University of Limerick
Department of Sociology Working Paper Series

Working Paper WP2015-02
September 2015

Brendan Halpin

Department of Sociology, University of Limerick

Multiple Imputation for Categorical Time-series

Available at `http://www.ul.ie/sociology/pubs/wp2015-02.pdf`

MICT: Multiple Imputation for Categorical Time-series

Brendan Halpin
Department of Sociology, University of Limerick
Ireland
brendan.halpin@ul.ie

Abstract

The `MICT` package provides a method for multiple imputation for categorical time-series data such as lifecourse or employment-status histories that preserves longitudinal consistency, using a monotonic series of imputations. It allows flexible imputation specifications, with a model appropriate to the target variable (`mlogit`, `ologit` etc.). Where transitions are substantially less frequent than once per time-unit, and where missingness tends to be consecutive (as is typical of lifecourse data), it produces imputations with better longitudinal consistency than `mi impute` or `ice`.

1 Missingness in longitudinal data

This paper describes an approach to multiple imputation for categorical cross-sectional time-series data, such as labour market or other life course histories. The approach focuses on filling gaps from their edges, using a monotonic series of imputations. It uses `mi impute mlogit` to carry out single imputations, but manages the sequencing of imputations independently of the `mi impute` infrastrucure. It respects the longitudinal consistency of the data in a way that is difficult or impossible to achieve with standard `mi impute` or `ice` approaches, while allowing flexible imputation models and full access to the power of Stata's `mi` post-imputation infrastructure. The method is implemented in the `MICT` package.

The typical application of this method is to imputing gaps in lifecourse data such as employment or fertility histories, where the state changes substantially less often than once per time-unit, and where missingness also tends to be consecutive. However, it is relevant for any categorical time-series data with relatively low transition rates and consecutive missingness.

1.1 Longitudinal data and missingness

The availability of longitudinal data such as labour market, family formation, or residential histories, is ever increasing, and methods for its analysis are becoming ever more common and widely used. However, longitudinal data is subject to missingness, often to a greater degree than cross-sectional data. While some methods can deal with missingness (e.g., duration models can "censor" data

from the first occurrence of missing observations onwards), others cannot, and require full data. Throwing away individual histories because of a small proportion of missingness is wasteful (even for duration models), and if missingness is not "completely at random" (MCAR), to use Rubin's term (1987), such deletion of cases may cause bias. Indeed, it is particularly likely that missingness is not random, in that volatile histories are disproportionately likely to result in missingness. This is not only because people who experience volatility will be more likely to miss data-recording opportunities (e.g., annual interviews) but also because volatile histories have more opportunity for incompleteness. For instance, if you are in the same job for ten years, missing an occasional interview will not impact the record, whereas if you have changed job 5 times since the last interview, there are far more opportunities for error to enter and for information to be lost.

2 Imputation as a solution

2.1 Multiple imputation as the state-of-the-art

Introduced by Rubin (1987), multiple imputation has become a standard way of dealing with missing data. Regression models predict incompletely observed variables using fully observed variables, and are used to impute values to replace the missing, drawn at random from the prediction distribution. Rubin's key insight is that if multiple imputations are drawn, creating multiple imputed data sets, and if the results of analyses on the multiple data sets are averaged, unbiased estimates of the desired quantity (such as a regression coefficient) can be made.[1] Stata has implemented a package of commands for creating, managing and analysing such imputations (see `mi impute`).

However, cross-sectional time-series data such as life-histories put a substantial strain on standard imputation. If in wide format (with one variable per time-unit), there are very many, very similar variables, all likely to be subject to missingness. Consider the example of five years of monthly employment statuses, yielding 60 very similar observations. Imputing each on the basis of all others will be computationally challenging, if not impossible, and it is (at best) difficult to define selective imputation models for each time-unit observation. Moreover, as will be shown below, while standard approaches perform well in terms of the distribution of individual variables, consecutive imputed variables will tend to vary too much relative to each other, and thus from a longitudinal point of view the imputed data will have transition rates that are significantly biased upwards.

Where a single variable is subject to missingness, imputation is straightforward. When multiple variables have missing values, the complication may arise that cases to be imputed have missing values in the predictor variables. It may be the case that variables to be imputed can be arranged in an order such that the first has no missing predictors, that the next has missing predictors only on the first, the next only on the first and second, and so on. If such a "monotonic" pattern of missingness is present, imputation can proceed following the same order, such that at each predictive step, all the predictors are either fully

[1] More strictly, parameter estimates are averaged across the multiple imputed data sets, and the variance depends both on the variance within and between analyses.

observed or already imputed. If so, imputation with multiple variables to be imputed becomes a simple extension of imputation of a single variable. However, such a monotonic pattern is not likely to emerge without there being a structural reason for it (attrition in longitudinal data is a typical example). In its absence, there are two approaches to multiple imputation with multiple variables subject to missingness, either modelling the joint distribution of the variables directly (JM), or multiple imputation by chained equations, MICE.

JM is attractive where the variables are, or can be transformed, such that the joint distribution is multivariate-normal. It is efficient and has good theoretical foundations. Where some variables are categorical, conventional practice is to use linear predictions of dummy variables. However, this has been shown to produce poor results (Allison, 2005; van Buuren, 2007). The alternative is so-called Fully Conditional Specification (FCS) where rather than modelling the joint distribution, the conditional distributions are modelled. This allows variable-specific imputation models to be used (including models appropriate to categorical data) and is thus more flexible. MICE implements FCS.

Monotonic imputation, where applicable, is also flexible, allowing variable-specific imputation models, and permitting forms such as logistic regression (van Buuren, 2007, p.224ff).

