Missing Data

Multiple Imputation
using the `mi` package in `R`

Outline of the Course

1) Missing Data Mechanisms. How are missing data generated and why should we care? Complete Case Analysis. Getting comfortable with R.
2) Simple missing data fixes: listwise deletion, available case, LVCF, mean imputation, dummy variable methods
3) More complicated missing data fixes: weighting, hotdecking, regression imputation
4) Building blocks and overview of multiple imputation (including regression imputation with noise)
5) **Multiple imputation in practice (software in R, simple analyses, and diagnostics)**
6) Multiple imputation in practice (more complicated models and considerations, more advanced diagnostics)
7) More advanced imputation and other missing data methods

Review of the FCS method for MI

Fully conditional specification (FCS) imputes *multivariate* missing data on a variable-by-variable basis as follows:

1. A simple imputation (e.g., random) is performed for *every* missing value;

2. Choose a variable with missing values, say Var1, to *impute*;

3. The *observed* values of Var1 in step 2 are regressed on the other variables. These regression models could be linear, logistic, Poisson, …

4. The *missing* values for Var1 are then replaced with predictions from the model in step 3;

5. Steps 2 – 4 are then repeated for *each* variable that has missing data. The loop through each of the variables constitutes one *iteration* or *cycle*;

6. Steps 2 through 4 are repeated for *several* cycles, with the imputations being updated at each cycle. The number of cycles to be performed can be specified by the researcher. At the end of these cycles the final imputations are retained, resulting in *one* imputed dataset.

Goal of MI

- Suppose we want to calculate some quantity Q that describes the population of interest (population mean, the population variance, population correlation, …).
- We can only calculate Q if the population data are fully known, but this is almost never the case.
- The goal of multiple imputation is to find an *estimate* $\hat{Q}$ that is *unbiased* and *confidence valid* (Rubin 1996).
- Confidence validity roughly means that the MI estimators should not underestimate the sampling variability.
- For example, a 95% confidence interval based on a MI estimator should have at least 95% coverage.

Three Sources of Variation

In MI we estimate Q repeatedly m times by $\hat{Q}_1, \ldots, \hat{Q}_m$.

Then the final estimate is obtained by combining the individual ones:

$$\bar{Q} = \frac{1}{m}\sum_{l=1}^{m} \hat{Q}_l$$

This process creates three sources of variance:

1. Variance caused by the fact that we are taking a sample rather than observing the entire population. This is the conventional *sampling* uncertainty;
2. Extra variance caused by the fact that there are missing values in the sample;
3. Extra simulation variance caused by the fact that $\bar{Q}$ itself is estimated for finite m.

Incorporating sampling uncertainty: PPD versus PMM

- Drawing an imputed value from the **Posterior Predictive Distribution** involves first drawing the coefficients of the model from their distribution (and the residual standard error as well) and then drawing imputed values from the predictive distribution that conditions on these values (colloquially this means adding noise to the prediction that would come from a model with these coefficients)
- **Predictive mean matching**: rather than drawing a new X from its predictive distribution, another option is to find the person in the dataset whose predicted value most closely matches the predicted value for the person with missing data and then substitute the first person's observed value for the missing value of the second (can be thought of as a kind of hotdecking)

Regression switching. Step 2: imputation

Impute $X_4^{(1)}$ using X_1, $X_2^{(s)}$, $X_3^{(s)}$

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	8	3
0	4	8	?
1	5	10	?
1	3	7	?
1	3	5	?
1	4	7	6
1	5	6	7
1	4	8	9
1	5	10	8

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	8	3
0	4	8	4.2
1	5	10	9.1
1	3	7	6.7
1	3	5	7.2
1	4	7	6
1	5	6	7
1	4	8	9
1	5	10	8

Impute X_2 using X_1, $X_3^{(s)}$, $X_4^{(1)}$

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	8	3
0	4	8	4.2
1	5	10	9.1
1	?	7	6.7
1	?	5	7.2
1	?	7	6
1	?	6	7
1	4	8	9
1	5	10	8

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	8	3
0	4	8	4.2
1	5	10	9.1
1	5.1	7	6.7
1	4.2	5	7.2
1	5.5	7	6
1	3.9	6	7
1	4	8	9
1	5	10	8

Multiple Imputation

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	?	3
0	4	?	?
1	5	10	?
1	?	7	?
1	?	5	?
1	?	7	6
1	?	6	7
1	4	8	9
1	5	?	8

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	8	3
0	4	8	2
1	5	10	12
1	5	7	9
1	4	5	8
1	4	7	10
1	3	6	7
1	4	8	9
1	5	?	8

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	7	3
0	4	8	1
1	5	10	9
1	4	7	9
1	5	5	8
1	5	7	6
1	3	6	7
1	4	8	9
1	5	?	8

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	7	3
0	9	8	2
1	5	10	10
1	5	7	9
1	4	5	8
1	5	7	6
1	3	6	7
1	4	8	9
1	5	?	8

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	8	3
0	4	9	3
1	5	9	11
1	4	7	9
1	4	5	9
1	5	7	6
1	4	6	7
1	4	8	9
1	5	?	8

Combining estimates across datasets (reminder):

Given estimates $Q_1 \ldots Q_M$ and their corresponding standard errors, $s_1, \ldots, s_M$

point estimate: $\theta = 1/m \sum_m Q_m$

variance: $W + (1 + m^{-1})B$

where,

$$W = 1/m \sum_m s_m^2$$

$$B = 1/(m-1) \sum_m (Q_m - \theta)^2$$

Estimates from each

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	8	3
0	4	8	4
1	5	10	12
1	5	7	11
1	4	5	9
1	4	7	10
1	3	6	7
1	4	8	9
1	5	?	8

$$\overline{X}_4 = 7.09$$

$$s_{\overline{X}_4} = 1.066$$

$$\overline{X}_4 = 5.82$$

$$s_{\overline{X}_4} = 0.923$$

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	7	3
0	4	8	1
1	5	10	7
1	4	7	8
1	5	5	6
1	5	7	6
1	3	6	7
1	4	8	9
1	5	?	8

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	7	3
0	9	8	2
1	5	10	10
1	5	7	9
1	4	5	8
1	5	7	6
1	3	6	7
1	4	8	9
1	5	?	8

$$\overline{X}_4 = 6.45$$

$$s_{\overline{X}_4} = 0.985$$

$$\overline{X}_4 = 6.72$$

$$s_{\overline{X}_4} = 1.001$$

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	8	3
0	4	9	3
1	5	9	11
1	4	7	9
1	4	5	9
1	5	7	6
1	4	6	7
1	4	8	9
1	5	?	8

Example: inference for the mean of X_4

- point estimate:

$$\theta = 1/m \sum_m Q_m$$
$$= 1/4\ (7.09+6.45+5.82+6.72)$$
$$= 6.52$$

- variance estimate

$$W + (1 + m^{-1})B = 0.99 + 1.25*0.29=1.35$$

where,

$$W=1/m \sum_m s_m^2$$
$$=1/4(1.066^2+.985^2+.923^2+1.001^2)=.99$$
$$B =1/(m-1) \sum_m (Q_m- \theta)^2$$
$$=1/3(.57^2 - .07^2 - .70^2 + .21^2) = .29$$

Common concerns with imputation

Addressing a common concern

- Some people feel very uncomfortable with MI because they feel it is “making up data”
- However the goal is not to find the “true” value that represents what we should have observed
- The goal is to use the imputations to better estimate model parameters
- The multiple imputations allow for variation in these estimates. When we use the combining rules we “average over” the missing data (rather than accepting them as the truth)
- Another way of thinking about it is as if we are appropriately re-weighting the complete case sample

Differences between packages

Different MI software packages vary somewhat in their exact implementation of this algorithm (e.g., in the order in which the variables are imputed), but the general strategy is the same.

The package `mi` uses a strategy similar to `MICE`: proceeding one-variable-at-a-time. It starts with median/mode for missing data, conducts a specified number of iterations, and cycles through until convergence. It then draws a bootstrap sample to create imputed values. What distinguishes `mi` from `MICE` is that `mi` adds algorithms related to semi-continuous data (such as age) and adds Bayesian models (Su, Gelman, Hill, & Yajima, 2011).

MI in practice using R, package `mi`

Installing the `mi` package

You should be able to install with the command

```
install.packages("mi")
```

If needed, first set the mirror.

Once package is installed you loaded it into R with

```
library(mi)
```

Documentation about the package here:

https://cran.r-project.org/web/packages/mi/mi.pdf

Basic steps when imputing using `mi`

1. Load the data
2. Create a missing_data object, look at the data and the missing data patterns
3. Examine the default choices for imputation models
4. Make changes to imputation models if necessary
5. Impute until converged
6. Plot diagnostics
7. Iterate between 4-6 if necessary
8. Run final pooled analysis

(1) Load the data

```
> data(nlsyV, package = "mi")
```

This extracts the nlsyV dataset from the mi package. This dataset pertains to children and their families in the United States. Variables are:

- ppvtr.36 -a numeric vector with data on the Peabody Picture Vocabulary Test administered at 36 months
- first - indicator for whether child was first-born
- b.marr - indicator if mother was married when child was born
- income - numeric data on family income in year after the child was born
- momage - a numeric vector with data on the age of the mother when the child was born
- momed - educational status of mother when child was born (1 = less than high school, 2 = high school graduate, 3 = some college, 4 = college graduate)
- romrace - race of mother (1 = black, 2 = Hispanic, 3 = white)

(1) Load the data

To read in other types of data files you can load the `foreign` package. Read data Stored by Minitab, S, SAS, SPSS, Stata, Systat, dBase, ...

```
> library(foreign)
> help(package="foreign")
```

and use one of the commands in that to read in data from a variety of formats.

