Methods for addressing missing data in health economic evaluations

XXXIX Spanish Health Economics Meeting

Pre-conference worshop: Practicals

11 June 2019

Make sure you have installed the latest version of `R` for your OS, which can be downloaded at `https://www.r-project.org/`. We recommend that you have installed the latest version of `Rstudio`, which provides an user-friendly interface with `R`, available at `https://www.rstudio.com/products/rstudio/download/`. You also need to download and install the latest version of `JAGS` from `https://sourceforge.net/projects/mcmc-jags/files/JAGS/4.x/`.

Exercise 1: Install missingHE and inspect the data

a Open `Rstudio` and install **missingHE** using the command `install.packages`. Then, use the command `library` to load the package. It is good practice to save your `R` script before running it as this allows to modify and re-run your code in case you want to change something. To create a new `R` script in `Rstudio`, select file tab → new file → `R` script. Now you can write, change and run your `R` code directly from your script file. Once you are done, you can name and save the script file for later use.

b Once **missingHE** has been loaded, the data should be available in the object `MenSS`, which you can directly access by typing `MenSS` in your command line. This is a pilot RCT on 159 young men at risk of sexually transmitted infections (STIs), who are assigned to either the control ($t = 1$) or the intervention ($t = 2$) group. The dataset contains the main variables used in the health economic analysis, which include individual-level QALYs and total costs (e_i, c_i), baseline utilities (u_{i0}) and other covariates. A general overview of the types of variables included in this dataset can be obtained using the command `str`. More information about this dataset can be accessed by typing `help(MenSS)`.

c Inspect the data with the command `summary` and check which variables are partially-observed and the proportions of missingness. Next, check whether the outcome variables present the typical complexities of HTA data. For example you can type `hist(Menss$e)` to look at the empirical distribution of the QALYs across both treatment groups. If you are intersted in the data for a single treatment group, for example the control, you can type `hist(MenSS$e[MenSS$t==1])`.

Exercise 2: Selection models

a As a startgin point, fit a bivariate Normal for costs and effects using the `selection` command from **missingHE**. This function has arguments which allow to fit selection models under different choices for the distributions of the outcomes and alternative assumptions about their missingness mechanisms. For example, the following command

```
> NN.sel=selection(data = MenSS, model.eff = e ~ u.0, model.cost = c ~ e,
+  model.me = me ~ 1, model.mc = mc ~ 1, type = "MAR", n.chains = 2,
+  n.iter = 10000, n.burnin = 1000, dist_e = "norm", dist_c = "norm",
+  save_model = TRUE)
```

runs a selection model in `JAGS`, which is saved in the object `NN.sel`. The model specifies a marginal Normal distribution for the effects (controlling for the baseline utilities) and a conditional Normal distribution for the costs (given the effects) – which effectively corresponds to a bivariate Normal model for (e, c). The model assumes a MAR (specifically MCAR) assumption on both missing data mechanisms and, by default, specifies vague priors on all parameters. The interpretation of the different arguments of the function is the following:

- `data`. The dataframe containing the variables for the economic analysis.
- `model.eff` and `model.cost`. The models for the effect and cost variables, respectively. These are provided using formulae expressions in conventional `R` linear modelling syntax, where the outcome and the covariates are placed to the left and right hand side of the symbol $\sim$, respectively. When there are multiple covariates, these can be additively placed to the right hand side of the formulae using the symbol `+`, while if there are no covariates, `1` should be used instead.
- `model.me` and `model.mc`. The models for the missing data indicators for the effect and cost variables, respectively. By default, these are modelled using Bernoulli distributions, where the missingness probabilities are linked to other variables using logistic-linear regressions. As for `model.eff` and `model.cost`, it is possible to include covariates in both formulae.
- `type`. The assumption about the missingness mechanisms, either `"MAR"` or `"MNAR"`. The latter can only be selected when the missingness probability for at least one outcome depends on the same variable (e.g. `model.me=me` $\sim$ `e`).
- `n.chains`, `n.iter`, `n.burnin`. Parameters related to the number of chains, iterations and burnin period to specify for the MCMC algorithm implemented by `JAGS`.
- `dist_e` and `dist_c`. Distributions selected for the effects and costs, respectively.
- `save_model`. Logical argument to specify whether the `JAGS` model file should be saved or not in the current working directory. More details about all the arguments of `selection` can be seen by typing `help(selection)`.

