? Methods

Introduction to missing data methods

Recai M. Yucel
Biostatistics and Epidemiology
School of Public Health
SUNY-Albany

Outline

- PART I –Introduction
  1. Missing-data problem
  2. Terminology
     - Types of missingness
     - Missingness mechanism
     - Examples
  3. Older methods
  4. Maximum likelihood [a]
  5. Multiple imputation

[a] will be covered very briefly

- PART II –Tools
  1. Methods for continuous incomplete data
     - NORM,PROC MI and IveWare
  2. Methods for categorical incomplete data
     - CAT and IveWare
  3. Methods for mixed incomplete data
     - Mix and IveWare
  4. Methods for multilevel incomplete data
     - Pan, MLWIN

1. The missing-data problem

THE PROBLEM

Most statistical analysis and estimation procedures were not designed to handle missing values

- Even small amounts of missing data cause great difficulty
- The missing-data aspect is a nuisance, not of primary interest
- Principled statistical solutions are computationally messy
- Ad hoc or unprincipled missing-data methods may do more harm than good (inefficiency, misleading uncertainty measures)

The goal

To make statistically valid inferences about population parameters from an incomplete dataset

- Not to estimate, predict, or recover missing values themselves
- Good to understand reasons for/ causes for missingness
- Good to avoid modeling the missing-data mechanism if possible
- Untestable assumptions are inevitable
- Sensitivity analyses are now standard part of any missing data procedure

History

Missing-data theory and practice

- **1**970's - ad-hoc procedures: case deletion, single imputation
- **1**980's - likelihood-based methods based on EM algorithm (Dempster, Laird and Rubin, 1977)
- **1**990's - multiple imputation; Bayes procedures using MCMC (Rubin, 1987)
- **2**000 + MI in complex problems (survey, genetic association studies), software development, diagnostics

c.1980 an important shift took place. Before that, missing data were seen as something to be *gotten rid of*. After that, missing data were seen as s source of uncertainty to be *averaged over*.

Key references

TECHNICAL:

- Dempster, Laird and Rubin (1977) article on EM algorithm
- Little and Rubin (1987, 2002) book on missing data: review of ad hoc methods, description of EM and likelihood-based methods
- Rubin (1987) book on multiple imputation
- Schafer (1997) book on MCMC and multiple imputation for missing-data problems
- Molenberghs and Kenward (2007) book on missing data in clinical studies

Key references (ctd.)

More subject-oriented

- Carpenter, J., Pocock, S. and Lamm, C. J. (2002) with special focus on missing data in clinical trials
- Ranhunathan (2004); Schafer and Graham (2002) : excellent summary on fundamentals of missing data
- Allison (2000, 2001)
- Genetic studies: Foulkes, Yucel and Li (2008); Fridley et al. (2009) (BMC); Servin and Stevens (2007) (PLoS Genetics)
- More comprehensive list and other resources are available at

  `www.missingdata.org.uk`

  maintained by James Carpenter

2. Terminology

What is a missing value?

Consider whether there is a well-defined "true" value underlying the missing-value code. For example:

- In a survey, a subject refuses to answer the income question
- In a prevention study, subject skips all substance-use (measurement error or unwillingness?)
- In a clinical trial, a patient drops out of a study b/c treatment doesn't work
- In genetic association studies, genotype data might be regarded missing

Considering a nonresponse to be a qualitatively different category ??? (Little and Rubin, 2002)

2. Terminology

Types of nonresponse

- **Unit nonresponse:** No data could be collected from the sampled unit (e.g. not at home, refused to participate, etc.). Traditionally handled by *reweighting*.
- **Item nonresponse:** Partial data collected for the unit, but some items missing (e.g. skipped the income question, unobservable genotype). Traditionally handled by imputation or weighting in simple settings.

In complicated datasets (e.g. panel studies), there are some intermediate situations:

- **"Wave" nonresponse:** subject is missing for one or more waves
- **Attrition or dropout** Subject leaves study and does not return

Patterns of missingness

1. Univariate problems: Only one variable subject to nonresponse

Units	variables X_1	X_2	. . .	X_p	Y
1	0	0	. . .	0	0
2	0	0	. . .	0	0
.	0	0	. . .	0	0
.	0	0	. . .	0	0
.	.....				
.	0	0	. . .	0	?
.	0	0	. . .	0	?
.	0	0	. . .	0	?
n	0	0	. . .	0	?

Patterns of missingness

2. Monotone patterns: If Y_j is missing, then $Y_{j+1}, \ldots, Y_p$ are missing as well

	variables						
Units	Y_1	Y_2	Y_3	.	.	.	Y_p
1	O	O	O	.	.	.	O
2	O	O	O	.	.	.	O
.	O	O	O	.	.	.	O
.	O	O	O	.	.	.	?
.	.....						
.	O	O	O	?	?	.	?
.	O	O	?	?	?	.	?
.	O	?	?	?	?	.	?
n	O	?	?	?	?	.	?

Patterns of missingness

3. Arbitrary patterns: Any set of variables may be missing for any unit

	variables				
Units	Y_1	Y_2	Y_3	. . .	Y_p
1	O	?	O	. . .	O
2	?	?	O	. . .	?
.	O	O	?	. . .	O
.	?	O	O	. . .	?
.	.....				
.	?	O	O	. . .	?
.	?	O	?	. . .	O
.	O	O	?	. . .	?
n	O	?	O	. . .	?

Missingness mechanisms

View response as a random process (Rubin, 1977). Not because we want to model it, but because we want to clarify the conditions under which we DO NOT have to model it!!!

$$
\begin{aligned}
Y &= \text{complete data} \\
&= (Y_{obs}, Y_{mis}) \\
R &= \text{indicators of response} \\
X &= \text{will always denote the completely-observed auxiliary data}
\end{aligned}
$$

Missingness mechanisms: MCAR

1. Missing completely at random (MCAR)- Probabilities of missingness unrelated to data

$$P(R \mid Y) = P(R)$$

- nonrespondents are like a random subsample
- rarely satisfied in practice
- may be refuted by examining Y_{obs} and R

MCAR Example

- After fielding and getting responses of a survey, a questionnaire of a study subject is lost
- The reason for missingness is completely random
- In other words, the probability that an observation is missing is not related to any of the study subject's characteristics
- Simple comparisons on characteristics observed and missing subjects provide a test for MCAR

Example: Missing genotype (Foulkes, Yucel and Li (2008)

- In population-based genetic association studies of unrelated individuals, genotype data can be useful in characterizing genotype-phenotype associations, or even gene-environment interactions
- Problem is that haplotypic phase (the alignment of alleles on a single chromosome) is unobservable
- This information can inform about the possible groupings of individual, giving rise to well-known estimation routines on clustered data
- Is this MCAR?? Perhaps...

Missingness mechanisms: MAR

2. Missing at random (MAR)- Probabilities of missingness may be related to Y_{obs} but not Y_{mis}

$$P(R \mid Y) = P(R \mid Y_{obs})$$

- **an unfortunate name: doesn't mean randomly missing!!!!!!**
- also called "ignorable nonresponse"
- cannot be refuted by examining Y_{obs} and R
- becomes more plausible as Y_{obs} is enriched
- under MAR, one does not need to model R
- good "default" assumption

MAR Example

- Suppose you field a survey and among many items asked, there are items pertaining to education and income
- Those who have between 5 and 15 years of education have complete income values
- Income is missing for a random sample of those who have less than 5 years and more than 15 years of education
- Conditional on education (which is observed), missing data on income are random

Missingness mechanisms: MNAR

3. Missing not at random (MNAR)- Probabilities of missingness may be related to Y_{mis}

- I ALWAYS HOPE THIS IS NEVER THE CASE!
- also called "nonignorable nonresponse"
- more difficult to handle than MAR
- requires explicit joint modeling of Y and R
- other un-verifiable assumptions need to be made
- methods will tend to be problem-specific

MNAR Example

- Suppose you have access to data at IRS (!!!)
- Examining the income for all those who have missing income indicate that they are not responding to income question because they have either high or low income
- Not obvious what to do because as the missingness probability depends on the unobserved characteristic
- If one can enrich the other collected data (e.g. geographical location, education, etc.) then MAR can be enforced
- In clinical studies, this may not be possible

Missingness mechanism: Notes

- In multivariate settings with arbitrary patterns of missingness, we often assume MAR as it is very difficult to posit any probability structure on missingness
- If missingness is less arbitrary (e.g. drop-out in longitudinal designs or planned monotone missingness), then we can get more adventurous
- If we are willing to impose a model on missingness, we should have a very strong reasoning and believe our model is correct
- MAR is often reasonable, especially in surveys

Example of the workshop: SBP Data

- To illustrate simple methods, we will work with an artificial data by Schafer and Graham (2002)
- Suppose that systolic blood pressure (SBP) of $N = 30$ subjects are recorded in January (X)
- Some have a second reading in February (Y) but others do not.
- Our data are simulated from a bivariate normal with means $\mu_x = \mu_y = 125$, standard deviations $\sigma_x = \sigma_y = 25$ and correlation $\rho = .60$

SBP Data

X	Y	X	Y
110	86	126	137
141	157	67	78
116	153	123	149
88	137	98	128
109	116	100	110
131	98	91	115
98	117	123	130
135	158	112	133
101	126	91	119
100	112	173	185
108	148	108	121
103	128	124	147
108	111	99	86
156	119	93	111
143	130	91	117

SBP: Imposing missing values under MCAR

- Randomly select 7 measurements taken in February (Y):

X	Y	X	Y
110	86	126	137
141	NA	67	78
116	153	123	NA
88	NA	98	NA
109	NA	100	NA
131	NA	91	NA
98	NA	123	NA
135	NA	112	NA
101	NA	91	NA
100	NA	173	NA
108	148	108	121
103	NA	124	NA
108	NA	99	NA
156	NA	93	NA
143	NA	91	117

SBP: Imposing missing values under MAR

- Those who have a higher than 130 ($X > 130$) (near-hypertension condition) reading in Jan. returned February for a second reading

X	Y	X	Y
110	NA	126	NA
141	157	67	NA
116	NA	123	NA
88	NA	98	NA
109	NA	100	NA
131	98	91	NA
98	NA	123	NA
135	135	112	NA
101	NA	91	NA
100	NA	173	185
108	148	108	NA
103	NA	124	NA
108	NA	99	NA
156	119	93	NA
143	107	91	NA

