Introduction to MICE and multilevel imputation

Utrecht, May 15, 2013

Stef van Buuren[1,2]

[1]Netherlands Organisation for Applied Scientific Rsearch TNO, Leiden

[2]Methodology and Statistics, FSBS, Utrecht University

May 15, 2013, Utrecht

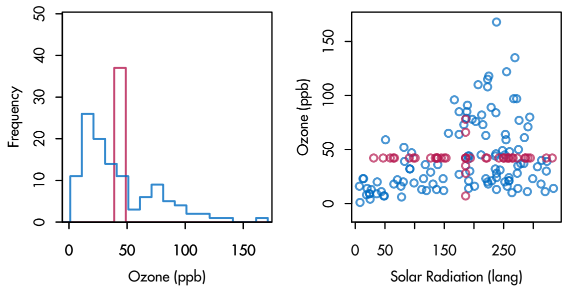
Software and examples

- R Download from `http://cran.r-project.org`
- R package: `mice` 2.17
- Optional: Install RStudio
- Example code: `doc` directory of the `mice` package
- Example code: `http://www.multiple-imputation.com/fimd.html`

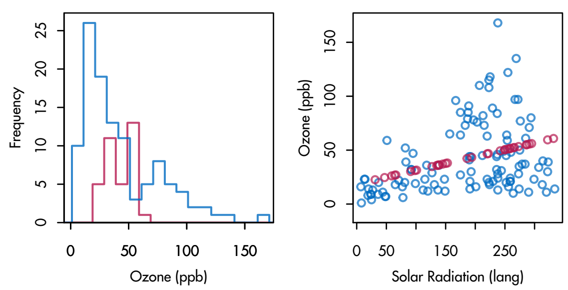
How to load the built-in code?

```
> doc <- file.path(path.package("mice"),"doc")
> dir(doc)

 [1] "JSScode.R"  "fimd1.r"    "fimd2.r"    "fimd3.r"
 [5] "fimd4.r"    "fimd5.r"    "fimd6.r"    "fimd7.r"
 [9] "fimd8.r"    "fimd9.r"    "index.html"

> edit(file = file.path(doc,"fimd1.r"))
```

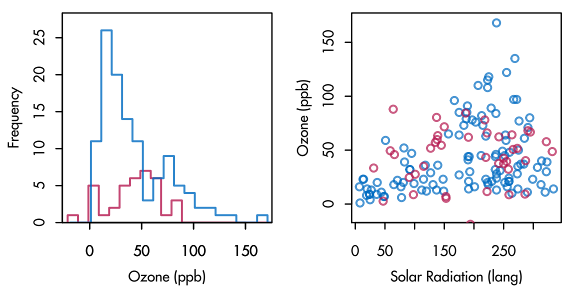

Mean imputation



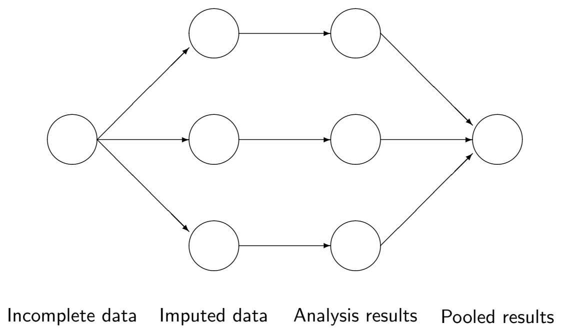
Regression imputation



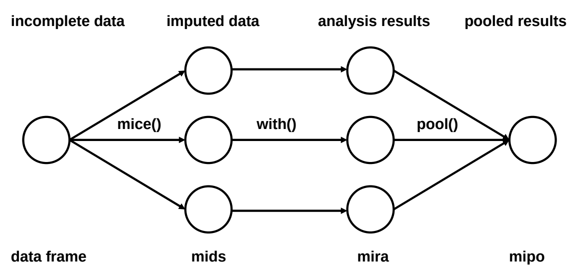
Stochastic regression imputation



Main steps used in multiple imputation



Steps in mice



Calculation (1)

```
> library(mice)
> options(digits = 3, width = 65)
> imp <- mice(nhanes, print = FALSE, m = 10, seed = 24415)
> fit <- with(imp, lm(bmi ~ age))
> est <- pool(fit)
> attributes(est)

$names
 [1] "call"   "call1"  "call2"  "nmis"   "m"      "qhat"
 [7] "u"      "qbar"   "ubar"   "b"      "t"      "r"
[13] "dfcom"  "df"     "fmi"    "lambda"

$class
[1] "mipo"   "mira"   "matrix"
```