b By setting the argument `save_model=TRUE`, the file of the `JAGS` model (named `selection.txt`) which has been written and run by the previous command is saved in your current working directory. Following the script in this file, without paying too much attention to the parts which have not been introduced in the course, make sure you can follow the code and match it with the slides from the lecture.

c Inspect the posterior distributions of the parameters of the model using the `print` function. The key quantities of interest for the economic analysis are the mean effects and costs in each treatment group (μ_{et}, μ_{ct}).

d Assess model convergence using different types of diagnostic plots, which can be obtained using the `diagnostic` function. For example, posterior density plots for the mean effects in both groups can be obtained by typing

```
> diagnostic(NN.sel, type = "denplot", param = "mu.e")
```

where, the first argument is the output of the `selection` function, while `type` and `param` indicate the type of diagnostic plot and family of parameters to display. The names for different types of diagnostics and families of parameters that can be selected can be seen by typing `help(diagnostic)`.

e Check the distribution of the imputed values for effects and costs in each group using the command `plot`.

```
> imp.NN=plot(NN.sel)
```

The object `imp.NN` is a list which contains the observed and imputed data for both effects and costs in each group. For example, an histogram of the imputed effects in the control arm can be obtained by typing `hist(imp.NN$‘imputed data‘$effects1)`

f Summarise the economic results from the model using the `summary` function. State your conclusions about the cost-effectiveness of the new intervention. To obtain graphical outputs to assess cost-effectiveness, you can use some of the functions from the **BCEA** package (which should be loaded first). Cost-effectiveness plane and cost-effectiveness acceptability curve plots can then be obtained by typing `ceplane.plot(NN.sel$cea)` and `ceac.plot(NN.sel$cea)`, respectively.

g Now, try to replicate the analysis by varying the assumptions of the model and compare the results across the alternative specifications. Possible choices to consider are:

- Add/remove covariates to/from the model of the effects and costs (`model.e` and `model.c`) using the covariates that are available in the MenSS dataset.
- Add/remove covariates to/from the model of the missing data indicators for the effects and costs (`model.me` and `model.mc`) using the variables that are available in the MenSS dataset.
- Explore MNAR assumptions for the effects and/or costs by setting `type="MNAR"` and including `e` and/or `c` into the formulae `model.me` and/or `model.mc`, respectively.
- Change the distributions of the effects and costs. Alternative choices are `"beta"` for Beta distributions (effects) and `"gamma"` or `"lnorm"` for Gamma and LogNormal distributions (costs). Note that values of ones and zeros are not allowed when using Beta and Gamma/LogNormal distributions, respectively. Thus, a small constant, e.g. 0.05, should be subtracted/added to the effect/cost data to be able to fit these distributions.

Exercise 3: Pattern mixture models

a As an alternative approach to handle missing data, now fit a bivariate Normal for costs and effects using the `pattern` command from **missingHE**. This function has arguments which allow to fit pattern mixture models under different choices for the distributions of the outcomes and alternative assumptions about their missingness mechanisms. For example, the following command

```
> NN.pat=pattern(data = MenSS, model.eff = e~u.0, model.cost = c~e,
+  type = "MAR", n.chains = 2, n.iter = 10000, n.burnin = 1000,
+  dist_e = "norm", dist_c = "norm", Delta_e = 0, Delta_c = 0,
+  save_model = TRUE)
```

runs a pattern mixture model in `JAGS`, which is saved in the object `NN.pat`. The model is a bivariate Normal under MAR, and is similar to the one fitted using the `selection` function. By default, the model specifies vague priors on all parameters and identify the distributions of the missing data using the parameters estimated from the completers (CC restriction). In `pattern`, the arguments `model.me` and `model.mc` from `selection` are replaced with `Delta_e` and `Delta_c`, which denote the sensitivity parameters used to identify the model. Under MAR, both values should be set to 0, while under MNAR different values can be used. More details about all the arguments of `pattern` can be seen by typing `help(pattern)`

b The argument `save_model=TRUE` saves the file of the `JAGS` model (named `pattern.txt`) in your current working directory. Following the script in this file, without paying too much attention to the parts which have not been introduced in the course, make sure you can follow the code and match it with the slides from the lecture.

c Inspect the posterior distributions of the key parameters of interest of the model using the `print` function.

d Assess model convergence and the distributions of the imputed data using the `diagnostic` and `plot` functions.

e Summarise the economic results from the model. State your conclusions about the cost-effectiveness of the new intervention.

f Now, try to replicate the analysis by varying the assumptions of the model and compare the results across the alternative specifications. As for `selection`, if Beta and/or Gamma/LogNormal distributions are specified for the effects and costs, it is necessary to subtract/add a small constant to the data to fit these distributions.

