Missing Data Analysis Application in R

```
library(mice)
```

Introduction

Any missing data method involves modeling assumptions. All methods have limitations – better to avoid missing values, or try to minimize the problem.

Missing data undermines randomization, the lynchpin of causal inferences in clinical trials.

Longitudinal studies often have drop-outs
- Move out of study catchment area
- Participation becomes too onerous

Some complete-data problems can be formulated with underlying unobserved data, allowing incomplete-data methods of analysis, e.g. EM algorithm
- Factor analysis – multivariate regression with unobserved regressors
- Mixed-effects models – random effects are unobserved "missing" data
- Genetics – genotypes are "missing"

Important to collect and use relevant covariate information
- Covariates related to missingness and main outcomes

Important to perform sensitivity analyses for nonignorable missing data.

Multiple Imputation (MI)

There are uncertainties associated with imputation and this uncertainty can be estimated by using MI. MI propagates imputation uncertainty and is more efficient than a single imputation.

The essential idea of multiple imputation is to reflect the uncertainty associated with missing data by imputing several values for each missing value, each imputed value drawn from the predictive distribution (i.e., from an *imputation model*) of the missing data, and therefore producing not one but several completed data sets.

Standard methods of statistical analysis are then applied in parallel to the M multiply imputed data sets (*analysis model*).

Parameters of interest are estimated along with their standard errors for each imputed data set. The estimates from the M analyses are combined using *Rubin's Rules*.

Estimated parameters are averaged across the M multiply imputed data sets.

The MI estimate is $\bar{\theta}_M = \frac{1}{M}\sum_{m=1}^{M}\widehat{\theta}_m$ where θ is the parameter of interest and $\widehat{\theta}_m$ is the estimate from the mth data set for $m = 1, \ldots, M$.

Standard errors are combined across imputed data sets, taking into account the variation among the estimates in the M data sets, thereby capturing the added uncertainty due to having to impute the missing data.

W_m is an estimate of the variance from the mth data set. $\bar{W}_M$ is the average within-imputation variance over the M imputed data sets. The MI estimate of the total variance is $T_M = \bar{W}_M + (1 + 1/M)B_M$ where $B_M = \frac{1}{M-1}\sum_{m=1}^{M}(\widehat{\theta}_m - \bar{\theta}_M)^2$ is the between-imputation variance.

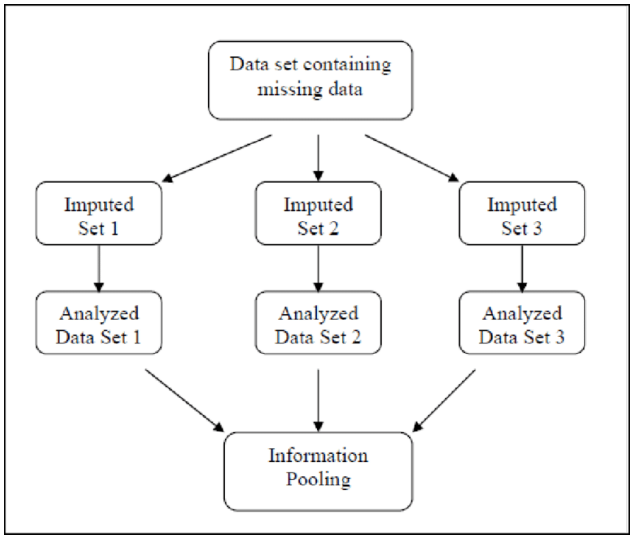


Figure 1: MI workflow

The estimated fraction of missing information (fmi) is

$$fmi = \frac{(1 + 1/M)B_M}{(1 + 1/M)B_M + \bar{W}_M}$$

The proportion of total variance that is due to the missing data is defined as $\lambda = (B + B/m)/T$.

The efficiency of the MI estimator relative to the maximally efficient maximum-likelihood estimator is $RE = \frac{m}{m - fmi}$.

If the number of imputations is very large, MI is as efficient as ML.

Even when the rate of missing information is high and the number of imputations modest, the relative efficiency of the MI estimator hardly suffers.

An important advantage is that the *imputation model* and *analysis model* can differ. In particular, auxiliary variables, V, that are not included in the analysis model can be used in the imputation model, so that the MAR assumption is more plausible.