Calculation (2)

```
> est$qhat

   (Intercept)   age
1         32.4 -2.77
2         32.6 -3.17
3         31.9 -2.97
4         30.8 -2.21
5         30.5 -2.21
6         31.0 -2.21
7         30.5 -2.48
8         32.3 -2.55
9         28.9 -1.42
10        28.1 -1.13
```


Calculation (3)

```
> summary(est)

              est   se     t    df Pr(>|t|) lo 95  hi 95 nmis
(Intercept) 30.89 2.43 12.70  9.93 1.83e-07 25.47 36.311   NA
age         -2.31 1.17 -1.98 12.12 7.08e-02 -4.85  0.229    0
              fmi lambda
(Intercept) 0.509  0.419
age         0.419  0.331

> est$qbar

(Intercept)         age
      30.89       -2.31

> est$lambda

(Intercept)         age
      0.419       0.331
```


Inspect the data

```
> library("mice")
> head(nhanes)

  age  bmi hyp chl
1   1   NA  NA  NA
2   2 22.7   1 187
3   1   NA   1 187
4   3   NA  NA  NA
5   1 20.4   1 113
6   3   NA  NA 184
```

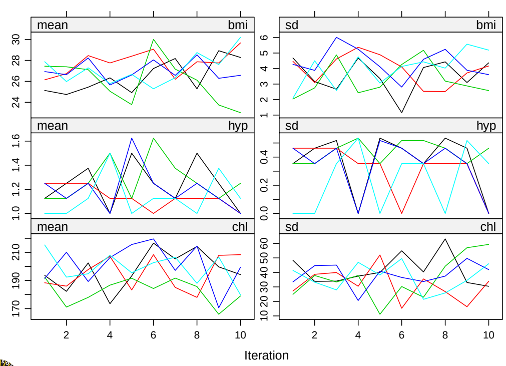

Inspect missing data pattern

```
> md.pattern(nhanes)

   age hyp bmi chl
13   1   1   1   1  0
 1   1   1   0   1  1
 3   1   1   1   0  1
 1   1   0   0   1  2
 7   1   0   0   0  3
     0   8   9  10 27
```


Multply impute the data

```
> imp <- mice(nhanes, print = FALSE, maxit = 10,
+     seed = 24415)
> plot(imp)
```


Inspect the trace lines for convergence



Fit the complete-data model

```
> fit <- with(imp, lm(bmi ~ age))
> est <- pool(fit)
> summary(est)

             est   se     t   df Pr(>|t|) lo 95 hi 95 nmis
(Intercept) 31.0 3.47  8.92 3.38  0.00182 20.59 41.33   NA
age         -2.3 1.40 -1.64 5.75  0.15369 -5.76  1.16    0
              fmi lambda
(Intercept) 0.806  0.718
age         0.643  0.537
```

What type of results are there

```
> attributes(est)
$names
 [1] "call"   "call1"  "call2"  "nmis"   "m"      "qhat"
 [7] "u"      "qbar"   "ubar"   "b"      "t"      "r"
[13] "dfcom"  "df"     "fmi"    "lambda"

$class
[1] "mipo"   "mira"   "matrix"
```


What is $\hat{Q}$?

```
> est$qhat
  (Intercept)    age
1        32.0 -2.737
2        32.7 -2.851
3        26.7 -0.821
4        30.0 -1.944
5        33.4 -3.132
```

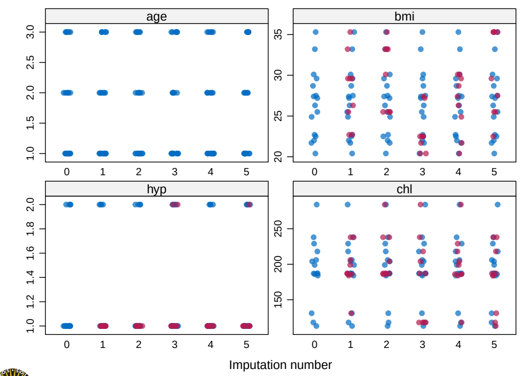

What is $\bar{Q}$?

```
> est$qbar
(Intercept)         age
       31.0        -2.3
> est$fmi
(Intercept)         age
      0.806       0.643
```

