

EP16: Missing Values in Clinical Research: Multiple Imputation

7. Convergence & Diagnostics

Nicole Erler

Department of Biostatistics, Erasmus Medical Center

✉ n.erler@erasmusmc.nl

Setting

In this section, we use imputed data from the following set-up:

```
library("mice")
imp0 <- mice(NHANES, maxit = 0,
             defaultMethod = c("norm", "logreg", "polyreg", "polr"))

meth <- imp0$method
meth["HyperMed"] <- ""
meth["BMI"] <- "~I(weight/height^2)"

pred <- imp0$predictorMatrix
pred[, "HyperMed"] <- 0

post <- imp0$post
post["creat"] <- "imp[[j]][,i] <- squeeze(imp[[j]][,i], c(0, 100))"
```

Setting

Knowing that we “forgot” to change the predictor matrix to prevent feedback from BMI to height and weight, we use the resulting mids object imp3 for demonstratin purposes:

```
imp3 <- mice(NHANES, method = meth, predictorMatrix = pred, post = post)
```

Setting

Knowing that we “forgot” to change the predictor matrix to prevent feedback from BMI to height and weight, we use the resulting mids object imp3 for demonstratin purposes:

```
imp3 <- mice(NHANES, method = meth, predictorMatrix = pred, post = post)
```

Additionally, we work with the improved imputation using the following additional settings:

```
pred[c("weight", "height"), "BMI"] <- 0
```

```
imp4 <- mice(NHANES, method = meth, predictorMatrix = pred, post = post,  
             maxit = 30)
```

Logged Events

Information on the automatic changes that were done by **mice** is returned as `loggedEvents`, which is part of the `mids` object.

`loggedEvents` is a `data.frame` and has the following columns:

<code>it</code>	iteration number
<code>im</code>	imputation number
<code>dep</code>	dependent variable
<code>meth</code>	imputation method used
<code>out</code>	names of altered or removed predictors

It can be obtained as

```
imp3$loggedEvents
```

Logged Events

Neither `imp3` nor `imp4` had any logged events.

To demonstrate `loggedEvents` we create a small dataset with some “mistakes” in it:

```
demo <- NHANES[, 1:5]           # first 5 variables from NHANES
demo$dupl <- demo[, 4]          # create a duplicate variable
demo$const <- 1                 # create a constant variable
demo$age[demo$gender == 'male'] <- NA # set age missing for all males
```

```
demoimp <- mice(demo)
```

```
## Warning: Number of logged events: 8
```

Logged Events

```
head(demoimp$loggedEvents)
```

##	it	im	dep	meth	out
## 1	0	0		constant	const
## 2	0	0		collinear	dupl
## 3	1	1	age	pmm	genderfemale
## 4	1	2	age	pmm	genderfemale
## 5	1	3	age	pmm	genderfemale
## 6	2	1	age	pmm	genderfemale

Before imputation (iteration 0):

- ▶ the constant variable was removed
- ▶ the duplicate variable was identified as collinear and removed.

During imputation:

- ▶ gender was removed from the model for age

Convergence

From a previous section of this course we know that **mice** uses an **iterative algorithm** and imputations from the first few iterations may not be samples from the “correct” distributions.

Convergence

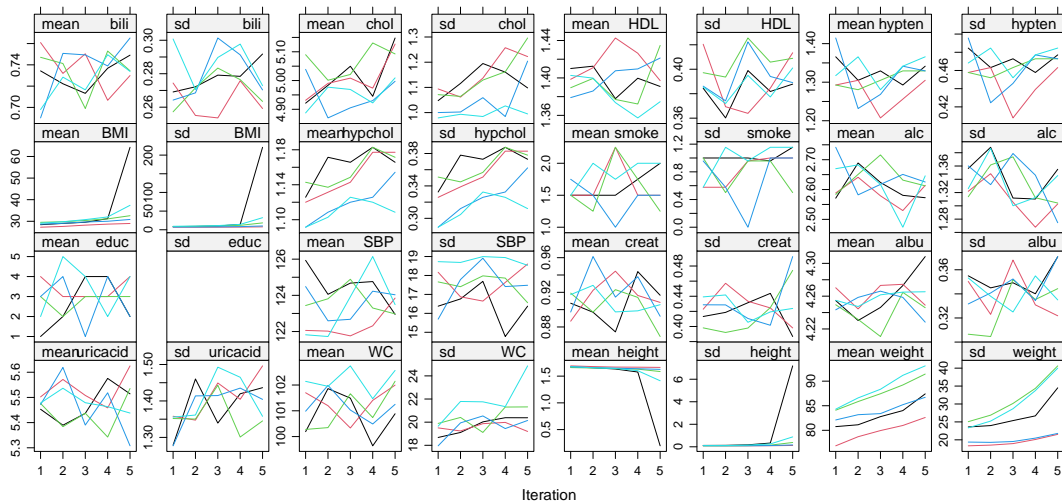
From a previous section of this course we know that **mice** uses an **iterative algorithm** and imputations from the first few iterations may not be samples from the “correct” distributions.

