Missing Data

More Complicated Fixes

Outline of 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

Notation

Let R be the **matrix** of variables R_1, …, R_p, corresponding to variables in our dataset, X_1, …, X_p, that indicate whether a given value of the corresponding X variable is observed (= 1) or missing (= 0)

X_1	X_2	X_3	X_4
0	2	5	2
0	1	3	?
0	2	4	3
0	3	5	?
1	5	?	?
1	4	?	12
1	6	11	?
1	6	12	16

R_1	R_2	R_3	R_4
1	1	1	1
1	1	1	0
1	1	1	1
1	1	1	0
1	1	0	0
1	1	0	1
1	1	1	0
1	1	1	1

Recall regression imputation from last time:

- Suppose only one variable, *Y*, has missing data.
- Begin by dividing the sample into those with variable *Y* observed, and those for whom *Y* is missing;
- Within the complete cases, build a model that predicts the values of that variable
- Apply that regression equation in the second group to predict (impute) the missing values
- This method becomes much more complicated when there are many variables with missing data

Recall regression imputation from last time:

- Conceptually, this is a good way to impute values in the sense that a great deal of information from the individual is used to predict the missing values;
- The higher the correlation between the predictors and *Y*, the better the imputation will be;
- This method forms the heart of the normal-model MI procedures;
- At best, this method will underestimate standard errors;
- Covariances are estimated without bias with this procedure (when certain conditions are met), but variances are too low.

Regression imputation in R

- Regression imputation is simply a form of prediction
- For instance, if the variable that is missing (here, *y*) can be predicted well by a linear regression then we would do something like the following:

```
Ry = as.numeric(!is.na(dat.obs))
dat.cc = dat.obs[Ry == 1, ]
dat.dropped = dat.obs[Ry == 0, ]
mod = lm(y ~ pred1 + pred2, data = dat.cc)
y.imp = predict(mod, newdata = dat.dropped)
dat.imp$var.w.miss[Ry == 0] = y.imp
```

