# Missing Data Presentation

Jiurui Tang

5/24/2018

# Introduction

# Motivating Example: Pregnancy, Infection, & Nutrition (PIN) Study

- Prospective study of risk factors for adverse birth outcomes in central NC prenatal care clinics
- We would like to study how cocaine use relates to small-for-gestational-age (SGA) births
- Covariates: maternal age, race, income, parity, and education
- Cocaine measured on questionnaire, in urine, and in hair

# Mathematical Notation

- Let $Y = (Y_{obs}, Y_{mis})$
- $r_i = 1$ if data for unit $i$ is missing, $r_i = 0$ otherwise
- $R = (r_1, ..., r_n)$
- $\theta$ denote the parameters associated with $Y$
- $\psi$ denote the parameters associated with $R$

# Types of Missing Data

# Missing Completely at Random (MCAR)

- Subjects with missing data are a complete random sample of study subjects
- e.g. hair cocaine missing for subset of women who were never asked to provide it for study logistical reasons
- $f(R|Y, \theta, \psi) = f(R|\psi)$

# Missing at Random (MAR)

- Missingness may depend on observed variables, but not on the missing values themselves
- e.g., hair samples missing more often for older women
- $f(R|Y, \theta, \psi) = f(R|Y_{obs}, \psi)$

# Missing not at Random/Nonignorable (MNAR)

- Missingness may depend on the missing values themselves and may also depend on observed values
- e.g., women who have used cocaine recently are less likely to provide biospecimens for drug testing
- $f(R|Y, \theta, \psi) = f(R|Y_{obs}, Y_{mis}, \psi)$

# Methods of Handling Missing Data

# Complete Case Anlysis

- Use only the set of observations with no missing values
- When missingness is MCAR, then the complete case estimator is unbiased
- Good approach when percentage of data missing is small ($<$ 5-10%)
- If there are many different variables with missing values, a large fraction of the observations may be dropped, resulting in inefficient use of information

# “Ad-hoc” Methods

- Creation of an indicator of missingness as a new variable
- Simple imputation with mean of observed values, or prediction from regression model
- Last value carried forward
- Hot deck imputation: replace missing value with an observed response from a “similar” subject
- Easy to implement, but have the potential to induce bias, not recommended

# Multiple Imputation

- Generate *p* possible values for each missing observation (typically 5-10 imputated datasets are created)
- Each of these datasets is analyzed using complete-data methods
- Combine the results of *p* separate analyses, this allows the uncertainty regarding the imputation to be taken into account
- Often the easiest solution for MCAR or MAR data
- One approach can be the Conditional Gaussian approach, alternatively, multiple imputation with chained equations (MICE) is used in most software

# MICE

- Also known as “fully conditional specification” or “sequential regression multiple imputation”
- Involve a variable-by-variable approach
- Imputation model is specified separately for each variable, using the other variables as predictors
- At each stage of the algorithm, an imputation is generated for the missing variable, then this imputed value is used in the imputation of the next variable
- Repeat this process for several cycles to obtain one imputed dataset
- May perform simple imputation for every missing value in the dataset in the initializing step, then set back simple imputation values to missing for the variable to be imputated at each stage

# MICE in R

We'll walk through a simple MICE example in R with NHANES dataset. It contains 25 obs and four variables: age (age groups: 20-39, 40-59, 60+), bmi (body mass index), hyp (hypertension status) and chl (cholesterol level).

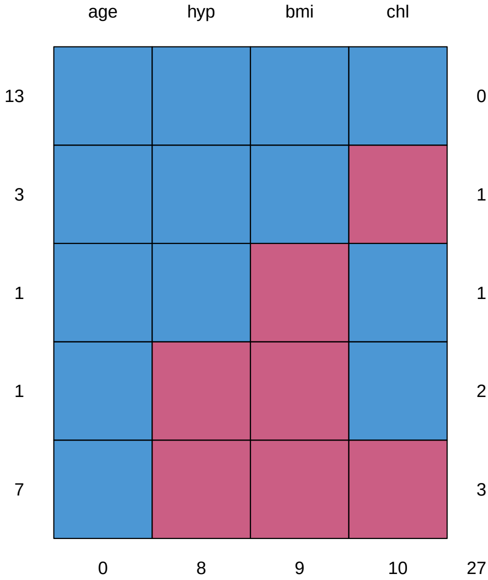
```r
# required libraries
library(mice)
library(VIM)
library(lattice)
library(NHANES)

# load data
data(nhanes)
str(nhanes)
```

```
## 'data.frame':    25 obs. of  4 variables:
##  $ age: num  1 2 1 3 1 3 1 1 2 2 ...
##  $ bmi: num  NA 22.7 NA NA 20.4 NA 22.5 30.1 22 NA ...
##  $ hyp: num  NA 1 1 NA 1 NA 1 1 1 NA ...
##  $ chl: num  NA 187 187 NA 113 184 118 187 238 NA ...
```

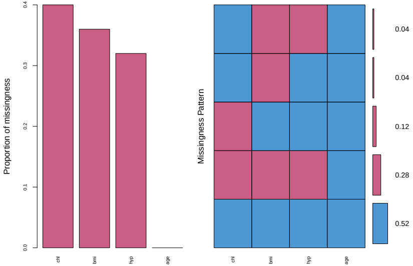
## Look at the pattern of missing data (in mice package)

```
md.pattern(nhanes)
```



Visualize missingness pattern in VIM package: This plot gives the frequencies for different combination of variables missing. Blue refers to observed data and red to the missing data.