SBP: Imposing missing values under MNAR

- Those recorded in February were those whose February measurements exceeded 130 ($Y > 130$) (Nurse may have decided to record only those who are in the hypertensive range)

X	Y	X	Y
110	NA	126	137
141	157	67	NA
116	153	123	149
88	137	98	NA
109	NA	100	NA
131	NA	91	NA
98	NA	123	130
135	158	112	133
101	126	91	NA
100	NA	173	185
108	148	108	NA
103	NA	124	147
108	NA	99	NA
156	NA	93	NA
143	NA	91	NA

R Code (SAS is MAR!)

```
# Below mvrnorm is the function simulating from a multivariate normal, but before set the random
# number generator seed so that we all have the same data
set.seed(1234)
library(MASS)
sbp= round(mvrnorm(n=30,mu=c(125,125), Sigma=matrix(c(25^2,375,375,25^2),2,2)))
sbp
> sbp
      [,1] [,2]
 [1,]  110   86
 [2,]  126  137
 [3,]  141  157
 [4,]   67   78

   omitted
# impose missing values on sbp [,2] (in our case it will be Y) under MCAR
miss=sample(1:30,size=30-7)
sbp.mcar=sbp
sbp.mcar[,2][miss]=NA
# now create MAR mechanism
#  Those who had a first SBP reading greater than 140 returned to give second reading
sbp.mar=sbp
sbp.mar[,2][sbp[,1]<130]=NA
# now create MNAR machanism
sbp.mnar=sbp
sbp.mnar[,2][sbp[,2]<130]=NA
```

Some comments on missingness mechanism

- Increasing sample mean and decreasing SD from MCAR to MAR to MNAR: Not a feature of MCAR, MNAR or MAR but commonly seen in practice (although our sample does not show this)

Table 1: Mean (SD)

X	Y	Y(MCAR)	Y(MAR)	Y(MNAR)
112.20	124.63	120	137.33	148.55
(22.23)	(23.46)	(29.13)	(24.32)	(15.37)

- In multivariate applications, not easy to figure these mechanisms!
- We often use MAR because mathematically convenient
- If possible, sensitivity tests are useful

3. Older methods

Simple case deletion

Omit incomplete cases from analysis, treating the remainder as the actual sample

- used by default in many statistical packages
- there could be various ways to do this in any given situation
  - when estimating a covariance matrix
  - listwise or pairwise deletion
  - complete cases or available cases

Case deletion –Pro

- in a few special cases, case deletion is the statistically "correct" (optimal) method
- does yield correct (**though not efficient**) inferences under MCAR
- can be the most practical solution when the portion of data discarded is "small" and relatively uninfluential
- nonparametric; makes no assumptions about the distribution of the *data*

Case deletion –Con

- nearly always inefficient
- can introduce biases if missingness is not MCAR
- often unclear which set of cases should be used for a particular analysis
- may discard unacceptably large portions of cases, especially in multivariate problems

Reweighting methods

More sophisticated form of case deletion; helps to reduce bias when missingness is not MCAR

- IDEA: Discard incomplete cases and reweight the complete ones so that they more closely resemble the population with respect to distribution of important (e.g. demographic) characteristics
- Often used to handle unit nonresponse in large surveys; nonresponse adjustments can be built into the survey weights

$$w_i = \frac{1}{P(\text{unit } i \text{ selected})} \times \frac{1}{P(\text{unit } i \text{ responds} \mid \text{selected})}$$

The first factor is determined by the sample design, the second factor must be estimated

Reweighting methods : Notes

- Weighting does not require a model for the data, but it may not be efficient
- Reviewed by Little and Rubin (1987) Ch. 4
- only works under MCAR
- examination of the variables for respondents and nonrespondents often reveals whether or MCAR holds

Single imputation

Fill in missing data with plausible values

- used by survey statisticians for 50+ years
- more efficient than case deletion, particularly for item nonresponse
- requires care to avoid data distortion
- adds fake information to dataset (single imputation), distorting uncertainty measures
- multiple imputation fixes the uncertainty problem
- See Little and Rubin (1987) for more discussion

Methods for single imputation

A. Mean substitution

Replace each missing value by mean of observed values

- preserves means
- distorts other aspects of the distribution (variance, quantiles,...)
- doubly disastrous effect on confidence intervals for mean (S^2 too small, n too large)
- distorts relationships among variables
- not recommended

Methods for single imputation (ctd.)

B. Hot deck imputation

Replace each missing value by a randomly drawn observed value

- similar to bootstrap
- preserves marginal distributions
- distorts relationships among variables
- covariate information can be included
- easiest to implement for problems of univariate missingness
- may produce strange results for certain types of problems (e.g. estimation of variances, quantiles)

Methods for single imputation (ctd.)

C. Regression methods

Replace each missing value by a predicted value from a regression model estimated from the observed data

Suppose we observe X and Y, X is completely observed and Y is only observed for n_{obs} cases.

- regress Y on X for cases $1, \ldots, n_{obs}$
- impute $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ for $i = n_{obs} + 1, \ldots, n$
- this WILL inflate correlations
- **Last observation carried forward :** even worse as it ignores regression to the mean

Methods for single imputation (ctd.)

C. Regression methods (CTD.)

Better idea: Replace each missing value by a predicted value plus a random residual

- impute $y_i = \hat{\beta} x_i + N(0, S^2)$, where $S^2 = MSE$
- requires a model
- assumes MAR
- becomes more difficult for multivariate missingness

Illustrating single imputation on SBP data

- Impute missing blood pressure readings of Y, under MAR mechanism
- Four methods
  - Mean substitution
  - Simple hot deck
  - Conditional mean imputation based on linear regression of Y on X
  - Drawing from the estimated predictive distribution of Y given X

Implementing via R

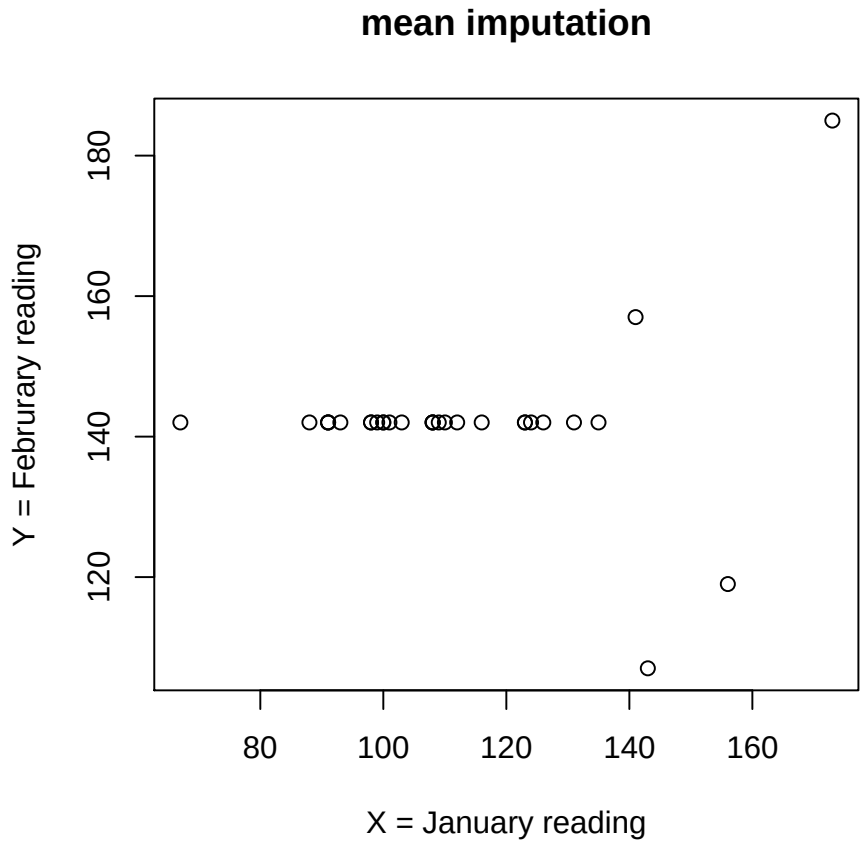
```
# mean imputation
sbp.mar[,2][sbp[,1]<140]=mean(sbp.mar[,2],na.rm=T)
plot(sbp.mar[,1],sbp.mar[,2],xlab="X = January reading",
ylab="Y = Februrary reading", main="mean imputation")

#simple hot deck imputation
## first re-create sbp.mar
sbp.mar=sbp
sbp.mar[,2][sbp[,1]<140]=NA
# observed values:
sbp.mar[,2][!is.na(sbp.mar[,2])]
# now sample of size=1 and replace teh NAs:
for(i in 1:30){sbp.mar[i,2][sbp[i,1]<140]=
    sample(sbp.mar[,2][!is.na(sbp.mar[,2])],size=1)}
# now look at the plot:
plot(sbp.mar[,1],sbp.mar[,2],xlab="X = January reading",
ylab="Y = Februrary reading", main="simple hot deck")
```