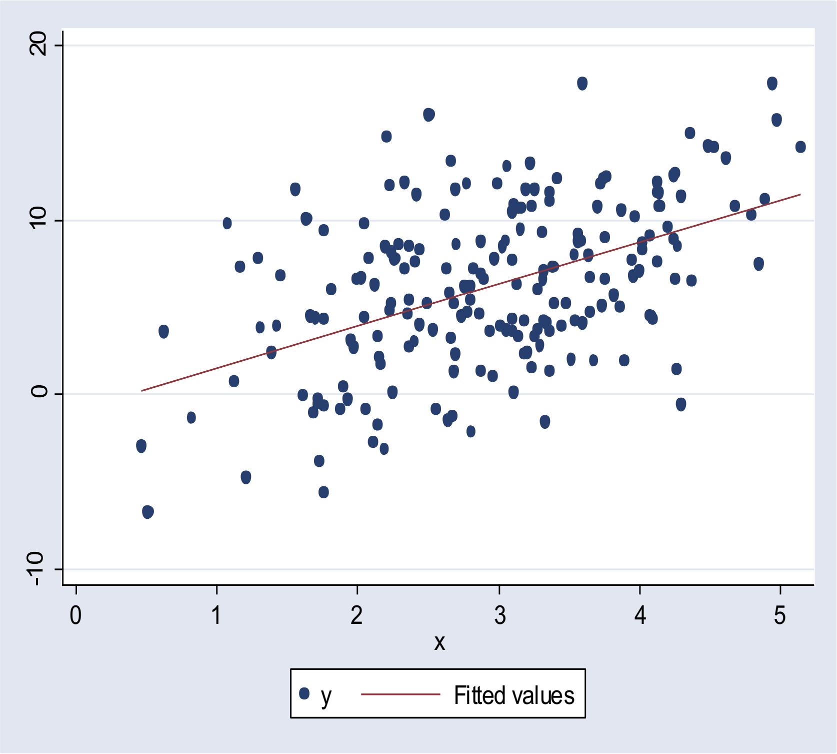
Regression imputation in SPSS

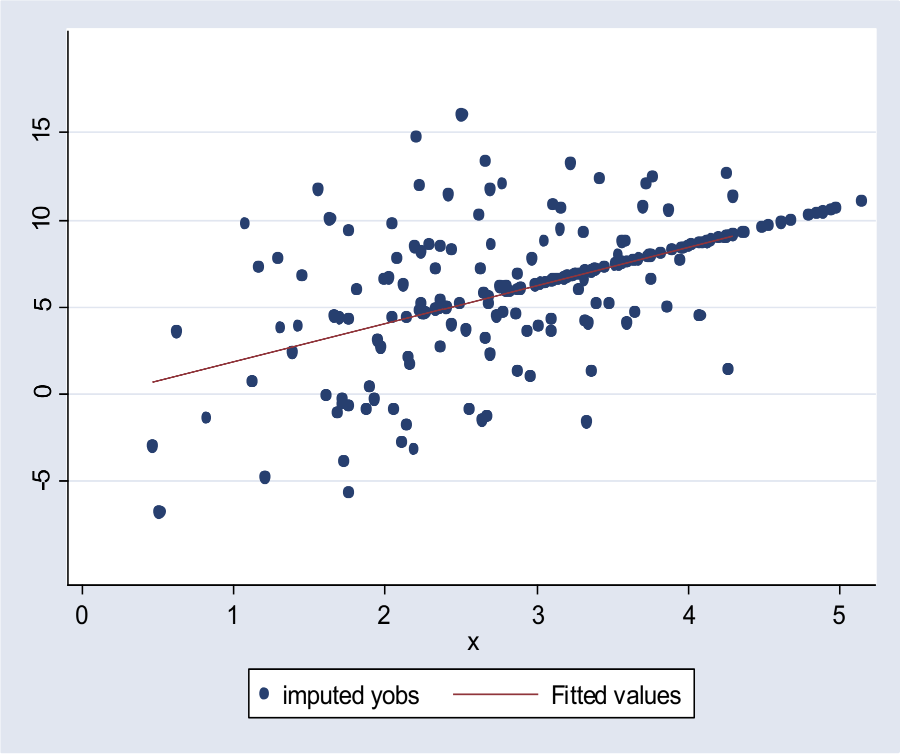
- Select Analyze -> Missing Value Analysis …, then select all numerical variables into Quantitative.
- In SPSS the regression imputation method is available through the MVA procedure when you select EM method.
- Checkmark “Save completed data” in the EM options.
- To impute the mean you need to recode the variable from Transform -> Recode into Same Variables …

Regression imputation issues

- When Y is present, there is always some difference between observed values and the regression line.
- However, with this imputation approach, the imputed values always fall right on the regression line.
- For this reason it is not a recommended method ☹

Illustration of regression imputation





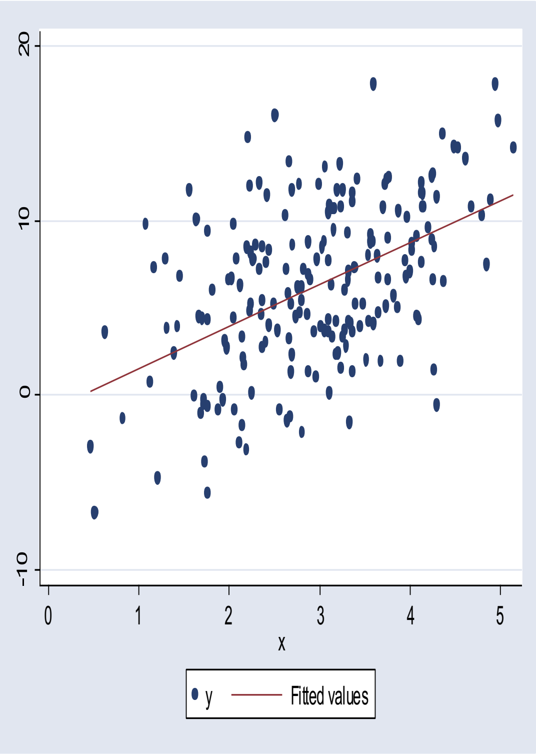
```
y |       Coef.   Std. Err.
x |    2.40121   .3060657
Root MSE = 4.15
```

```
y |       Coef.   Std. Err.
x |    2.195098   .2451244
Root MSE = 3.32
```

