# Incomplete Data Analysis - Assignment 3

Yile Shi (s2168022)

2022/3/31

## Question 1

**(a) - Solution**

Function `cc()` in `mice` package returns the complete cases in a dataset. Thus, we count the number of complete cases in dataset 'nhanes' and further compute the percentage of incomplete cases by $1 - \frac{\#complete\ cases}{\#all\ cases}$.

As a result, we obtain the percentage of incomplete cases is 48% (12 cases out of 25).

```
# the percentage of incomplete cases
1 - nrow(cc(nhanes)) / nrow(nhanes)
```

```
## [1] 0.48
```

**(b) - Solution**

Recall that to estimate the variance of $\hat{\theta}^{MI}$, the multiple imputation of $\theta$, we calculate the following statistics:

1. **between-imputation variance:** $B = \frac{1}{M-1} \sum_{i=1}^{M} \left( \hat{\theta}^{(i)} - \hat{\theta}^{MI} \right)^2$

2. **within-imputation variance:** $\overline{U} = \frac{1}{M} \sum_{i=1}^{M} \hat{U}^{(i)}$, where $\hat{U}^{(i)}$ is the estimated variance of $\hat{\theta}^{(i)}$

Then we obtain the **total variance** $V^{MI} = \overline{U} + \left(1 + \frac{1}{M}\right) B$. Note that $M$ denotes the number of imputed datasets by `mice` ($M = 5$ by default).

According to the work by Buuren et al. 2011, the proportion of total variance that is attributed to the missing values is

$$\lambda = \frac{B + \frac{B}{M}}{V^{MI}}$$

We yield the following results with the standard MICE procedure - `mice()`, `with()` and `pool()`. Note that in step 2 (`with()`), we fit the normal linear regression of `bmi` over `age`, `hyp` and `cyl`.

As mentioned before, we look at the column `lambda` to get the proportion of variance due to missing data for each parameter (3 digits, seed $= 1$):

1. $\lambda_{age} = 0.686$

2. $\lambda_{hyp} = 0.350$

3. $\lambda_{chl} = 0.304$

Based on the proportion of variance due to missing data, we conclude that `age` appears to be the most affected by the non-response with the largest value of $\lambda$.

```
# impute dataset with mice
imp.list <- mice(nhanes, printFlag = FALSE, seed = 1)

# linear model to predict bmi
bmi.fit <- with(imp.list, lm(bmi ~ age + hyp + chl))

# pool the results
bmi.pool <- pool(bmi.fit)
bmi.pool$pooled
```

```
##         term m    estimate         ubar            b            t dfcom
## 1 (Intercept) 5 19.61789252 10.588884721 0.8662133972 11.628340797    21
## 2         age 5 -3.55287155  0.744810536 1.3585594461  2.375081872    21
## 3         hyp 5  2.19701748  2.886391704 1.2976511612  4.443573098    21
## 4         chl 5  0.05378081  0.000287288 0.0001046083  0.000412818    21
##          df        riv     lambda       fmi
## 1 16.936189 0.09816483 0.08938989 0.1807424
## 2  3.528053 2.18884032 0.68640637 0.7824821
## 3  9.035494 0.53949067 0.35043452 0.4583762
## 4 10.228828 0.43694808 0.30408063 0.4092932
```

**(c) - Solution**

We repeat same analysis on $\lambda$ by using different random seeds from 2 to 6. The result are shown in the data frame below.

We can observe that the conclusions in 1.(b) do not remain the same for different seeds. The values of $\lambda$ and the parameter most affected by the non-response vary dramatically.

For seed 2, 3 and 6, `age` has the largest value of $\lambda$ (0.403, 0.590, 0.655 of each, 3 digits) and is considered to be most affected by the non-response, which is consistent with conclusions in 1.(b), using random seed 1. On the other hand, `chl` and `hyp` take the largest proportion of variance for seed 4 and 5 respectively, with values 0.331 and 0.594 of each.

Recall that the larger amount of missing data, the larger the variability of values of $\lambda$ will be. Since nearly half (48%) of cases in the dataset are incomplete, it is reasonable to have different conclusions every time we adjust the random seed.

```r
# create a data frame to store the results from different seeds
bmi.lambda <- data.frame(age = bmi.pool$pooled[2, 10],
                         hyp = bmi.pool$pooled[3, 10],
                         chl = bmi.pool$pooled[4, 10],
                         row.names = "seed.1")

# repeat analysis with different seeds
for (i in seq(2, 6)){
  imp.list <- mice(nhanes, printFlag = FALSE, seed = i)
  bmi.fit <- with(imp.list, lm(bmi ~ age + hyp + chl))
  bmi.pool <- pool(bmi.fit)
  df <- data.frame(age = bmi.pool$pooled[2, 10],
                   hyp = bmi.pool$pooled[3, 10],
                   chl = bmi.pool$pooled[4, 10],
                   row.names = paste("seed.", i, sep = ""))
  bmi.lambda <- rbind(bmi.lambda, df)
}
bmi.lambda
```

```
##                age       hyp       chl
## seed.1 0.6864064 0.3504345 0.3040806
## seed.2 0.4033924 0.1430995 0.2959966
## seed.3 0.5895051 0.4101152 0.5621346
## seed.4 0.2189333 0.1961083 0.3305334
## seed.5 0.4511896 0.5942866 0.2346065
## seed.6 0.6549523 0.2960364 0.5196295
```

**(d) - Solution**

From 1.(c), we observe that the parameter most affected by the non-response varies as random seed changes. Thus, we expect that increasing the number of imputed datasets can lead to more consistent and stable results which has low dependence on the choice of random seed.

We keep using same random seeds from 1 to 6, but change the number of imputed datasets, $M$, from 5 to 100. Again, we obtain a data frame with values of $\lambda$ for each parameter with different random seed.