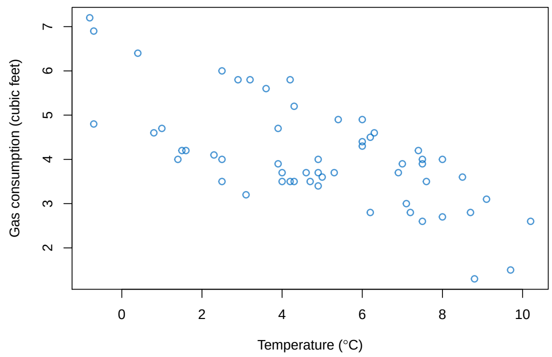

Stripplot of observed and imputed data

```
> stripplot(imp, pch = 20, cex = 1.2)
```

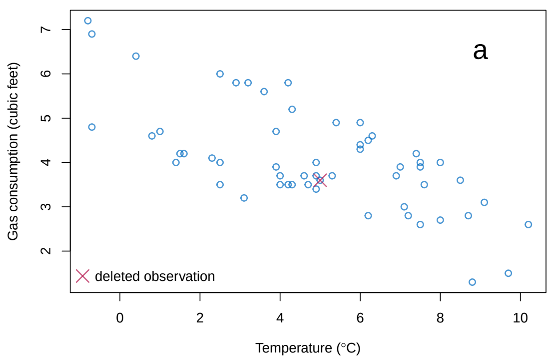

Stripplot of observed and imputed data



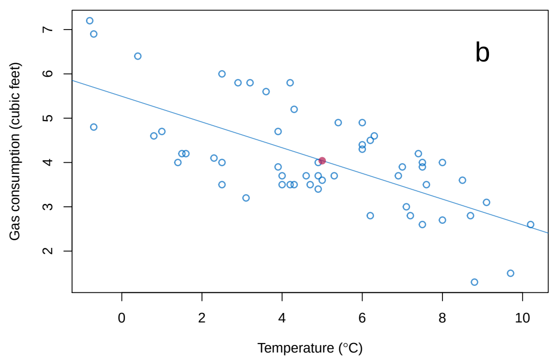
Relation between temperature and gas consumption



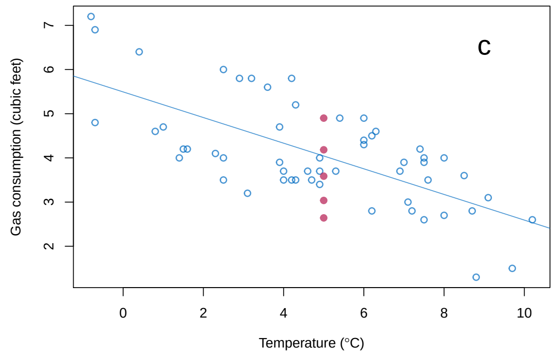
We delete gas consumption of observation 47



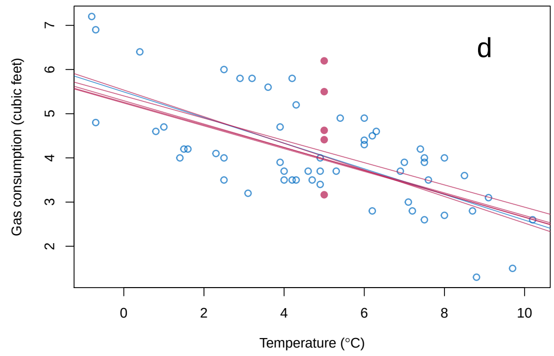
Predict imputed value from regression line



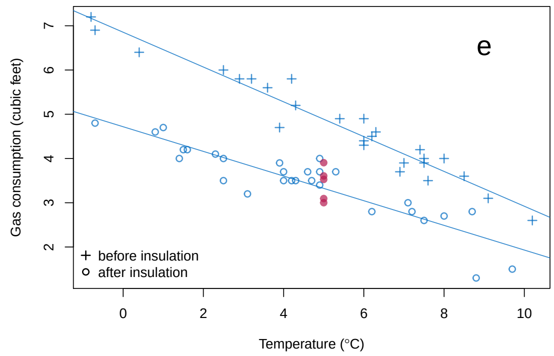
Predicted value + noise



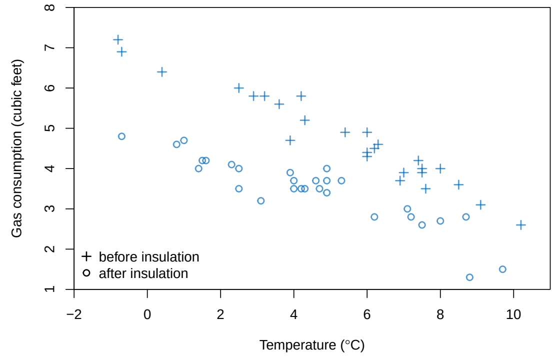
Predicted value + noise + parameter uncertainty