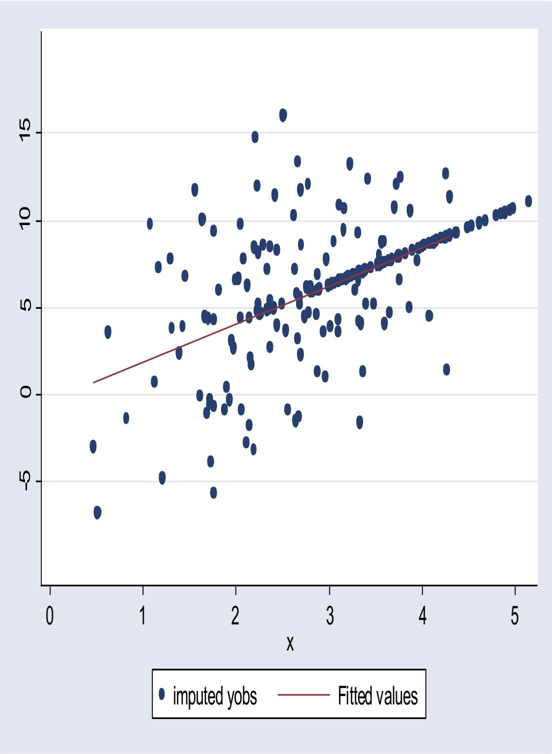
Regression imputation with noise

- Adds back in most of the variability that standard regression imputation removes
- However, can be difficult to implement when there are complex patterns of missing data

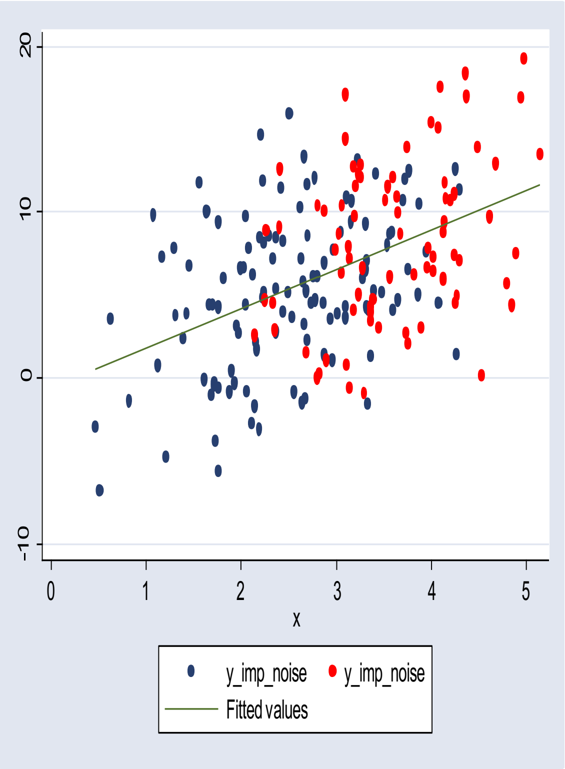
Illustration of regression imputation with noise



```
y |    Coef.   Std. Err
x |  2.4012    .30607
Root MSE = 4.15
```



```
y |   Coef.   Std. Err
x | 2.1951   .24512
Root MSE = 3.32
```



```
y |     Coef.     Std. Err.
x |   2.374209     .327025
Root MSE = 4.43
```

Regression imputation with noise: R

- You already know how to get regression imputations without noise
- Now we just need to add the "noise"
- Create *noise* by using the `rnorm( )` command with n=number of imputed values, mean=0 and sd=the residual standard error [summary(mod)$sigma]

  noise = rnorm(length(y.imp),0,summary(mod)$sigma)
- Just add the random normal draw to your predicted values

  > y.imps = y.imp + noise

How to stochastically impute for binary data?

- Regression imputation can be referred to more generally as “stochastic imputation” (stochastic is a fancy word for random)
- In this case we use the same principle as with imputation using linear regression, but by the `glm` function.
- `mod = glm(y ~ pred1 + pred2, data = dat.cc, family=“binomial”)`
- Then obtain the probabilities of observing 1:
- ps = predict(mod, newdata=dat.dropped, type=“response”)
- Then use these probabilities to generate Bernoulli
- observations:
- y.imps = rbinom(sum(Ry==0), 1, ps)

Stochastic imputation for unordered categorical

First fit a multinomial regression to get a matrix of probabilities (with number of columns equal to the number of categories in your variable)

```
> library(nnet)
> mmod = multinom(y ~ pred1 + pred2,data = dat.cc)
> ps = predict(mmod, type = "prob", newdata =
  dat.dropped)
```

… now use the probabilities to impute new values

(assumes ps is a matrix of probabilities obtained from fitted model combined with predictor data for observations you want to impute for (as in last slide))

```
> k = 5     # k is the number of categories
> cat.imps = numeric(nimp)
# nimp is the number to be imputed (sum(Ry==0))
> for(i in 1:nimp)
{
 cat.imps[i] = sum(rmultinom(1, 1, ps[i,])*c(1:k))
}
```

The part within the parentheses will create a vector of 0's except the one category where the draw occurred and that slot will now show its category number rather than a 1. So when you sum all you get back is that category number.