(2) Create a `missing_data` object, then look at the data and the missing data patterns

This class is similar to a **`data.frame`**, but is customized for the situation in which variables with missing data are being modeled for multiple imputation.

```
mdf = missing_data.frame(nlsyV)
summary(mdf)
image(mdf)
hist(mdf)
```

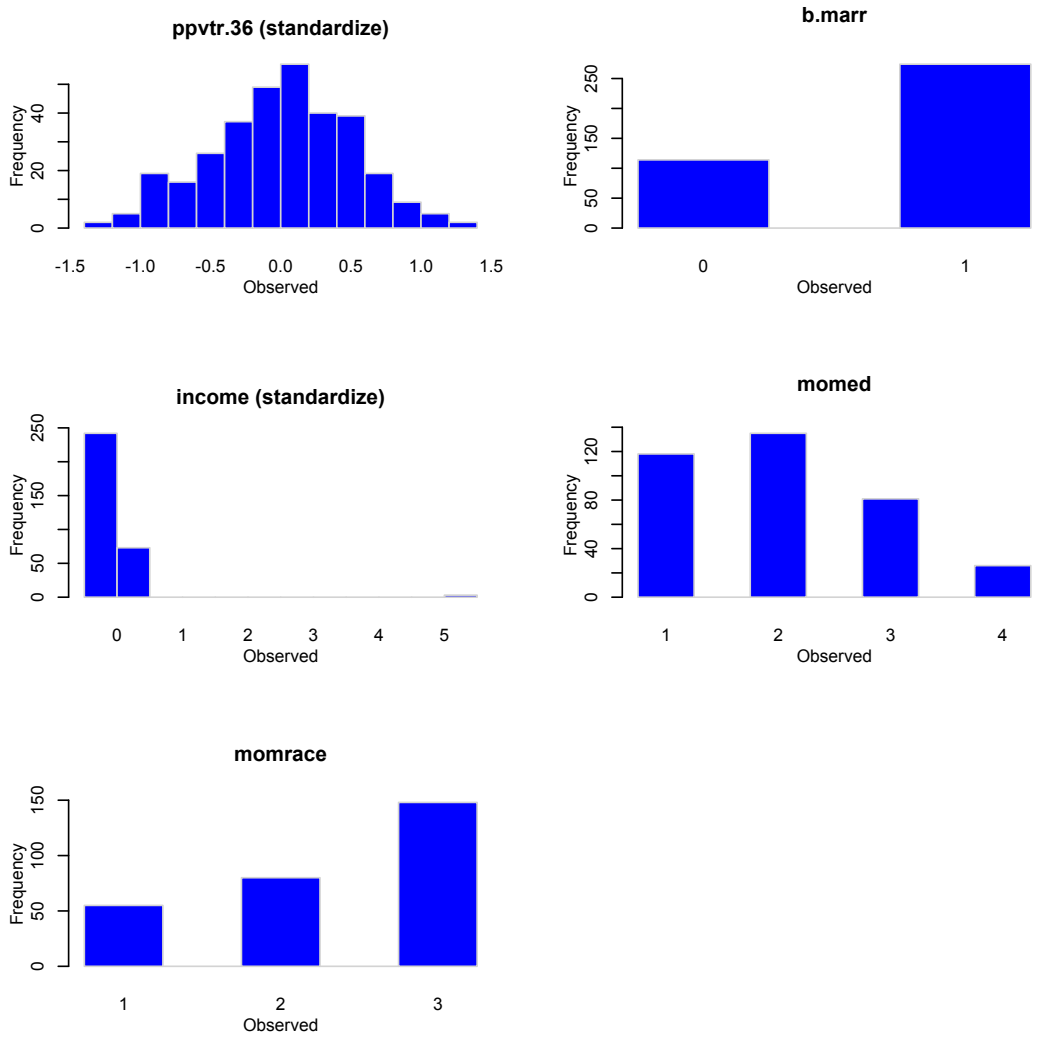
`summary` produces the same result as the `summary` method for a `data.frame`

```
summary(mdf)
```

```
   ppvtr.36       first     b.marr         income            momage
 Min.   : 41.00   0:226   0   :114   Min.   :       0   Min.   :16.00
 1st Qu.: 74.00   1:174   1   :274   1st Qu.:   8590   1st Qu.:22.00
 Median : 87.00           NA's: 12   Median :  17906   Median :24.00
 Mean   : 85.94                      Mean   :  32041   Mean   :23.75
 3rd Qu.: 99.00                      3rd Qu.:  31228   3rd Qu.:26.00
 Max.   :132.00                      Max.   :1057448   Max.   :32.00
 NA's   : 75.00                      NA's   :     82

 momed      momrace
 1   :118   1   : 55
 2   :135   2   : 80
 3   : 81   3   :148
 4   : 26   NA's:117
 NA's: 40
```

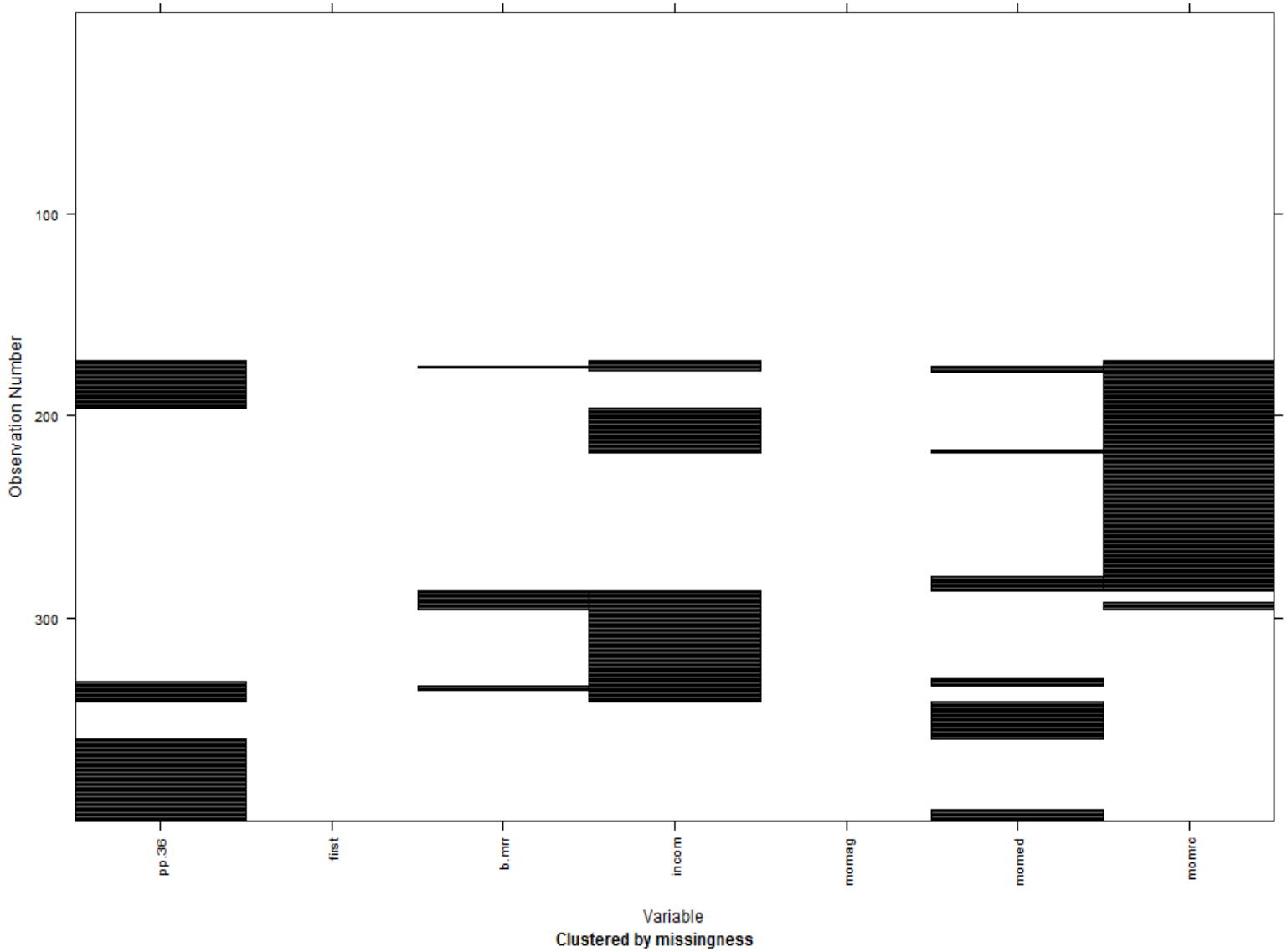
`hist` shows histograms of the observed variables that have missingness:

```
hist(mdf)
```

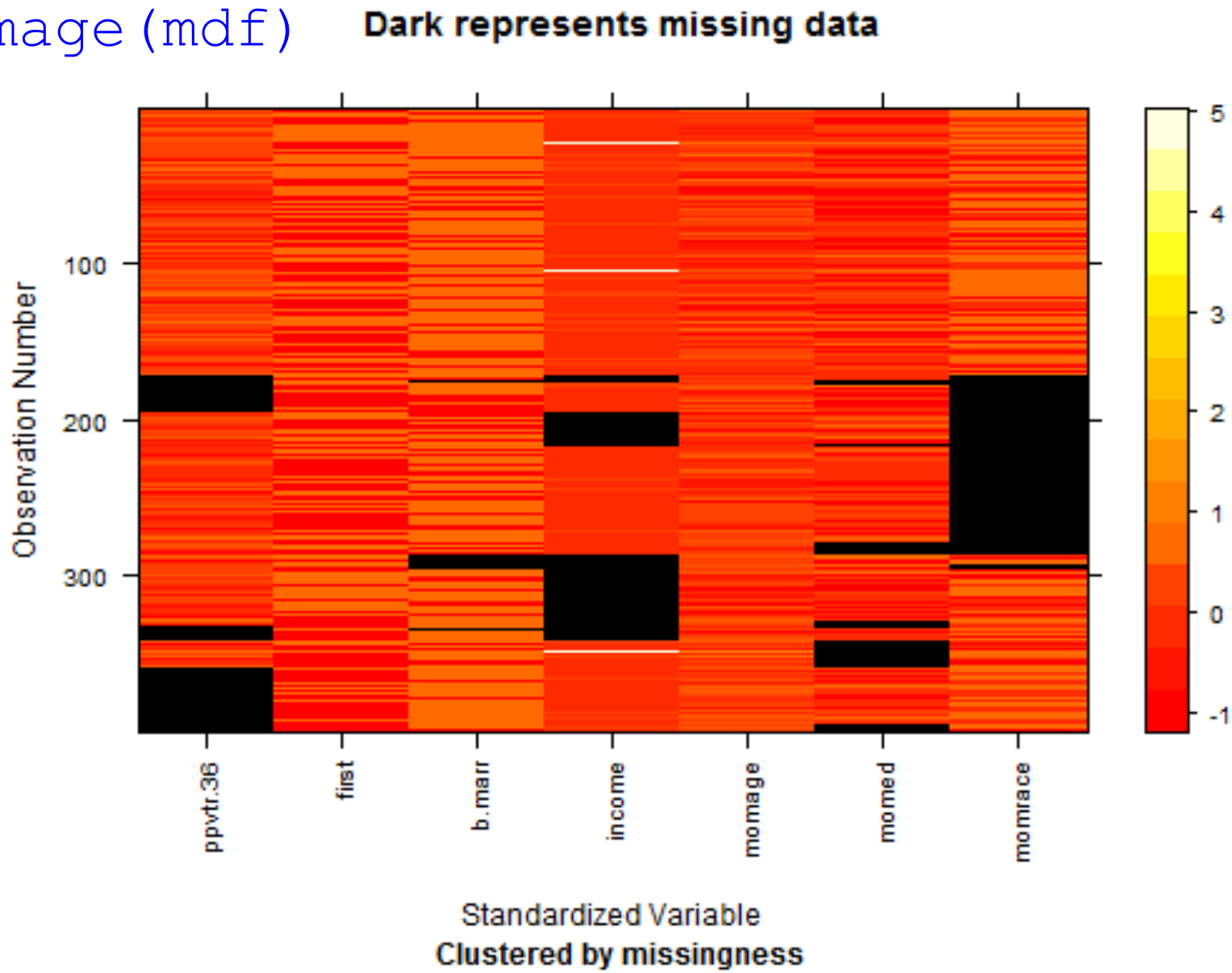


```
> image(mdf, grayscale=TRUE)
```

plots an image to visualize the pattern of missingness



```
image(mdf)
```


Dark represents missing data
Observation Number
100
200
300
ppvtr.36
first
b.marr
income
momage
momed
momrace
5
4
3
2
1
0
-1
Standardized Variable
Clustered by missingness