Marked incompatibility between the data model and imputation model may be a concern. *Any variable that will be included the analysis model should also be included in the imputation model.*

Multiple imputation cannot preserve features of the data that are not represented in the imputation model.

It is important to insure that the imputation model is consistent with the intended analysis.

In addition to variables that will be in the analysis model, include variables in the imputation model that make the assumption of ignorable missingness reasonable.

Finding variables that are highly correlated with a variable that has missing data will likely improve the quality of imputations, as will variables that are related to missingness.

Use all relevant variables, even ones not used in the analysis model.

There is nothing wrong in using the response variable to help impute missing values of explanatory variables. Think of imputation as a pure prediction problem.

Make sure that the imputation model captures relevant features of the data. For example, imputations will not preserve nonlinear relationships and interactions among the variables, unless we make special provision for these features of the data.

Joint vs. Conditional Distribution Imputation Strategies

Conditional imputation has the ability to specify individual regression models for different types of variables. This approach is sometimes called a fully conditional specification or multiple imputation via chained equations.

- Types of variables
  - Continuous (Normal)
  - Categorical (Logistic or generalized logistic)
  - Count (Poisson)
  - Mixed or semi-continuous (Logistic/Normal)
  - Ordinal (Ordered probit)
- Parametric or semi-parametric regression models
- Restrictions
  - Regression model is fitted only to the relevant subset
- Bounds
  - Draws from a truncated distribution from the corresponding regression model
- Models each conditional distribution. There is no guarantee that a joint distribution exists with these conditional distributions.

Example of MI in R using mice

We will use the NHANES example from the **mice** package. It is a small data set with 4 variables and 25 observations that was extracted from the much larger NHANES data. It is automatically loaded with the **mice** package, so you can refer to it without 'reading' it in. `age` is grouped such that 1=20-39, 2=40-59, 3=60+. `chl` is measured in mg/dL and `hyp` is 1=no, 2=yes. `bmi` is body mass index.

Here again is our workflow, but with the R functions listed.

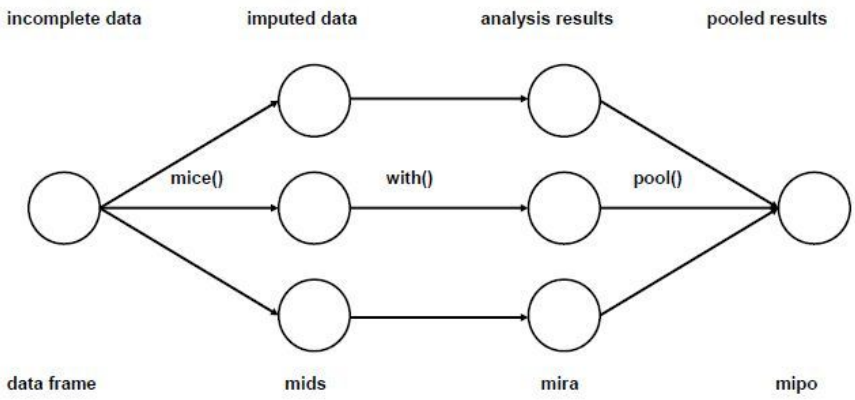


Figure 2: MI workflow

Our goal is to fit a regression model of cholesterol on age and bmi. But as we see below we have some missing values. If we fit a regression model, 12 observations (nearly half of our sample!) are deleted.

```
summary(nhanes)
```

```
##       age            bmi             hyp             chl       
##  Min.   :1.00   Min.   :20.40   Min.   :1.000   Min.   :113.0  
##  1st Qu.:1.00   1st Qu.:22.65   1st Qu.:1.000   1st Qu.:185.0  
##  Median :2.00   Median :26.75   Median :1.000   Median :187.0  
##  Mean   :1.76   Mean   :26.56   Mean   :1.235   Mean   :191.4  
##  3rd Qu.:2.00   3rd Qu.:28.93   3rd Qu.:1.000   3rd Qu.:212.0  
##  Max.   :3.00   Max.   :35.30   Max.   :2.000   Max.   :284.0  
##                 NA's   :9       NA's   :8       NA's   :10
```