Multiple Imputation

Multiple Imputation

- What is it?
  - Description
  - What are its strengths and weaknesses
  - Assumptions?
- How do we do it?
  - Generally
  - Software
- Why should we trust it?

Multiple Imputation: what it is

- MI uses observed data to impute missing values that reflect both sampling variability and model uncertainty
- MI creates several imputations for each missing value and thus creates several completed datasets
- Compete data analyses can be performed on each imputed dataset
- The answers from each dataset are combined using standard rules

Multiple Imputation

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

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

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

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

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

MI: strengths

- Maintains entire dataset
- Uses all available information
- Weak (more plausible) assumptions about the missing data mechanism
- Properly reflects two kinds of uncertainty about the missing values (so, confidence intervals have correct coverage properties)
  - Sampling uncertainty
  - Model uncertainty
- Maintains relationships between variables
- One set of imputed datasets can be used for *many* analyses (allowing for release, for example, of public use imputed datasets)

MI: weaknesses

- Can be more complex to implement (though with current software this is becoming less and less of an issue)
- Have to rely on modeling assumptions

MI: assumptions about the missing data mechanism

- Most commonly we assume that data are *missing at random* which means that if two observations had the same values for all their observed covariates we would expect them to be equally likely to have missing values on the others.
- Note that this is an untestable assumption and is distinct from decisions about how to *predict* missing values

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

MI: non-ignorable missing data mechanisms?

- Can MI also handle "non-ignorable" missing data mechanism:

$$p(R|X) = p(R|X^{obs},Z)$$

  where X^{obs} denotes the portion of the data matrix that is observed, and Z denotes something partially or completely unobserved
- Typically (that is with existing software) no, however in many situations MAR is often a good approximation to NMAR (particularly if we are able to include many variables in our imputation model)
- In theory MI could be augmented to handle certain types of non-ignorable models (it's on our list for `mi`)

MI: How does it work?

1. Specify a model for the complete data, and fit
2. Use this model to predict missing values
3. Repeat (1)-(2) *M* times to create *M* complete datasets
4. Perform your desired analysis in each dataset, then…

Types of MI models

- We have to specify a model for the data that guides how predictions are made. These vary across software packages but can be separately broadly into
  - Joint models (Multivariate Normal, Loglinear, General Location Model)
  - Conditional models (Different model for each variable with missing data – this is like the regression imputations we did)

Recall: Combining estimates across datasets

Given estimates Q_1 , … , Q_M and their corresponding standard errors s_1,…, s_M

point estimate: $\theta = 1/\mathrm{m} \sum_{\mathrm{m}} \mathrm{Q}_{\mathrm{m}}$

variance: $\bar{U} + (1 + \mathrm{m}^{-1})\mathrm{B}$

where,

$\bar{U} = 1/\mathrm{m} \sum_{\mathrm{m}} \mathrm{s}_{\mathrm{m}}^{2}$

$\mathrm{B} = 1/(\mathrm{m}-1) \sum_{\mathrm{m}} (\mathrm{Q}_{\mathrm{m}} - \theta)^{2}$

Multiple Imputation:

X1	X2	X3
✓	✓	✓
✓	✓	✓
✓	✓	✓
✓	✓	✓
✓	?	✓
✓	?	✓
✓	✓	?
✓	✓	?
✓	?	?
✓	?	?

