An Algorithm for Creating Models for Imputation Using the MICE Approach:

An application in Stata

Rose Anne Medeiros
rosem@ats.ucla.edu

Statistical Consulting Group
Academic Technology Services
University of California, Los Angeles

2007 West Coast Stata Users Group meeting

Outline

1. Introduction
   - Motivation
2. The Programs
   - -pred_eq-
   - -check_eq-
3. What does it look like?
   - Syntax
   - Examples
4. Closing odds and ends

Imputation methods

- Imputation involves replacing missing values in a data matrix with plausible values
- All imputations are based on some sort of model (however simple or complex)
- The quality of the imputation, and the substantive analyses that follow, all depend on the quality of the imputation model

Multivariate Imputation by Chained Equations I

- Multivariate Imputation by Chained Equations (MICE) uses a series of univariate analyses to predict missing values
    - For each variable to be imputed, imputed values are drawn from a conditional distribution based on univariate regression models
    - This process is repeated multiple times, so that previous estimated values are used in subsequent rounds of estimation
    - At least in theory, this should converge to a stable multivariate solution

Multivariate Imputation by Chained Equations II

- An important feature of the MICE approach is that even though all the estimates are interrelated, there is an equation for each variable imputed by model.
- Described in detail in van Buuren et al. (1999).
- In Stata this is implemented with the package -ice-, as well as MICE (R), and IVEware (available as a SAS macro and as a stand-alone package).

The imputation and analysis process

Steps for researchers:

1. Obtain data
2. Build imputation model
3. Run imputation model and create multiple imputed datasets
4. Run analyses on imputed data

Steps for data distributers:

1. Obtain data
2. Build imputation model
3. Run imputation model and create multiple imputed datasets
4. Release data for use by researchers

Building imputation models

- Imputation models should contain as many "predictor" variables as possible, since the greater the number of variables the greater the amount of information from which to make estimations (Rubin 1996, van Buuren, Boshuizen & Knook 1999).
- One way to approach this is to use all other variables in a dataset to predict missing values on a given variable. But...
  - This is not practically feasible in datasets with many variables.
  - Unnecessary since at least some variables are likely to contain redundant information.

Building imputation models

- Imputation models should contain as many "predictor" variables as possible, since the greater the number of variables the greater the amount of information from which to make estimations (Rubin 1996, van Buuren, Boshuizen & Knook 1999).
- One way to approach this is to use all other variables in a dataset to predict missing values on a given variable. But...
  - This is not practically feasible in datasets with many variables.
  - Unnecessary since at least some variables are likely to contain redundant information.

Building imputation models

- Imputation models should contain as many "predictor" variables as possible, since the greater the number of variables the greater the amount of information from which to make estimations (Rubin 1996, van Buuren, Boshuizen & Knook 1999).
- One way to approach this is to use all other variables in a dataset to predict missing values on a given variable. But...
  - This is not practically feasible in datasets with many variables.
  - Unnecessary since at least some variables are likely to contain redundant information.

- One solution to this is to use a subset of the "best" predictors to predict missing values in each variable with missing data.
  - Here "best" is defined as those *n* variables with the highest bivariate correlations with the variable being predicted.
  - Another possible definition of "best" is all potential predictors with a correlation over some criterion value (van Buuren, Boshuizen & Knook 1999).

- This takes care of issues related to the number of predictors, however, there end up being a number of practical problems with the equations this generates, specifically:
    - Collinearity between selected predictors (redundant information).
    - Lack of variance in the variable being predicted when all predictors are non-missing.
    - Predictors which perfectly predict binary variables. (With other types of dependent variables, perfect predictors do not prevent estimation.)
    - Inability to estimate errors (zeros on the diagonal of the VCE matrix).

- This takes care of issues related to the number of predictors, however, there end up being a number of practical problems with the equations this generates, specifically:
  - Collinearity between selected predictors (redundant information).
  - Lack of variance in the variable being predicted when all predictors are non-missing.
  - Predictors which perfectly predict binary variables. (With other types of dependent variables, perfect predictors do not prevent estimation.)
  - Inability to estimate errors (zeros on the diagonal of the VCE matrix).