Traceplots can be used to visually assess **convergence**.

In **mice**, the function `plot()` produces traceplots of the mean and standard deviation (across subjects) per incomplete variable.

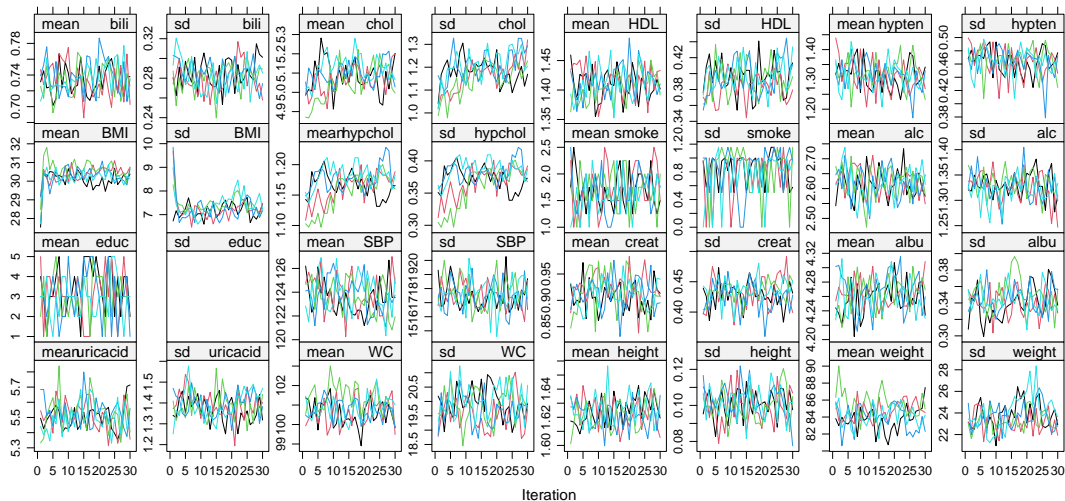
Convergence

```
plot(imp3, layout = c(8, 4))
```



Convergence

```
plot(imp4, layout = c(8, 4))
```



Convergence

Strong trends and traces that show **correlation** between variables indicate **problems of feedback**. This needs to be investigated and resolved in the specification of the `predictorMatrix`.

Weak trends may be artefacts that often disappear when the imputation is performed with more iterations.

Diagnostics

When MCMC chains have converged, the **distributions of the imputed and observed values** can be compared to investigate differences between observed and imputed data.

Note:

Plots usually show the **marginal** distributions of observed and imputed values, which do not have to be identical under MAR.

But:

The **conditional** distributions (given all the other variables in the imputation model) of the imputed values are assumed to be the same as the conditional distributions of the observed data.