Now we can find that `age` always takes the largest proportion of variance due to missing cases (largest value of $\lambda$) in 5 of 6 seeds. For seed 3, the variable `chl` becomes the parameter which is affected by the non-response, with $\lambda = 0.328$. It appears that the conclusions become more consistent with $M = 100$, as we expected.

As far as I am concerned, I prefer the analyses with $M = 100$.

Recall that the **total variance** of $\theta$ (`bmi` in this question) is calculated by

$$V^{MI} = \overline{U} + \left(1 + \frac{1}{M}\right)B$$

where $B = \frac{1}{M-1} \sum_{i=1}^{M} \left(\hat{\theta}^{(i)} - \hat{\theta}^{MI}\right)^2$ and $\overline{U} = \frac{1}{M} \sum_{i=1}^{M} \hat{U}^{(i)}$.

Therefore, large value of $M$ can reduce the values of both $\overline{U}$ and $B$, and further the total variance, increasing the reliability of the estimates. In terms of high accuracy of predictions, larger number of imputed datasets is a better choice.

However, time consumption and low statistical efficiency can be the problems of large value of $M$. When imputing large scaled dataset with a large number of missing values, large $M$ will relatively increase the running time. Moreover, the imputation procedure arrives at stable and statistically significant conclusions when $M$ reaches a certain value. Increasing $M$ beyond this value does not significantly influence the conclusions but consumes resource and time.

Overall, I still consider $M = 100$ as a better choice as the dataset in this question only have 25 observations and 100 imputed datasets can be easily handled.

```
# initialize the data frame
bmi.lambda.100 <- NULL

# repeat analysis with m = 100
for (i in seq(1, 6)){
  imp.list <- mice(nhanes, m = 100, printFlag = FALSE, seed = i)
  bmi.fit <- with(imp.list, lm(bmi ~ age + hyp + chl))
  bmi.pool <- pool(bmi.fit)
  df <- data.frame(age = bmi.pool$pooled[2, 10],
                   hyp = bmi.pool$pooled[3, 10],
                   chl = bmi.pool$pooled[4, 10],
                   row.names = paste("seed.", i, sep = ""))
  bmi.lambda.100 <- rbind(bmi.lambda.100, df)
}
bmi.lambda.100
```

```
##              age       hyp       chl
## seed.1 0.4324680 0.2915346 0.3217837
## seed.2 0.4031077 0.2825108 0.2939693
## seed.3 0.3093072 0.2425105 0.3281911
```

```
## seed.4 0.3943223 0.2565132 0.2835232
## seed.5 0.3322570 0.2893046 0.2461956
## seed.6 0.4430300 0.2860700 0.3113085
```

## Question 2

**Solution**

As required, we apply stochastic regression imputation (SRI) and bootstrap sampling to generate the data in step 2. To calculate the empirical coverage probability, according to **NOTE 1**, we define a counter to count the times when the ground truth value of $\beta_1$, i.e. 3, is covered in its 95% empirical confidence interval for each method and further obtain the probability with the corresponding frequency.

From the results below, we observe the empirical coverage probability of bootstrap imputation is larger than the probability of SRI (0.95 vs 0.88). The reason for this is that stochastic regression imputation does not incorporate the variability of function weights, which means the uncertainty of imputed values is not considered, unlike bootstrap imputation. Thus, generally the 95% confidence intervals obtained using SRI are more narrow and less likely to contain the ground truth value.

```r
# read dataset into R
load("dataex2.Rdata")

# initialize counters for empirical coverage probabilities of two methods
count.sri <- count.boot <- 0

# MICE standard procedure for each dataset
for (i in 1:100){

  # step 1 - mice()
  imp.sri <- mice(dataex2[, , i], m = 20, seed = 1,
                  printFlag = FALSE, method = "norm.nob")
  imp.boot <- mice(dataex2[, , i], m = 20, seed = 1,
                   printFlag = FALSE, method = "norm.boot")

  # step 2 - with()
  fit.sri <- with(imp.sri, lm(Y ~ X))
  fit.boot <- with(imp.boot, lm(Y ~ X))

  # step 3 - pool()
  pool.sri <- summary(pool(fit.sri), conf.int = TRUE)
  pool.boot <- summary(pool(fit.boot), conf.int = TRUE)

  # if the ground truth, 3, is covered in the 95% CI of beta_1 ...
  if (pool.sri$`2.5 %`[2] <= 3 & 3 <= pool.sri$`97.5 %`[2]){
    count.sri = count.sri + 1
  }
  if (pool.boot$`2.5 %`[2] <= 3 & 3 <= pool.boot$`97.5 %`[2]){
    count.boot = count.boot + 1
  }
}

# report the empirical coverage probability
data.frame(sri = count.sri / 100, bootstrap = count.boot / 100,
           row.names = "prob.")
```

```
##       sri bootstrap
## prob. 0.88      0.95
```

## Question 3

**Solution**

We assume a general linear regression model $\boldsymbol{y} = \boldsymbol{x}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ for the multiple imputation procedure. Here,

1. $\boldsymbol{y} = (y_1, y_2, \cdots, y_m)$: observations of response variable

2. $\boldsymbol{x} = (\boldsymbol{1}, \boldsymbol{x_1}, \boldsymbol{x_2}, \cdots, \boldsymbol{x_n})$: covariate matrix

3. $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \cdots, \beta_n)$: coefficients

4. $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2, \cdots, \epsilon_m),\ \epsilon_i \sim \mathcal{N}(0, \sigma^2), i = 1, \cdots, m$: noise

We start with the scenario mentioned in (i):

## Question 4

**Solution**

# Question 5

**(a) - Solution**

**(b) - Solution**