- This takes care of issues related to the number of predictors, however, there end up being a number of practical problems with the equations this generates, specifically:
    - Collinearity between selected predictors (redundant information).
    - Lack of variance in the variable being predicted when all predictors are non-missing.
    - Predictors which perfectly predict binary variables. (With other types of dependent variables, perfect predictors do not prevent estimation.)
    - Inability to estimate errors (zeros on the diagonal of the VCE matrix).

- This takes care of issues related to the number of predictors, however, there end up being a number of practical problems with the equations this generates, specifically:
  - Collinearity between selected predictors (redundant information).
  - Lack of variance in the variable being predicted when all predictors are non-missing.
  - Predictors which perfectly predict binary variables. (With other types of dependent variables, perfect predictors do not prevent estimation.)
  - Inability to estimate errors (zeros on the diagonal of the VCE matrix).

- This takes care of issues related to the number of predictors, however, there end up being a number of practical problems with the equations this generates, specifically:
  - Collinearity between selected predictors (redundant information).
  - Lack of variance in the variable being predicted when all predictors are non-missing.
  - Predictors which perfectly predict binary variables. (With other types of dependent variables, perfect predictors do not prevent estimation.)
  - Inability to estimate errors (zeros on the diagonal of the VCE matrix).

The Two Parts

The solution is implemented in two related programs:

- **`pred_eq`** selects sets of n predictors for each variable with missing values.
- **`check_eq`** checks the equations for problems that tend to cause errors in -ice-.
- The algorithm implemented in this package is similar to that discussed by van Buuren, Boshuizen and Knook (1999).

The Two Parts

The solution is implemented in two related programs:

- **`pred_eq`** selects sets of n predictors for each variable with missing values.
- **`check_eq`** checks the equations for problems that tend to cause errors in -ice-.
- The algorithm implemented in this package is similar to that discussed by van Buuren, Boshuizen and Knook (1999).

- The two programs are designed to be used with -ice-, as a result:
  - As much as possible the syntax for the commands are similar (above and beyond what is typical in Stata).
  - Where appropriate, it has options similar to those in -ice-, e.g. **cmd(***cmdlist***)** and **substitute(***sublist***)**
  - Uses the same criteria for selecting the type of regression used to estimate the model
  - Outputs equations in an ice-friendly format
  - Will even produce a (draft) command for -ice- based on the options specified.

- The two programs are designed to be used with -ice-, as a result:
  - As much as possible the syntax for the commands are similar (above and beyond what is typical in Stata).
  - Where appropriate, it has options similar to those in -ice-, e.g. **`cmd(`***`cmdlist`***`)`** and **`substitute(`***`sublist`***`)`**
  - Uses the same criteria for selecting the type of regression used to estimate the model
  - Outputs equations in an ice-friendly format
  - Will even produce a (draft) command for -ice- based on the options specified.

- The two programs are designed to be used with -ice-, as a result:
  - As much as possible the syntax for the commands are similar (above and beyond what is typical in Stata).
  - Where appropriate, it has options similar to those in -ice-, e.g. **`cmd(`***`cmdlist`***`)`** and **`substitute(`***`sublist`***`)`**
  - Uses the same criteria for selecting the type of regression used to estimate the model
  - Outputs equations in an ice-friendly format
  - Will even produce a (draft) command for -ice- based on the options specified.

- The two programs are designed to be used with -ice-, as a result:
  - As much as possible the syntax for the commands are similar (above and beyond what is typical in Stata).
  - Where appropriate, it has options similar to those in -ice-, e.g. **`cmd(`**`cmdlist`**`)`** and **`substitute(`**`sublist`**`)`**
  - Uses the same criteria for selecting the type of regression used to estimate the model
  - Outputs equations in an ice-friendly format
  - Will even produce a (draft) command for -ice- based on the options specified.

- The two programs are designed to be used with -ice-, as a result:
  - As much as possible the syntax for the commands are similar (above and beyond what is typical in Stata).
  - Where appropriate, it has options similar to those in -ice-, e.g. **`cmd(`**`*cmdlist*`**`)`** and **`substitute(`**`*sublist*`**`)`**
  - Uses the same criteria for selecting the type of regression used to estimate the model
  - Outputs equations in an ice-friendly format
  - Will even produce a (draft) command for -ice- based on the options specified.