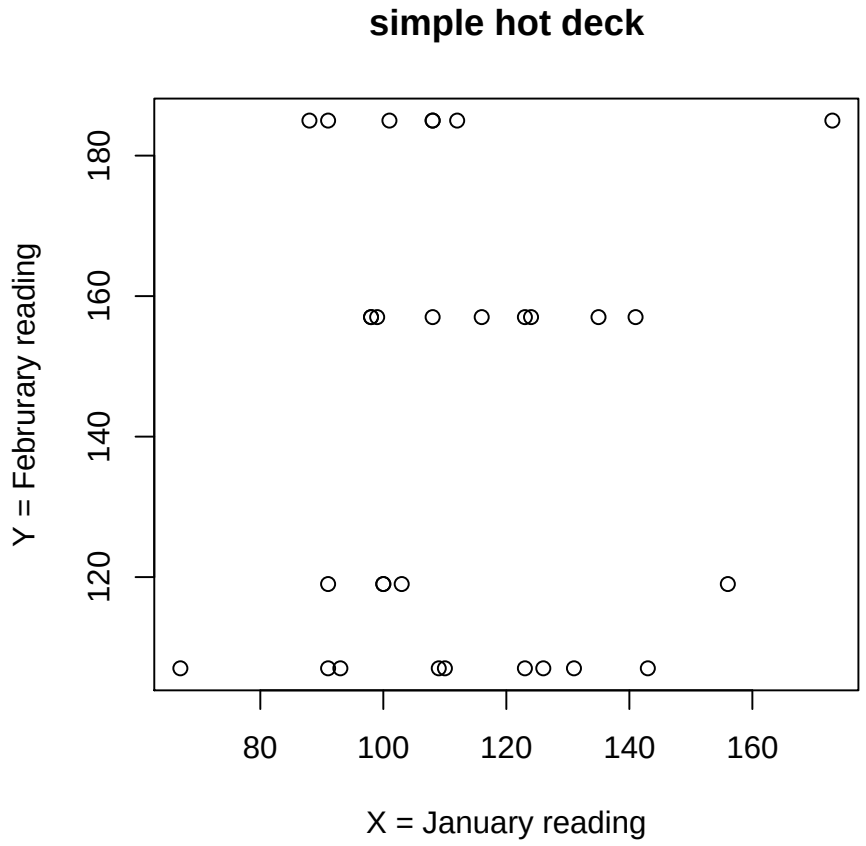
```
# Conditional mean
# first fit a regression model, and predict values of missing Y for the observed X values
reg=lm(sbp.mar[,2]~sbp.mar[,1])
for(i in 1:30){sbp.mar[i,2][sbp[i,1]<140]=reg$coef[1]+reg$coef[2]*sbp.mar[i,1]}
plot(sbp.mar[,1],sbp.mar[,2],xlab="X = January reading",
ylab="Y = Februrary reading", main="conditional mean imputation")

# imputing from predictive distribution (in our case this would be just adding
# a normal variate with
# variance equal to the residual variance
sbp.mar=sbp
sbp.mar[,2][sbp[,1]<140]=NA
for(i in 1:30){sbp.mar[i,2][sbp[i,1]<140]=reg$coef[1]+reg$coef[2]*sbp.mar[i,1]+rnorm(n=1,mean=0,sd=34.74)}
plot(sbp.mar[,1],sbp.mar[,2],xlab="X = January reading",
ylab="Y = Februrary reading", main="Predictive distribution imputation")
```

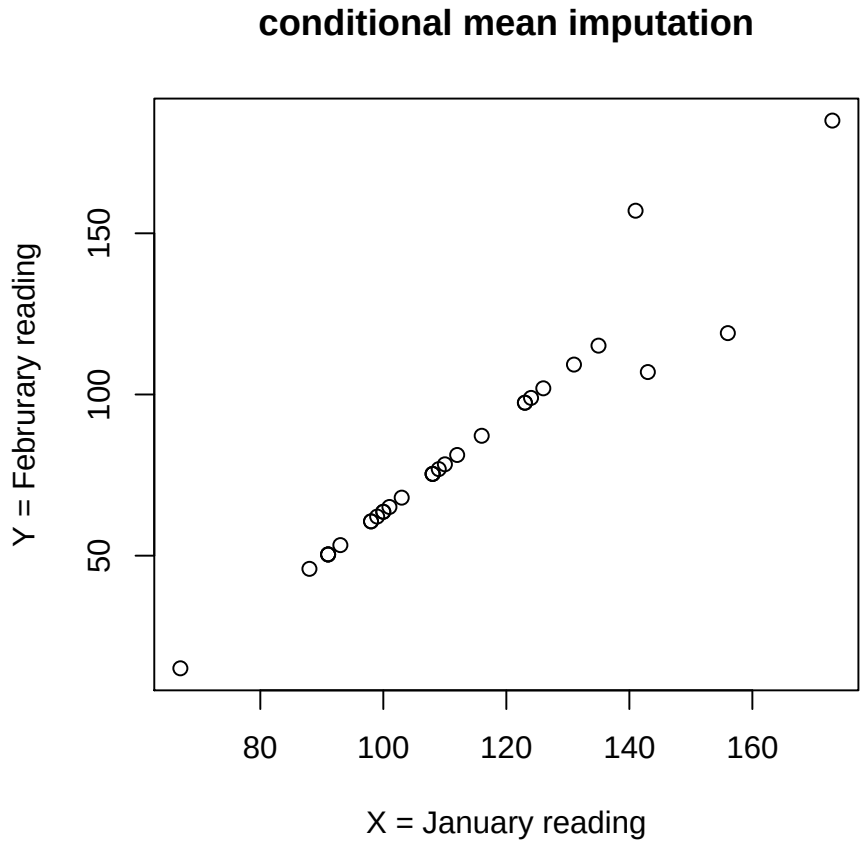
Relationship after mean imputation



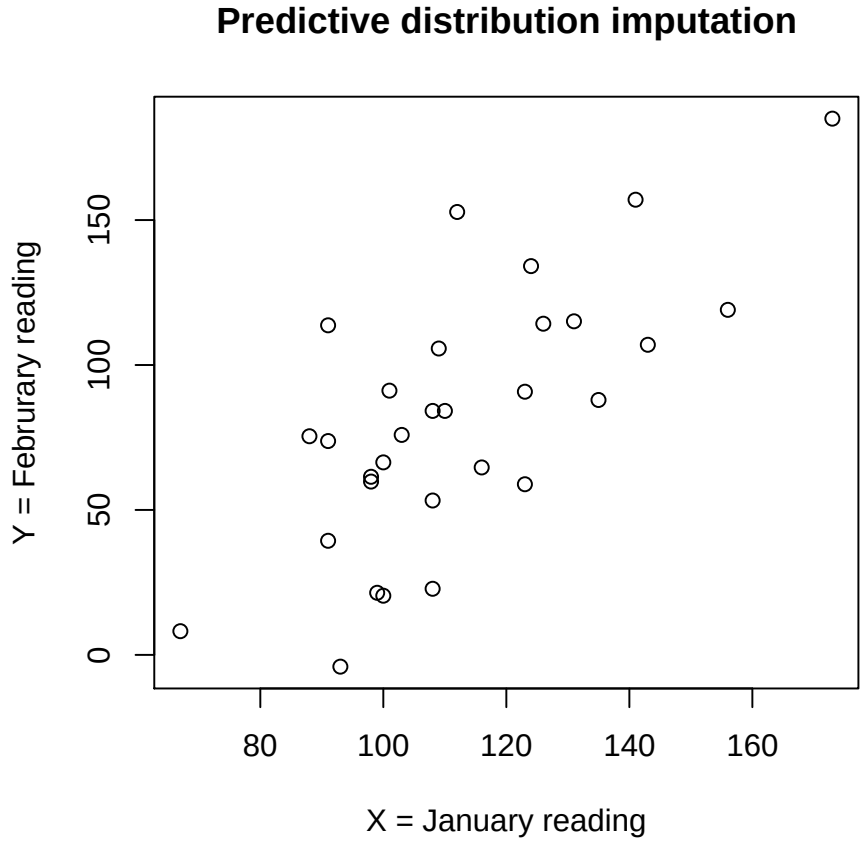
Relationship hot deck imputation



Relationship after conditional mean imputation



Relationship after imputing from a predictive distribution



Problem with single imputation

FUNDAMENTAL ISSUE

Whenever missing data are replaced by one set of imputed values, later analyses will not reflect **missing-data uncertainty**

- sample size is overstated
- confidence intervals too narrow
- Type 1 error rated too high

The problem becomes worse as the **rates of missing information** and the **number of parameters** increase

Problem with single imputation (ctd.)

Example: Scalar Inference

- 30% missingness
- confidence intervals for a scalar quantity (regression coefficient, odds ratio, relative risk, etc.)

Nominal coverage	90%	95%	99%
Actual coverage	77%	85%	94%

- In extreme cases such as our running example, performance become dramatically bad, in fact coverage seen as 0% when the misisngness mechanism is MNAR
- See Graham and Schafer (2002) for detailed discussion

Problem with single imputation (ctd.)

Example: Multiparameter Inference

- 30% missingness
- significance levels for testing a ten component null hypothesis (e.g. regression F-test)

Nominal α	10%	5%	1%
Actual α	57%	45%	25%

4. Maximum Likelihood

Estimate parameters of interest directly from the observed data, maximizing

$$\begin{aligned} f(Y;\theta) &= \text{distribution of complete data} \\ L(\theta;Y_{obs}) &= \int f(Y;\theta)dY_{mis} \end{aligned}$$

- assumes MAR; response mechanism is not modeled
- fully parametric
- approximately unbiased in large samples and is highly efficient
- Confidence intervals or regions are often computed using well known result

$$\hat{\theta} \sim N(\theta, [-l''(\hat{\theta})]^{-1})$$

Maximum Likelihood (ctd.)

- requires special algorithms such as EM
- PROBLEM-SPECIFIC
- standard errors may be difficult to get (observed information matrix, not expected)

Maximum Likelihood (ctd.)

EXAMPLES A few software products are available for calculating ML

- NORM, CAT, MIX, PAN
- SAS PROC MI, counter-inutitive but it actually calculates ML when number of imputations is specified as zero (examples will follow)
- SAS PROC MIXED for unbalanced longitudinal data with missing responses (does not handle missing covariates, for that refer to PAN)
- AMOS, Mx, Mplus for linear models and structural equations with incomplete data

COMMENTS

- these all assume MAR
- Missing data might not be properly accounted for if important "causes" of missingness are not in the model

Maximum likelihood: EM

Computing MLE in missing-data problems

- Often requires iterative computation
- A general method for MLE in missing data was described by Dempster, Laird and Rubin (DLR) (1977) *("solve an intractable incomplete-data problem by iteratively solving an easier complete-data problem")*

Likelihood-based methods: EM of DLR

Features of EM

- Very stable; guaranteed to increase $l(\theta; Y)$
- Convergence rate is linear
- high rates of missing information can make it converge painfully slow!
- SEs are **not** automatic byproduct of EM, unlike Fisher scoring
- wide variety of applications including non-missing data problems such as random-effects models, latent class models, etc.

General comments

- In principle, ML is the most efficient answer
- In nearly all cases, the ML estimation routines available today assume MAR
- ML estimation with incomplete data is available to users in only a fairly small group of models
- ML algorithms for incomplete data can be complicated
- ML estimation for a range of regression models with missing covariates and nonignorable missing response by Ibrahim and colleagues

Example: Finding ML via SAS PROC MI

```
data sbpmar;
input x y @@;
datalines;
110 .   126 .
141 157 67 .
116 . 123 .
88 . 98 .
109 . 100 .
131 . 91 .
98 . 123 .
135 . 112 .
101 . 91 .
100 . 173 185
108 . 108 .
103 . 124 .
108 . 99 .
156 119 93 .
143 107 91 .
;
* now use PRC MI to find the ML in the presence of missing values
proc mi data=sbpmar seed=1234 simple nimpute=0;
em itprint outem=outem;
var x y;
run;
* This example is not all that interesting! But used as illustration!;
```

EM of SAS PROC MI: Output

Model information

Model Information

Data Set	WORK.SBPMAR
Method	MCMC
Multiple Imputation Chain	Single Chain
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	0
Number of Burn-in Iterations	200
Number of Iterations	100
Seed for random number generator	1234

EM of SAS PROC MI: Output

Missing data patterns

Missing Data Patterns

Group	x	y	Freq	Percent	Group Means x	Group Means y
1	X	X	4	13.33	153.250000	142.000000
2	X	.	26	86.67	105.884615	.

