6502535 -0.3497465 0.5918120 -0.4081880 0.6778814 -0.3221186
## 2:  5000 0.6553896 -0.3446104 0.5895332 -0.4104668 0.6818220 -0.3181780
## 3: 10000 0.6599010 -0.3400990 0.5937168 -0.4062832 0.6801734 -0.3198266
## 4: 50000 0.6575508 -0.3424492 0.5918658 -0.4081342 0.6772848 -0.3227152
```

Here, as n increases, the mean of the CCA estimates doesn't converge to the true value and the bias doesn't converge towards 0. Hence, we can say that the CCA estimation of these regression coefficients is biased for Figure 1D where the missingness of the exposure depends on observable and explainable variables and ignoring them tends to cause bias.

Question 6

Question 7

Question 8

Question 9

The assignment took 11.5 hours to complete.