-pred_eq-: Generating the equations

- Predictors are selected based on bivarate correlations with the variable being predicted.
- The number of predictors can be user specified.
  - The default is 20.
  - In general, this should be set as high as is practical.
- Allows for special handling of nominal variables.
  - Accepts substitutions of a series of dummy variables (via a list of the same format -ice- takes).
  - Optionally uses Stata's built in command -tetrachoric-
  - If you have installed -polychoric- (by Stas Kolenikov), this can also be specified.

-pred_eq-: Generating the equations

- Predictors are selected based on bivarate correlations with the variable being predicted.
- The number of predictors can be user specified.
  - The default is 20.
  - In general, this should be set as high as is practical.
- Allows for special handling of nominal variables.
  - Accepts substitutions of a series of dummy variables (via a list of the same format -ice- takes).
  - Optionally uses Stata's built in command -tetrachoric-
  - If you have installed -polychoric- (by Stas Kolenikov), this can also be specified.

-pred_eq-: Generating the equations

- Predictors are selected based on bivarate correlations with the variable being predicted.
- The number of predictors can be user specified.
  - The default is 20.
  - In general, this should be set as high as is practical.
- Allows for special handling of nominal variables.
  - Accepts substitutions of a series of dummy variables (via a list of the same format -ice- takes).
  - Optionally uses Stata's built in command -tetrachoric-
  - If you have installed -polychoric- (by Stas Kolenikov), this can also be specified.

-check_eq- I

1. Drops highly collinear predictors
2. If dep does not vary when all predictors are non-missing this is reported to the user and the equation is not checked further.
   - The equation is still printed, as this should not be a problem for -ice-.
   - Optionally, predictors can be dropped to maximize the number of categories of the dependent variable. (option: drop_preds)
3. For binary variables, or those specified to be used with -logit-, the program checks for perfect prediction and attempts to determine which predictor perfectly predicts the outcome. If it is able to do so, the predictor is dropped.

-check_eq- II

4. Checks for zeros on the diagonal of the VCE matrix. If they exist -check_eq- will drop predictors to attempt to remedy this. This the default but it can be turned off.
5. If the boot option is specified equation is rerun using -bootstrap- to check for errors.

Using -pred_eq- and -check_eq- together

- -pred_eq- will automatically pass equations to -check_eq-.
- However, the user might want to use -pred_eq- to select highly correlated predictors, and then augment these equations with additional variables.
- For this reason -check_eq- will also accept equations directly from the user.

Command syntax

pred_eq *varlist* [*if*] [*in*] [, np(#) noeqlist macros nochcheck_eq show_unchecked_eq noeq(*varlist*) only(*varlist*) drop_preds ice substitute(*substitute_string*) cmd(*command_list*) maxdrop(#) polychoric tetrachoic polycriteria(option) tetcriteria(option)]

check_eq [*if*] [*in*] [, mac(*global_macro_name*) eq(*equation_list*) noeqlist macros cmd(*command_list*) detail substitute(*substitute_string*) nosubdrop drop_preds detail maxdrop(#) boot(*varlist*)]

A familiar example

- auto.dta modified to have missing data on 9 of the 11 numeric variables.
- I also created three variables that are duplicates of other variables. (These have no missing values.)

```
gen mpg2 = mpg
gen headroom2 = headroom
gen turn2 = turn
```

Description of missing data

```
     Variable        # Miss        Total      Miss/Total
-------------------------------------------------------
        price             4           74     .054054
          mpg             0           74           0
        rep78            15           74     .202703
     headroom             3           74     .040541
        trunk             7           74     .094595
       weight            10           74     .135135
       length             7           74     .094595
         turn             1           74     .013514
 displacement             5           74     .067568
   gear_ratio             0           74           0
      foreign             8           74     .108108
         mpg2             0           74           0
    headroom2             0           74           0
        turn2             0           74           0
```