EM of SAS PROC MI: Output

Univariate statistics and pairwise correlations

Univariate Statistics

Variable	N	Mean	Std Dev	Minimum	Maximum	---Missing Values-- Count	Percent
x	30	112.20000	22.23914	67.00000	173.00000	0	0.00
y	4	142.00000	35.72114	107.00000	185.00000	26	86.67

Pairwise Correlations

	x	y
x	1.000000000	0.607944569
y	0.607944569	1.000000000

EM of SAS PROC MI: Output

Initial Parameter Estimates for EM

TYPE	_NAME_	x	y
MEAN		112.200000	142.000000
COV	x	494.579310	0
COV	y	0	1276.000000

EM of SAS PROC MI: Output

MI Procedure, Iteration history and EM (MLE) estimates

The MI Procedure

EM (MLE) Iteration History

Iteration	-2 Log L	y
0	246.717168	142.000000
1	246.588776	142.000000
2	246.529711	141.633274
	.....	
	.....	
196	244.756075	91.026681
197	244.755094	90.936872
198	244.754131	90.847915
199	244.753186	90.759803
200	244.752258	90.672527

EM (MLE) Parameter Estimates

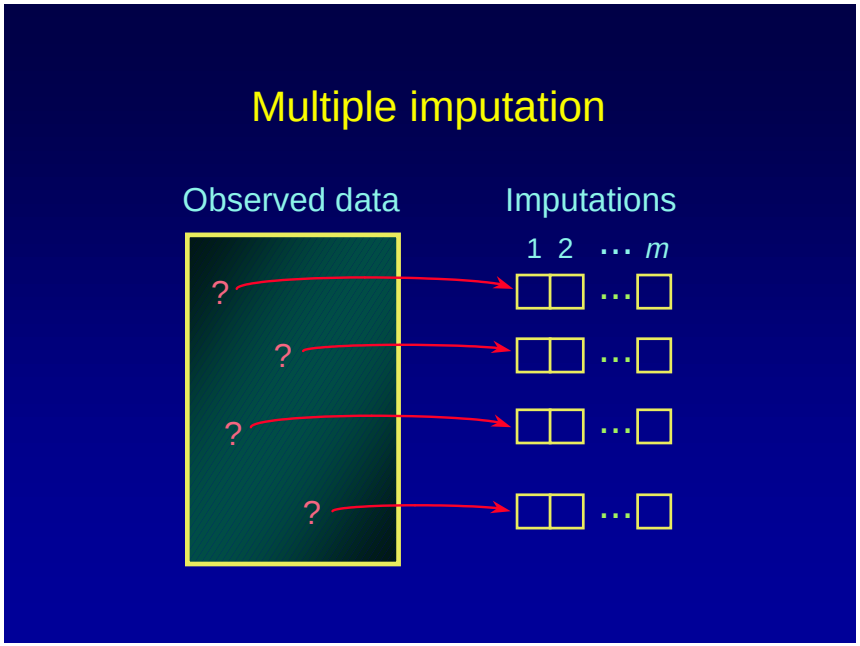
TYPE	_NAME_	x	y
MEAN		112.200000	90.672527
COV	x	478.093333	605.342212
COV	y	605.342212	1378.238156

EM (Posterior Mode) Estimates

TYPE	_NAME_	x	y
MEAN		112.200000	82.917007
COV	x	434.630303	626.579673
COV	y	626.579673	1248.142498

5. Multiple imputation

A simulation-based approach to missing data



Multiple imputation (ctd.)

- Retains much of the attractiveness of single imputation from a conditional distribution but solves the problem of understating uncertainty
- Replaces each missing value by a list of m simulated values values
- Each of the m data sets is analyzed in the same fashion by a complete data method; the results are then combined following Rubin's rules

Rubin (1987) calls this the *repeated-imputation* inference method

Multiple imputation (ctd.)

- works with standard complete-data analysis methods
- One set of imputations may be used for many analyses
- can be highly efficient, even for small m
  - The efficiency of an estimator based on m imputations is $(1 + \gamma/m)^{-1}$, where γ is the rate of missing information

Efficiency of multiple imputation (%)					
			γ		
m	0.1	0.3	0.5	0.7	0.9
3	97	91	86	81	77
5	98	94	91	88	85
10	99	97	95	93	92
20	100	99	98	97	96

Multiple imputation: RULES for combining estimates

After obtaining m imputations of Y_{mis}, analyze the m completed datasets and combine the results

RUBIN'S (1987) RULES FOR SCALAR ESTIMATES

$$\begin{aligned}
\hat{Q} &= \text{complete-data point estimate} \\
\hat{U} &= \text{complete-data variance estimate} \\
\bar{Q} &= m^{(-1)} \sum_{t=1}^{m} \hat{Q}^{(t)} \\
B &= (m-1)^{-1} \sum_{t=1}^{m} (\hat{Q}^{(t)} - \bar{Q})^2 \\
&= \text{Between imputation variance}
\end{aligned}$$

$$\begin{aligned}
\bar{U} &= m^{(-1)} \sum_{t=1}^{m} U^{(t)} \\
&= \text{Within imputation variance} \\
T &= \bar{U} + (1 + m^{-1})B \\
&= \text{Total variance}
\end{aligned}$$

Interval estimate is $\bar{Q} \pm t_\nu \sqrt{T}$, where

$$\nu = (m-1)\left[1 + \frac{\bar{U}}{(1+m^{-1})B}\right]^2$$

Multiple imputation: RULES for combining estimates

- Degrees of freedom vary from $m-1$ to ∞, depending on relative sizes of $\bar{U}$ and $(1+m^{-1})B$
- Relative increase in variance due to nonresponse is estimated by

$$r = \frac{(1+m^{-1})B}{\bar{U}}$$

- Fraction of missing information is estimated by

$$\frac{r+2/(\nu+3)}{r+1}$$

Note': this estimate can be noisy for small n

Multiple imputation: RULES for combining estimates

Additional methods available for multidimensional estimands

- Combining point estimates and covariance matrices (Li, Raghunathan, and Rubin, 1991)
- Combining p-values (Li et al., 1991)
- Combining likelihood-ratio test statistics (Meng and Rubin, 1992)

All methods reviewed in Schafer (1997, ch. 4)

Multiple imputation

Proper multiple imputation

- The validity of MI rests upon how the imputations are created and how that procedure relates to the model used to subsequently analyze the data
- Creating MI's often requires special algorithms (Schafer, 1997; Schafer and Yucel, 2002)
- In general, they should be drawn from a distribution for the missing data that reflects uncertainty about the parameters of the data model; $P(Y_{mis} \mid Y_{obs}, \theta)$.
- Simulate m **independent** plausible values for the paramaters $\theta^{(1)}, \theta^{(2)}, \ldots, \theta^{(m}$ then draw the missing data for each $\theta^{(i)}$, $i = 1, 2, \ldots, m$

Multiple imputation (ctd.)

- "Independence" is key to get to the multiply imputed datasets
- Assess whether the m versions are independent using time series plots (PART II)
- Use each of the m data to fit the models and combine the results (More on this in Part II)

Multiple imputation: Imputation models

- The imputation model should include
  - variables crucial to the analysis
  - variables that are highly predictive of them
  - variables that are highly predictive of the missingness
  - variables that describe special features of the sample design (probability surveys)
- Relationships that are the subject of future analyses should be present
- May rely on extra information
- Approximate models are usually okay

Creating MIs

Example 1

$$
\begin{array}{rcl}
x_1, x_2, \ldots, x_n & & \text{observed covariate} \\
y_1, y_2, \ldots, y_n & \sim & N(\beta_0 + \beta_1 xi, \sigma^2) \\
y_1, \ldots, y_a & & \textit{observed} \\
y_{a+1}, \ldots, y_n & & \textit{missing}
\end{array}
$$

$$
X = \begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ . & . \\ . & . \\ . & . \\ 1 & x_n \end{pmatrix} \quad y = \begin{pmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_n \end{pmatrix} \quad \hat{\beta} = \begin{pmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{pmatrix}
$$

Creating MIs : Example 1 ctd.

- Calculate $\hat{\beta} = (X^T X)^{-1} X^T y$, $\hat{\epsilon} = y - X\hat{\beta}$
- Draw $\sigma^2 \sim \hat{\epsilon}^T \hat{\epsilon} / \chi^2_{a-2}$
- Draw $\beta \sim N(\hat{\beta}, \sigma^2 (X^T X)^{-1})$
- Draw $y_i \sim N(\beta_0 + \beta_1 x_i, \sigma^2),\ i = a+1, \ldots, n$
- Repeat m times

Creating MIs

EXAMPLE 2

$$\begin{aligned} y_1, y_2, \ldots, y_n &\sim F \\ y_1, \ldots, y_a &\quad \textit{observed} \\ y_{a+1}, \ldots, y_n &\quad \textit{missing} \end{aligned}$$

Question: Can we create proper multiple imputations nonparametrically?