Missing data patterns

There are 20 missing data patterns whose frequency table is

```
> tabulate(mdf@patterns)
[1] 172  61  35  34  18  18  20   2   6   7   6   6   3   2   2   2   3   1   1   1

> levels(mdf@patterns)
[1] "nothing"                          "momrace"                                "ppvtr.36"
[4] "income"                           "momed"                                  "ppvtr.36, momrace"
[7] "income, momrace"                  "income, momed"                          "ppvtr.36, momed"
[10] "momed, momrace"                  "ppvtr.36, income"                       "b.marr, income"
[13] "ppvtr.36, income, momrace"       "ppvtr.36, b.marr, income"               "ppvtr.36, income, momed"
[16] "income, momed, momrace"          "b.marr, income, momrace"                "ppvtr.36, momed, momrace"
[19] "ppvtr.36, income, momed, momrace" "ppvtr.36, b.marr, income, momed, momrace“
```

Type `mdf@patterns` to access the corresponding pattern for every observation

(3) Examine defaults to see if they make sense

```
> show(mdf)

                        type missing method  model
ppvtr.36          continuous      75    ppd linear
first                 binary       0   <NA>   <NA>
b.marr                binary      12    ppd  logit
income            continuous      82    ppd linear
momage            continuous       0   <NA>   <NA>
momed    ordered-categorical      40    ppd ologit
momrace  ordered-categorical     117    ppd ologit

              family     link transformation
ppvtr.36    gaussian identity    standardize
first           <NA>     <NA>           <NA>
b.marr      binomial    logit           <NA>
income      gaussian identity    standardize
momage          <NA>     <NA>    standardize
momed    multinomial    logit           <NA>
momrace  multinomial    logit           <NA>
```

Ordered and unordered categorical variables

Ordered and unordered categorical variables require special attention

- If such a variable has any missing data it should be included in your dataset as a single variable with multiple levels
- If these variables are coded as "factors" in R then the `mi` program will understand that they are categorical (you can convert using the as.factor() command)
- Otherwise you can explicitly change the status using the change() command in the `mi` package
- unordered categoricals will be imputed using multinomial logit
- ordered categoricals will be imputed using ordered logit

(3) Examine defaults to see if they make sense

```
> show(mdf)
```

variable name

```
                          type missing method  model
ppvtr.36             continuous      75    ppd linear
first                    binary       0   <NA>   <NA>
b.marr                   binary      12    ppd  logit
income               continuous      82    ppd linear
momage               continuous       0   <NA>   <NA>
momed      ordered-categorical      40    ppd ologit
momrace    ordered-categorical     117    ppd ologit
```

variable `type` determines the default imputation model and transformation. Options are continuous, binary, unordered-categorical, ordered-categorical, positive-continuous, nonnegative-continuous, proportion.

(3) Examine defaults to see if they make sense

```
> show(mdf)
```

number of missing values

```
                          type missing method  model
ppvtr.36             continuous      75    ppd linear
first                    binary       0   <NA>   <NA>
b.marr                   binary      12    ppd  logit
income               continuous      82    ppd linear
momage               continuous       0   <NA>   <NA>
momed      ordered-categorical      40    ppd ologit
momrace    ordered-categorical     117    ppd ologit
```

method used to impute after model is fit:
drawing from the posterior predictive distribution (ppd) or predictive mean matching (pmm)
(can also do mean, median or conditional mean imputation)

(3) Examine defaults to see if they make sense

```
> show(mdf)
```

```
                          type missing method  model
ppvtr.36            continuous      75    ppd linear
first                   binary       0   <NA>   <NA>
b.marr                  binary      12    ppd  logit
income              continuous      82    ppd linear
momage              continuous       0   <NA>   <NA>
momed     ordered-categorical      40    ppd ologit
momrace   ordered-categorical     117    ppd ologit
```

model used to fit data: specification corresponds to the "family" argument in standard glm models in R (e.g. "normal" implies standard linear regression, "binomial" implies logistic regression, in this matrix "logit" standards for either logistic regression, ordered logit, or multinomial logit models, depending on the variable type)

(3) Examine defaults to see if they make sense

```
> show(mdf)

                        type missing method  model
ppvtr.36          continuous      75    ppd linear
first                 binary       0   <NA>   <NA>
b.marr                binary      12    ppd  logit
income            continuous      82    ppd linear
momage            continuous       0   <NA>   <NA>
momed    ordered-categorical      40    ppd ologit
momrace  ordered-categorical     117    ppd ologit

               family     link transformation
ppvtr.36    gaussian identity    standardize
first            <NA>     <NA>           <NA>
b.marr      binomial    logit           <NA>
income      gaussian identity    standardize
momage           <NA>     <NA>    standardize
momed    multinomial    logit           <NA>
momrace  multinomial    logit           <NA>
```

In the absence of a model with a clear "buzzword" (like "probit") the user can define supported generalized linear model by specifying the family and link.

(3) Examine defaults to see if they make sense

```
> show(mdf)

                        type missing method  model
ppvtr.36          continuous      75    ppd linear
first                 binary       0   <NA>   <NA>
b.marr                binary      12    ppd  logit
income            continuous      82    ppd linear
momage            continuous       0   <NA>   <NA>
momed    ordered-categorical      40    ppd ologit
momrace  ordered-categorical     117    ppd ologit

               family     link transformation
ppvtr.36    gaussian identity    standardize
first           <NA>     <NA>           <NA>
b.marr      binomial    logit           <NA>
income      gaussian identity    standardize
momage          <NA>     <NA>    standardize
momed    multinomial    logit           <NA>
momrace  multinomial    logit           <NA>
```

transformation that has been performed to the data: some (log, standardize) are built in, others can be user-specified

(3) Examine defaults to see if they make sense

> show(mdf)

```
                        type missing method default.model transformation
ppvtr.36          continuous      75    ppd        normal    standardize
first                 binary       0   <NA>          <NA>           <NA>
b.marr                binary      12    ppd         logit           <NA>
income            continuous      82    ppd        normal    standardize
momage            continuous       0   <NA>          <NA>    standardize
momed    ordered-categorical      40    ppd         logit           <NA>
momrace  ordered-categorical     117    ppd         logit           <NA>
```

(4) Make changes to imputation models

```
> mdf <- change(mdf, y = c("momed", "momrace"),
  what = "type", to = "un")
> show(mdf)
```

```
                          type missing method  model
ppvtr.36             continuous      75    ppd linear
first                    binary       0   <NA>   <NA>
b.marr                   binary      12    ppd  logit
income               continuous      82    ppd linear
momage               continuous       0   <NA>   <NA>
momed    unordered-categorical      40    ppd mlogit
momrace  unordered-categorical     117    ppd mlogit
```

(5) Impute until converged

```
> imputations <- mi(mdf)
> converged <- mi2BUGS(imputations)
> print(converged)
> plot(converged)
```

```
Chain 1
Iteration: 0.....
Iteration: 1.....
Iteration: 2.....
Iteration: 3.....
Iteration: 4.....
Iteration: 5.....
Iteration: 6.....
Iteration: 7.....
Iteration: 8.....
Iteration: 9.....
Iteration: 10.....
Iteration: 11.....
Iteration: 12.....
Iteration: 13.....
Iteration: 14.....
Iteration: 15.....
Iteration: 16.....
Iteration: 17.....
Iteration: 18.....
Iteration: 19.....
Iteration: 20.....
Iteration: 21.....
Iteration: 22.....
Iteration: 23.....
Iteration: 24.....
Iteration: 25.....
Iteration: 26.....
Iteration: 27.....
Iteration: 28.....
Iteration: 29.....
Iteration: 30.....
Estimating models on completed data
```

Output from

`imputations <- mi(mdf)`

for the first chain

What does convergence mean?

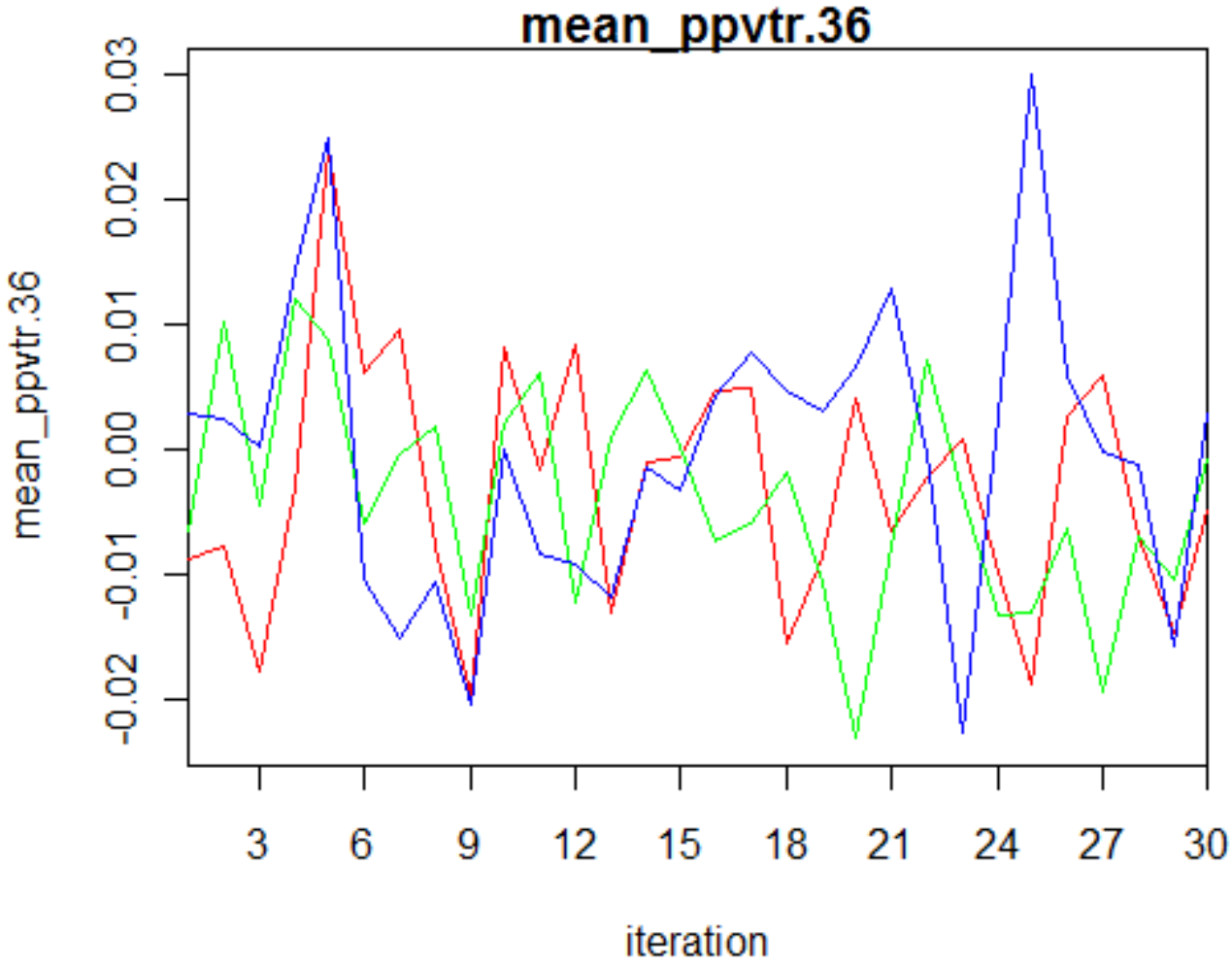
- We run four chains to check convergence
- Each chain represents the process described earlier
  - first we fill in starting values
  - we proceed by stochastically imputing each variable with missing data in turn.
  - after each of these variables has been imputed once we have achieved one iteration of that chain
  - At that end of each iteration we save 2 statistics for each variable with missing data: the mean of the imputed values, the standard deviation of the imputed values.
  - Then we cycle through again to complete the second iteration. And so on.