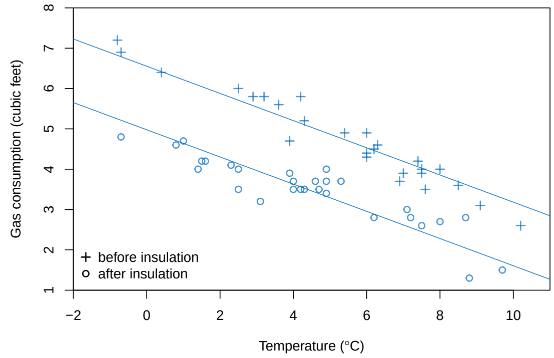
Imputation based on two predictors



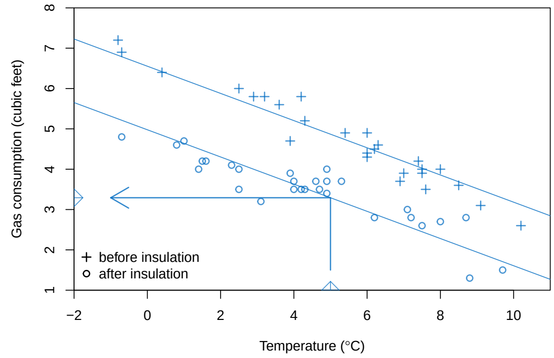
Predictive mean matching: Y given X



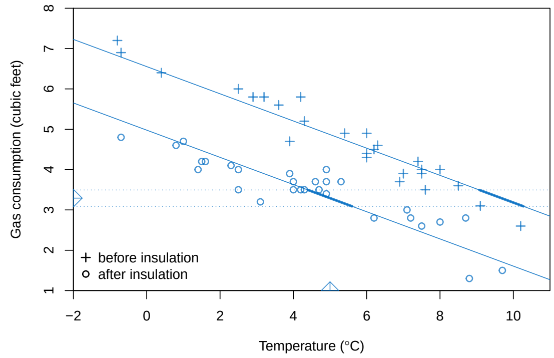
Add two regression lines



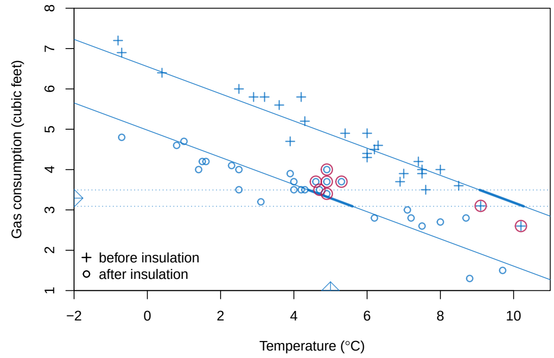
Predicted given 5° C, 'after insulation'



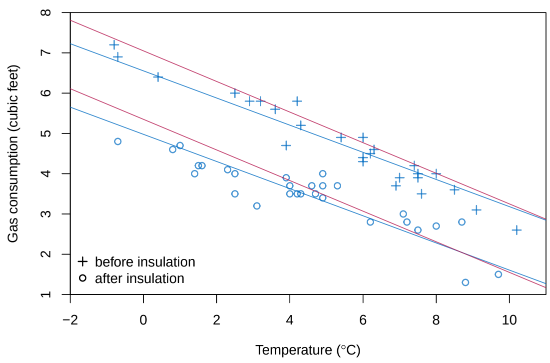
Define a matching range $\hat{y} \pm \delta$