Diagnostics

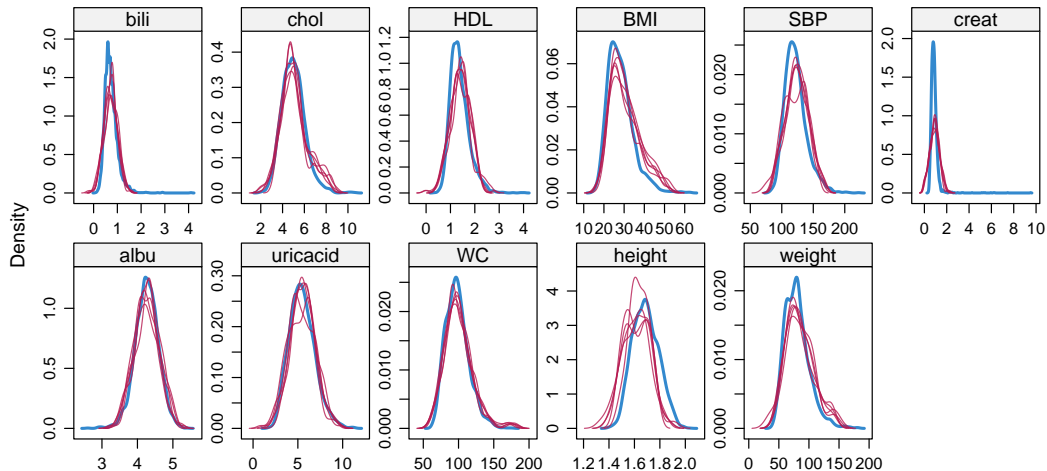
mice provides several functions for visual diagnosis of imputed values:

- ▶ `densityplot()` (for large datasets and variables with many NAs)
- ▶ `stripplot()` (for smaller datasets and/or variables with few NAs)
- ▶ `bwplot()`
- ▶ `xyplot()`

These functions create [lattice graphics](#), which can be modified analogously to their parent functions from the **lattice** package.

Diagnostics

```
densityplot(imp4)
```

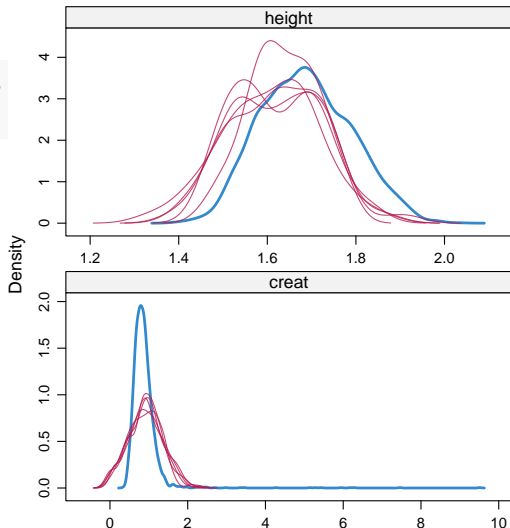


Diagnostics

```
densityplot(imp4, ~ height + creat,  
            layout = c(1, 2))
```

The `densityplot()` shows that

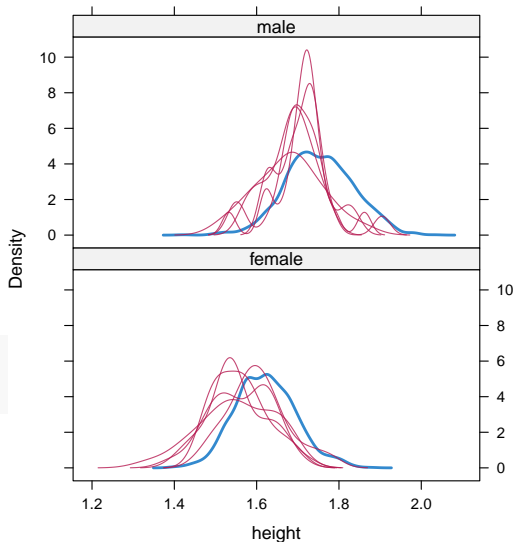
- ▶ imputed values of `height` are smaller than the observed values
- ▶ the distribution of the imputed values of `creat` is wider than the distribution of the observed values



Diagnostics

In some cases **differences** in distributions **can be explained by strata** in the data, however, here, **gender** does not explain the difference in observed and imputed values.

```
densityplot(imp4, ~height|gender,  
            layout = c(1, 2))
```



Diagnostics

As an alternative, we might consider **race** to explain the differences

```
densityplot(imp4, ~height|race)
```

```
## Error: need at least 2 points to select a bandwidth automatically
```

Diagnostics

As an alternative, we might consider `race` to explain the differences

```
densityplot(imp4, ~height|race)
```

```
## Error: need at least 2 points to select a bandwidth automatically
```

```
with(NHANES, table(race = race, "height missing" = is.na(height)))
```