What does convergence mean?

- Each one of the lines in the plots on the next slides follows the values for one of these statistics across iterations
- The other chains are created in the same way except each starts with a different set of starting values and so takes a different path.
- Once we have reached convergence it means that we are imputing from the "stationary distribution" (which we hope is the right distribution).

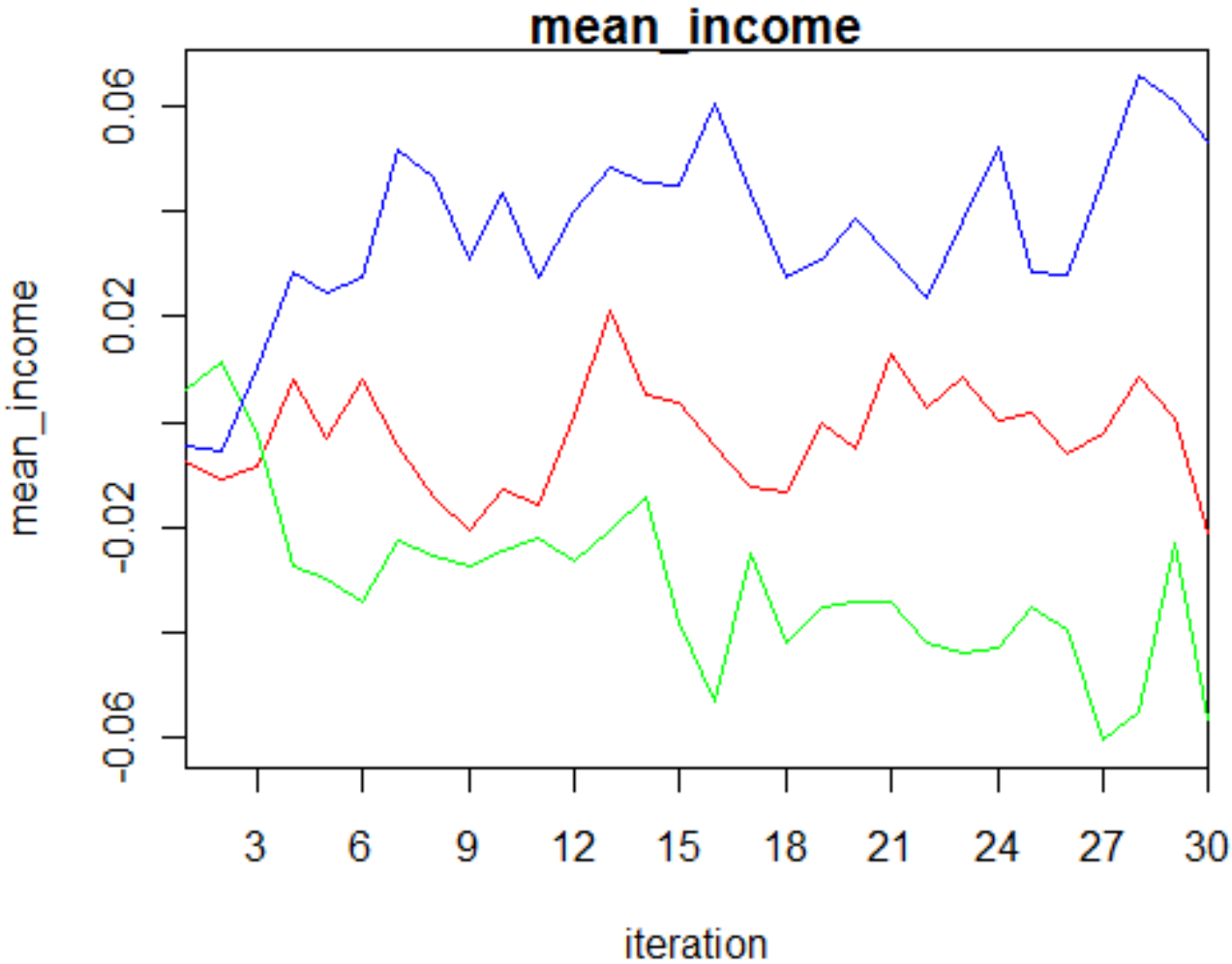
- We can visually assess whether the chains have converged by seeing whether they mix (they cover the same territory).
- We can numerically assess this with the Rhat statistic. And yes we'd like them to be close to 1 (ideally less than 1.1).

What does convergence mean?



now looking at convergence for the income imputations

```
> traceplot(converged)
```



not so good…
how can we quantify?

Rhat statistic (also called "estimated potential scale reduction") (Gelman and Rubin)

$$\hat{R} = \sqrt{\frac{\frac{N-1}{N}W + \frac{1}{N}B}{W}}$$

$$B = N\,\mathrm{var}(\bar{x}^m)$$

$$W = \frac{1}{m}\sum_m \mathrm{var}(x^m)$$

- x is a statistic of your choice; we look at the mean and sd of the *completed* data for each variable with missing data
- N is number iterations per chain
- M is number of chains, m indicates the mth chain

Convergence diagnostics: pay attention to “Rhat”

```
Rhats(imputations)
mean_ppvtr.36   mean_b.marr   mean_income
0.9998835       1.1728611     1.2706806

mean_momed  mean_momrace
1.0267368     1.0233137

sd_ppvtr.36     sd_b.marr     sd_income
0.9932876     1.1723309     1.0581659
sd_momed    sd_momrace
1.0291674     0.9840442
```

What if not converged? Try some more iterations…

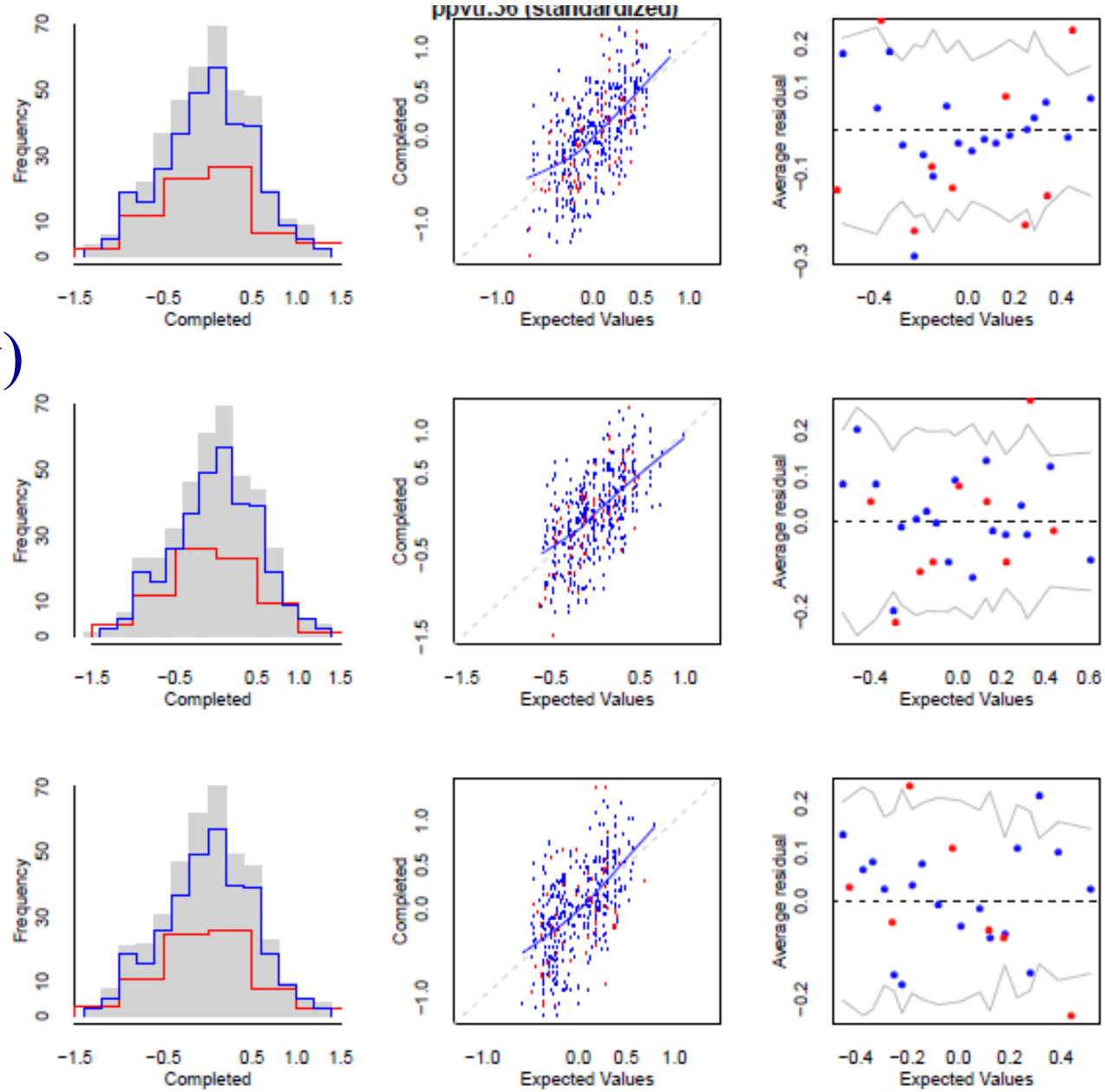
```
> imputations <- mi(imputations)
```