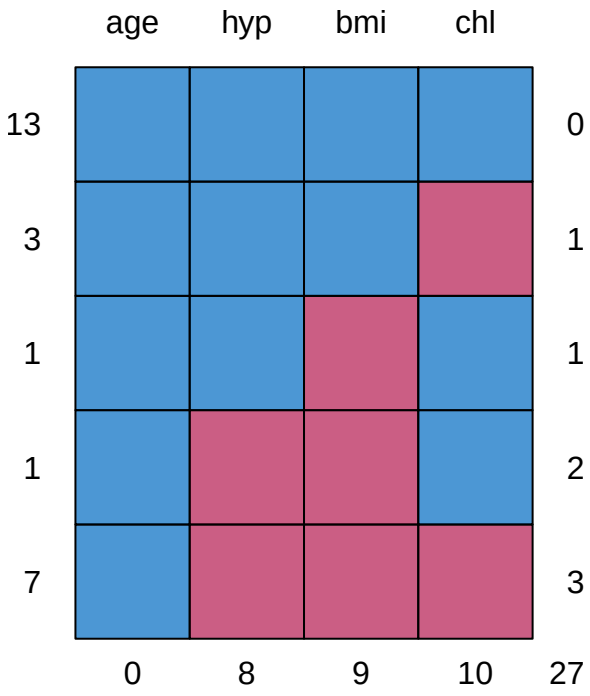
```
summary(lm(chl ~ age + bmi, data=nhanes))
```

```
##
## Call:
## lm(formula = chl ~ age + bmi, data = nhanes)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -31.187 -19.517  -0.310   6.915  60.606
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -80.194     58.772  -1.364 0.202327
## age           53.069     11.293   4.699 0.000842 ***
## bmi            6.884      1.846   3.730 0.003913 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.67 on 10 degrees of freedom
##   (12 observations deleted due to missingness)
## Multiple R-squared:  0.7318, Adjusted R-squared:  0.6781
## F-statistic: 13.64 on 2 and 10 DF,  p-value: 0.001388
```

The first step is to examine the missingness.

Examine missing data patterns

```
md.pattern(nhanes)  # from mice package
```




```
##    age hyp bmi chl
## 13   1   1   1   1  0
## 3    1   1   1   0  1
## 1    1   1   0   1  1
## 1    1   0   0   1  2
## 7    1   0   0   0  3
##      0   8   9  10 27
```

A '1' or the color blue indicates observed and '0' or the color red indicates missing. The numbers on the left side column tell you how many individuals are observed for that pattern and the numbers on the right side column tell you how many variables are missing for that particular pattern. Thus, for example, there are 3 individuals who are observed on `age`, `hyp`, and `bmi` but missing on `chl`. Along the bottom are the number of individuals missing on each variable. Thus, there are 10 individuals who are missing values on `chl` and 9 who are missing values on `bmi`.

```
md.pairs(nhanes)    # from mice package
```

```
## $rr
##     age bmi hyp chl
## age  25  16  17  15
## bmi  16  16  16  13
## hyp  17  16  17  14
## chl  15  13  14  15
##
## $rm
##     age bmi hyp chl
## age   0   9   8  10
## bmi   0   0   0   3
## hyp   0   1   0   3
## chl   0   2   1   0
##
## $mr
##     age bmi hyp chl
```

```
## age   0   0   0   0
## bmi   9   0   1   2
## hyp   8   0   0   1
## chl  10   3   3   0
##
## $mm
##     age bmi hyp chl
## age   0   0   0   0
## bmi   0   9   8   7
## hyp   0   8   8   7
## chl   0   7   7  10
```

`r` denotes observed and `m` denotes missing. The first letter after `$` refers to the variable in the rows and the second to the variable in the columns. Thus, for example, in the `$rm` table, we see that there are 9 individuals whose age is observed but their bmi is not.

Perform the multiple imputation

```
imp <- mice(nhanes, m=20, maxit=25, seed=42, print=FALSE)
```