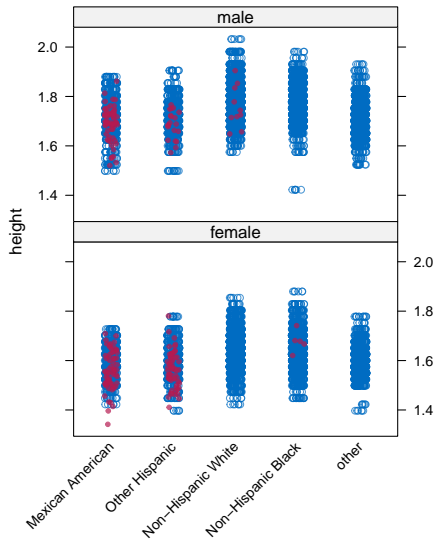
##		height missing	
##	race	FALSE	TRUE
##	Mexican American	233	26
##	Other Hispanic	252	16
##	Non-Hispanic White	884	2
##	Non-Hispanic Black	618	1
##	other	451	0

There are not enough missing values of `height` per categories of `race` to estimate densities.

Diagnostics

In that case, a `stripplot()` may be better suited.

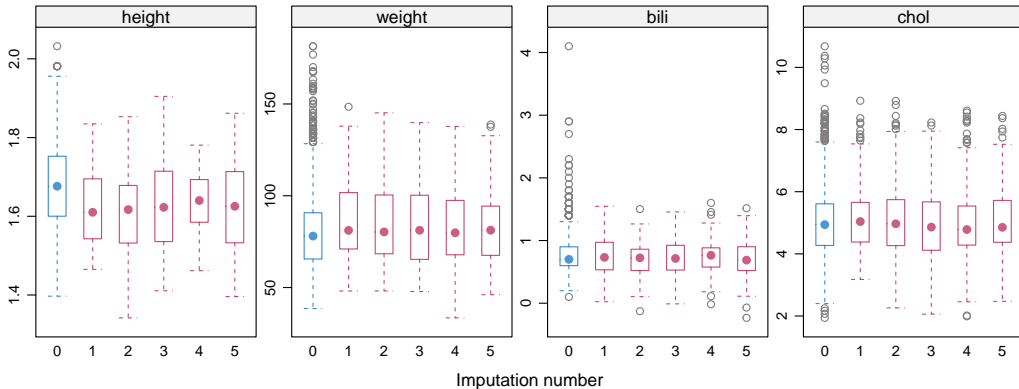
```
stripplot(imp4, height ~ race|gender,  
          pch = c(1, 20), layout = c(1, 2),  
          scales = list(x = list(rot = 45)))
```



Diagnostics

Alternatively, observed and imputed data can be represented by box-and-whisker plots:

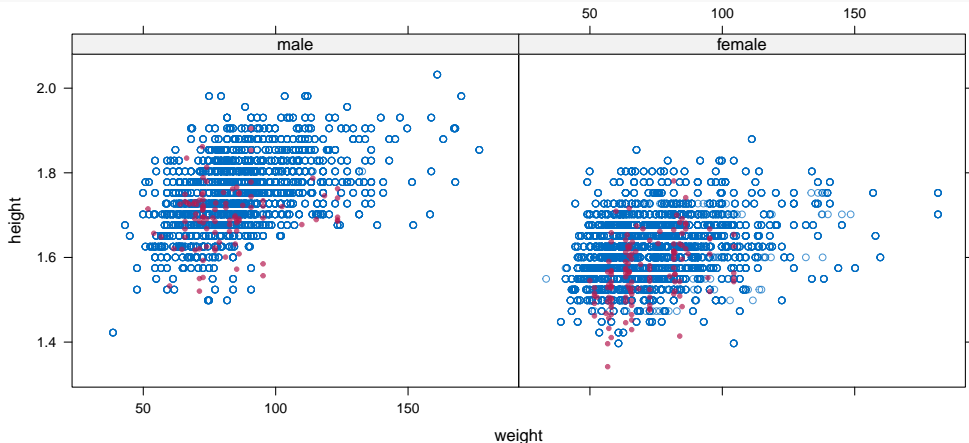
```
bwplot(imp4, height + weight + bili + chol ~.imp)
```