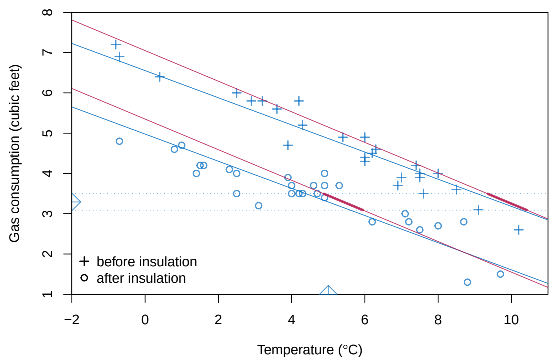
Select potential donors



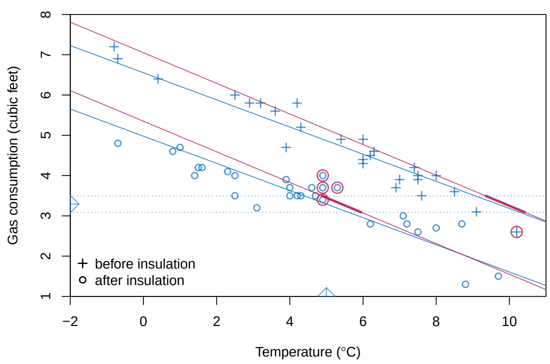
Bayesian PMM: Draw a line



Define a matching range $\hat{y} \pm \delta$



Select potential donors

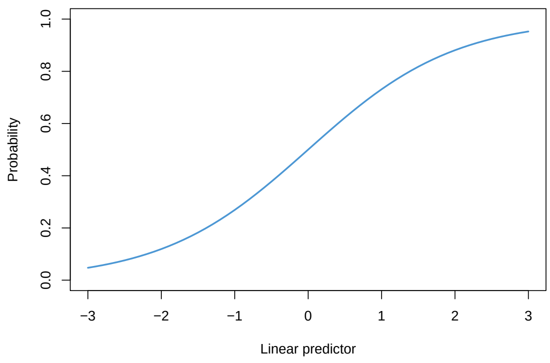


Imputation of a binary variable

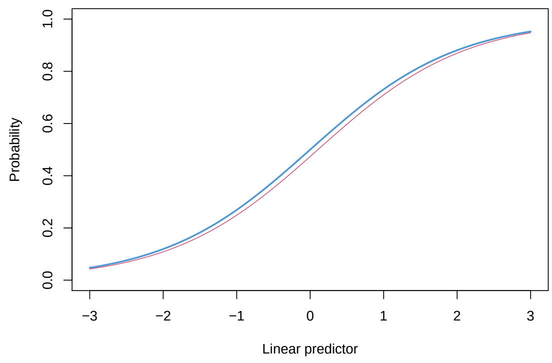
- *logistic regression*

$$\Pr(y_i = 1|X_i, \beta) = \frac{\exp(X_i\beta)}{1 + \exp(X_i\beta)}. \qquad (1)$$

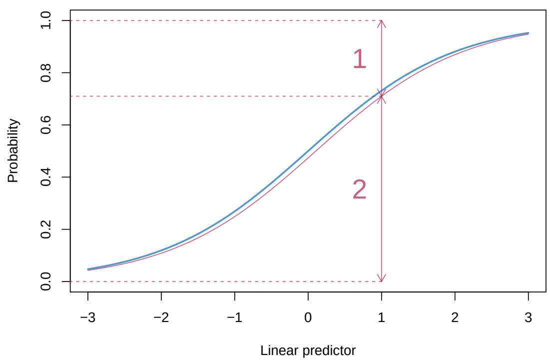
Fit logistic model



Draw parameter estimate



Read off the probability

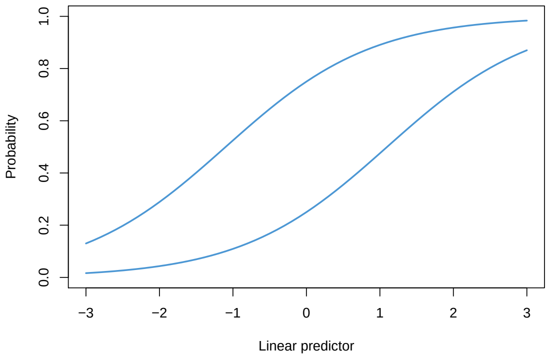


Impute ordered categorical variable

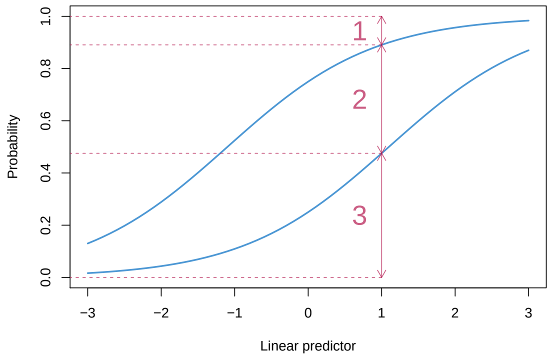
- K ordered categories $k = 1, \ldots, K$
- *ordered logit model*, or
- *proportional odds model*

$$\Pr(y_i = k|X_i, \beta) = \frac{\exp(\tau_k + X_i\beta)}{\sum_{k=1}^{K} \exp(\tau_k + X_i\beta)} \qquad (2)$$