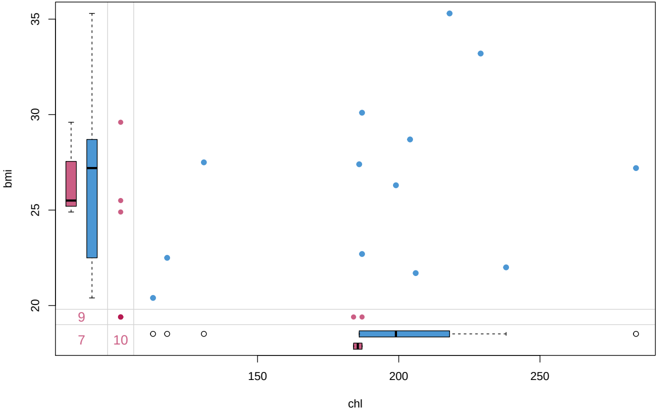
```
nhanes_aggr <- aggr(nhanes,col=mdc(1:2),numbers=TRUE,sortVa
                  labels=names(nhanes),cex.axis=.7, gap=3
                  ylab=c("Proportion of missingness","Mis
```



```
##
## Variables sorted by number of missings:
## Variable Count
##      chl  0.40
```

The margin plot of the pairs using VIM package

```
marginplot(nhanes[, c("chl", "bmi")], col = mdc(1:2), cex.
```



Blue box plots summarise the distribution of observed data given the other variable is observed, and red box plots summarise the distribution of observed data given the other variable is missing.

# Imputation & Analyzing imputated datasets

```r
imp <- mice(nhanes, m=5, printFlag=FALSE, maxit = 40, seed=
# The output imp contains m=5 completed datasets

fit.mi <- with(data=imp, exp = lm(chl ~ age + bmi))
#Each dataset can be analysed using function with()

combFit <- pool(fit.mi) # Combine all the results of the 5

round(summary(combFit),2)
```

```
##             estimate std.error statistic    df p.value
## (Intercept)    11.30     59.66      0.19 10.61    0.85
## age2           43.00     17.94      2.40 13.47    0.03
## age3           54.51     19.99      2.73 10.93    0.02
## bmi             5.82      2.03      2.87 11.52    0.01
```

Increase the number of imputations to $m = 20$ to see whether we get similar results

```
imp20 <- mice(nhanes, m=20, printFlag=FALSE, maxit = 30, se
fit.mi20 <- with(data=imp20, exp = lm(chl ~ age + bmi))
combFit <- pool(fit.mi20)
round(summary(combFit),2)
```

```
##             estimate std.error statistic    df p.value
## (Intercept)    12.21     58.37      0.21 14.34    0.84
## age2           47.56     18.70      2.54 13.34    0.02
## age3           64.69     22.21      2.91 10.62    0.01
## bmi             5.69      1.99      2.86 14.93    0.01
```

The results are not much changed. MI works for as low as $m = 5$ for this example.

## Examine imputations created by MICE

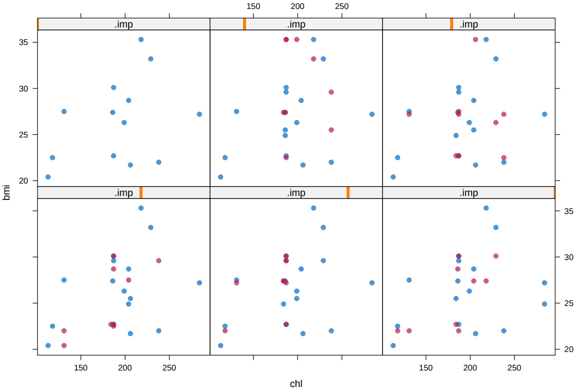
```r
# Extract the imputations for bmi
imp$imp$bmi
```

```
##       1    2    3    4    5
## 1  35.3 27.2 20.4 29.6 22.0
## 3  35.3 27.2 30.1 30.1 22.0
## 4  27.4 22.7 27.5 27.4 27.4
## 6  27.4 22.7 22.7 27.4 22.7
## 10 22.5 27.5 22.7 22.7 27.4
## 11 35.3 35.3 22.0 22.0 22.0
## 12 25.5 26.3 22.5 27.2 28.7
## 16 33.2 27.2 29.6 27.2 30.1
## 21 29.6 22.5 28.7 29.6 30.1
```

```r
# Extract the second complete dataset
imp_2 <- complete(imp, 2)
head(imp_2)
```

