An informal tutorial on the `ice` command for chained equations imputation in Stata

Patrick Royston, MRC Clinical Trials Unit, London

Nordic Stata Users’ Meeting, September 7th, 2007
Multiple, multivariate imputation with \texttt{ice}: the basic algorithm

- The imputation model is $x_1, \ldots, x_k$
- Some observations are assumed to be missing at random (MAR)
- Initialise – fill in missing values at random
- Apply \texttt{uvis} to $x_1$ regressing on $x_2, \ldots, x_k$
- Replace missing values in $x_1$
- Repeat for $x_2, \ldots, x_k$ on the other $x$’s (cycle 1)
- Repeat for about 10 cycles
- Do $m$ times to give $m$ imputed datasets with complete observations
We might have started like this:

```
icce varlist [if exp] [in range] [weight],
    [ saving(filename [, , replace]) m(#) cmd(cmdlist) cycles(#) boot[(varlist)]
    draw[(varlist)] seed(#) dryrun
    eq(eqlist) passive(passivelist)
    noshoweq substitute(sublist)
    interval(intlist) replace
    genmiss(string) dropmissing
    other_options ]
```
Instead, we will start with a real-life problem:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>survtime</td>
<td>347</td>
<td>395.44</td>
<td>431.53</td>
<td>2.00</td>
<td>2268.00</td>
</tr>
<tr>
<td>censdead</td>
<td>347</td>
<td>0.93</td>
<td>0.26</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>age</td>
<td>347</td>
<td>58.62</td>
<td>10.12</td>
<td>28.92</td>
<td>79.89</td>
</tr>
<tr>
<td>sex</td>
<td>347</td>
<td>0.32</td>
<td>0.47</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>who</td>
<td>347</td>
<td>2.97</td>
<td>0.72</td>
<td>2.00</td>
<td>4.00</td>
</tr>
<tr>
<td>rem</td>
<td>347</td>
<td>1.57</td>
<td>0.50</td>
<td>1.00</td>
<td>2.00</td>
</tr>
<tr>
<td>mets</td>
<td>346</td>
<td>1.84</td>
<td>0.37</td>
<td>1.00</td>
<td>2.00</td>
</tr>
<tr>
<td>lung</td>
<td>346</td>
<td>0.45</td>
<td>0.50</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>res</td>
<td>256</td>
<td>4.28</td>
<td>0.64</td>
<td>2.00</td>
<td>5.00</td>
</tr>
<tr>
<td>grade</td>
<td>224</td>
<td>3.49</td>
<td>0.87</td>
<td>2.00</td>
<td>5.00</td>
</tr>
<tr>
<td>t_mt</td>
<td>346</td>
<td>129.42</td>
<td>421.11</td>
<td>0.00</td>
<td>6405.00</td>
</tr>
<tr>
<td>wcc</td>
<td>324</td>
<td>8.72</td>
<td>4.14</td>
<td>3.10</td>
<td>55.20</td>
</tr>
<tr>
<td>haem</td>
<td>324</td>
<td>12.27</td>
<td>1.91</td>
<td>7.60</td>
<td>17.50</td>
</tr>
<tr>
<td>scal</td>
<td>306</td>
<td>2.46</td>
<td>0.21</td>
<td>2.10</td>
<td>3.74</td>
</tr>
<tr>
<td>wt</td>
<td>268</td>
<td>73.16</td>
<td>13.45</td>
<td>29.80</td>
<td>132.00</td>
</tr>
<tr>
<td>maxdtu</td>
<td>204</td>
<td>9.70</td>
<td>3.57</td>
<td>2.50</td>
<td>22.00</td>
</tr>
<tr>
<td>esr</td>
<td>169</td>
<td>49.61</td>
<td>35.06</td>
<td>1.00</td>
<td>154.00</td>
</tr>
<tr>
<td>visc</td>
<td>50</td>
<td>1.90</td>
<td>0.26</td>
<td>1.50</td>
<td>2.50</td>
</tr>
</tbody>
</table>
Prognostic factors in advanced kidney cancer (MRC RE01 trial)

- 350 patients, of whom 347 had follow-up information and 322 died
- 16 potential prognostic factors
  - 12 with some missing data
  - 4 binary
  - 3 ordered categorical
  - 0 nominal
  - 9 continuous
- Survival time, censoring indicator
Steps of an \texttt{ice} analysis

- Discard observations with missing response
- Summarise the variables
- Weed out variables with “too much” missingness
  - !!Subjective
  - Some use rule of thumb that >50\% missing is unacceptable
- Distinguish types of variable
  - binary, ordinal, nominal, continuous, time-to-event with censoring
  - Treated differently
- Construct the \texttt{ice} command and run it.
Binary variables

• Imputed using logistic regression
• Consider discarding sparse variables
  • Hardly any 1’s (or 0’s)
  • Useless predictors in many analyses
  • Can give problems in imputation
• Otherwise generally trouble-free
  • Be aware of the “perfect prediction” problem (Ian White)
  • Fixed automatically for logistic regression in ice
  • Could affect ordinal logistic regression
Ordinal variables

- Good choice of regression command is often `ologit`
  - note: *not* the default – that is `mlogit`
- *(May)* need to deal with dummy variables
  - Include in main `varlist`
  - May need `passive()` and `substitute()` options
- Generally fairly trouble-free
Nominal (unordered categorical) variables

- Default and only sensible choice of regression command is `mlogit`
- Need to deal with dummy variables correctly
  - Include in main `varlist`
  - Use `passive()` and `substitute()` options
- May need to combine sparse categories
Continuous variables

- Normality assumed – must be checked
- If necessary, transform variable to approximate normality
- Stata’s `lnskew0` command is useful here
  - Must back-transform after imputation
  - Check range of imputed values is valid
- If can’t find suitable transformation, use the `match()` option
  - no longer assumes normality
Time-to-event with censoring

- To reduce bias, essential to include the outcome variable in the chained-equation imputation models
- What functional form for \(_t\) in model?
- Include \(_d\)?
- Van Buuren et al. (1999): use \(\ln(_t)\) & \(_d\)
  - No theoretical underpinning
- White (unpub): use \(H0(_t)\) and \(_d\) if have single binary variable in imputation model (for use with later Cox PH model)
Some tips

- **Use the** `dryrun` **option and carefully check the equations** `ice` **has created**

- **Sometimes you want to “tailor” specific equations according to subject-matter knowledge or detailed investigation**
  - Use the `eq()` option to do this

- **Compare the distribution of imputed and observed values – they should generally be roughly similar**
  - Use the `genmiss()` option to mark the missing observations
Now we’ll do an example in Stata. The finished product:

```
ice age sex who rem mets lung grade
g2 g3 g4 res r1 r2 r4 ln_t_mt ln_wcc
haem ln_scal ln_wt ln_maxdtu esr ln_visc
H0 _d, saving(re01i) m(5) match(esr)
cmd(grade:mlogit, res:ologit) genmiss(M)
seed(1001) substitute(grade:g2 g3 g4,
res:r1 r2 r4)
passive(g2:grade==2 \
g3:grade==3 \\
g4:grade==4 \r1:res==1 \r2:res==2 \r4:res==4)
```