EP16: Missing Values in Clinical Research: Multiple Imputation

8. Analysis of Imputed Data, Pooling & More

Nicole S. Erler

Department of Biostatistics, Erasmus Medical Center

✉ n.erler@erasmusmc.nl

Analysing imputed data

Once we have confirmed that our imputation was successful, we can move on to the **analysis of the imputed data**.

For example, we might be interested in the following logistic regression model:

```r
glm(DM ~ age + gender + hypchol + BMI + smoke + alc,
    family = "binomial")
```

Analysing imputed data

Once we have confirmed that our imputation was successful, we can move on to the **analysis of the imputed data**.

For example, we might be interested in the following logistic regression model:

```
glm(DM ~ age + gender + hypchol + BMI + smoke + alc,
    family = "binomial")
```

To fit the model on each of the imputed datasets, we do not need to extract the data from the `mids` object, but can use `with()`.

```
mod1 <- with(imp4, glm(DM ~ age + gender + hypchol + BMI + smoke + alc,
                       family = "binomial"))
```

`mod1` is an object of class `mira`.

Pooling the Results

Pooled results can be obtained using `pool()` and its summary.

```r
res1 <- summary(pool(mod1), conf.int = TRUE)
# round(res1, 3)
```

Pooling the Results

Pooling with `mice::pool()` is available for most types of models.

It extracts the model coefficients and variance-covariance matrices using `tidy()` from the package **broom**. Hence, pooling using the `pool()` function from **mice** only works for models of classes for which a method `tidy()` exists.

An alternative is offered by the package **mitools** and the function `MIcombine()`.

Functions for Pooled Results

mice currently has two functions available for evaluating model fit / model comparison

For **linear** regression models the pooled R^2 can be calculated using `pool.r.squared()`.

```
mod2 <- with(imp4, lm(SBP ~ DM + age + hypten))
pool.r.squared(mod2, adjusted = TRUE)
```

```
##                est     lo 95     hi 95 fmi
## adj R^2 0.3265363 0.2957747 0.3573434 NaN
```

The argument `adjusted` specifies whether the adjusted R^2 or the standard R^2 is returned.

Functions for Pooled Results

The function `pool.compare()` allows comparison of **nested models** (i.e., models where one is a special case of the other, with some parameters fixed to zero) using a **Wald test**.

Example: To test if `smoke` has a relevant contribution to the model for `DM` from above we re-fit the model without `smoke` and compare the two models:

```r
mod3 <- with(imp4, glm(DM ~ age + gender + hypchol + BMI + alc,
                       family = "binomial"))
# Wald test
pool.compare(mod1, mod3)$pvalue
```

```
## Warning: 'pool.compare' is deprecated.
## Use 'D1' instead.
## See help("Deprecated")

##           [,1]
## [1,] 0.8179913
```

Functions for Pooled Results

The package **miceadds** extends **mice**, for example with the following functionality:

Combine χ^2 or F statistics from multiply imputed data:

```
miceadds::micombine.chisquare(dk, df, ...)
miceadds::micombine.F(values, df1, ...)
```

These functions take vectors of statistics computed on each imputed dataset and pool them.

Functions for Pooled Results

The package **miceadds** extends **mice**, for example with the following functionality:

Combine χ^2 or F statistics from multiply imputed data:

```
miceadds::micombine.chisquare(dk, df, ...)
miceadds::micombine.F(values, df1, ...)
```

These functions take vectors of statistics computed on each imputed dataset and pool them.

Calculate correlation or covariance of imputed data:

```
miceadds::micombine.cor(mi.res, ...)
miceadds::micombine.cov(mi.res, ...)
```

These functions take `mids` objects as input.

Extract Imputed Data

The function complete() allows **extraction of the imputed data** from a mids object:

```
mice::complete(data, action = 1, include = FALSE, ...)
```

- data: the mids object
- action:
  - 1, …, m (single imputed dataset)
  - "long": long format (imputed data stacked vertically)
  - "broad": wide format (imputed data combined horizontally; ordered by imputation)
  - "repeated": (like "broad", but ordered by variable)
- include: include the original data?
  (if action is "long", "broad" or "repeated")

Extract Imputed Data

The function `mids2spss()` allows the **export of imputed data** (`mids` objects) to SPSS.

```r
mids2spss(imp4,
          filedat = "datafile.txt", # the file containing the data
          filesps = "importsyntax.sps", # syntax to get .sav from .txt
          silent = TRUE, ...
)
```

Data from `mids` objects can also be exported to MPLUS using `mids2mplus()`.

Combining `mids` objects

To **increase the number of imputed datasets** without re-doing the initial *m* imputations, a second set of imputations can be done and the two `mids` objects combined using `ibind()`.

```
# same syntax as before, but different seed
imp4b <- update(imp4, post = post, maxit = 20, seed = 456)
imp4combi <- ibind(imp4, imp4b)
# check the new number of impute datasets:
imp4combi$m
```

```
## [1] 10
```

Adding variables to `mids` objects

The function `cbind.mids()` allows us to **add columns** to a `mids` object. The extra columns can either be a `data.frame`, `matrix`, `vector` or `factor` or another `mids` object.

For example data columns that should be part of the imputed data for completeness, but are not needed in the imputation.

```r
extravar <- rnorm(nrow(NHANES))
impextra <- mice::cbind.mids(x = imp4, extravar = extravar)
```

Note:

`cbind()` just adds columns to the data, you need to make sure they are **sorted correctly** so that the rows of the new data are from the same subjects as the corresponding rows in the imputed data.

Your Turn!

Practical

How to Analyse Data after Multiple Imputation **html**