Fit ordered logit model



Read off the probability



Other types of variables

- Count data
- Semi-continuous data
- Censored data
- Truncated data
- Rounded data

Univariate imputation in `mice` - flat data

Method	Description	Scale type
pmm	Predictive mean matching	any*
norm	Bayesian linear regression	numeric
norm.nob	Linear regression, non-Bayesian	numeric
norm.boot	Linear regression, bootstrap	numeric
norm.predict	Best linear prediction	numeric
mean	Unconditional mean imputation	numeric
ri	drawn indicator for MNAR data	numeric
sample	Simple random sample	any
cart	Classification and regression trees	any

[1]

[1] Van Buuren (2012). Chapter 3.

Imputation in `mice` - flat data - categorical

Method	Description	Scale type
logreg	Logistic regression	factor, 2 levels*
logreg.boot	Logistic regression, bootstrap	factor, 2 levels
polyreg	Multinomial logit model	factor, > 2 levels*
polr	Ordered logit model	ordered, > 2 levels*
lda	Linear discriminant analysis	factor
quadratic	Quadratic relations	numeric

[2]

[2] Van Buuren (2012). Chapter 3.

Multivariate Imputation by Chained Equations (MICE)

- MICE algorithm
- Specify imputation model for each incomplete column
- Fill in starting imputations
- And iterate
- Model: Fully Conditional Specification (FCS)

Fully Conditional Specification : Con's

- Theoretical properties only known in special cases
- Cannot use computational shortcuts, like sweep-operator
- Care needed in building and checking the model

Fully Conditional Specification : Pro's

- Extremely flexible
- Easy to communicate
- Subset selection of predictors
- Splits missing data and complete data problem
- Modular, can preserve valuable work
- Appears to work quite well in practice

Fully Conditional Specification (FCS): Software

R	mice, transcan, mi
SPSS V17	procedure multiple imputation
SAS	IVEware, SAS 9.3
STATA	ice command, multiple imputation command
Stand-alone	Solas, Mplus

Multilevel imputation

Univariate imputation in mice two-level data

Method	Description	Scale type
2l.norm	Two-level linear model, heteroskedastic	numeric
2l.pan	Two-level linear model, homoskedastic	numeric
2lonly.mean	2nd level, class mean	numeric
2lonly.norm	2nd level, normal model	numeric
2lonly.pmm	2nd level, predictive mean matching	any
2l.logit	2nd level, logistic regression	binary*
2lmixed.logit	2nd level, mixed level predictors	binary*

* in development, by Ross Boylan

Linear mixed-effects model

$$y_j = X_j\beta + Z_j u_j + \epsilon_j$$

y_j $(n_j \times 1)$ outcomes in class j

X_j $(n_j \times p)$ design matrix, β fixed effects

Z_j $(n_j \times q)$ design matrix, u_j random effects $u_j \sim N(0, \Omega)$

ϵ_j $(n_j \times 1)$ residuals, $\epsilon_j \sim N(0, \sigma_j^2 I(n_j))$

$\sigma_j^2 = \sigma^2$, homogeneity

$\epsilon_j \perp u_j$, independence

Linear two-level model

$$\begin{aligned} y_j &= Z_j\beta_j + \epsilon_j \\ \beta_j &= W_j\beta + u_j \end{aligned}$$

y_j $(n_j \times 1)$ outcomes in class j

Z_j $(n_j \times q)$ level-1 predictors, β_j varying coefficients

W_j $(q \times p)$ level-2 predictors, β fixed effect

u_j $(q \times 1)$ random effects, $u_j \sim N(0, \Omega)$

ϵ_j $(n_j \times 1)$ residuals, $\epsilon_j \sim N(0, \sigma_j^2 I(n_j))$

$\sigma_j^2 = \sigma^2$, homogeneity

$\epsilon_j \perp u_j$, independence

Where are the missing data?

$$\begin{aligned} y_j &= Z_j\beta_j + \epsilon_j \\ \beta_j &= W_j\beta + u_j \end{aligned}$$

y_j standard problem in multilevel analysis

Z_j non-standard problem at first level

W_j non-standard problem at 2nd level

j missing class variable, non-standard

Where are the missing data? Missing y_j

$$\begin{aligned} y_j &= Z_j\beta_j + \epsilon_j \\ \beta_j &= W_j\beta + u_j \end{aligned}$$

- Laird and Ware (1982) EM-algorithm
- Jennrich and Schluchter (1986), speed-up
- Verbeke and Molenberghs (2000), many applications
- Daniels and Hogan (2008), MNAR for longitudinal case

Where are the missing data? Missing y_j

$$\begin{aligned} y_j &= Z_j\beta_j + \epsilon_j \\ \beta_j &= W_j\beta + u_j \end{aligned}$$

1. The missing data are confined to y_j,
2. The MAR assumption is plausible,
3. Any factors in the MAR mechanism are included into the multilevel model,
4. The multilevel model for the complete data is correctly specific.