Step 1: Run -pred_eq-

```
pred_eq price-foreign mpg2 headroom2 turn2, np(5) ///
   show_unchecked_eq nocheck_eq
```

```
Depending upon the number of variables and the options selected,
pred_eq may take a while to run.
9 variables need equations.

Unchecked equations.
eq(foreign :  gear_ratio displacement turn2 weight turn,/*
*/ displacement :  weight gear_ratio length turn2 turn,/*
*/ turn :  turn2 weight length displacement mpg2,/*
*/ length :  weight turn2 turn displacement mpg,/*
*/ weight :  length turn2 displacement turn mpg2,/*
*/ trunk :  length weight headroom2 headroom turn2,/*
*/ headroom :  headroom2 trunk length displacement weight,/*
*/ rep78 :  foreign turn2 turn weight displacement,/*
*/ price :  displacement weight length mpg2 mpg)
```

Step 2: Edit the equations and run -check_eq-

```
check_eq , eq(foreign :  gear_ratio displacement turn2 weight
turn,/*
*/ displacement :  weight gear_ratio length turn2 turn,/*
*/ turn :  turn2 weight length displacement mpg2,/*
*/ length :  weight turn2 turn displacement mpg,/*
*/ weight :  length turn2 displacement turn mpg2,/*
*/ trunk :  length weight headroom2 headroom turn2,/*
*/ headroom :  headroom2 trunk length displacement weight
gear_ratio,/*
*/ rep78 :  foreign turn2 turn weight displacement,/*
*/ price :  displacement weight length mpg2 mpg foreign)
```

Final equations.

```
eq(price :  displacement weight length mpg2 foreign,/*
*/ rep78 :  foreign turn2 weight displacement,/*
*/ headroom :  trunk length displacement weight gear_ratio,/*
*/ trunk :  length weight headroom2 turn2,/*
*/ weight :  length displacement turn mpg2,/*
*/ length :  weight turn2 displacement mpg,/*
*/ turn :  weight length displacement mpg2,/*
*/ displacement :  weight gear_ratio length turn2,/*
*/ foreign :  gear_ratio displacement turn2 weight)
```

A more complex example

- The data come from a study of relationship behavior in college students
- A small subset of the dataset that inspired this project
- 26 variables total: 7 background variables, 19 variables on relating to respondent behavior
- 374 cases (84 have at least one missing value).

What happens if I just try to run -ice-?

```
ice a04az-ccpss1i psep-pdead engaged married using "ice test",
substitute(a07:  psep pdivorced pother pdead, a10:  engaged
married)
```

```
   #missing |
     values |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        290       77.54       77.54
          1 |         12        3.21       80.75
          2 |          3        0.80       81.55
          3 |          2        0.53       82.09
          4 |          1        0.27       82.35
          6 |          1        0.27       82.62
          8 |          1        0.27       82.89
         10 |          1        0.27       83.16
         12 |          1        0.27       83.42
         15 |          2        0.53       83.96
         16 |          1        0.27       84.22
         17 |          2        0.53       84.76
         19 |         20        5.35       90.11
         20 |          4        1.07       91.18
         22 |         30        8.02       99.20
         23 |          3        0.80      100.00
------------+-----------------------------------
      Total |        374      100.00
   Variable | Command | Prediction equation
------------+---------+-------------------------------------------------------
      a04az | regress | a05az a06az a08 a03a ccnes1i ccnep1i ccncs1i ccncp1i
            |         | ccnes2i ccnep2i ccnes3i ccnep3i ccncs2i ccncp2i
            |         | ccncs3i ccncp3i ccsms1i ccsmp1i ccsms2i ccsmp2i
            |         | ccsms3i ccsmp3i ccpss1i psep pdivorced pother pdead
            |         | engaged married
      a05az | regress | a04az a06az a08 a03a ccnes1i ccnep1i ccncs1i ccncp1i
            |         | ccnes2i ccnep2i ccnes3i ccnep3i ccncs2i ccncp2i
            |         | ccncs3i ccncp3i ccsms1i ccsmp1i ccsms2i ccsmp2i
            |         | ccsms3i ccsmp3i ccpss1i psep pdivorced pother pdead
            |         | engaged married
        a07 | mlogit  | a04az a05az a06az a08 a03a ccnes1i ccnep1i ccncs1i
            |         | ccncp1i ccnes2i ccnep2i ccnes3i ccnep3i ccncs2i
            |         | ccncp2i ccncs3i ccncp3i ccsms1i ccsmp1i ccsms2i
            |         | ccsmp2i ccsms3i ccsmp3i ccpss1i engaged married
                 <output omitted>
       psep |         | [Passively imputed from (a07==2)]
  pdivorced |         | [Passively imputed from (a07==3)]
     pother |         | [Passively imputed from (a07==4)]
      pdead |         | [Passively imputed from (a07==6)]
    engaged |         | [Passively imputed from (a10==2)]
    married |         | [Passively imputed from (a10==3)]
------------------------------------------------------------------------------
```

```
Imputing
Error 430 encountered while running -uvis-
I detected a problem with running uvis with command mlogit on response
a07 and covariates a04az a05az a06az a08 a03a ccnes1i ccnep1i ccncs1i
ccncp1i ccnes2i ccnep2i ccnes3i ccnep3i ccncs2i ccncp2i ccncs3i
ccncp3i ccsms1i ccsmp1i ccsms2i ccsmp2i ccsms3i ccsmp3i ccpss1i
engaged married.
The offending command resembled:
uvis mlogit a07 a04az a05az a06az a08 a03a ccnes1i ccnep1i ccncs1i
ccncp1i ccnes2i ccnep2i ccnes3i ccnep3i ccncs2i ccncp2i ccncs3i
ccncp3i ccsms1i ccsmp1i ccsms2i ccsmp2i ccsms3i ccsmp3i ccpss1i
engaged married ,
With mlogit, try combining categories of a07, or if appropriate,
use ologit
convergence not achieved
r(430);

end of do-file

r(430);
```