Diagnostics

The function `xyplot()` allows multivariate investigation of the imputed versus observed values.

```
xyplot(imp4, height ~ weight | gender, pch = c(1, 20))
```



Diagnostics

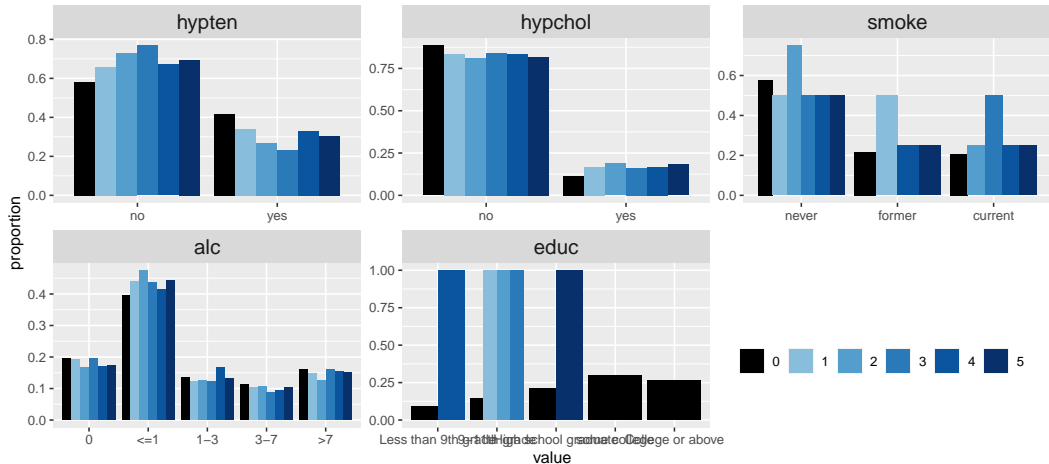
All of the above graphs displayed only continuous imputed variables. For categorical variables we can compare the proportion of values in each category.

mice does not provide a function to do this, but we can write one ourselves, as for instance the function `propplot()`, for which the syntax can be found here:

<https://gist.github.com/NErler/0d00375da460dd33839b98faeee2fdab>

Diagnostics

```
propplot(imp4)
```



Diagnostics

`smoke` and `educ` have **very few missing values** (4 and 1)

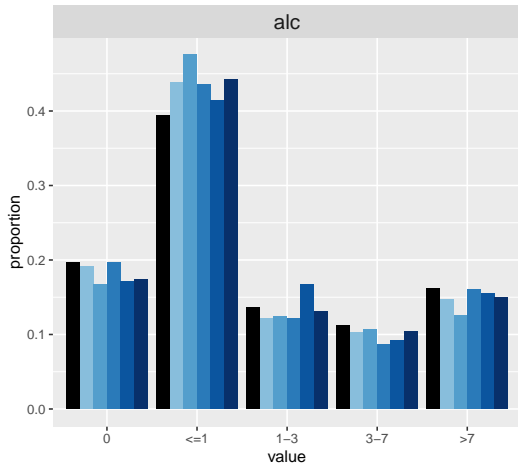
➡ no need to worry about differences

Diagnostics

`smoke` and `educ` have **very few missing values** (4 and 1)

➔ no need to worry about differences

`alc`: missing values are imputed in the category “ ≤ 1 ” more often than we would expect from the observed data



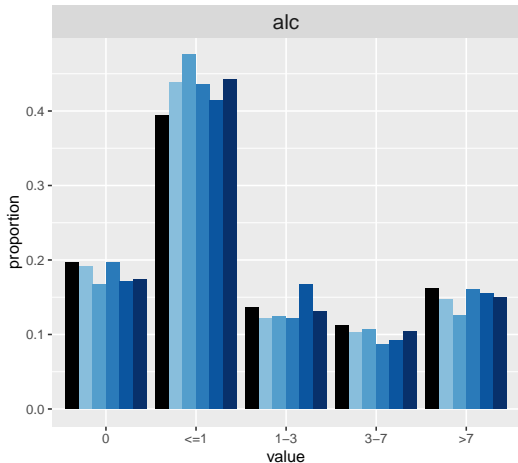
Diagnostics

`smoke` and `educ` have **very few missing values** (4 and 1)