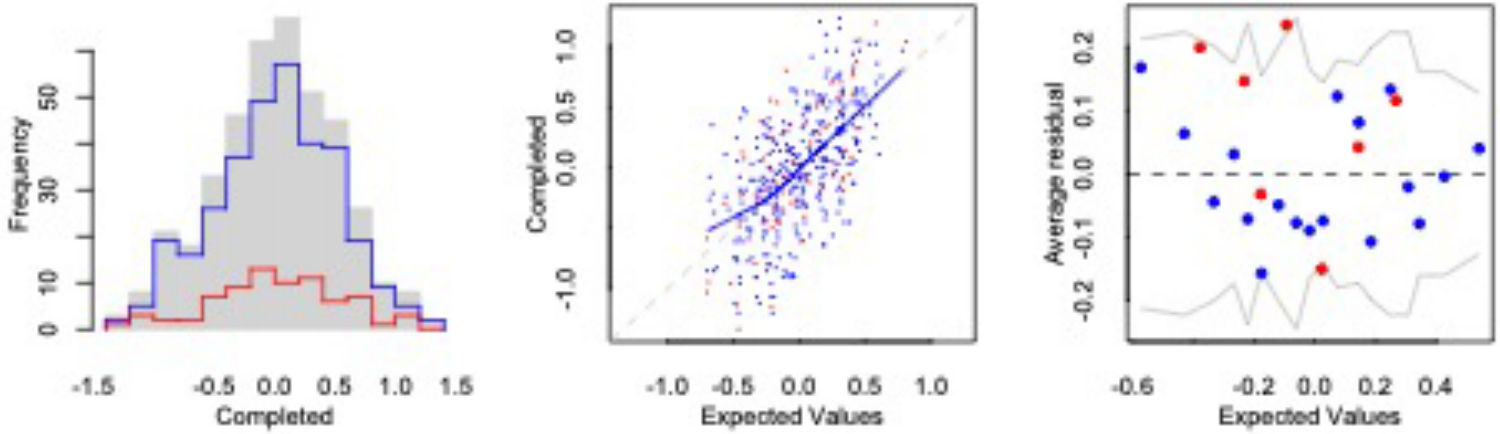
```
> Rhats(imputations)
mean_ppvtr.36   mean_b.marr   mean_income    mean_momed  mean_momrace
0.9971384     1.0074346     1.0236755     0.9934136     1.0027449

sd_ppvtr.36     sd_b.marr     sd_income      sd_momed    sd_momrace
0.9945078     1.0070225     0.9953870     0.9922516     0.9969038
```

(6) Plot diagnostics

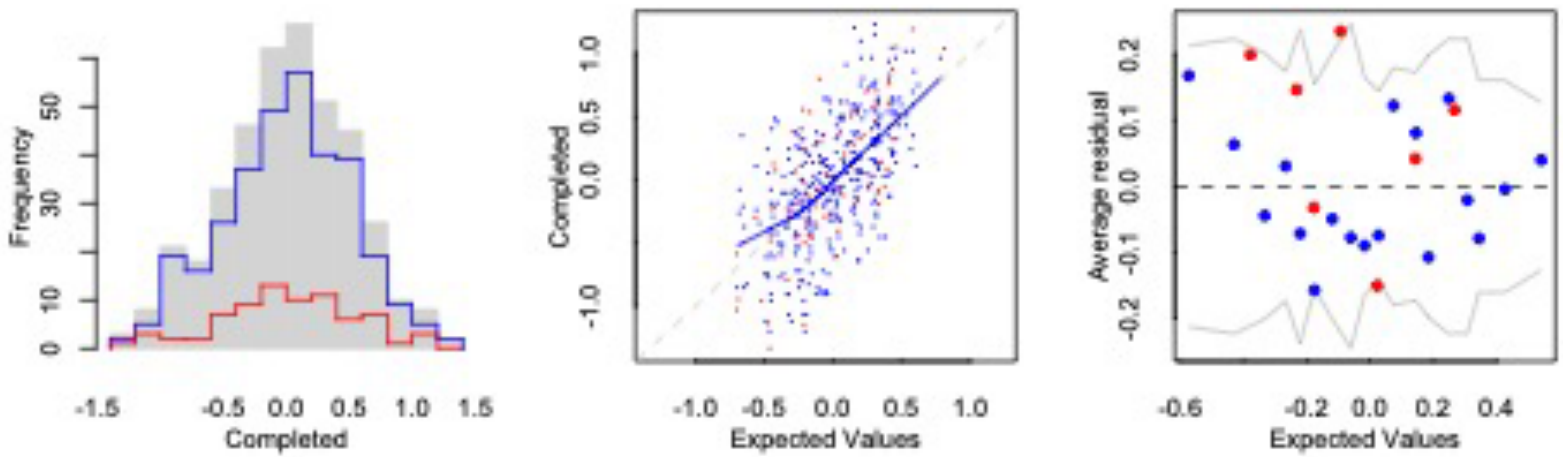
```
> plot(imputations)
> hist(imputations)
```

one set of plots (a row) for each chain

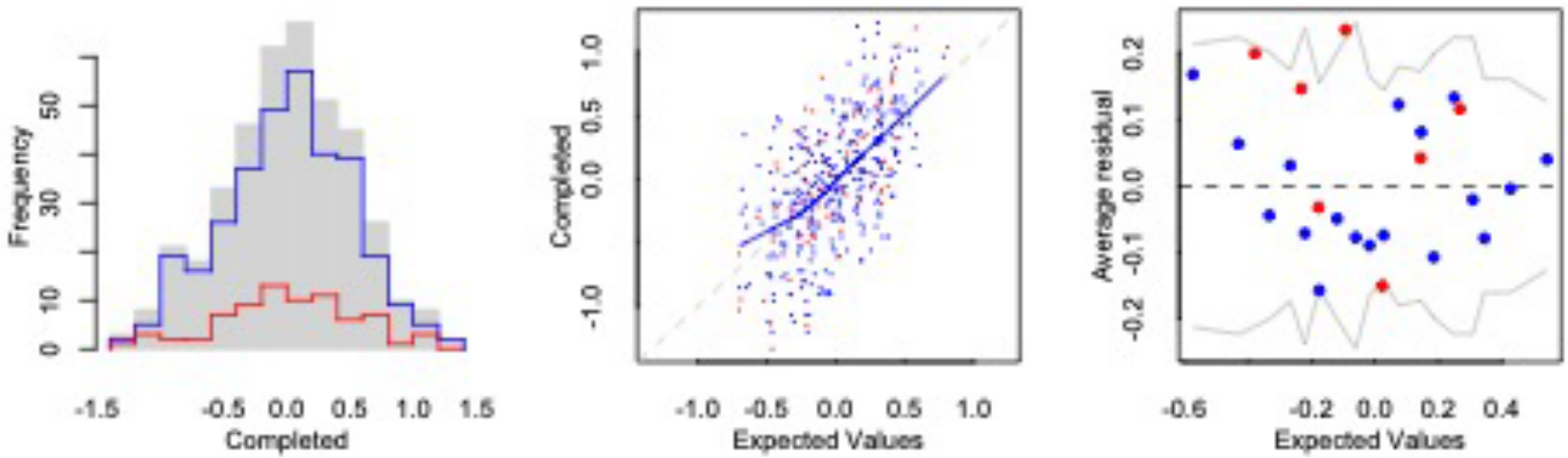




overlaid histograms: grey shaded for completed, blue outline for observed, red outline for imputed