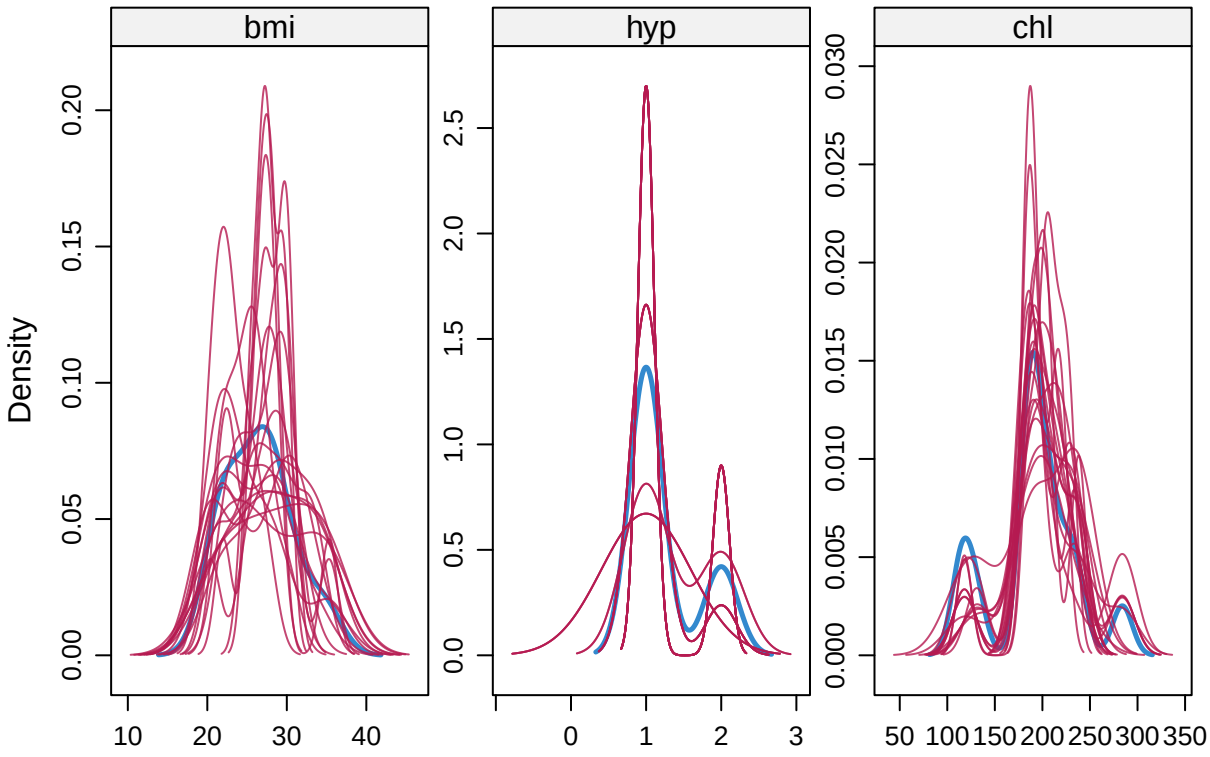
You should make sure that the imputations are plausible. For example, you do not want negative BMIs.

```
imp$imp$bmi
```

```
##       1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 1  27.2 27.2 22.5 30.1 30.1 22.0 33.2 30.1 27.5 35.3 27.2 35.3 29.6 33.2 27.2
## 3  29.6 28.7 22.7 33.2 22.0 29.6 30.1 22.0 27.2 33.2 30.1 35.3 22.0 28.7 30.1
## 4  25.5 27.5 20.4 22.5 22.5 22.5 22.5 24.9 27.4 20.4 25.5 22.7 25.5 20.4 20.4
## 6  25.5 27.4 21.7 21.7 21.7 21.7 21.7 20.4 25.5 21.7 24.9 20.4 25.5 25.5 22.5
## 10 27.4 22.5 20.4 28.7 27.5 22.5 22.7 26.3 28.7 29.6 27.4 26.3 33.2 27.4 28.7
## 11 27.2 30.1 27.5 27.2 24.9 27.2 30.1 33.2 26.3 33.2 29.6 29.6 27.2 28.7 35.3
## 12 27.5 20.4 24.9 27.5 25.5 20.4 20.4 35.3 22.0 27.4 27.4 25.5 20.4 22.7 22.5
## 16 29.6 27.2 25.5 33.2 25.5 26.3 29.6 29.6 28.7 24.9 29.6 22.0 33.2 20.4 28.7
## 21 29.6 22.0 22.5 35.3 26.3 35.3 30.1 35.3 35.3 27.5 30.1 33.2 29.6 22.5 22.0
##      16   17   18   19   20
## 1  35.3 30.1 35.3 29.6 27.5
## 3  27.2 28.7 30.1 29.6 27.2
## 4  24.9 20.4 20.4 24.9 27.4
## 6  20.4 25.5 22.7 25.5 22.5
## 10 28.7 28.7 27.4 29.6 27.5
## 11 30.1 20.4 33.2 29.6 27.4
## 12 27.4 20.4 24.9 22.5 27.2
## 16 29.6 30.1 27.2 27.5 22.7
## 21 30.1 27.2 33.2 22.7 22.0
```