➔ no need to worry about differences

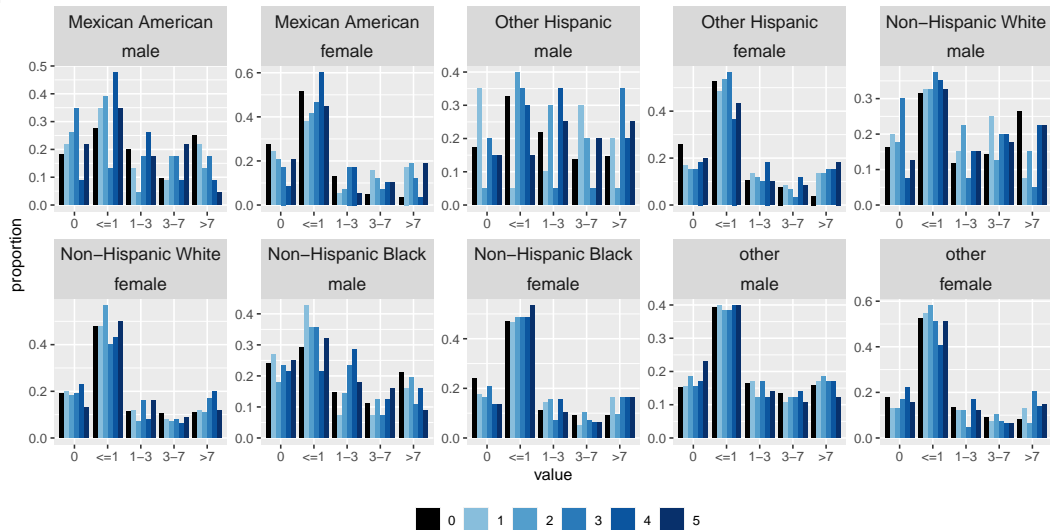
`alc`: missing values are imputed in the category " ≤ 1 " more often than we would expect from the observed data

If we expect that `gender` and `race` might explain the differences for `alc`, we can include those factors into the plot.



Diagnostics

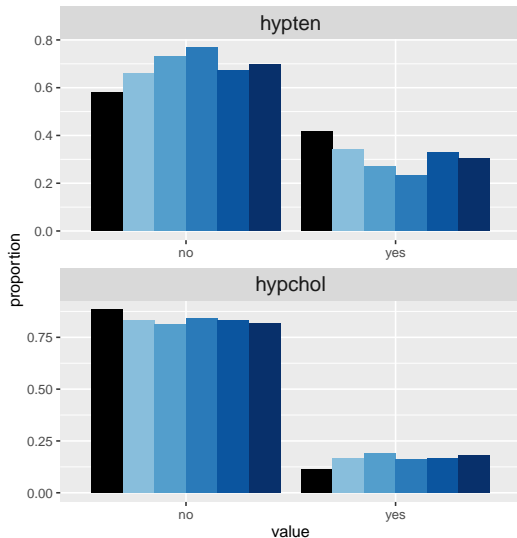
```
propplot(imp4, formula = alc ~ race + gender)
```



Diagnostics

We also see that

- ▶ `hypten` is less frequent and
- ▶ `hypchol` a bit more frequent, in the imputed data compared to the observed.



Diagnostics

Since hypertension is more common in older individuals, we may want to investigate if `age` can explain the differences in imputed values of `hypten`.

```
round(sapply(split(NHANES[, "age"], addNA(NHANES$hypten)), summary), 1)
```

##	no	yes	<NA>
## Min.	20.0	20.0	20.0
## 1st Qu.	28.0	47.0	30.0
## Median	38.0	59.0	38.5
## Mean	40.7	56.9	41.5
## 3rd Qu.	51.0	68.0	50.8
## Max.	79.0	79.0	78.0

The distribution of `age` in participants with missing `hypten` is very similar to the distribution of `age` in participants without `hypten`.

Diagnostics

Plotting the proportions of observed and imputed **hypten** separately per quartile of **age**:

```
propplot(imp4, formula = hypten ~ cut(age, quantile(age), include.lowest = T))
```

