# Practical 11:
# Missing data I

# Practical 1: Impact of missing data, missingness mechanisms, and intro to multiple imputation

## Objectives

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

1. appreciate the impact missing data can have, in terms of both bias and efficiency (precision) of estimates
2. know how to investigate missingness in a variable
3. do simple multiple imputation in Stata for a single partially observed variable

## 1 Missingness in a binary variable - the 'class size' study

For the first exercise, we use the 'class size' data set. These are derived from the class size study carried out by Peter Blatchford and colleagues at the Institute of Education (London) and kindly made available to us.

Load the file `classp3.dta`:

```
use classp3, clear
```

The variables in the `classp3` data set should now appear in the `variables` window. A description of the variables is given in Table 1. We will come to the variable `sen_m` below.

You can also type

```
describe
* and/or
codebook
```

to get a fuller description of the data set and each of the variables.

Note that the test scores have been normalised. This was done as follows. For each test, the pupils' results were ranked. Then for observation in rank order $i$, where $n$ pupils sat the test, the normalised result was calculated as the inverse normal of $i/(n+1)$. Since many pupils got the same marks there are ties in the data.

| Variable name | Details |
|---|---|
| `uniqueid` | Unique pupil identifier |
| `nmatpre` | Pre-reception maths score |
| `nlitpost` | Post-reception literacy score |
| `sen` | Special educational needs (1=yes, 0=no) |
| `sen_m` | Special educational needs variable, with a number of missing values |

Table 1: Variables in the class size data set

We are now ready to carry out some analyses. We will look at the effect of `nmatpre` on `nlitpost`, adjusting for `sen`. First we will analyse all the data using the complete `sen` variable; then we will use the `sen_m` variable which has missing data and do a complete case analysis. Then we will use the missing indicator method. Lastly we will use multiple imputation. As you go along, complete Table 2.

### *1.1 Full data analysis*

To fit the model of interest to the full data (i.e. without missingness), proceed as follows:

```
* full data analysis
regress nlitpost nmatpre sen
```

| Variable | Original data Coeff. (S.E.) | Complete case analysis Coeff. (S.E.) | Multiple imputation Coeff. (S.E.) |
|---|---|---|---|
| Pre reception numeracy (`nmatpre`) | 0.58 (0.012) | | |
| Special educational needs (`sen`) | −0.432 (0.043) | | |

Table 2: Class size data: results of (i) original data analysis; (ii) complete case analysis, and (iv) multiple imputation

### *1.2 Complete case analysis*

Now look at the `sen_m` variable where there is missing data, and carry out a complete case analysis using:

```
* complete case analysis
regress nlitpost nmatpre sen_m
```

Note Stata's default is a complete case analysis. How many observations have missing data in this variable? How do the complete case analysis results compared to the full data results, both in terms of bias, and precision? Under what assumptions is this complete case analysis unbiased?

### *1.3 Investigating missingness in* `sen_m`

To understand the complete case results we will investigate the missingness in `sen_m` variable. One way of doing this is to generate a binary indicator of missingness (or observation) and fit a logistic regression model:

```
gen r=(sen_m!=.)
logistic r nlitpost nmatpre
```

What can you conclude from these results regarding the missingness mechanism of `sen_m`? We now try adding `sen_m` to the logistic regression:

```
logistic r nlitpost nmatpre sen_m
```

Why is Stata unable to fit this model? Instead we cheat and use the `sen` variable:

```
logistic r nlitpost nmatpre sen
```

What can you now conclude about the missingness mechanism for `sen_m`? In practice (i.e. with only `sen_m`) what could you conclude about the missingness mechanism?

We will continue with the school test score data that we used earlier. Recall that the `sen_m` variable contains missing values.

## 2 Multiple Imputation

### *2.1 Preliminaries*

First, re-load the dataset `classp3.dta` we were using earlier. Before we can do any imputation, we must first tell Stata how we want the multiple imputations to be stored (we choose the wide form here):

```
mi set wide
```

Next we need to register variables. At a minimum, we must register any variables which we want to go on to impute. Here this is just the `sen_m` variable:

```
mi register imputed sen_m
```

You can also register so called 'passive' and 'regular' variables. A passive variable is a variable which will vary from one imputation to the next, and whose value is a function of imputed variables or other passive variables. Regular variables are variables that will have the same value as in each imputation. Thus far I have never encountered a situation where I've needed to register regular variables, although Stata recommend you do, as Stata can then spot certain errors and prevent them (e.g. if you mistakenly modified a regular variable in the imputed datasets).

## 3 Generating the imputations

In the lecture we looked at how imputation works for a normal linear regression model. Since `sen_m` is binary, we will use a logistic regression imputation model rather than a linear regression model. The steps taken are conceptually the same, with slight modifications due to the fact we are using a logistic model. The steps are (which Stata will do for us!):

1. Fit the logistic regression imputation model to the complete cases, obtaining estimates and SEs of the log odds ratios.
2. For each $m = 1, .., M$, draw a new set of log ORs, based on a normal approximation to the parameters' posterior distribution (MI was originally derived from a Bayesian perspective).
3. For imputation $m = 1, .., M$:
   (a) For each subject with a missing value of `sen_m`, calculate their predicted probability of having special education needs, based on the logistic model and parameters drawn in the previous step.
   (b) For each such subject, randomly impute 0/1 from a Bernoulli (binary) distribution, with probability of 'success' as calculated in the previous step.

```
mi impute logit sen_m nlitpost nmatpre, add(10) rseed(4921)
```

This command asks Stata to impute `sen_m` using a logistic regression imputation model, with `nlitpost` and `nmatpre` as covariates. The `add(10)` option asks Stata to add 10 imputations to our dataset. The `rseed` option sets Stata's random number seed. If you set the seed to 4921 as shown, you should get exactly the same results as given in the solution sheet. Browse the dataset to examine the new variables which have been added and check that you understand what they are. The way Stata has stored/structured the dataset is a consequence of our choice of `wide` when we `mi set` the data.

## 4 Fitting models to the imputations

Lastly we fit our model of interest to the 10 imputations using `mi estimate`:

```
mi estimate: regress nlitpost nmatpre sen_m
```

The `mi estimate` command fits our specified model to each of the 10 imputations, collects the parameter estimates and standard errors, and combines these using Rubin's rules. The `mi estimate` command can be used with many of Stata's estimation commands - for more information type `help mi estimate`.

Write the MI estimates you have obtained into the Table from Practical 1. How do they compare with those based on the full data?

## 5 Conclusions

We have seen that missing data can have serious adverse consequences, both from a bias perspective and in terms of the precision of our estimates (standard errors and confidence intervals). To tackle the problem, we must first understand where data are missing in our study, and use both the data and our contextual knowledge to guide us as to what are plausible assumptions regarding missingness. With missingness in only a single variable, logistic regression can be used to investigate how missingness depends on the other fully observed variables in the dataset. Alternatively, we can compare the distributions of these variables between those with $R = 1$ and those with $R = 0$.

For the data here, we saw that the complete case analysis was biased, but we were only able to draw this conclusion by comparing its results to the 'full data' results. In practice, we do not have the 'full data' results to compare to!

We have seen that imputing a single partially observed variable, with all the others fully observed, is relatively easy to do in Stata. We have focused on the mechanics of how it works. When using MI in applied projects things are rarely as simple as the situation/dataset we have used here. The difficulties lie in carefully choosing imputation models so that they are reasonable for the data at hand, and carefully exploring and thinking about the MAR assumption.