Original Dataset

X1	X2	X3
✓	✓	✓
✓	✓	✓
✓	✓	✓
✓	✓	✓
✓	*	✓
✓	*	✓
✓	✓	*
✓	✓	*
✓	*	*
✓	*	*

theta1

X1	X2	X3
✓	✓	✓
✓	✓	✓
✓	✓	✓
✓	✓	✓
✓	*	✓
✓	*	✓
✓	✓	*
✓	✓	*
✓	*	*
✓	*	*

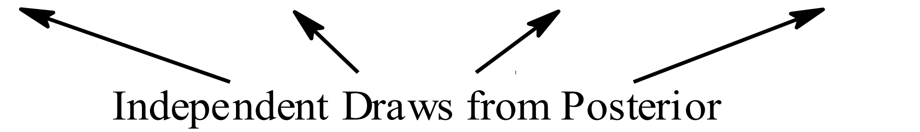
theta2

X1	X2	X3
✓	✓	✓
✓	✓	✓
✓	✓	✓
✓	✓	✓
✓	*	✓
✓	*	✓
✓	✓	*
✓	✓	*
✓	*	*
✓	*	*

theta3

X1	X2	X3
✓	✓	✓
✓	✓	✓
✓	✓	✓
✓	✓	✓
✓	*	✓
✓	*	✓
✓	✓	*
✓	✓	*
✓	*	*
✓	*	*

theta4



Independent Draws from Posterior

* represent imputed values

MI Success Stories: NHANES simulation study

NHANES (National Health and Nutrition Examination Survey) III has produced 5 imputed datasets spanning approximately 30 variables. To evaluate how reasonable these might be the following test was performed

1. Complete case data was pooled from four NCHS examination surveys (n=31,847)
2. From this population, stratified samples (n=6000) were drawn (using the same sampling plan as NHANES)
3. Missing values were imposed using an ignorable mechanism that mimicked actual NHANES missing data patterns
4. MI always did at least as well and generally did better than complete case analyses
5. When MI failed (departures from 95% coverage) it could generally be traced back to departures from the normality assumptions

(note that this was done using earlier, more crude versions of MI)

Large-scale surveys now using MI

- National Health and Nutrition Evaluation Survey (NHANES) National Center for Health Statistics
- Fatal Accident Report System (FARS) Department of Transportation for the National Highway Traffic Safety Administration
- Survey of Consumer Finances (SCF) Federal Reserve Bank
- Others…

More success?

- For another similar example see:

Raghunathan & Paulin (1998) “ Multiple Imputation of Income in the Consumer Expenditure Survey: Evaluation of Statistical Inference” in *ASA Proceedings of the Business and Economic Statistics Section*, pp. 1-10

Software

(Some) MI Software

- Amelia (stand-alone, shareware, MVN): http://gking.harvard.edu/amelia
- Solas (stand-alone, $$$, propensity scores)
- SAS: PROC MI, PROC MIANALYZE ($$$, MVN)
- Stata: mi command ($$$, MVN)
- MICE (R), **mi (R)**, ICE(Stata, $$$), IVEWARE (SAS, $$$) (all shareware) [these all are “chained equation” or “regression switching” approaches]
- Schafer: cat, norm, mix (Splus missing data library) ($$$, Bayesian, choice of loglinear, MVN, GLOM)
- Schafer: norm, pan (stand-alone, shareware)
- **For more info see www.multiple-imputation.com**

mi (R) -- Regression switching

- Primary Advantage
  - Don’t have to specify a joint distribution, just all of the conditionals, so can have very flexible functional form -- important for fitting big models and for interactions
- Potential Disadvantage
  - May not correspond to a valid joint distribution (shorthand: may not work right): but simulation studies seem to indicate that this either doesn’t happen or doesn’t hurt results in practice) (also ongoing work to help diagnose in practice)
  - Have to think a little harder to make sure the conditionals you specify make sense (can use diagnostics)
- Similar packages (ICE, MICE, IVEWARE) run in Stata, Splus/R and SAS (they came before this package)

Regression switching (aka "chained equations", "iterative imputation")

- The building block of this approach is analogous to the regression imputations with noise discussed previously
- This method can easily handle complicated missing data patterns
- This method can accommodate different types of data structures
- This method accounts for model uncertainty in addition to sampling uncertainty

Regression switching: basic algorithm

1. First missing values are temporarily replaced with “starting values” generated as random draws from the empirical distribution of the variable
2. *For each variable with missing data*
   a) Draw an imputed value for each given all other variables in the dataset (or a specified subset) using both observed and imputed (or starting) values. For now, think of this as regression imputation with noise though the model can be more general than linear regression.
   b) Replace the starting value (or previously imputed value) with the imputed value from (a)
3. Repeat step (2) *C* times (where *C* might equal 10, 20, 30…). We'll talk more next week about how to determine this #
4. To obtain a newly imputed dataset repeat steps (1)-(3) *M* times (where *M* might equal 5)

Regression switching. Step 1: starting values

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	6
1	5	10	7
1	3	7	2
1	3	5	7
1	4	7	6
1	5	6	7
1	4	8	9
1	5	10	8

For observation 6, value for X_2 was replaced by a random draw from: 3,4,3,4,5,4,5

Regression switching. Step 2: Regression imputation

There are two major choices in Step 2 for each variable X_k

- What conditional model to fit for $E[X_k \mid X_{(-k)}]$

  (where $X_{(-k)}$ is the vector of all variables in the dataset except the k^{th})

  If your data do not seem appropriate for linear regression (e.g. not continuous) other types of models can be used. For `mi` these include: linear regression, logistic regression, multinomial logit, ordered logit, and (overdispersed) Poisson.
- How to move from the model fit to an imputation that reflects both
  - model uncertainty, and
  - sampling uncertainty?