- Draw n values from $y_1, \ldots, y_a$ with replacement
- Subsample $n - a$ values from these with replacement
- Repeat m times

Called "approximate Bayesian bootstrap" (ABB) by Rubin (1987)

- extension of this used by SOLAS (Statistical Solutions, 1998)

Creating MIs

More Generally We need to obtain independent draws $Y_{mis}^{(1)}, \ldots, Y_{mis}^{m}$

$$P(Ymis \mid Yobs) = \int P(Ymis \mid Yobs, \theta) P(\theta \mid Yobs) d\theta$$

where θ = parameters of the imputation model

- $P(\theta \mid Yobs)$ is often intractable
- special computational methods needed (Markov chain Monte Carlo, MCMC)
- NORM uses a class of MCMC, Data Augmentation, to accomplish this
- Augments the unknowns (missing data and parameters) from their prospective distributions and iterate for a number of cycles

- Must have large number of iterations to ensure that draws of missing data are independent
  - time series plots of θ
  - autocorrelation functions for θ

Creating MIs':Practical Advice

- Be conservative; its better to overestimate the number of cycles needed
- Before trying data augmentation, its a good idea to run EM
- Parameter estimates from EM are good starting values for DA
- Convergence behavior of DA is usually resembles that of EM

  Rule-of-thumb: If EM converges by t cycles, DA will almost certainly converge by t cycles as well

Notes on multiple imputation and EM

- Statistical model needed in both for complete data (normal, etc.)
- Assumptions on missingness mechanism is needed (MAR)
- Often MI is preferred as the "practitioners" could focus on analyses of a more substantive importance using the "completed" data
- In some problems (e.g. clustered data) EM requires more complex statistical algorithms
- Prior distribution for parameters for MI (default noninformative priors are usually okay)

Software for MI

Software for imputation under a multivariate normal model

- **NORM** (Schafer 1999):Free Windows program. Uses data augmentation.
- **Amelia** (Gary King et al., 2001): Free program; also available as macros for GAUSS. Uses importance resampling rather than DA.
- **PROC MI** (SAS version 8.2): New "experimental" SAS procedure equivalent to NORM, also implements other methods for creating multiple imputations
- **S+ Missing Data (S-PLUS Version 6) and R library norm**: MI for Gaussian model; equivalent to NORM

Software for MI(ctd.)

Software for imputation under other models

- **CAT, MIX:** Schafers old S-PLUS functions for categorical data, mixed continuous and categorical data; now obsolete (part of SPlus library(missing) and R libraries cat and mix)
- **PAN Schafer and Yucel (2002):** program for longitudinal data; under development. Currently available as R library pan.
- **SOLAS (Statistical Solutions, Inc.):** Multiple imputation by two methods
  - Propensity-score method with approximate Bayesian bootstrap (Lavori, Dawson and Shera, 1995). This can be dangerous! (Allison, 2000)

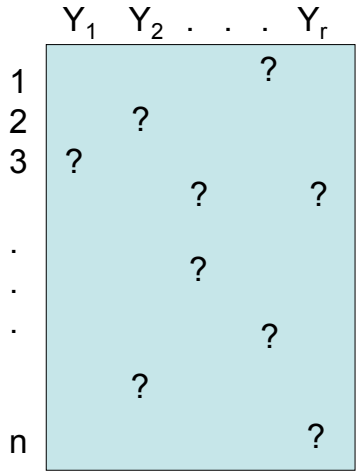
  - Model-based method using a sequence of regression models; requires missingness pattern to be monotone.

- **MICE and IveWare:** S+ and R functions for approximate MI using "chained regression equations". IveWare is a set of fortran routines and available as SAS macro. MICE is available from

  `http:\\www.multiple-imputation.com`

Part II

1. Methods for continuous incomplete data



? : missing values (MAR)

n units are independent and
Identically distributed N(0,Σ)

1. Methods for continuous incomplete data

- NORM (SAME AS PROC MI with MCMC option)
  - Computational algorithm
  - What to watch for
  - Example and application using NORM
- SAS PROC MI
- IveWare

Imputation model : The normal model

$$\begin{aligned} Y &= \text{matrix of complete data} \\ &= (y_1, y_2, \ldots, y_n)^T \end{aligned}$$

Assume that

$$y_1, y_2, \ldots, y_n \mid \theta \sim N_p(\mu, \Sigma),$$

where $\theta = (\mu, \Sigma)$ is unknown.

- Each variable is assumed to be normal
- Units are regarded as independently drawn from the same population (no strata or clusters, in surveys may use these as random variables)
- Only simple (pairwise) associations among variable; no interaction

Most real data depart from these assumptions. But we may still be able to use the normal model to produce good-quality imputations.

The normal model: How about non-normal variables?

- If a variable is skewed, apply a transformation (sqrt, log, etc)
- If a variable is binary or ordinal, impute under the normal model and round off imputed values to the nearest (OBSERVED) category
- Better methods are available (Yucel, Yulei and Zaslavsky, 2008) but the gain is minimal

Simulations show that multiple imputation is quite robust to model misspecification (Schafer 1997, ch. 6)

Completely observed categorical data A k-level categorical variable with no missing values may be included in a normal model

- Replace the variable with $k-1$ dummy codes
- Inferences are not harmed because the variable is never imputed

The normal model: Complex Associations

- Normal model implies that each variable has an additive, linear regression on the other variables. Imputations will not reflect
  - nonlinear relationships
  - interactions
- If Y_1 and Y_2 are completely observed, then we can include higher-order variables Y_1^2, $Y_1 Y_2$, etc. without harm
- If missingness on Y_1 and Y_2 is mild, we may include higher-order variables in the imputation model, but remove them from the imputed dataset and recalculate them
- For these IveWare could be an alternative

Data Example: Adolescent alcohol prevention trial (AAPT)

THIS EXAMPLE IS INTENDED FOR NORM DEMO!

Longitudinal study of substance use in Los Angeles area schools (Hansen and Graham, 1991)

- $n = 3574$ students
- Let's focus on the first wave of data
- Look for effects of POSCON and NEGCON on reported alcohol use, controlling for other covariates
- Missing values due to nonresponse and attrition; some variables missing by 33% by design (MAR is ok for missing by design)

Data Example: AAPT

STRATEGY

1. Impute missing values $m = 10$ times using NORM software
   - Include a dummy for sex
2. Fit a linear regression model (10 times) to predict the alcohol use given covariates (POSCON, NEGCON and sex) to see the effects of POSCON and NEGCON
3. Combine estimated coefficients and standard errors using Rubin's rule

NORM: http://www.stat.psu.edu/ jls/misoftwa.html

Free Windows program for multiple imputation using Bayesian methods based on a multivariate normal model. It acts as a

- **Pre-processor,** filling in the missing values so that other statistical programs can make full use of your data
- **post-processor**, combining the output (estimates and standard errors) from m statistical analyses to produce a single set of results

NORM's basic functions

- **Summarize** rates and patterns of missing values
- **EM algorithm** for efficient estimation of means, variances and covariances from an incomplete data set
- **Data augmentation** procedure for creating proper multiple imputations of missing values
- **Series plots** to diagnose the convergence behavior of data augmentation
- Facility for **combining the results** of a multiply-imputed data analysis, using Rubin.s (1987) rules for MI inference

All functions are invoked through an easy-to-use graphical Windows interface

NORM : Features

- no limit to the number of cases or variables (sometimes crashes after 60 variables, better to use the "missing" library of Splus, or "norm" library of R)
- interactively **tabulate and plot** variables
- **include or exclude** variables from the model or imputed data sets
- apply **transformations** to improve normality
- automatic **rounding** of imputed values to any precision, or to the variable's nearest observed values
- extensive on-line help manual

NORM handout

- Screen shots of Norm software

SAS PROC MI

- SAS procedure for creating multiple imputations
- Assumes sampled units to be independent (no clustering), so important to incorporate design variables as much as possible
- Several methods are available for creating multiple imputations
- Method of choice depends on the missingness patterns

SAS PROC MI : Choosing the imputation method

- **Univariate or Monotone missingness:** regression imputation assuming a multivariate normality, or propensity score method.
- **Arbitrary missing patterns:** A markov chain monte carlo (mcmc option) that relies on the NORM techniques
- More information available at

  `http://support.sas.com/rnd/app/da/new/802ce/stat/chap9/index.htm`

SAS PROC MI : Example code

```
* first let's look at our unintersting example;
proc mi data=sbpmar seed=1234 out=outmi nimpute=10
Round= 1 1;
var x y;
    mcmc timeplot(mean(y)) acfplot(mean(y));
run;

* because our example pertains to univariate missingness;
* we can apply propensity method (nonparametric) or monotone method as well;
* the latter is based on sequential and parametric;

* First propensity method;
proc mi data=sbpmar seed=1234 simple out=outmi;
monotone method=propensity;
var x y;
run;
* Now the monotone;
proc mi data=sbpmar seed=1234 simple out=outmi;
mcmc impute=monotone;
var x y;
run;


* Example 2: Many variables in a hypothetic "sample" data
```

```
*multiply impute 10 datasets;
proc mi data=sample seed=32984 out=outmi nimpute=10
   Minimum =94 42   0 0 -26  0 0 0 1 0  1    16  1 1 1 0 0
   maximum=221 106  5 1  32  1 1 6 5 1  3.64 30  4 4 6 3 1
   Round=    1 1    1 1 .01  1 1 1 1 1 .01    1  1 1 1 1 1;
   var bps bpd   sesum
        racewb agecent  marriedYN Finhs comsum exer
        smoke trustsc decsum employ  finance  house
         Noncompsc smoke_race;
    mcmc timeplot(mean) acfplot(mean);
   run;
```

- "seed" initializes the random number generator
- "outmi" is the output data that will contain the 10 imputed data, it will have a variable called

  ``_Imputation_''

  referring to the number of the imputed data
- Maximum and minimum are used to determine the ranges of the (observed) variables

- "round" indicates the units to round variables in the imputation
- var statements identifies the variables to be analyzed, all variables must be numeric.