## Checking imputations visually

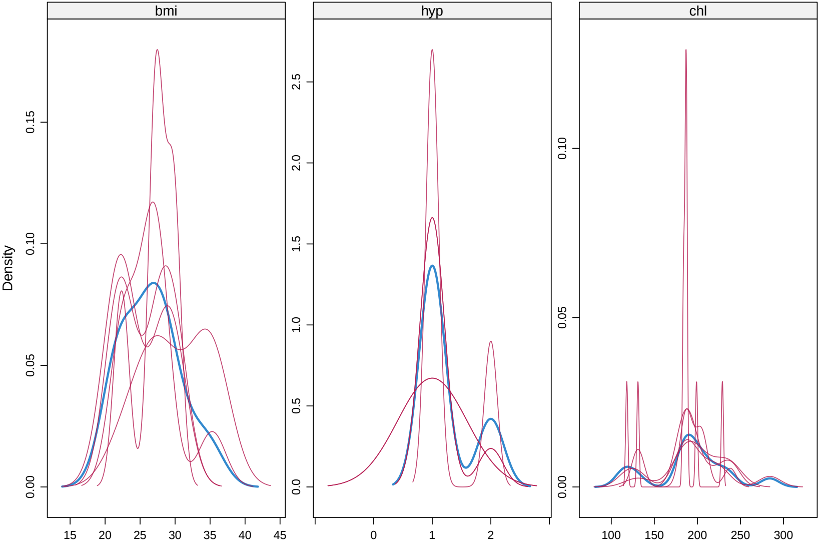
```
# scatterplot (of chl and bmi) for each imputed dataset
xyplot(imp, bmi ~ chl | .imp, pch = 20, cex = 1.4)
```



We expect the red points (imputed data) have almost the same shape as blue points. Blue points are constant across imputed datasets, but red points differ from each other, which represents our uncertainty about the true values of missing data.

# Checking imputations visually

```r
# To detect interesting differences in distribution between
densityplot(imp)
```



What if we only know BMI as a binary indicator?

```
# Create an indicator variable of BMI, coded as 1 if BMI >=
# NA if BMI is missing
nhanes$obese <- ifelse(is.na(nhanes$bmi),NA,ifelse(nhanes$b
nhanes_new <- nhanes[,-which(names(nhanes) %in% "bmi")]
nhanes_new$obese <- as.factor(nhanes_new$obese)

# MICE
imp_new <- mice(nhanes_new, m=5, printFlag=FALSE, maxit = 4
fit.mi_new <- with(data=imp_new, exp = lm(chl ~ age + obese
combFit_new <- pool(fit.mi_new)
round(summary(combFit_new),2)
```

```
##             estimate std.error statistic   df p.value
## (Intercept)   147.84     20.90      7.07 6.48    0.00
## age2           33.66     23.22      1.45 7.38    0.18
## age3           44.49     26.17      1.70 6.01    0.13
## obese1         32.84     19.63      1.67 8.44    0.13
```

# How mice() actually imputes values

```
# See the univariate imputation model for each incomplete
# for your data as default
imp_new$meth
```

```
##      age      hyp      chl    obese
##       ""    "pmm"    "pmm" "logreg"
```

```
# Possible imputation models provided by mice()
methods(mice)
```

```
##  [1] mice.impute.2l.bin       mice.impute.2l.lmer
##  [3] mice.impute.2l.norm      mice.impute.2l.pan
##  [5] mice.impute.2lonly.mean  mice.impute.2lonly.norm
##  [7] mice.impute.2lonly.pmm   mice.impute.cart
##  [9] mice.impute.jomoImpute   mice.impute.lda
## [11] mice.impute.logreg       mice.impute.logreg.boot
## [13] mice.impute.mean         mice.impute.midastouch
## [15] mice.impute.norm         mice.impute.norm.boot
```

Change the default imputation methods

```
# Since hyp is binary, we want to change the default imputa
meth=imp_new$meth
meth = c("", "logreg", "pmm","logreg")
nhanes_new$hyp <- as.factor(nhanes_new$hyp)
imp_new2 <- mice(nhanes_new, m=5, printFlag=FALSE, maxit =
imp_new2$method
```

```
##      age      hyp      chl    obese
##       "" "logreg"    "pmm" "logreg"
```

```
# Visiting sequence
imp_new2$vis
```

```
## [1] "age"   "hyp"   "chl"   "obese"
```

```
# Predictor matrix: Which variables does mice() use as pre
# for imputation of each incomplete variable?
imp_new2$pred
```

```
##       age hyp chl obese
## age     0   1   1     1
## hyp     1   0   1     1
## chl     1   1   0     1
## obese   1   1   1     0
```

```
# We can specify relevant predictors as follows:
# Suppose that hyp is considered irrelevant as a predictor
pred=imp_new2$pred;
pred[, "hyp"] = 0
pred
```

```
##       age hyp chl obese
## age     0   0   1     1
## hyp     1   0   1     1
## chl     1   0   0     1
```