Accounting for model uncertainty

- Accounting for model uncertainty.

  Typically we reflect this type of uncertainty by drawing regression coefficients from an appropriate distribution (e.g. a normal distribution centered at the estimated coefficient and using the s.e. to determine the variance) rather than using the point estimates of the coefficients as the "truth"
- Accounting for sampling uncertainty
  - Drawing from the posterior predictive distribution
  - Predictive mean matching

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

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

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

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

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

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

X_1	X_2	X_3	X_4
0	3	8	2
0	4	9	3
0	3	7	3
0	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

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

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

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

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

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

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

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, look at the data and 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)
```

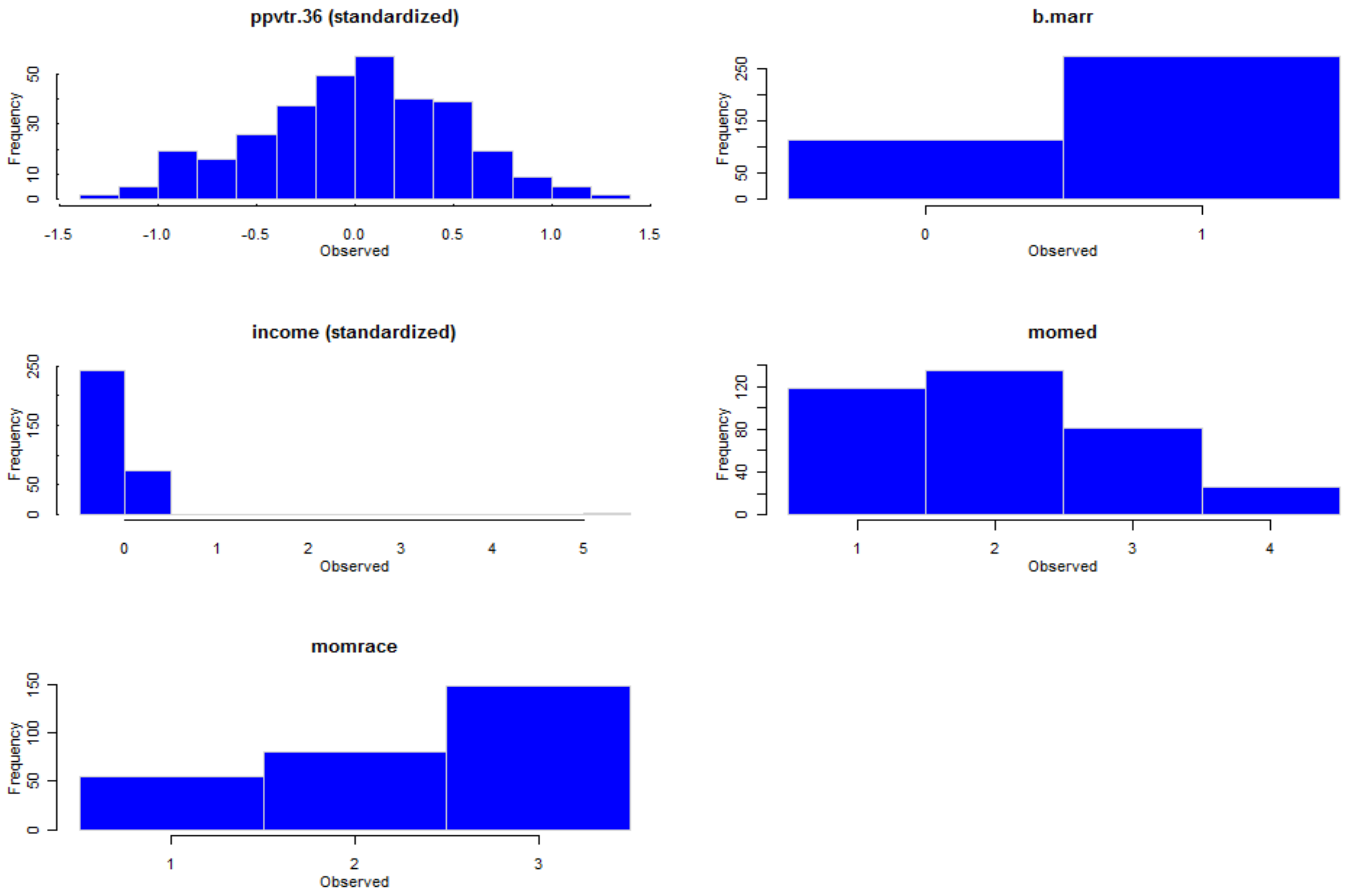
`summary` produces the same result as the `summary` method for a `data.frame`