SAS PROC MI : Results from SBP.MAR data

Multiple Imputation Variance Information

	Variance			
Variable	Between	Within	Total	DF
y	4288.539519	99.106847	4816.500317	0.5279

Multiple Imputation Variance Information

Variable	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
y	47.599067	0.982747	0.910519

Multiple Imputation Parameter Estimates

Variable	Mean	Std Error	95% Confidence Limits		DF	Minimum	Maximum
y	92.756667	69.401011	.	.	0.5279	-41.366667	194.966667

SAS PROC MI : fitting models

```
* Example 1;
proc reg data=outmi outest=outmodel1 covout;
  model x=y;
  by _Imputation_;
run;
* now combine these regressions;
proc print data=outmodel1(obs=10);
    var _Imputation_ _Type_ _Name_;
        Intercept y;
proc mianalyze data=outmodel1;
     var Intercept y;
run;

* Example 2;
*regression on bps and bpd;
proc reg data=outmi2 outest=outmodel5 covout;* noprint;
  model bps bpd=  smoke  trustsc decsum racewb agecent  marriedYN
       Finhs smoke_race;
  by _Imputation_;
  run;
```

- Each imputed data set is used to fit a regression
- Important to use

‘‘`by _Imputation_`’’

so that the reg. proc goes through the 10 imputed dataset

SAS PROC MI : Example code (ctd.)

```
*Sort by Dependent variable so MI Analyze can examine both dependent variables;
proc sort data=outmodel5;
  by _depvar_;
  run;
*set ods file name;
ods rtf file='d:\MI Model5.rtf' style=minimal;
*combine parameter estimates and standard errors using Rubin's rules;
  proc mianalyze data=outmodel5 edf=158;
  var intercept   smoke  trustsc decsum racewb agecent  marriedYN
       Finhs smoke_race  ;
        by _depvar_;
  run;
```

- Finally, PROC MIANALYZE combines the 10 sets of regression coefficients and SEs

Regressing y on x using available cases

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   -83.620    209.079  -0.400    0.728
sbp.mar[, 1]    1.472      1.360   1.083    0.392

Residual standard error: 34.74 on 2 degrees of freedom
  (26 observations deleted due to missingness)
Multiple R-squared: 0.3696, Adjusted R-squared: 0.05439
F-statistic: 1.173 on 1 and 2 DF,  p-value: 0.3921
```

SAS PROC MI : Output

Model Information

Data Set	WORK.OUTMODEL1
Number of Imputations	10

Multiple Imputation Variance Information

	Variance			
Parameter	Between	Within	Total	DF
Intercept	885.449655	127.448880	1101.443501	11.509
y	0.062924	0.006049	0.075266	10.642

Multiple Imputation Variance Information

Parameter	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	7.642238	0.900239	0.917411
y	11.442773	0.931415	0.914795

Multiple Imputation Parameter Estimates

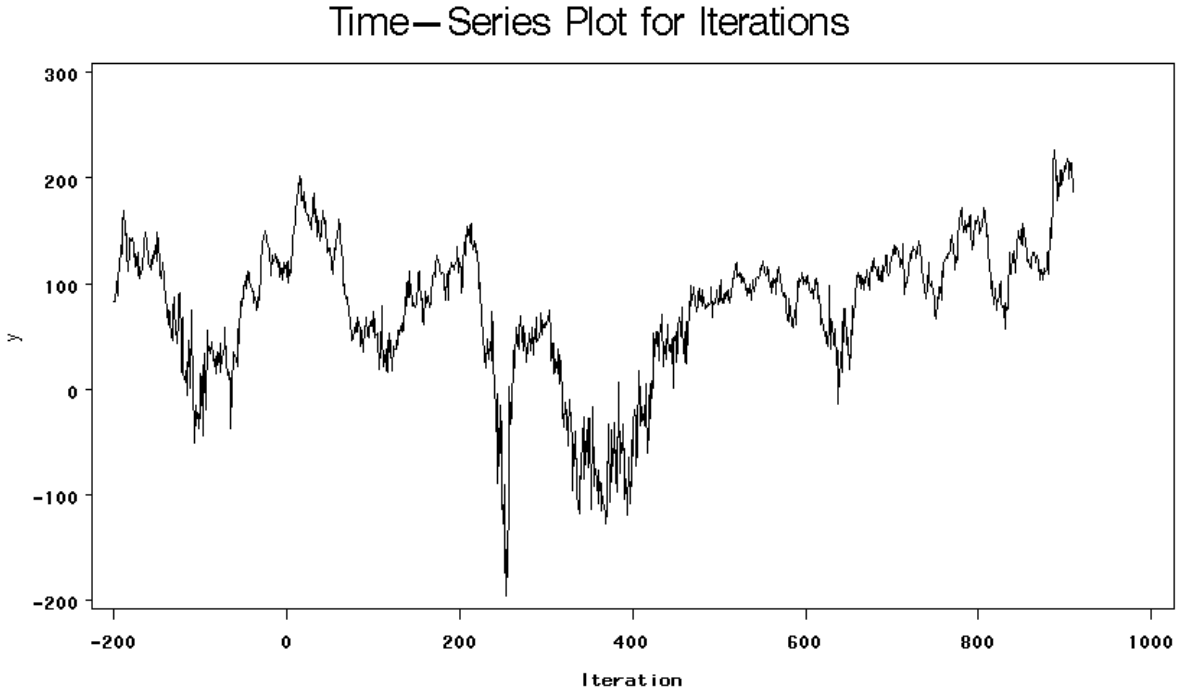
Parameter	Estimate	Std Error	95% Confidence Limits		DF	Minimum	Maximum
Intercept	94.772869	33.188002	22.11911	167.4266	11.509	49.573262	156.265705
y	0.262961	0.274346	-0.34336	0.8693	10.642	-0.226017	0.655549

Multiple Imputation Parameter Estimates

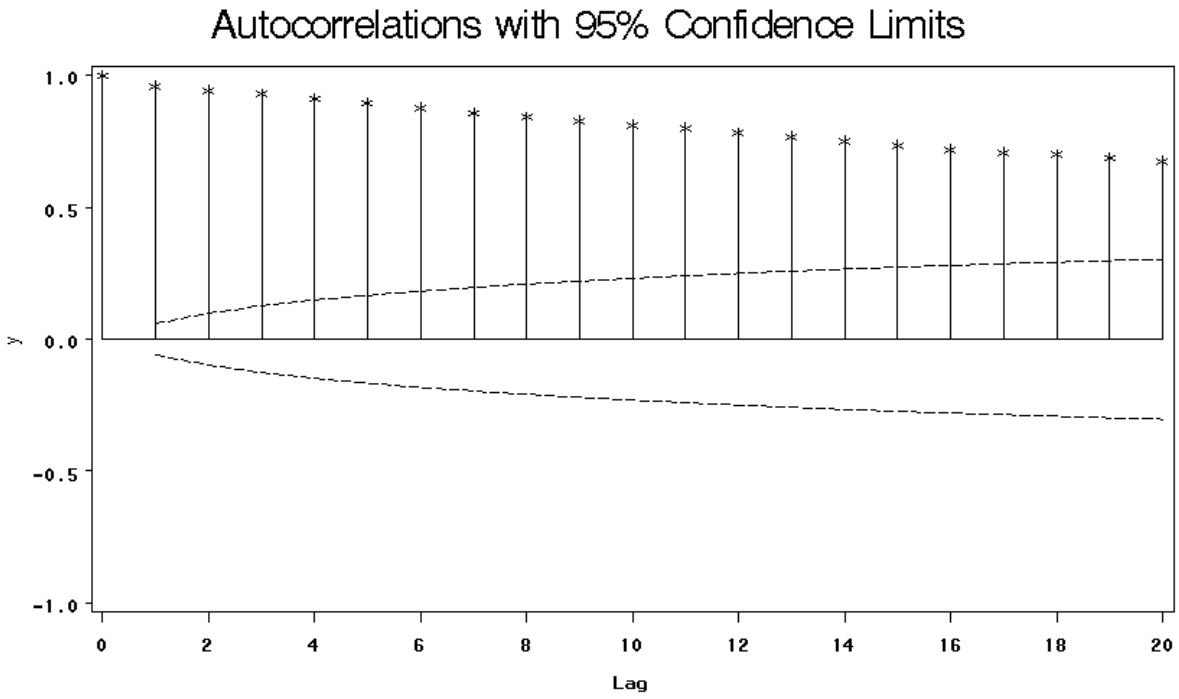
Parameter	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	0	2.86	0.0150
y	0	0.96	0.3591

Monitoring convergence

Time series plot (ideally we DO NOT want to see any pattern):

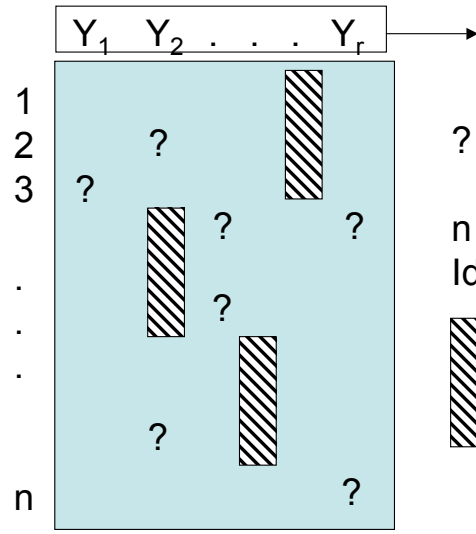


Autocorrelation plot (ideally we want correlations to die down within a few iterations):



Usually iterations die down around the same number of iterations of EM.

IveWare:



ALL KINDS OF VARIABLES:
CTS, BINARY, ORDINAL, ETC.

? : missing values (MAR)

n units are independent and
Identically distributed N(0,Σ)

: skip patterns

IveWare: http://www.isr.umich.edu/src/smp/ive/

- SAS Macro developed by Raghunathan and colleagues at the University of Michigan
- Primarily deals with missing values in surveys
- Also has an option for calculating accurate estimates of variances in complex sample surveys, incorporating sampling weights
- No clustering, units are assumed to be independent

IveWare: Imputing Survey Data

What to do with Skip patterns (restrictions), bounds

- Survey items on "mammography" or "smoking" are not applicable to all individuals
- Should we set the values of men on mammography to missing?
- Probably better to assume a model on the correct sub-sample
- On some other items with incomplete data, appropriate models (such as truncated probability distribution) could be more appropriate (e.g. "years of smoking")

IveWare : Idea

- Imputation on a variable by variable basis conditioning on all observed variables
- Creates imputations through a sequence of multiple hierarchical regressions with covariates including all other variables (observed or imputed)
- The sequence of imputation occurs in a cyclical manner, overwriting previously drawn values
- Type of regression models depends on the variable being imputed

IveWare (ctd.)

Models

- linear regression for continuous variables
- logistic and polytomous regression for binary and ordinal variables
- multinomial regression for nominal
- two-part model for semi-continuous variables

IveWare (ctd.)

- Draws from "pragmatic" conditional distributions
- Cannot be derived from a joint distribution
- Difficult to assess analytically appropriateness of this "approximation"
- Useful for creating multiple imputations especially in highly multivariate problems

IveWare: Modules

- **IMPUTE:** creates multiply imputed datasets using the sequential approach
- **DESCRIBE:** estimates population parameters (means, subgroup differences), nice feature for complex sample surveys
- **REGRESS:** fits a variety of models for data resulting from a complex sample design
- **SASMOD:** allows users to take into account complex sample design features when analysing data with several SAS procs.
- All three modules (DESCRIBE, REGRESS, SASMOD) allow missing values and preform multiple imputation analysis

2. Methods for categorical incomplete data: CAT

- Splus (part of library "missing") and R (library "cat") function developed by Schafer (1997)
- Imputation model is a loglinear model
- Markov-Chain Monte Carlo method for simulating from this loglinear model
- More description available at

  `http://cran.r-project.org/doc/packages/cat.pdf`

- As the number of categorical variables increases, becomes almost impossible to function
- IveWare should be considered in dealing with large number (>10) of categorical variables with missing values

CAT: R functions

- Number of functions is available in the R library called CAT, some of the most useful ones are:
  - **prelim.cat:** prepares the data by grouping and sorting (by missingness patterns, etc.) to be used by the following functions
  - **em.cat or ecm.cat:** Finds ML estimate or posterior mode of cell probabilities under the saturated model
  - **da.cat:** simulates unknown parameters from the observed-data posterior (or likelihood) under a saturated model. Could be used with the imp.cat() function
  - **imp.cat:** Performs single random imputation of missing values in a categorical dataset under a value of cell probabilities (from em.cat or da.cat)
  - **mi.inference:** Combines estimates and SEs from m completed-data analyses

CAT: R functions

IMPORTANT NOTE ON DATA MATRIX X:

- The rows of x correspond to observational units, and the columns to variables
- Missing values are denoted by NA
- The categorical variables must be coded with consecutive positive integers starting with 1. For example, a binary variable must be coded as 1,2 rather than 0,1.

CAT: Example code