To fit the model under MNAR, the argument `type` must be set to `"MNAR"` and specific values for the lower and upper bounds of the distributions of the sensitivity parameters for the effects and/or costs in both treatment groups must be provided. Specifically, under MNAR, `pattern` assumes Uniform distributions for Δ_e and Δ_c, whose hyperprior values must be provided by the user. For example, assuming that the partially-observed individuals are associated with an average decrease in the QALYs between 0.1 and 0.2 with respect to the completers, we can include this information into the model by creating the 2×2 matrix

```
> prior.Delta.e=matrix(NA, nrow =  2, ncol = 2)
> prior.Delta.e[,1]=c(-0.2, -0.2)
> prior.Delta.e[,2]=c(-0.1, -0.1)
```

where the rows and columns represent the treatment group and range of values, respectively. The object `prior.Delta.e` can then be passed to the argument `Delta_e` in the `pattern` function.

Exercise 4: Hurdle models

a Although hurdle models are not, technically speaking, missingness models, they allow to explore the impact on conlcusions of alternative assumptions about the proportions of individuals who can be potentially associated with a structural value. This is particularly useful in HTA, where structural values typically occur in both outcomes (e.g. one for QALYs and zero for costs). You can use the function `hurdle` in **missingHE** to fit hurdle models to HTA data. The following code fits a bivariate Normal model using an hurdle approach to handle structural ones and zeros in the effects and costs.

```
> NN.hur=hurdle(data = MenSS, model.eff = e ~ u.0, model.cost = c ~ e,
+  model.se = se ~ 1, model.sc = sc ~ 1, type = "SCAR", se = 1, sc = 0,
+  n.chains = 2, n.iter = 10000, n.burnin = 1000,
+  dist_e = "norm", dist_c = "norm", save_model = TRUE)
```

The `hurdle` function has similar arguments to the `selection` function, with only few exceptions. There are three main differences:

- The formulae for the models of the missing data indicators (`model.me` and `model.mc`) are replaced with two formulae for the models of the structural data indicators (`model.se` and `model.sc`). The values of these indicators may be equal to 1 (structural value), 0 (non-structural value) or missing (unobserved value).
- The argument `type` is now related to the assumptions about the **structural value mechanism** and can be set to either `"SCAR"` (structural completely at random) or `SAR` (structural at random). These are different assumptions compared with those about the missingness mechanisms as it is possible to have structural values even when there are no missing data.
- There are two additional arguments: `se` and `sc`. They respectively indicate the value in the effect and cost data that should be treated as structural by the model. If there are no structural values for one of the two outcomes, it is possible to set either `se=NULL` or `sc=NULL` to indicate that the hurdle approach should only be used for the other outcome. More information about the arguments of the function `hurdle` can be accessed by typing `help(hurdle)`.

b The argument `save_model=TRUE` saves the file of the `JAGS` model (named `hurdle.txt`) in your current working directory. Following the script in this file, without paying too much attention to the parts which have not been introduced in the course, make sure you can follow the code and match it with the slides from the lecture.

c Inspect the posterior distributions of the key parameters of interest of the model using the `print` function.

d Assess model convergence and the distributions of the imputed data using the `diagnostic` and `plot` functions.

e Summarise the economic results from the model. State your conclusions about the cost-effectiveness of the new intervention.

f Now, try to replicate the analysis by varying the assumptions of the model and compare the results across the alternative specifications. In contrast to both `selection` and `pattern`, when there are structural values in the outcomes, the function `hurdle` allows to fit Beta and Gamma/LogNormal distributions to the effects and costs without the need to rescale the data.

It is possible to fit the model under MNAR by providing the vectors of the structural value indicators to `hurdle`. For example, let us assume that we want to assess the impact on the results under the assumption that all the individuals with a unit baseline utility are also associated with a unit QALYs (i.e. structural ones). We can generate the desired structural value indicators using the `ifelse` function.

```
> d_e=ifelse(MenSS$e==1, 1, 0)
> d_e[MenSS$u.0==1 & is.na(MenSS$e)]=1
```

The first line creates the variable `d_e`, taking value one, zero and missing for each individual when $e_i = 1$, $e_i < 1$ and $e_i = \text{NA}$, respectively. The second line, replaces the values of `d_e` with 1 when $u_{i0} = 1$ and e_i is missing. We can then pass this indicator variable to the optional argument `d_e` in the `hurdle` function and fit the model under MNAR.