Visualize this by overlaying the distribution for each of the 20 imputations on the observed data distribution.

```
densityplot(imp)
```



You can obtain the complete data using `complete()` which will give you the first imputation. If you wanted the second imputation, you would use `complete(imp, 2)`.

```
mice::complete(imp)
```

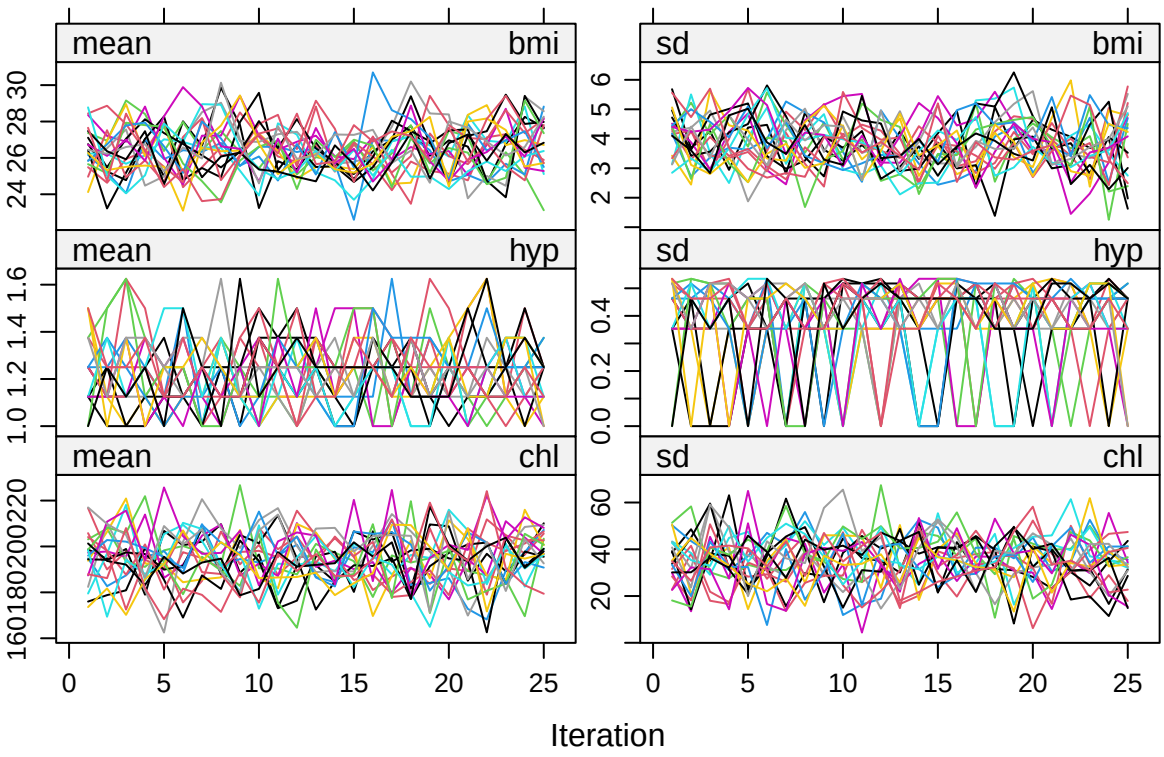
```
##    age  bmi hyp chl
## 1    1 27.2   1 187
## 2    2 22.7   1 187
## 3    1 29.6   1 187
## 4    3 25.5   2 284
## 5    1 20.4   1 113
## 6    3 25.5   1 184
## 7    1 22.5   1 118
## 8    1 30.1   1 187
## 9    2 22.0   1 238
## 10   2 27.4   1 218
## 11   1 27.2   1 187
## 12   2 27.5   1 218
## 13   3 21.7   1 206
## 14   2 28.7   2 204
## 15   1 29.6   1 187
## 16   1 29.6   1 238
## 17   3 27.2   2 284
## 18   2 26.3   2 199
## 19   1 35.3   1 218
## 20   3 25.5   2 199
## 21   1 29.6   2 199
## 22   1 33.2   1 229
## 23   1 27.5   1 131
## 24   3 24.9   1 184
## 25   2 27.4   1 186
```

```
nhanes
```

```
##    age  bmi hyp chl
## 1    1   NA  NA  NA
## 2    2 22.7   1 187
## 3    1   NA   1 187
## 4    3   NA  NA  NA
## 5    1 20.4   1 113
## 6    3   NA  NA 184
## 7    1 22.5   1 118
## 8    1 30.1   1 187
## 9    2 22.0   1 238
## 10   2   NA  NA  NA
## 11   1   NA  NA  NA
## 12   2   NA  NA  NA
## 13   3 21.7   1 206
## 14   2 28.7   2 204
## 15   1 29.6   1  NA
## 16   1   NA  NA  NA
## 17   3 27.2   2 284
## 18   2 26.3   2 199
## 19   1 35.3   1 218
## 20   3 25.5   2  NA
## 21   1   NA  NA  NA
## 22   1 33.2   1 229
## 23   1 27.5   1 131
## 24   3 24.9   1  NA
## 25   2 27.4   1 186
```