Running -pred_eq- and -check_eq- in one step.

```
pred_eq a04az-ccpss1i, np(5) substitute(a07:  psep pdivorced
pother pdead, a10:  engaged married)

Depending upon the number of variables and the options
selected, pred_eq may take a while to run.

26 variables need equations.

Progress:  Checking equations.

Problems experienced creating prediction equation.
Make changes by hand.  Current equation:
logit a08 ccnes2i ccnep2i ccncp1i ccncs1i ccncp3i The problem
is most likely more than one x variable perfectly predicts
y.
```

- Program produces 11 messages about the equations.
- The detail option expands the amount of information given.

Running -pred_eq- and -check_eq- in one step.

```
pred_eq a04az-ccpss1i, np(5) substitute(a07:  psep pdivorced
pother pdead, a10:  engaged married)
```

```
Depending upon the number of variables and the options
selected, pred_eq may take a while to run.

26 variables need equations.

Progress:  Checking equations.

Problems experienced creating prediction equation.
Make changes by hand.  Current equation:
logit a08 ccnes2i ccnep2i ccncp1i ccncs1i ccncp3i The problem
is most likely more than one x variable perfectly predicts
y.
```

- Program produces 11 messages about the equations.
- The detail option expands the amount of information given.

Running -pred_eq- and -check_eq- in one step.

```
pred_eq a04az-ccpss1i, np(5) substitute(a07:  psep pdivorced
pother pdead, a10:  engaged married)
```

```
Depending upon the number of variables and the options
selected, pred_eq may take a while to run.

26 variables need equations.

Progress:  Checking equations.

Problems experienced creating prediction equation.
Make changes by hand.  Current equation:
logit a08 ccnes2i ccnep2i ccncp1i ccncs1i ccncp3i The problem
is most likely more than one x variable perfectly predicts
y.
```

- Program produces 11 messages about the equations.
- The detail option expands the amount of information given.