```
#
# Example 1 Based on Schafer's p. 329 and ss. This is a toy version,
# using a much shorter length of chain than required. To
# generate results comparable with those in the book, edit
# the ## Not run: line below and comment the previous one.
#
data(belt)
attach(belt.frame)

s <- prelim.cat(x=belt[,-7],counts=belt[,7])

m <- c(1,2,5,6,0,1,2,3,4,0,3,4,5,6,0,1,3,5,0,1,4,6,0,2,4,6)

theta <- em.cat(s,margins=m, # excruciantingly slow; needs 2558
maxits=5000) # iterations.
rngseed(1234)
#
# Now ten multiple imputations of the missing variables B2, I2 are
# generated, by running a chain and taking every 2500th observation.
# Prior hyperparameter is set at 0.5 as in Schafer's p. 329
#
# Impute from parameters obtained from em.cat

imp<- imp.cat(s,theta)
```

3. Methods for categorical mix data: MIX

- Splus (part of library "missing") and R (library "mix") function developed by Schafer (1997)
- Imputation model is a general location model
- Think of this model as a model for contingency table cell probabilities (multinomial) and in each cell, multivariate normality for the variables
- Markov-Chain Monte Carlo method for simulating from this loglinear model

Methods for categorical mix data: MIX (ctd.)

- More description available at

  `http://cran.r-project.org/doc/packages/mix.pdf`

- As the number of categorical variables increases, becomes almost impossible to function
- IveWare should be considered in dealing with large number (>10) of categorical variables with missing values

MIX: R functions

- Similar to CAT:
  - **prelim.mix:** prepares the data by grouping and sorting (by missingness patterns, etc.) to be used by the following functions
  - **em.mix or ecm.mix:** Finds ML estimate or posterior mode of cell probabilities under the saturated model
  - **da.mix:** simulates unknown parameters from the observed-data posterior (or likelihood) under a saturated model. Could be used with the imp.cat() function
  - **imp.mix:** Performs single random imputation of missing values under a value of cell probabilities (em.mix or da.mix)
  - **loglik.mix:** Calculates the observed-data loglikelihood under the general location model at a userspecified parameter value.
  - **mi.inference:** Combines estimates, SEs from m completed-data analyses

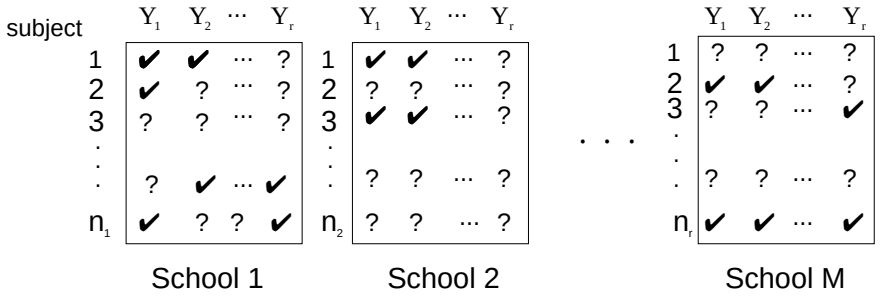
MIX: R functions

Important note on data matrix x:

- The rows of x correspond to observational units, and the columns to variables
- Missing values are denoted by NA
- The categorical variables must be in the first p columns of x, and they must be coded with consecutive positive integers starting with 1.

4. Methods for multilevel incomplete data

Multivariate clustered (longitudinal) data



$\surd$ = **observed value**

? = **missing value**

$Y_1, Y_2, \ldots, Y_r$ = set of r response variables

Data Nature

Important Relationships

- Relationships among variables $Y_1, \ldots, Y_r$ within a unit at each time
- Growth or change in any variable Y_j within an individual across time points
- Relationships between response variables $Y_1, \ldots, Y_r$ and any additional subject level (non-time varying) covariates included in the model
- sampling design producing correlated units

Example 1

ADOLESCENT ALCOHOL PREVENTION TRIAL Longitudinal study of adolescent substance use in Los Angeles area (Hansen and Graham, 1991).

- Cohort of 3,574 children from 45 schools
- Substance-use attitudes and behaviors measured yearly by questionnaire in grades 5-10
- Typical nonresponse due to absenteeism, attrition, etc.
- Some missingness by design

	Percent missing by grade					
	5	6	7	8	9	10
DRINKING	2	24	24	33	35	44
POSCON	47	55	62	100	66	63
NEGCON	48	56	62	100	100	100

Possible imputation method with NORM

- "Unstack" responses and time-varying covariates apply NORM

```
Subj.                        Variables
----   ------------------------------------------------
  1   SEX ALC_5... ALC_10 POS_5...POS_10 NEG_5...NEG_10
  2   SEX ALC_5... ALC_10 POS_5...POS_10 NEG_5...NEG_10
  .                      ...
  .                      ...
  .                      ...
  M   SEX ALC_5... ALC_10 POS_5...POS_10 NEG_5...NEG_10
-------------------------------------------------------
```

- preserves arbitrary covariance structures
- becomes unwieldly for more than few time points and covariates
- requires small number of common time point

Imputing with PAN

- Estimation and imputation routines for two-level multivariate linear mixed model
  - preserves "main effects" of time-varying covariates on time-varying response
  - correctly reflects sampling design or correlated data into imputations
  - therefore, standard errors are not understated

Two-level model

$$y_i = X_i\beta + Z_i b_i + \epsilon_i, \quad i = 1, \ldots, m$$

y_i = $(n_i \times r)$ matrix of responses

CONTAINS ALL THE VARIABLES WITH MISSING VALUES

X_i = $(n_i \times p)$ matrix of covariates

NO MISSING VALUES

Z_i = $(n_i \times q)$ matrix of covariates

β = $(p \times r)$ regression coefficients

$vec(b_i) \sim N_{qr}(0, \psi)$ random effects

$vec(\epsilon_i) \sim N_{n_i r}(0, \Sigma \otimes I_{n_i})$ errors

ψ : unstructured or block diagonal

e.g. r blocks of size $q \times q$

Two-level model: creating imputations

MCMC algorithm (Gibbs sampler)

- Unknown quantities $Y_{mis}, b_i, \theta = (\beta, \Sigma, \Psi)$
- If b_i's are known, this would be just a fixed model drawing θ is easy
- For fixed θ draw b_i
- For fixed b_i, θ, draw Y_{mis}

Prior distributions

$$\begin{aligned} \beta &\sim \text{uniform on } \mathcal{R}^{\sqrt{\nabla}} \text{ (improper)}, \\ \Psi^{-1} &\sim Wishart(\nu_1, \Lambda_1), \nu_1 \geq rq, \\ \Sigma^{-1} &\sim Wishart(\nu_2, \Lambda_2), \nu_2 \geq r, \end{aligned}$$

where

$$\begin{aligned} \nu_1^{-1}\Lambda_1^{-1} &: \text{prior guess for } \Psi, \text{and} \\ \nu_2^{-1}\Lambda_2^{-1} &: \text{prior guess for } \Sigma. \end{aligned}$$

Example

Adolescent alcohol prevention trial

Examined a cohort of $m = 3574$ children and focused on three variables

- DRINKING (Y_1) : composite measure of self-reported alcohol use
- POSCON (Y_2) : perceived positive consequences of use
- NEGCON (Y_3) : perceived negative consequences of use

Jointly model POSCON and NEGCON on DRINKING, using all data from grades 5-10

Example (ctd.)

	Grade					
	5	6	7	8	9	10
DRINKING	−1.43	−1.12	−0.57	0.09	1.29	1.97
	(1.33)	(1.96)	(2.73)	(3.47)	(4.40)	(4.78)
POSCON	1.30	1.34	1.48	—	1.84	1.96
	(0.61)	(0.62)	(0.74)	—	(0.89)	(0.91)
NEGCON	2.94	3.05	3.07	—	—	—
	(0.76)	(0.75)	(0.77)	—	—	—

- Apply log transformation to DRINKING to make the constant variance assumption more plausible
- use a multivariate linear mixed model with intercepts and slopes varying by individual

Assessing convergence

Autocorrelation functions for several elements of θ

- analyzed each imputed dataset using 2-level growth model for DRINKING
- combine 10 sets of coefficients and standard errors using Rubin's (1987) methods

Estimated coefficients, standard errors, degrees of freedom and percent missing information from multiply-imputed growth-curve analysis

	est.	SE	df	% missing
intercept	−1.380	0.052	34	58
grade (1=5th, . . . , 6=10th)	0.139	0.0077	121	28
sex (0=female, 1=male)	0.08	0.028	251	19
sex × grade	−0.013	0.007	94	32
POSCON	1.113	0.035	21	67
NEGCON	−0.245	0.033	46	46

Recent research

- 3-level model (both longitudinal and clustered) is also available as R function (Yucel, 2008)
- R package mlmmm (Yucel, 2008)
- Models with random covariances (Yucel, 2008)
- Extending IveWare to handle clustering will be available as R package called SHRIMP (Yucel, Schenker and Raghunathan, 2007)

R code for pan