You should also check for *convergence*.

```
plot(imp)
```



Fit the analysis model to each of the imputed data sets:

```
fit <- with(imp, lm(chl ~ age + bmi))
```

Pool (i.e., combine) the results across the imputed data sets:

```
summary(mice::pool(fit))
```

```
##          term   estimate std.error  statistic       df    p.value
## 1 (Intercept) -17.621852 67.742300 -0.2601307 11.76138 0.79926094
## 2         age  33.076891 11.144760  2.9679320 10.50561 0.01339944
## 3         bmi   5.778044  2.027238  2.8502055 13.51864 0.01321579
```

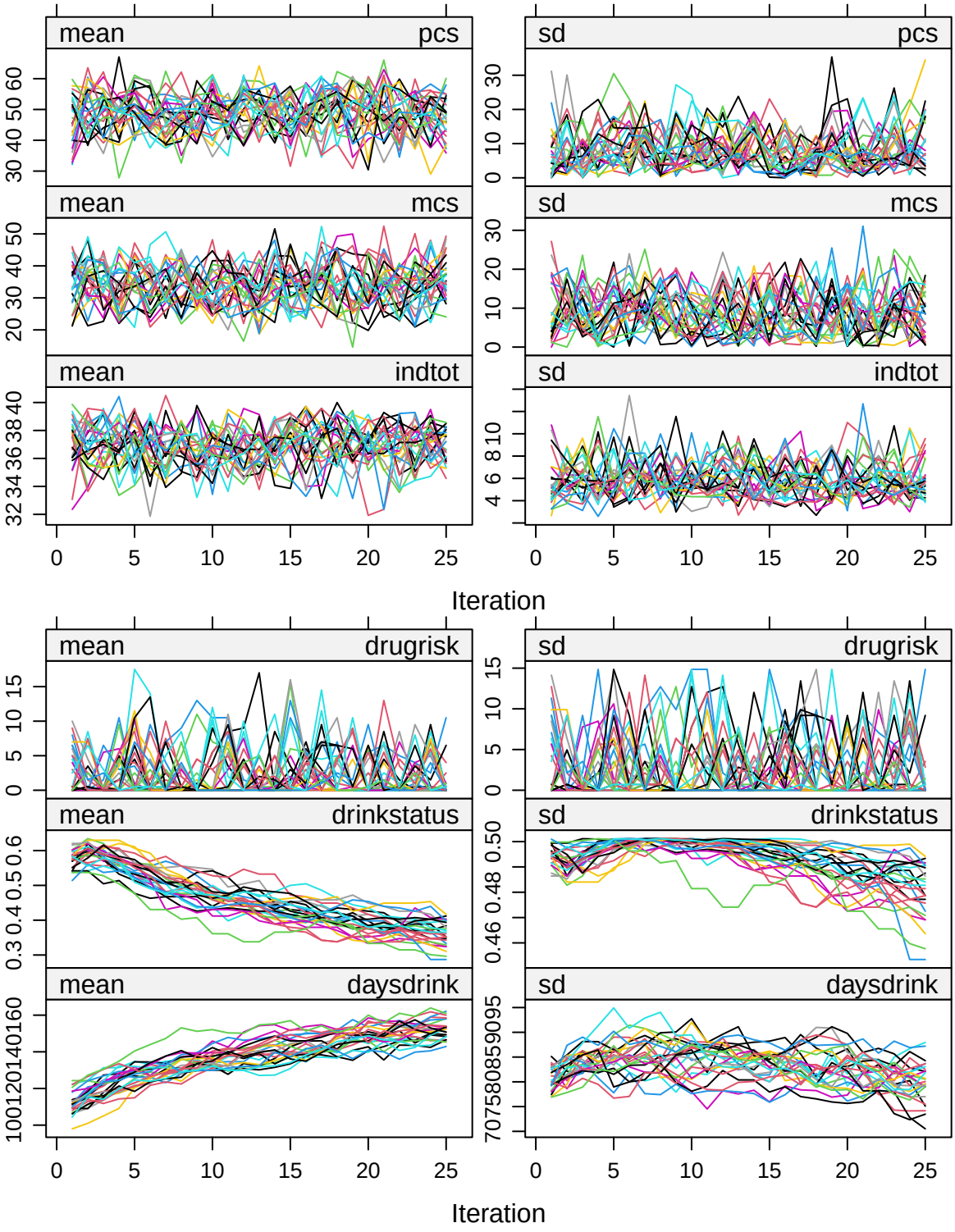
```
pool.r.squared(fit)
```

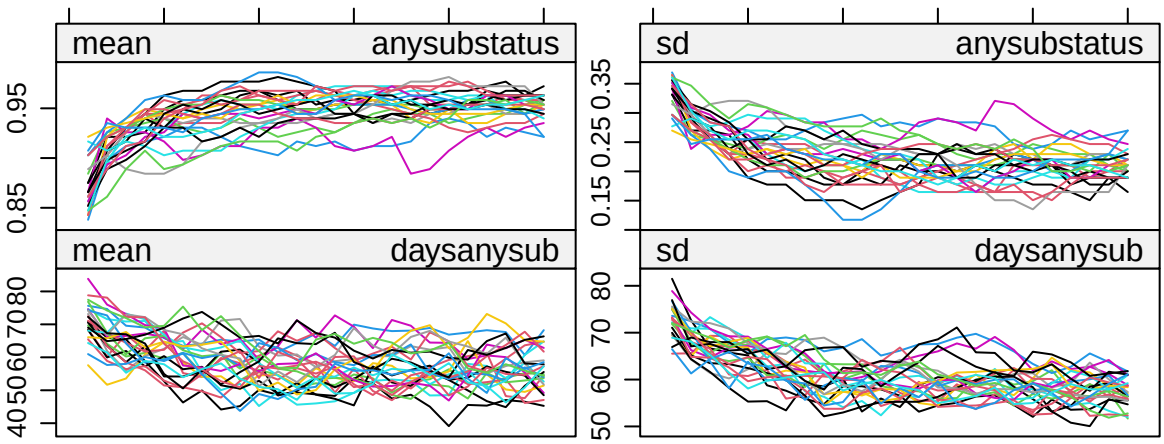
```
##           est      lo 95     hi 95       fmi
## R^2 0.4691349 0.08470467 0.7748879 0.4079589
```