[3]

[3]See: van Buuren S (2011) Multiple imputation of multilevel data. In Hox J, J. & Roberts J, K. (Eds), *The Handbook of Advanced Multilevel Analysis*. Routledge, Milton Park, UK, pp. 173-196.

Where are the missing data? Missing Z_j

$$\begin{aligned} y_j &= Z_j\beta_j + \epsilon_j \\ \beta_j &= W_j\beta + u_j \end{aligned}$$

- Take complete level-1 cases: `MLwiN`, `HLM`, `PROC MIXED`, `nlme`, `arm`, `MIXED`, and so on
- FIML: Full Information Maximum Likelihood, `Mplus`

Where are the missing data? Missing W_j

$$\begin{aligned} y_j &= Z_j\beta_j + \epsilon_j \\ \beta_j &= W_j\beta + u_j \end{aligned}$$

- Take complete level-2 classes (standard, but very wasteful)

Where are the missing data? Missing j

$$\begin{aligned} y_j &= Z_j\beta_j + \epsilon_j \\ \beta_j &= W_j\beta + u_j \end{aligned}$$

- Missing class variable j: usually case deletion.
- Multiple imputation of the class variable appears straightforward, but has never been tried.

Joint modeling approach

$$\begin{aligned} y_j &= Z_j\beta_j + \epsilon_j \\ \beta_j &= W_j\beta + u_j \end{aligned}$$

- Put all incomplete level-1 variables in y_j
- y_j, β_j, β, u_j and ϵ_j become matrices
- Assume $\epsilon_j \sim N(0, \Omega_{1j})$
- Assume $u_j \sim N(0, \Omega_2)$
- Standard case: Assume $\Omega_{1j} = \Omega_1$ for all j
- Generalizes Schafer's joint models to multilevel data

Joint modeling approach: Software

- `PAN`, `mlmmm` (Schafer & Yucel, 2002; Yucel 2008; Yucel 2011)
- `REALCOM-IMPUTE` (Carpenter, Goldstein & Kenward, 2011; Carpenter & Kenward, 2013, Ch. 9)
- Extensions to categorical data have been proposed
- Imputes both at level-1 and level-2
- Requires $\epsilon_j \sim N(0, \Omega_1)$ and $u_j \sim N(0, \Omega_2)$

Fully Conditional Specification

$$\begin{aligned} y_j &= Z_j\beta_j + \epsilon_j \\ \beta_j &= W_j\beta + u_j \end{aligned}$$

- Impute variable-by-variable using univariate multilevel model
- Allow for σ_j^2, heterogenous error variance per class
- Iteratively draw β, u_j , Ω and σ_j^2 by Markov Chain Monte Carlo

$$\dot\beta \sim p(\beta|u_j, \sigma^2) \quad (3)$$
$$\dot u_j \sim p(u_j|\beta, \Omega, \sigma^2) \quad (4)$$
$$\dot\Omega \sim p(\Omega|u_j) \quad (5)$$
$$\dot\sigma_j^2 \sim p(\sigma^2|\beta, u_j) \quad (6)$$

- On converge, draw $\dot y_j$ given the current values of the parameters.

Fully Conditional Specification: Does it work?

$$\begin{aligned} y_j &= Z_j\beta_j + \epsilon_j \\ \beta_j &= W_j\beta + u_j \end{aligned}$$

- Ian White's note (personal communication):
- y_1 and y_2 are two level-1 response variables
  1. $p(y_1|y_2, x)$ depends on the cluster mean of y_2, $\bar y_{2,j}$
  2. regression weight for $\bar y_{2,j}$ depends on n_j
  3. spread of $p(y_1|y_2, x)$ depends on n_j
- Thus, ignoring the multilevel structure for level-1 variables can create invalid imputations
- On the other hand, almost nothing is known about the impact of violations

Fully Conditional Specification: Simulations

- Zhao and Yucel (2009) compared two methods:
  - JM: Multivariate linear mixed-effect model (PAN)
  - FCS: Univariate generalized linear mixed-effect model (Gibbs sampler)
- Their conclusions
  - For continuous variables, FCS performed "reasonably well"
  - For binary variables, FCS outperforms PAN in almost all scenarios
  - Moderate missingness rates do not impact performance
  - Role of the priors negligible in most settings.