Final equations.

```
eq(a04az : a05az a06az a07 a03a ccsmp3i,/*
*/ a05az : a04az a06az ccsms3i ccsmp1i ccsms1i,/*
*/ a06az : a04az a05az a07 ccnep2i ccncp1i,/*
*/ a03a : a10 a07 a08 ccncs3i ccncs2i,/*
*/ ccnes1i : ccnep1i ccncs1i ccncp1i ccnes2i ccnep2i,/*
*/ ccnep1i : ccnes1i ccncp1i ccncs1i ccnep2i ccnes2i,/*
*/ ccncs1i : ccncp1i ccnes1i ccnep1i ccnes2i ccnep2i,/*
*/ ccncp1i : ccncs1i ccnes1i ccnep1i ccnes2i ccnep2i,/*
*/ ccnes2i : ccnep2i ccncp1i ccncs1i ccnes1i ccnep1i,/*
*/ ccnep2i : ccnes2i ccncp1i ccnep1i ccncs1i ccnes1i,/*
<output omitted>
*/ ccsmp1i : ccsms1i ccsmp3i ccsms3i ccsmp2i ccsms2i,/*
*/ ccsms2i : ccsms3i ccsmp2i ccsmp3i ccsms1i ccsmp1i,/*
*/ ccsmp2i : ccsmp3i ccsms2i ccsms3i ccsmp1i ccncs1i,/*
*/ ccsms3i : ccsms2i ccsmp3i ccsmp2i ccsms1i ccsmp1i,/*
*/ ccsmp3i : ccsmp2i ccsms3i ccsms2i ccsmp1i ccsms1i,/*
*/ ccpss1i : ccsmp3i ccsmp1i ccsms1i ccsmp2i ccncs2i)
```

Summary

- Together the two programs create and check equations that can be used with -ice-
- Can save considerable time when the alternative is to create imputation models for a large number of variables by hand, or diagnose and fix errors iteratively with -ice-.
- -pred_eq- and especially -check_eq- *can* take a considerable amount of time to run. But think about what they do:
  - -pred_eq- runs pairwise correlations between each variable to be imputed, and all possible predictors.
  - -check_eq- at the very least runs one regression for every variable to be imputed, if there are any problems, it does often considerably more work.

Summary

- Together the two programs create and check equations that can be used with -ice-
- Can save considerable time when the alternative is to create imputation models for a large number of variables by hand, or diagnose and fix errors iteratively with -ice-.
- -pred_eq- and especially -check_eq- *can* take a considerable amount of time to run. But think about what they do:
    - -pred_eq- runs pairwise correlations between each variable to be imputed, and all possible predictors.
    - -check_eq- at the very least runs one regression for every variable to be imputed, if there are any problems, it does often considerably more work.

Summary

- Together the two programs create and check equations that can be used with -ice-
- Can save considerable time when the alternative is to create imputation models for a large number of variables by hand, or diagnose and fix errors iteratively with -ice-.
- -pred_eq- and especially -check_eq- *can* take a considerable amount of time to run. But think about what they do:
    - -pred_eq- runs pairwise correlations between each variable to be imputed, and all possible predictors.
    - -check_eq- at the very least runs one regression for every variable to be imputed, if there are any problems, it does often considerably more work.

Possible additions

- van Buuren, Boshuizen and Knook (1999) suggest including variables that predict missingness as predictor variables. This is currently not implemented (although the user could easily include them by hand), but may be implemented as an option in later versions.
- May allow a criterion correlation level (e.g. $r \geq 0.2$) for selection of predictors.
- Updates to -njc-. :)

Acknowledgements

- Ian White for a number of helpful comments and suggestions, including pointing out several unnecessary components of earlier versions of the program.
- Maarten Buis read and commented on drafts of the help files.
- Patrick Royston for helpful comments on the package.

Credit where credit is due:

- The ado file which outputs the equations is heavily based upon Jeroen Weesie's -wraplist-.
- Various parts of the program also borrowed from Patrick Royston's -ice-.

References

van Buuren S., H. C. Boshuizen and D. L. Knook. 1999. Multiple imputation of missing blood pressure covariates in survival analysis. Statistics in Medicine 18:681-694.

Royston P. 2004. Multiple imputation of missing values. Stata Journal 4(3):227-241.

Royston P. 2005a. Multiple imputation of missing values: update. Stata Journal 5: 188-201.

Royston P. 2005b. Multiple imputation of missing values: update of ice. Stata Journal 5: 527-536.

Rubin, D. B., 1996. Multiple Imputation After 18+ Years. Journal of the American Statistical Association 91: 473-489.