Checking diagnostics

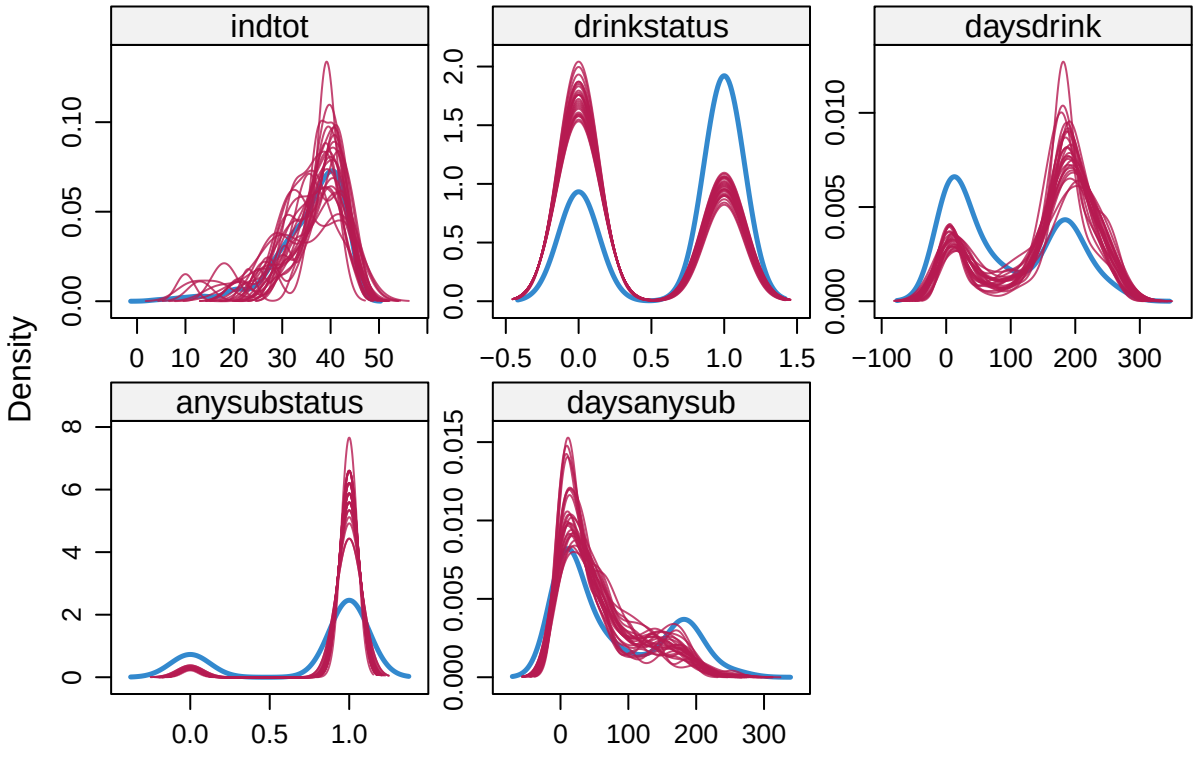
The example above was well-behaved. Here is a pathological example.

```
helpmiss <- read.csv("helpmiss.csv", header = TRUE)
dat <- dplyr::select(helpmiss, homeless, pcs, mcs, cesd, indtot, pss_fr, drugrisk, satreat, drinkstatus
imp <- mice(dat, m=25, maxit=25, seed=42, print=FALSE)
plot(imp)
```


mean
pcs
mean
mcs
mean
indtot
sd
pcs
sd
mcs
sd
indtot
Iteration
mean
drugrisk
mean
drinkstatus
mean
daysdrink
sd
drugrisk
sd
drinkstatus
sd
daysdrink
Iteration




```
densityplot(imp)
```



Non-ignorable missingness

Nonignorable mechanisms can be included in a missing-data analysis but this is a difficult modeling problem. Software for fitting non-ignorable models is not widely available. Designing to avoid nonignorable missing

data is preferable if possible. Two generic modeling strategies are pattern-mixture models and selection models.

Selection models are:
- more natural substantive formulation of model if inference concerns the entire population
- more common approach in literature
- sensitive to specification of the form of the missing-data mechanism, which is often not well understood

Pattern-mixture models:
- are more natural when interest is in population strata defined by missing-data pattern
- are closer to the form of the data and sometimes simpler to fit
- can avoid specifying the form of the missing data mechanism, which is incorporated indirectly via parameter restrictions

But often little is known about the missing-data mechanism and results may be sensitive to the model formulation. Parameters of missing-data are often unidentified or weakly identified from the data. Parameters of MNAR models often cannot be reliably estimated – identifiability requires structural assumptions that are often questionable. As a result, it may be more appropriate to do a sensitivity analysis by fixing weakly identified parameters at different values. *Varying certain parameters in a sensitivity analysis is the preferred approach.* In many (not all) situations, it would be reasonable to choose an MAR primary model, and look at MNAR models via a sensitivity analysis to assess plausible deviations from MAR.

With substantial missing data, sensitivity analyses to assess robustness to alternative analysis models are needed. Sensitivity analysis is a scientific way of attempting to reflect uncertainty arising from potentially MNAR missing data. Deciding on how to implement and interpret a sensitivity analysis is challenging. The need and importance of sensitivity analysis increases with the amount of potentially MNAR missing data. This reinforces the need to limit missing data in the design and implementation stage. Avoiding substantial amounts of missing data is key!