[4]

[4] Zhao, E., & Yucel, R. M. (2009). *Performance of Sequential Imputation Method in Multilevel Applications*. ASA 2009 Proceedings of the Survey Research Methods Section. Alexandria, VA.

Fully Conditional Specification: Simulations

- Van Buuren (2011) compared four methods:
  - CC: Complete Case Analysis (CC)
  - FF: Multiple Imputation Flat File
  - SC: Multiple imputation treating class as fixed factor
  - ML: Multiple multilevel imputation (`2l.norm`)
- Conclusions
  - CC is bad strategy with missing data in Z_j
  - FF biases the ICC downwards, SC biases upwards, ML is about right
  - Smaller classes makes the problem more difficult
  - ML is a considerable improvement over CC, and better than FF and SC. However, coverage often fails to achieve nomimal level

5

[5] See: van Buuren S (2011) Multiple imputation of multilevel data. In Hox J, J. & Roberts J, K. (Eds), *The Handbook of Advanced Multilevel Analysis*. Routledge, Milton Park, UK, pp. 173-196.

Two-level imputation methods: Software I

- `mice.impute.2l.norm()` in `mice`.
  - Implemented the MCMC method
  - Assumes $\sigma_j^2 \neq \sigma^2$
  - Best method, but not so fast
- `mice.impute.2l.pan()` in `mice`.
  - Uses Schafer's pan method
  - Faster than `mice.impute.2l.norm()`
  - Assumes homogeneity $\sigma_j^2 = \sigma^2$, and thus less flexible
- Both impute first level variables, clustered in classes
- At present, no dedicated multilevel methods for discrete variables

Two-level imputation methods: Software II

- `mice.impute.2lonly.mean()`
  - Fills in the class mean
  - Use only for repair
- `mice.impute.2lonly.norm()`
  - Numerical data
  - Draw value from normal distribution
- `mice.impute.2lonly.pmm()`
  - Any data
  - Draw value by predictive mean matching
- All impute 2nd level variables
- All 'string out' the imputed value to all members in the class

Calling `mice.impute.2l.norm()` from `mice()`

```
> library("mice")
> popmis[1:3,]

  pupil school popular sex texp const teachpop
1     1      1      NA   1   24     1        7
2     2      1      NA   0   24     1        7
3     3      1       7   1   24     1        6

> ini <- mice(popmis, maxit=0)
> pred <- ini$pred
> pred["popular",] <- c(0, -2, 0, 2, 2, 2, 0)
> imp <- mice(popmis, meth = c("","","2l.norm","","","",""),
+             pred = pred, maxit=1, m = 2, seed = 71152)

 iter imp variable
  1   1  popular
  1   2  popular
```


Predictive mean matching with class variable

- A practical alternative: `mice.impute.pmm()`
  - Include class variable as dummy
  - Use predictive mean matching for flat files
  - Check ICC before and after imputation
  - Can also be used for discrete data
  - Has 'worked well' in some cases (i.e. similar ICC's)
  - Little is yet known about statistical properties

How bad is ignoring the multilevel structure?

Table: Intra-class correlation under flat file imputation methods.

vars orig	truth	observed	norm	normclass	pmm
popular	0.363	0.340	0.276	0.360	0.362
popteach	0.341	0.314	0.253	0.339	0.346
texp	1.000	1.000	0.435	0.999	1.000

6

[6] Thanks to Gerko Vink

Conclusions

- Can we apply multiple imputation to multilevel data?
- Depends on where the missing data are: y_j, Z_j, W_j, j
- Joint modeling: `REALCOM-IMPUTE`, `PAN`
- Fully conditional specification: `mice`
- Not yet covered: categorical level-1 variables, pooling of random effect, White's theoretical problem, coverage not always optimal
- Current software slow, and some pecularities
- If possible, make the 'wide' data matrix
- Flat predictive mean matching emerged as a remarkable alternative

Further reading

1. Van Buuren, S. and Groothuis-Oudshoorn, C.G.M. (2011). `mice`: Multivariate Imputation by Chained Equations in `R`. Journal of Statistical Software, 45(3), 1–67.
2. Van Buuren, S. (2012). Flexible Imputation of Missing Data. Chapman & Hall/CRC, Boca Raton, FL.

Flexible Imputation of Missing Data (FIMD)