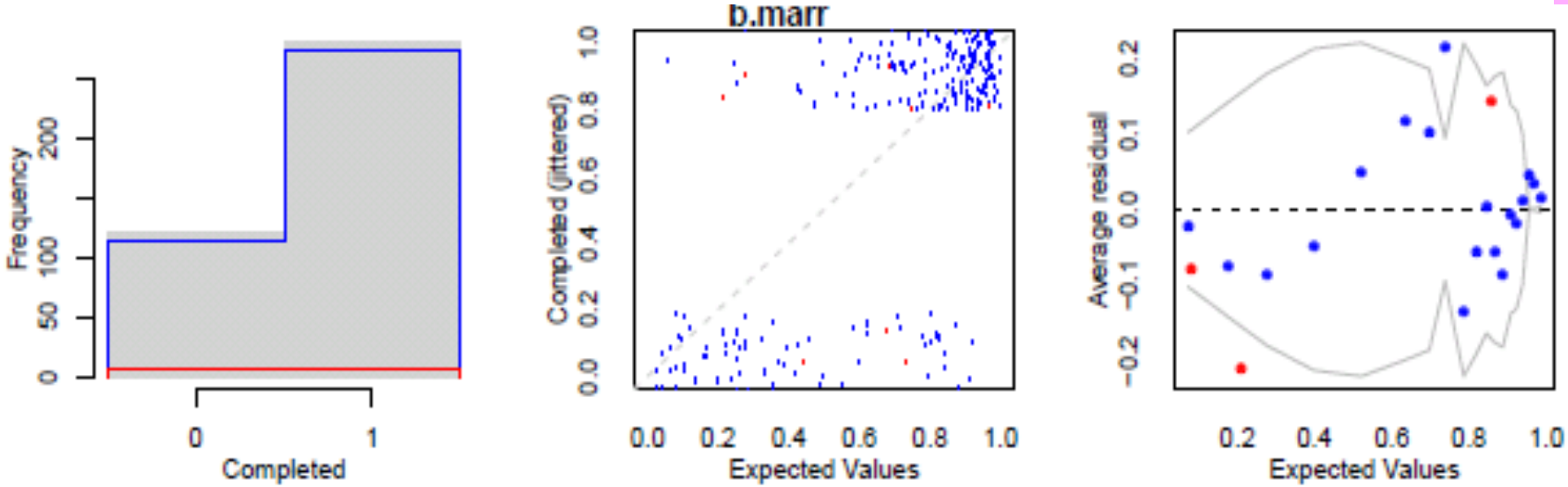


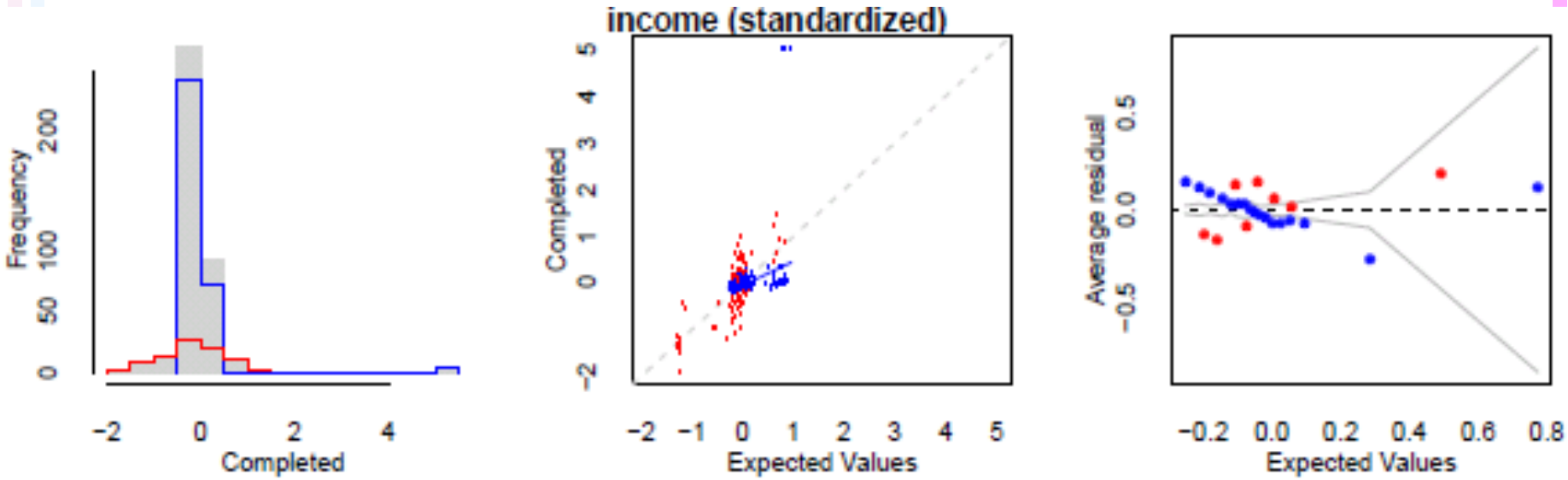
expected values from fitted models plotted against observed (blue) and imputed (red) data points

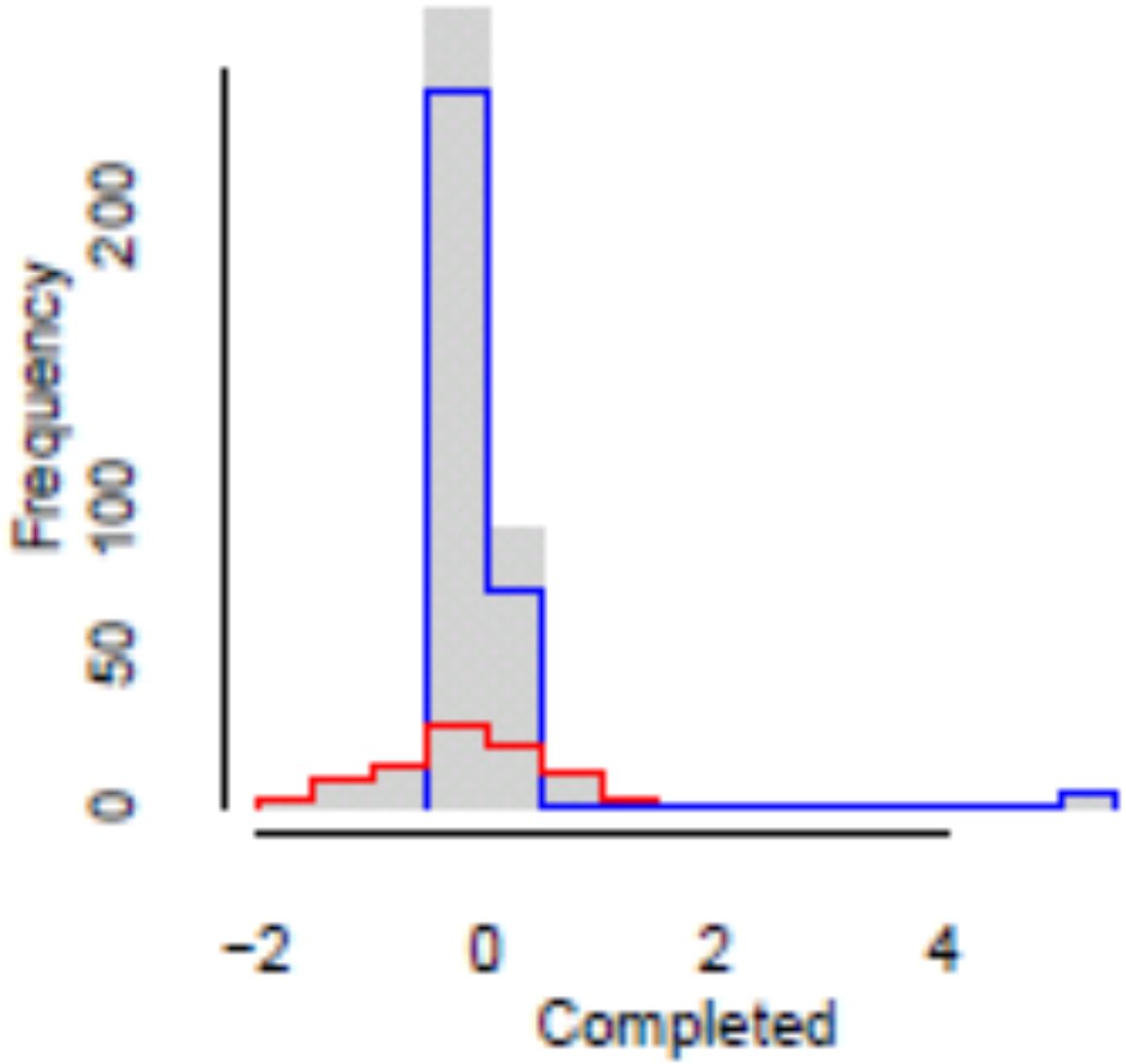


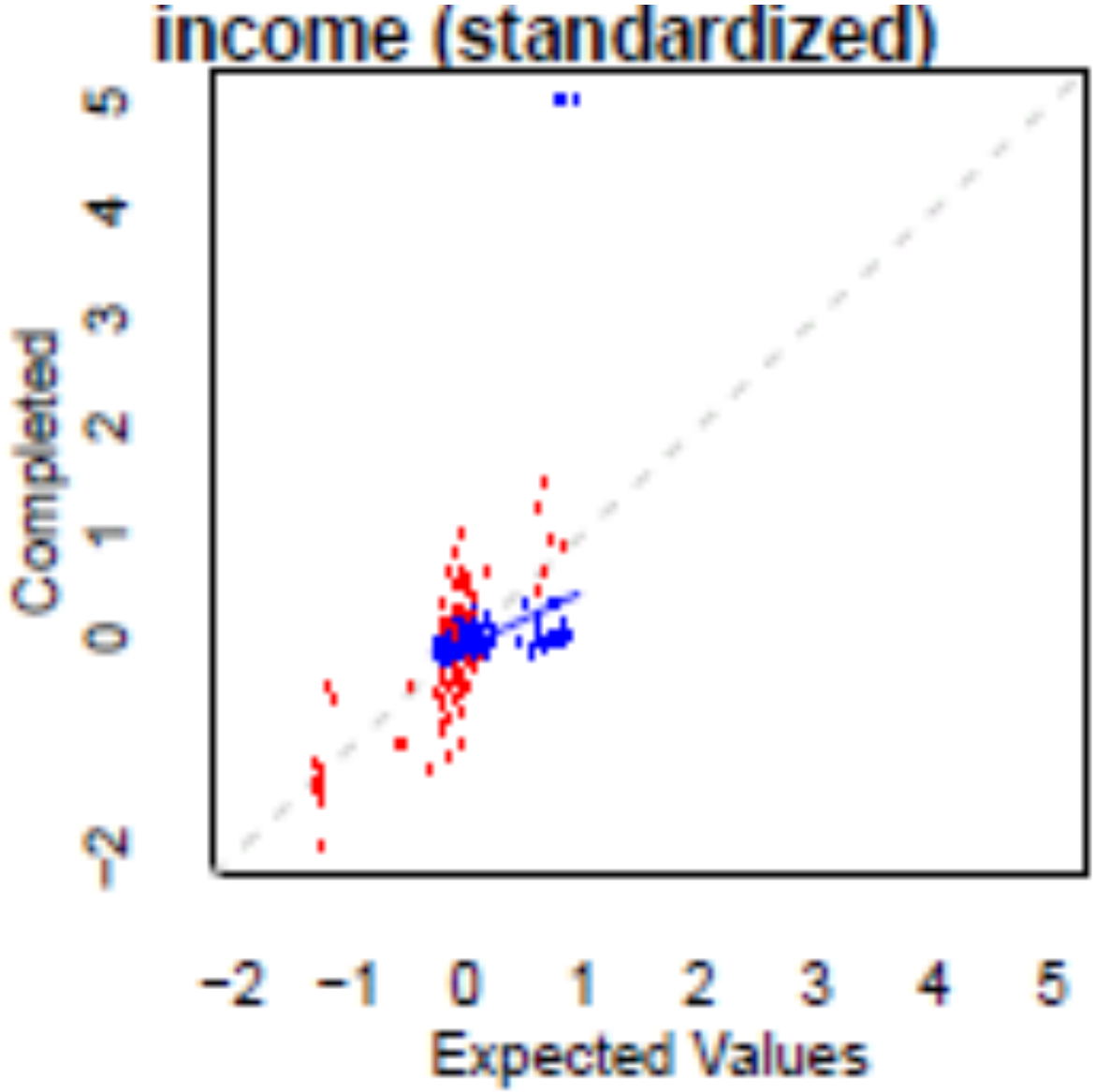
Binned residual plot that plots the average of residuals in bins against the expected values with 95% error bounds. Each point in a binned residual plot is the average of the points that fall in each "bin" (interval of the variable on the *x*-axis) from a standard residual point.