```
library(pan)
library(foreign)
data.restore("mglmm.dump")
#
# get missingness rates by grade
for(i in 1:6) print(round(100*mean(is.na(y[occ==i,1])),1))
for(i in 1:6) print(round(100*mean(is.na(y[occ==i,2])),1))
for(i in 1:6) print(round(100*mean(is.na(y[occ==i,3])),1))

par(mfrow=c(4,6))
for(i in 1:6) hist(y[occ==i,1],density=-1)
for(i in 1:6) print(round(mean(y[occ==i,1],na.rm=T),2))
for(i in 1:6) print(round(sqrt(var(y[occ==i&!is.na(y[,1]),1])),2))
#
#
#########################################################################
#  y = Nxr matrix of responses
#  subj = Nx1 vector of subject indicators coded as 1,2,...,m
#  occ = Nx1 vector of occasion (time) indicators coded as 1,2,...,nmax
#  pred = Nxpcol matrix of predictors
#  xcol = px1 vector of integers indicating the columns of pred in X
#  zcol = qx1 vector of integers indicating the columns of pred in Z
#########################################################################
# log transformation to the alc use
```

```
y[!is.na(y[,1]),1]<-log(y[!is.na(y[,1]),1]+2)
prior<-list(a=3,Binv=3*diag(rep(1,3)),c=6,Dinv=6*diag(rep(1,6)))
#
tmp1<-pan(y,subj,pred,xcol,zcol,prior,seed=123456,iter=1000)
tmp2<-pan(y,subj,pred,xcol,zcol,prior,seed=2256,iter=1000,start=tmp1$last)
tmp3<-pan(y,subj,pred,xcol,zcol,prior,seed=8006,iter=1000,start=tmp2$last)
tmp4<-pan(y,subj,pred,xcol,zcol,prior,seed=5456,iter=1000,start=tmp3$last)
tmp5<-pan(y,subj,pred,xcol,zcol,prior,seed=4524,iter=1000,start=tmp4$last)
tmp6<-pan(y,subj,pred,xcol,zcol,prior,seed=5602,iter=1000,start=tmp5$last)
tmp7<-pan(y,subj,pred,xcol,zcol,prior,seed=7820,iter=1000,start=tmp6$last)
tmp8<-pan(y,subj,pred,xcol,zcol,prior,seed=9824,iter=1000,start=tmp7$last)
tmp9<-pan(y,subj,pred,xcol,zcol,prior,seed=4353,iter=1000,start=tmp8$last)
tmp10<-pan(y,subj,pred,xcol,zcol,prior,seed=5363,iter=1000,start=tmp9$last)
tmp11<-pan(y,subj,pred,xcol,zcol,prior,seed=444,iter=1000,start=tmp10$last)

# check the convergence
psi<-array(0,c(6,6,11000))
psi[,,   1:1000]<-tmp1$psi
psi[,,1001:2000]<-tmp2$psi
psi[,,2001:3000]<-tmp3$psi
psi[,,3001:4000]<-tmp4$psi
psi[,,4001:5000]<-tmp5$psi
psi[,,5001:6000]<-tmp6$psi
psi[,,6001:7000]<-tmp7$psi
psi[,,7001:8000]<-tmp8$psi
psi[,,8001:9000]<-tmp9$psi
psi[,,9001:10000]<-tmp10$psi
psi[,,10001:11000]<-tmp11$psi
```

```
sigma<-array(0,c(3,3,11000))
sigma[,,   1:1000]<-tmp1$sigma
sigma[,,1001:2000]<-tmp2$sigma
sigma[,,2001:3000]<-tmp3$sigma
sigma[,,3001:4000]<-tmp4$sigma
sigma[,,4001:5000]<-tmp5$sigma
sigma[,,5001:6000]<-tmp6$sigma
sigma[,,6001:7000]<-tmp7$sigma
sigma[,,7001:8000]<-tmp8$sigma
sigma[,,8001:9000]<-tmp9$sigma
sigma[,,9001:10000]<-tmp10$sigma
sigma[,,10001:11000]<-tmp11$sigma

beta<-array(0,c(4,3,11000))
beta[,,   1:1000]<-tmp1$beta
beta[,,1001:2000]<-tmp2$beta
beta[,,2001:3000]<-tmp3$beta
beta[,,3001:4000]<-tmp4$beta
beta[,,4001:5000]<-tmp5$beta
beta[,,5001:6000]<-tmp6$beta
beta[,,6001:7000]<-tmp7$beta
beta[,,7001:8000]<-tmp8$beta
beta[,,8001:9000]<-tmp9$beta
beta[,,9001:10000]<-tmp10$beta
beta[,,10001:11000]<-tmp11$beta
#
par(mfrow=c(6,6))
for(i in 1:6){ for(j in 1:6)
  plot(1:3000,psi[i,j,1:3000],type="l")}
```

```
for(i in 1:6){ for(j in 1:6)
  acf(psi[i,j,1001:11000],lag.max=200)}
#
sex<-matrix(pred[,3],ncol=6,byrow=T)[,1]

reshape<-function(y){
newy<-t(y)
dim(newy)<-c(18,3574)
t(newy)}
orig<-cbind(sex,reshape(y))
#
imp1<-cbind(sex,reshape(tmp2$y))
imp2<-cbind(sex,reshape(tmp3$y))
imp3<-cbind(sex,reshape(tmp4$y))
imp4<-cbind(sex,reshape(tmp5$y))
imp5<-cbind(sex,reshape(tmp6$y))
imp6<-cbind(sex,reshape(tmp7$y))
imp7<-cbind(sex,reshape(tmp8$y))
imp8<-cbind(sex,reshape(tmp9$y))
imp9<-cbind(sex,reshape(tmp10$y))
imp10<-cbind(sex,reshape(tmp11$y))
# roundoff is a function for rounding the cat. variables imputed under normal
roundoff<-function(imp){
imp[,c(2,5,8,11,14,17)]<-round(imp[,c(2,5,8,11,14,17)],5)
imp[,c(3,4,6,7,9,10,12,13,15,16,18,19)]<-
round(imp[,c(3,4,6,7,9,10,12,13,15,16,18,19)])
        for(i in c(3,4,6,7,9,10,12,13,15,16,18,19)){
imp[imp[,i]<1,i]<-1
imp[imp[,i]>4,i]<-4}
```

```
imp}
imp1<-roundoff(imp1)
imp2<-roundoff(imp2)
imp3<-roundoff(imp3)
imp4<-roundoff(imp4)
imp5<-roundoff(imp5)
imp6<-roundoff(imp6)
imp7<-roundoff(imp7)
imp8<-roundoff(imp8)
imp9<-roundoff(imp9)
imp10<-roundoff(imp10)
# the following is a function for outputting data
output<-function(imp,filename){
sink(filename)
cat(paste(format(imp[,1]),format(imp[,2]),format(imp[,3]),
format(imp[,4]),format(imp[,5]),format(imp[,6]),
format(imp[,7]),format(imp[,8]),format(imp[,9]),
format(imp[,10]),format(imp[,11]),format(imp[,12]),
format(imp[,13]),format(imp[,14]),format(imp[,15]),
format(imp[,16]),format(imp[,17]),format(imp[,18]),
format(imp[,19])),sep="\n")
sink()
invisible()}
output(imp1,"imp1.dat")
output(imp2,"imp2.dat")
output(imp3,"imp3.dat")
output(imp4,"imp4.dat")
output(imp5,"imp5.dat")
output(imp6,"imp6.dat")
```

```
output(imp7,"imp7.dat")
output(imp8,"imp8.dat")
output(imp9,"imp9.dat")
output(imp10,"imp10.dat")
#
par(mfrow=c(4,6))
hist(orig[,2],density=-1)
hist(orig[,5],density=-1)
hist(orig[,8],density=-1)
hist(orig[,11],density=-1)
hist(orig[,14],density=-1)
hist(orig[,17],density=-1)

hist(orig[,3],density=-1)
hist(orig[,6],density=-1)
hist(orig[,9],density=-1)
frame();frame()
hist(orig[,15],density=-1)
hist(orig[,18],density=-1)

hist(orig[,4],density=-1)
hist(orig[,7],density=-1)
hist(orig[,10],density=-1)

par(mfrow=c(4,6))
hist(imp1[,2],density=-1)
hist(imp1[,5],density=-1)
hist(imp1[,8],density=-1)
hist(imp1[,11],density=-1)
```

```
hist(imp1[,14],density=-1)
hist(imp1[,17],density=-1)

hist(imp1[,3],density=-1)
hist(imp1[,6],density=-1)
hist(imp1[,9],density=-1)
hist(imp1[,12],density=-1)
hist(imp1[,15],density=-1)
hist(imp1[,18],density=-1)

hist(imp1[,4],density=-1)
hist(imp1[,7],density=-1)
hist(imp1[,10],density=-1)
hist(imp1[,13],density=-1)
hist(imp1[,16],density=-1)
hist(imp1[,19],density=-1)
#############################################################################
# now analyze the imputed datasets
library(lmm)
result<-as.list(1:10)
for(i in 1:10){
   fna<-paste("imp",format(i),".dat",sep="")
   tmp<-matrix(scan(fna),ncol=19,byrow=T)
   alc<-tmp[,c(2,5,8,11,14,17)]
   m<-nrow(tmp)
   poscon<-tmp[,c(3,6,9,12,15,18)]
   negcon<-tmp[,c(4,7,10,13,16,19)]
   y<-as.vector(t(alc))
   poscon<-as.vector(t(poscon))
```

```
   negcon<-as.vector(t(negcon))
   #
   occ<-matrix(1:6,nrow(alc),ncol(alc),byrow=T)
   sex<-cbind(tmp[,1],tmp[,1],tmp[,1],tmp[,1],tmp[,1],tmp[,1])
   subj<-1:nrow(alc)
   subj<-cbind(subj,subj,subj,subj,subj,subj)
   occ<-as.vector(t(occ))
   subj<-as.vector(t(subj))
   sex<-as.vector(t(sex))
   pred<-cbind(int=1,time=occ,sex=sex,time.sex=occ*sex,poscon=poscon,
      negcon=negcon)
   xcol<-1:6
   zcol<-1:2
   #
   result[[i]]<-ecme(y,subj,occ,pred,xcol,zcol,method=3)}
#
est<-as.list(1:10)
SE<-as.list(1:10)
for(i in 1:10){
est[[i]]<-result[[i]]$beta
SE[[i]]<-sqrt(diag(result[[i]]$cov.beta))}
res<-mi.inference(est,SE)
res<-cbind(est=round(res$est,4),SE=round(res$std.err,4),
   df=round(res$df),pval=round(res$signif,3),
   pctminf=round(100*res$fminf))
```