```
summary(mdf)
```

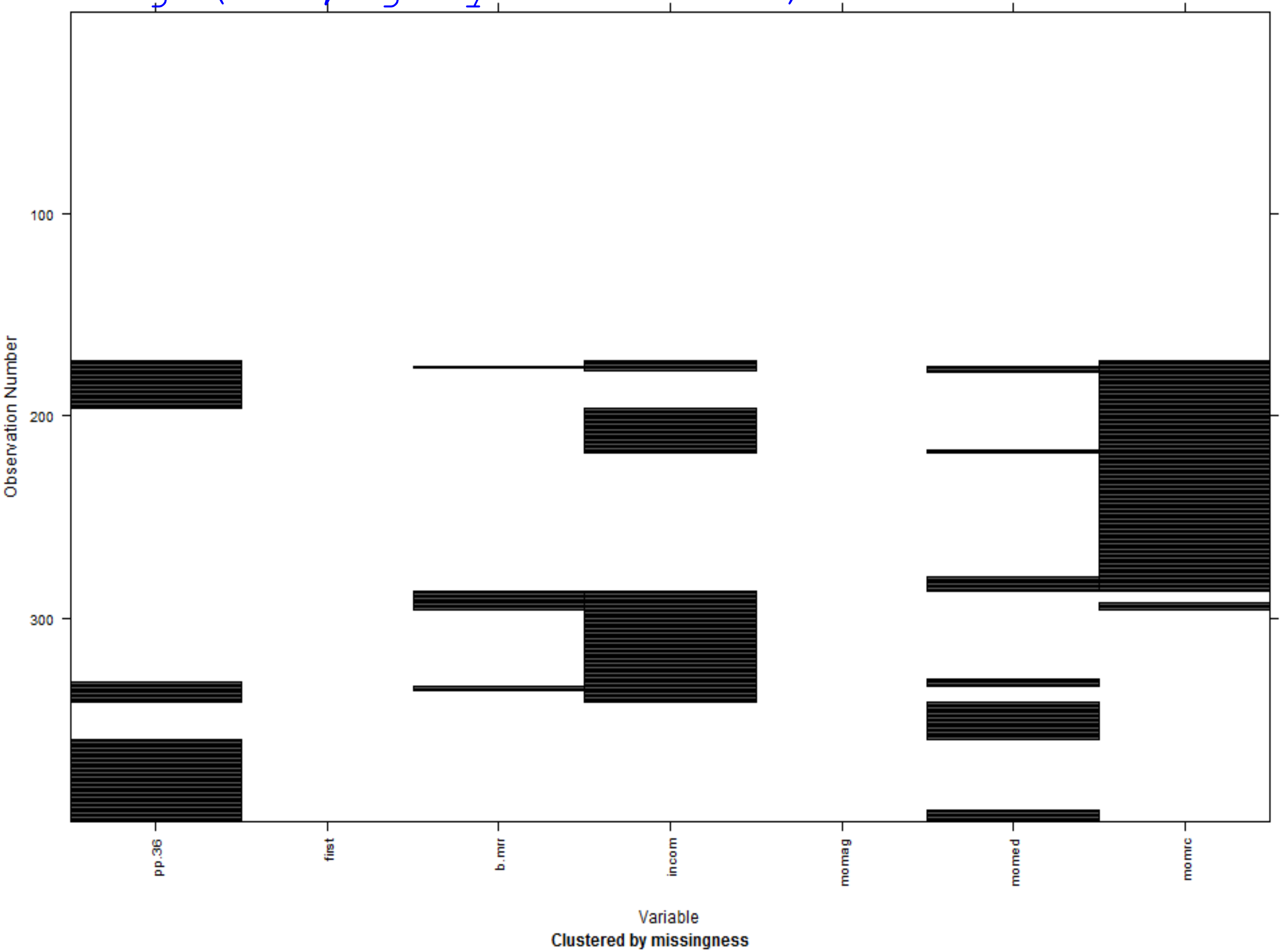
```
   ppvtr.36       first    b.marr        income           momage
 Min.   : 41.00   0:226   0   :114   Min.   :      0   Min.   :16.00
 1st Qu.: 74.00   1:174   1   :274   1st Qu.:   8590   1st Qu.:22.00
 Median : 87.00           NA's: 12   Median :  17906   Median :24.00
 Mean   : 85.94                      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:

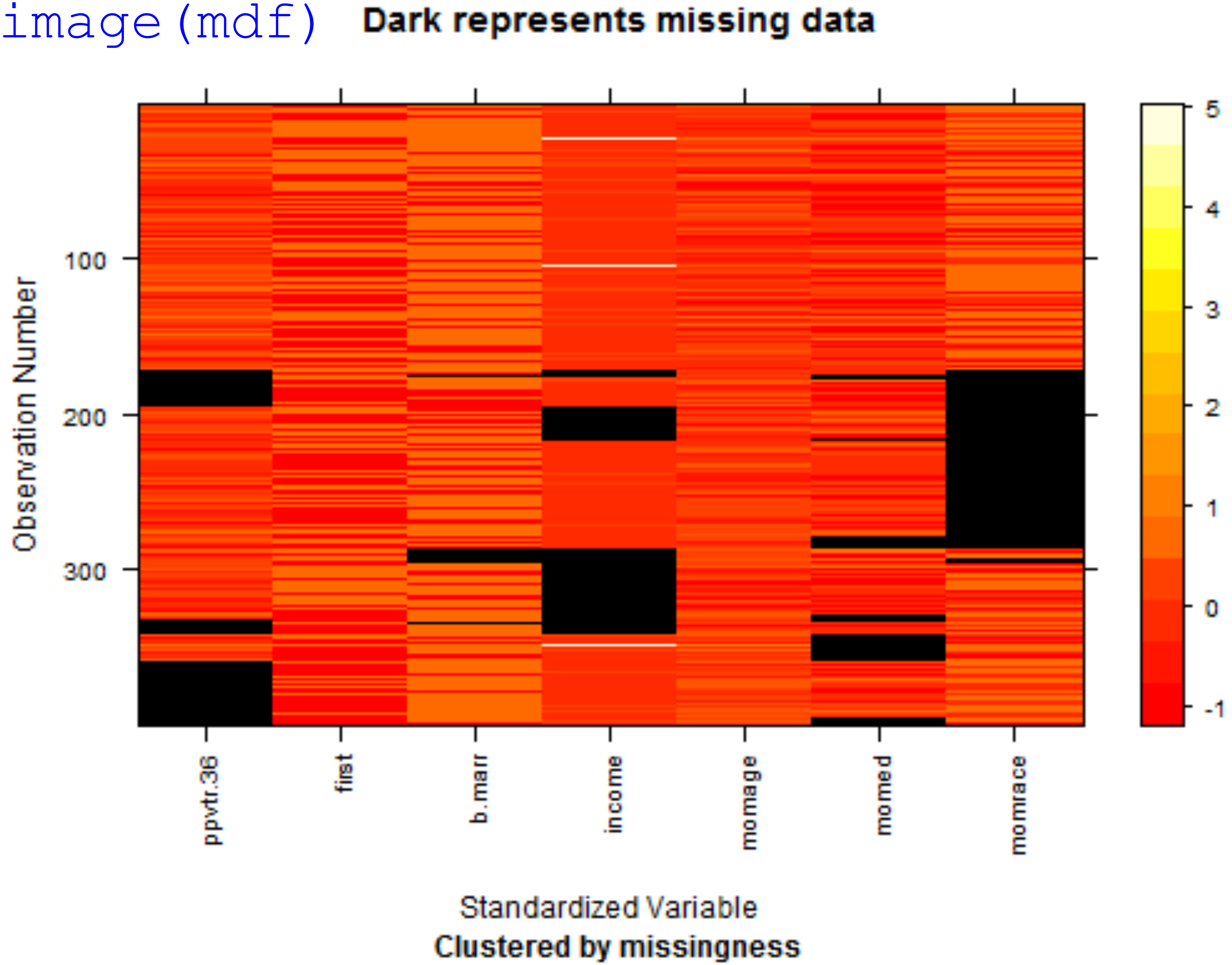
`hist(mdf)`



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



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

(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>
```

(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.

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 (default is unordered 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 probit
- importantly, both will be defined as separate indicators when used as predictors in the other imputation models

(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.