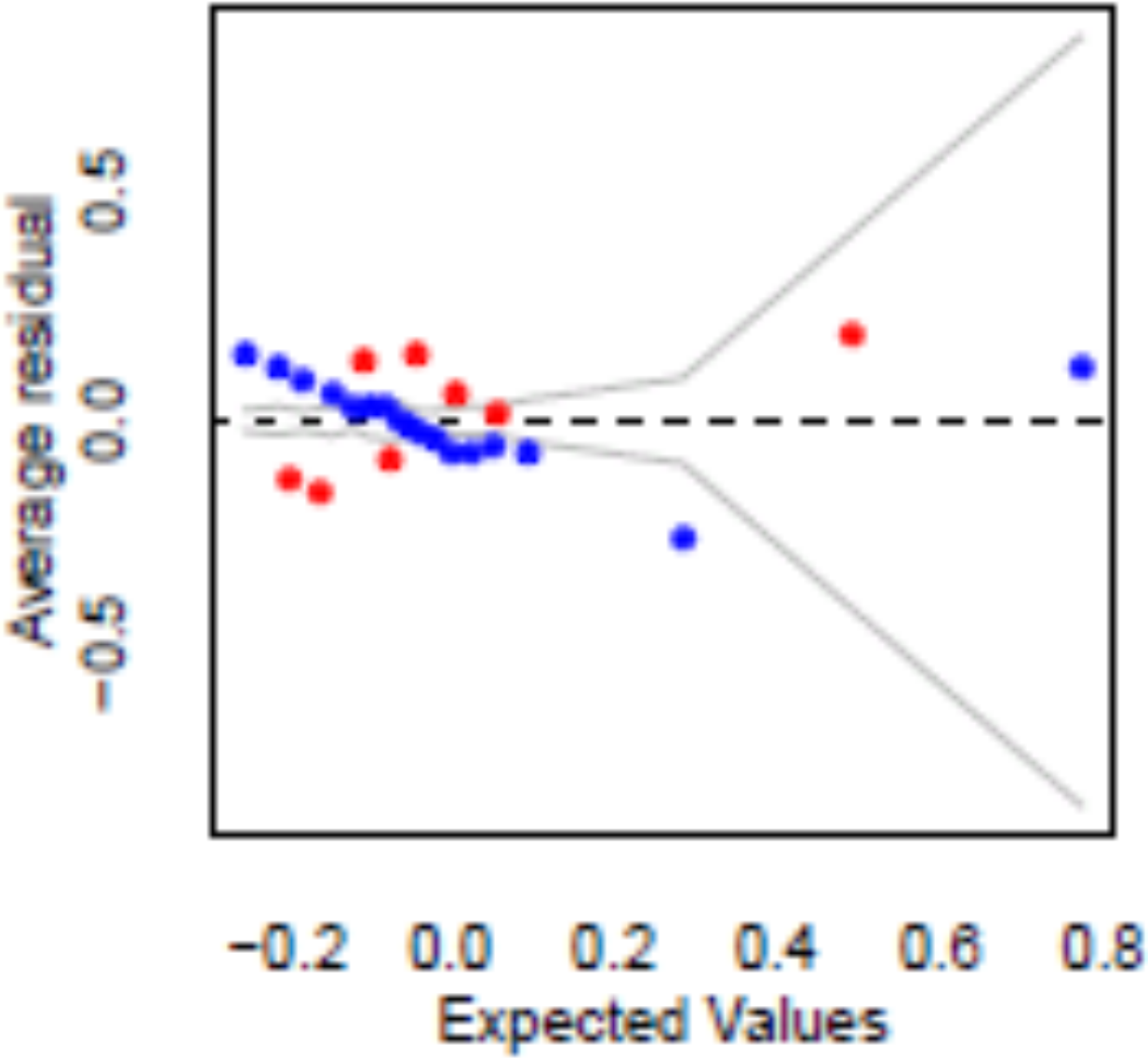
Ref: Gelman,Goegebeur, Tuerlinckx, and Van Mechelen (2000)

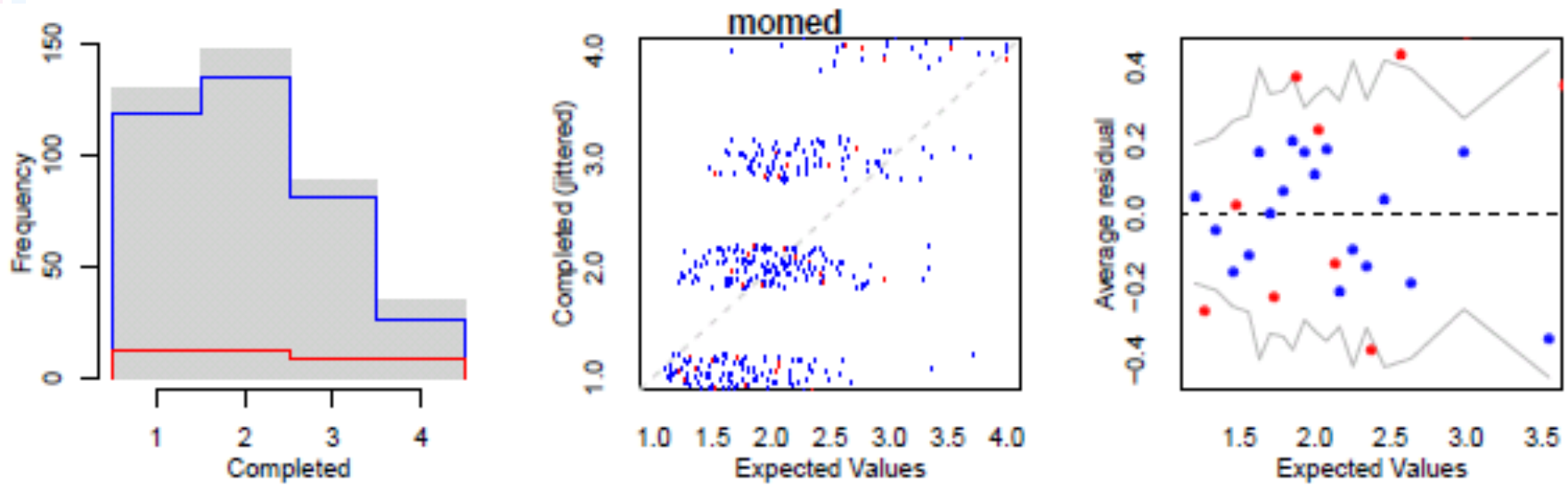

Frequency
0
50
100
200
0
1
Completed
b.marr
Completed (jittered)
0.0
0.2
0.4
0.6
0.8
1.0
Expected Values
Average residual
−0.2
−0.1
0.0
0.1
0.2
0.2
0.4
0.6
0.8
1.0
Expected Values

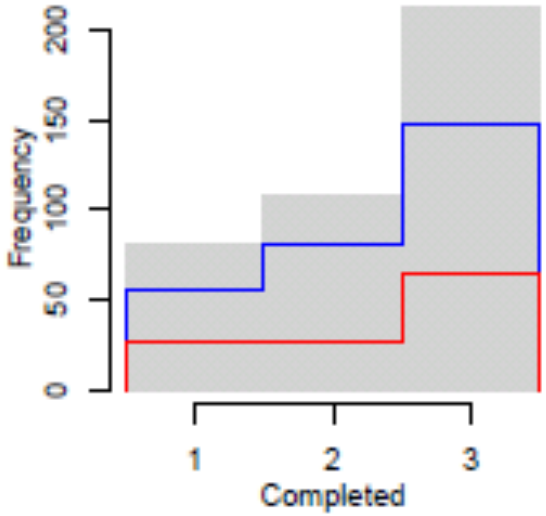

income (standardized)
Frequency
0
50
100
200
−2
0
2
4
Completed
Completed
−2
0
1
2
3
4
5
−2
−1
0
1
2
3
4
5
Expected Values
Average residual
−0.5
0.0
0.5
−0.2
0.0
0.2
0.4
0.6
0.8
Expected Values

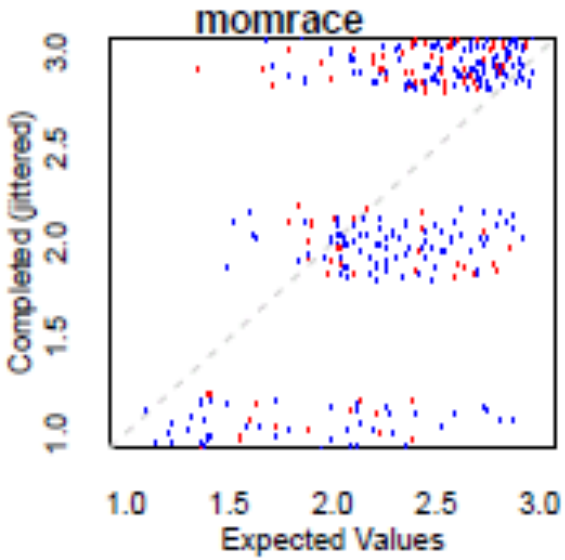

Frequency
0
50
100
200
−2
0
2
4
Completed

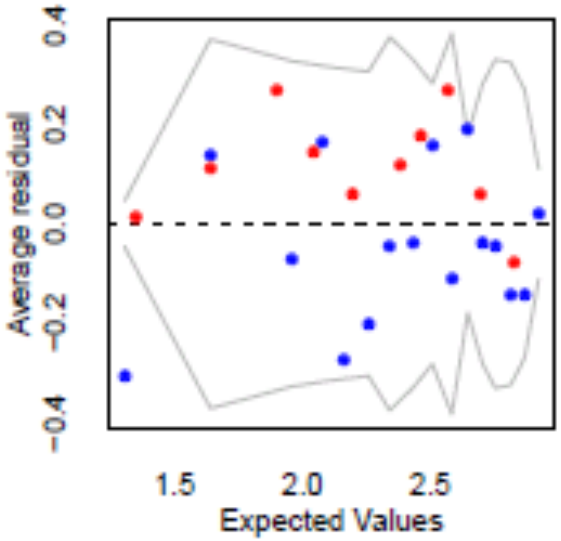

income (standardized)
Completed
Expected Values
−2
0
1
2
3
4
5
−2
−1
0
1
2
3
4
5


Average residual
0.5
0.0
−0.5
−0.2
0.0
0.2
0.4
0.6
0.8
Expected Values


momed
Frequency
0
50
100
150
1
2
3
4
Completed
Completed (jittered)
1.0
2.0
3.0
4.0
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Expected Values
Average residual
−0.4
−0.2
0.0
0.2
0.4
1.5
2.0
2.5
3.0
3.5
Expected Values


