Practical 12:
Missing data II

Practical 2: An introduction to multiple imputation for clustered data

Objectives

The objectives are that, at the end of this practical, you should understand

1. how to do multiple imputation in a multilevel/clustered data setting.

1 Multiple imputation with multi-level/clustered data

In this practical we will use simulated data which is motivated by the class size dataset introduced earlier. The data contain (simulated) observations of standardized scores of pre-reception year maths and literacy scores and post-reception literacy scores. The data have a two-level hierarchical structure, with children clustered within schools.

The dataset is called `prac2data.dta`. Open the dataset using the File/Open menu or using the command line (assuming you have changed Stata's working directory to the appropriate directory): `use prac2data.dta, clear`

1.1 Summarising the missing data

To summarize the number of missing values per variable, type:

```
misstable summarize nlitpre nlitpost nmatpre
```

As the command shows, there is only missing data in the nmatpre variable. These missing values have been simulated completely at random (MCAR).

2 Complete case analyses

Since we know that the data are MCAR, we can obtain unbiased estimates from complete case analyses. In this practical we will focus on two models of interest.

2.1 Random-effects model for nmatpre

We first fit a model for the nmatpre variable to investigate whether children from the same school have more similar scores to each other than children from different schools. To do this, we fit a simple linear mixed or random-effects model (also sometimes called a variance components model). This is a linear model with a random school effect, and is often called the random-intercepts model. We fit the model using Stata xtmixed command:
`xtmixed nmatpre || schoolid:`

2.2 Mixed-effects model for nlitpost

Now, our model of interest we fit a regression model for nlitpost with `nlitpre` and `nmatpre` as covariates. Since the observations are clustered within school, we again include a random-school intercept, using the `xtmixed` command:
`xtmixed nlitpost nlitpre nmatpre || schoolid:`

What is your interpretation of the estimated effects for `nlitpre` and `nmatpre` on `nlitpost`? Compare the estimate for `sd( cons)` with the one from the earlier model - would you expect them to be similar?

3 Multiple imputation ignoring clustering

Now we impute the missing nmatpre values using an imputation model which completely ignores the clustering. We first use the `mi set` and `mi register` commands:

```
mi set wide
mi register imputed nmatpre
mi impute regress nmatpre nlitpre nlitpost, add(10) rseed(4231)
```

We now fit our two models of interest to the imputed data using mi estimate: First we fit the random-intercepts model to `nmatpre`:

```
mi estimate: xtmixed nmatpre || schoolid:
```

Write your estimates and compare with the complete case estimates. Are there any substantive differences, and if so, do they make sense, given the way in which you have imputed the missing nmatpre values?

We now fit the second of our models of interest:

```
mi estimate: xtmixed nlitpost nlitpre nmatpre || schoolid:
```

Write your estimates into Table below and compare the estimates with the complete case estimates. You will need another similar table for the Variance component model.

	Complete cases	MI ignoring clustering	MI fixed cluster	Multi-level MI
Fixed intercept				
`nlitpre`				
`nmatpre`				
Between-school SD				
Within-school SD				

Table 1: Parameter estimates (standard errors) from fitting the observed model to the complete cases and multiple imputation models

4 Including the cluster variable as a fixed effect

We now try imputing using Stata's mi commands but now including the school id variable as a fixed effect:

```
*including cluster as fixed effect
use prac4data, clear
mi set wide
mi register imputed nmatpre
mi impute regress nmatpre nlitpre nlitpost i.schoolid, add(10) rseed(4231) noisily
mi estimate: xtmixed nmatpre || schoolid:
mi estimate: xtmixed nlitpost nlitpre nmatpre || schoolid:
```

Again, write your estimates into the corresponding Tables, and compare the estimates. Can you suggest reasons for any substantive differences or lack of differences? Why might using fixed effects for the cluster variable school be a bad idea?

5 Multi-level multiple imputation using jomo

Lastly we will use the jomo package to impute the missing nmatpre values using an appropriate multi-level (random-effects) imputation model.

No prior R knowledge is required for this practical.

5.1 Importing data in R

To begin with, launch `RStudio` from the command window. `RStudio` is simply an editor for R.

First, change your working directory. To do this, just go to `Session -> Change Working Directory -> Choose Directory` and select the folder where you have the data.

Next, open a new R script. To do this, simply go to `File -> New file -> R script`. Then, install `jomo` and `foreign`. The last one is a package that is very useful to upload in R data that are saved in `SAS` or `Stata` format. To do this, copy in your new script the following code:

```
install.packages("foreign")
install.packages("jomo")
library(foreign)
library(jomo)
```

Then select all and either click on the `Run` button at the top right corner of the script window, or just type `ctrl+R`. You then need to upload your data:

```
data<-read.dta("prac2data.dta")
attach(data)
```

5.2 Performing the imputations

Then, choose which variables will be imputed (Y) and which will be used only as covariates (X) in the imputation model, being fully observed.

```
Y<-data.frame(nmatpre)
intercept<-rep(1,length(nlitpre))
X<-data.frame(intercept,nlitpost,nlitpre)
Z<-data.frame(intercept)
```

Run imputation process with `jomo` and save the data in `Stata` format:

```
imput<-jomo(Y=Y,X=X,Z=Z,clus=schoolid)
colnames(imput)[6]<-"schoolid"
write.dta(imput,"Imputations.dta")
```

5.3 Importing into Stata and fitting models of interest

Now go back to `Stata` and simply run these commands to load the imputed datasets:

```
use Imputations, clear
mi import flong, m(Imputation) id(id) imputed(nlitpre)
```

Finally, fit the substantive model and apply Rubin's rules to get our final MI estimates:

```
mi estimate: xtmixed nmatpre || schoolid:
mi estimate: xtmixed nlitpost nlitpre nmatpre || schoolid:
```

How do the parameter estimates compare with the complete case estimates and those based on a single-level imputation model? How do the standard errors compare with those from the complete case analyses?