Frequency
0
50
100
150
200
1
2
3
Completed


momrace
Completed (jittered)
1.0
1.5
2.0
2.5
3.0
Expected Values


Average residual
−0.4
−0.2
0.0
0.2
0.4
1.5
2.0
2.5
Expected Values

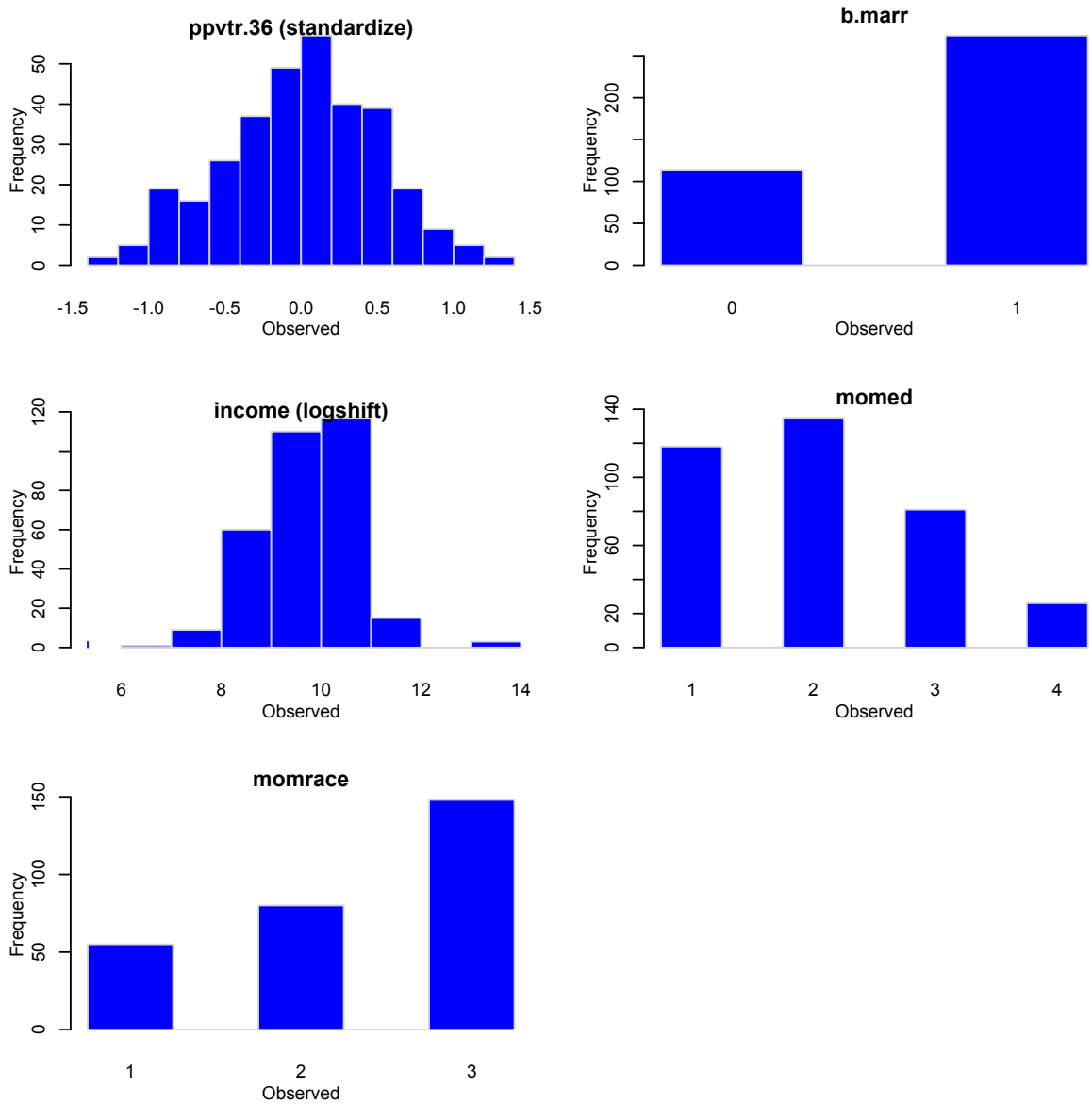
Iterate steps (4)-(6) (if necessary)

Let's treat income as "non-negative continuous," a type that creates two new variables to replace the original

1) an indicator variable for whether the observation is 0 or not
2) the second forces a log transformation for the positive values and treats the 0 values as missing

```
mdf <- change(mdf, y = "income", what = "type", to = "nonn")

                          type missing method  model
ppvtr.36             continuous      75    ppd linear
first                    binary       0   <NA>   <NA>
b.marr                   binary      12    ppd  logit
income   nonnegative-continuous      82    ppd linear
momage               continuous       0   <NA>   <NA>
momed     unordered-categorical      40    ppd mlogit
momrace   unordered-categorical     117    ppd mlogit
```

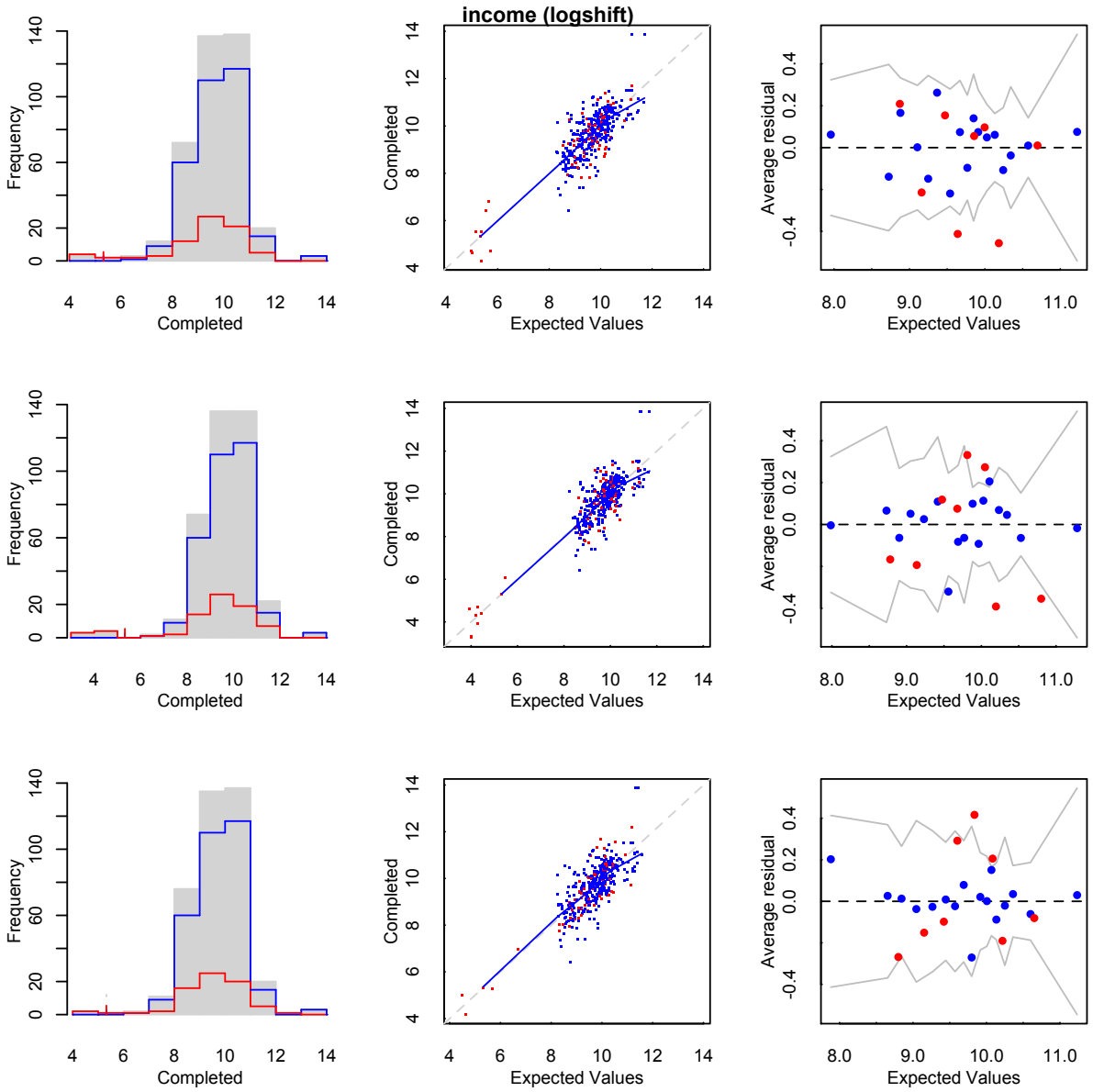
Pre-imputation



Now after running 60 iterations…..

```
> imputations <- mi(mdf, n.iter=60)
```

```
> Rhats(imputations)
mean_ppvtr.36   mean_b.marr   mean_income    mean_momed  mean_momrace
0.9954509     0.9933845     1.0142938     1.0259769     1.0164465

sd_ppvtr.36     sd_b.marr     sd_income      sd_momed    sd_momrace
0.9969167     0.9931634     1.0482143     1.0329785     0.9969921
```


income (logshift)
Frequency
Completed
Expected Values
Average residual

Iterate steps (4)-(6) (if needed)

Now using PMM to impute

```
mdf <- change(mdf, y = "income", what ="imputation_method",
to = "pmm")
```

```
> show(mdf)
type missing method  model
ppvtr.36             continuous      75    ppd linear
first                    binary       0   <NA>   <NA>
b.marr                   binary      12    ppd  logit
income   nonnegative-continuous      82    pmm linear
momage               continuous       0   <NA>   <NA>
momed     unordered-categorical      40    ppd mlogit
momrace   unordered-categorical     117    ppd mlogit
```

What is PMM doing:

1. Generate predicted values for *x* for *all* cases, *both* those with data missing on *x* and those with data present.

2. For each case with missing *x*, identify a set of cases with observed *x* whose *predicted* values are close to the predicted value for the case with missing data.

3. From among those close cases, randomly choose one and assign its *observed* value to substitute for the missing value.

Results: PMM produces imputed values that are much more like real values. If the original variable is skewed, the imputed values will also be skewed.

Paul Allison has criticized the method:
https://statisticalhorizons.com/predictive-mean-matching

(7) Run pooled analysis

(Let's use 5 imputed datasets)

```
> analysis <- pool(ppvtr.36 ~ first + b.marr + scale(income) +
   momage + momed + momrace, imputations, m=5)
> display(analysis)

glm(formula = ppvtr.36 ~ first + b.marr + scale(income) +
    momage + momed + momrace, data = imputations, m = 5)
              coef.est coef.se
(Intercept)    72.36     7.00
first1          3.59     1.63
b.marr1         4.74     1.97
scale(income)   0.66     0.80
momage         -0.06     0.28
momed2          4.03     1.89
momed3          9.00     2.28
momed4         14.36     3.51
momrace2       -5.41     2.45
momrace3       13.58     2.27
n = 400, k = 10
residual deviance = 87938.5, null deviance = 139952.0
   (difference = 52013.5)
overdispersion parameter = 219.8
residual sd is sqrt(overdispersion) = 14.83
```

Compare to complete case analysis

```
> glm(formula = ppvtr.36 ~ first + b.marr + scale(income) + momage
   + factor(momed) + factor(momrace), family = gaussian, data =
   nlsyV)
                  coef.est coef.se
(Intercept)       64.98    10.67
first              5.27     2.72
b.marr             6.20     3.37
scale(income)      0.62     1.24
momage             0.18     0.45
factor(momed)2     4.21     3.04
factor(momed)3     9.51     3.67
factor(momed)4     7.42     6.36
factor(momrace)2 -3.89      4.19
factor(momrace)3 14.15      3.89
---
  n = 172, k = 10
  residual deviance = 40100.9, null deviance = 63870.6 (difference
   = 23769.7)
```

more on mi options

mi(y, model, ...) defaults:

- n.iter = 30
- n.chain = (depends on processor)
- max.minutes = 20
- seed = NA,
- verbose = TRUE

next week…

- more mi options
- extracting datasets
- more diagnostics
- …