MICE works as follows: In the first round, all missing values are imputed with a minimal model (hot-decking, or regressions using fully observed predictors only). Then the imputations are replaced by better imputations based on a full model using observed and imputed values. This process is repeated for a number of cycles, with the effect that the influence of the poor-quality first-round imputations is diluted. It has been implemented for R by Van Buuren et al (1999; 2007; 2011) and for Stata by Royston (2004; 2009)

More recent versions of Stata have incorporated equivalent functionality in the core `mi impute` infrastructure. These are excellent implementations, providing both good imputations and tools that make MI easy to use, but they do not suit time-series data, in neither wide nor long format. In wide format, there are very many poorly distinguished variables with wide incidence of missing: it is hard to write adequate imputation models, and they will tend to have severe problems in estimation. In long format, the relevant predictors are lags and leads, and the infrastructure is not adapted to using and updating lagged and leading variables.

One other package (for R and Stata) addresses imputation of longitudinal data: Amelia (Honaker and King, 2010). This is explicitly written with a view to respecting the longitudinal logic of time-series. However, it implements the JM approach to imputation, so while its authors make passing reference to imputing categorical variables, it is likely to generate poorer imputations for categorical data than packages that can use appropriate forms of model, such as binary or multinomial logistic.

3 Filling gaps in life course data

The approach presented is well adapted for the sort of data typically seen in lifecourse histories: a categorical state space observed on a regular basis (e.g., monthly) over a reasonably extended period (e.g., a small number of years), with transitions occuring at a relatively low rate. Hence we observe spells in

states that are significantly longer than one observation. Similarly, missing values will also tend to occur in runs (as gaps), due to a mixture of data collection problems (respondent absence at consecutive data collection points) and data structure (e.g., part or complete spells being non-reported or mis-reported). In effect, this data is typified by having a discrete state space, and conceptually continuous time that is approximated by discrete observations (e.g., monthly), and will usually be collected retrospectively at one or more well-separated time-points, often in terms of start and end dates of spells. Such data contains a relatively high amount of redundancy, such that information in the observed proportion is a good predictor of the missing part, and the runs of missingness mean missing observations tend to one or more missing neighbours.

Of course, forms of data other than lifecourse histories may well also have these features, and the approach is equally valid for them. Data with higher transition rates, or truly discrete time, will be harder to impute, because there is more variation from observation to observation.

3.1 A gap-filling algorithm

When data contains multiple observations per individual, the long format (one observation per person–time-unit) is natural, though (particularly when all individuals should be observed for the same time-span) the wide format is also appropriate, if a little less natural. However Stata's `mi impute` infrastructure is not designed to deal with data in a long format, requiring imputation variables to be in the same record. The method reported here exploits the `mi impute mlogit` command, but it uses the long format, and handles the management of predictor variables (in particular lags and leads of the target state), the storing of imputated values, and the sequencing of the imputations independently of Stata's `mi` infrastructure.

Conceiving of the data as longitudinal in nature reduces the many variables to a single state variable (indexed by time), and throws the focus on gaps, rather than individual missing variables. We can use a single form of model to predict all candidate observations. However, the presence of gaps means that no single model can apply to all time-points (e.g., the state at t+1 will not be observed for all cases). By focusing directly on gaps, we can nonetheless define a family of models that can be estimated in a monotonic series. We start by imputing the first (or, equally validly, the last) element of the longest gap using data from the nearest observed points, before and after. That is, for the first element of a gap, it is predicted by data from the immediately preceding element, and the first observed element after the gap. If we restrict ourselves to internal gaps, the lag and lead data are guaranteed not to be missing.

Once the first element of the longest gap has been filled, it holds that for the next-longest gap, the lag and lead data are either observed or already imputed. Thus we continue by imputing the last element of the next-longest gap, using the last observed (or imputed) value before the gap, and the immediately subsequent value. The process continues (alternating between first and last elements of the gaps) until all internal gaps are filled. Gaps at the start and end of the series can be imputed using an analogous approach, that uses only information from respectively after and before the gap.

We can illustrate it with the following example, with two sequences containing a six and three-element gap respectively (see Figure 1). We begin (in line

	Six unit gap	Three unit gap
Observed data with gaps	XX.....XXX	XXX...XXXXX
Impute first element of 6-unit gaps	XXi.....XXX	XXX...XXXXX
Impute last element of 5-unit gaps	XXI....iXXX	XXX...XXXXX
Impute first element of 4-unit gaps	XXIi...IXXX	XXX...XXXXX
Impute last element of 3-unit gaps	XXII..iIXXX	XXX..iXXXXX
Impute first element of 2-unit gaps	XXIIi.IIXXX	XXXi.IXXXXX
Impute only element of 1-unit gaps	XXIIiIIXXX	XXXIiIXXXXX

X: observed data; .: missing data; i: data being imputed; X: observed data used as predictor; I: imputed data used as predictor; I: previously imputed data.

Figure 1: Gap-filling sequence of imputation

2) by imputing the first element of six-unit gaps, using information pertaining to the last $(t-1)$ and next observed $(t+6)$ time points. Nothing happens to the shorter sequence. Then we impute the fifth element of five-unit gaps, using $t+1$ and $t-5$), and then the first element of four-unit gaps. The next step affects both sequences since the six-unit gap has been reduced to three, and imputes the third element of three-unit gaps. The process continues until all gaps are filled.

The data that can be used to predict must at least include the state at the prior and subsequent observed time-points. It should also include summaries of the prior and subsequent experience, and it can contain any other data keyed to these time-points (i.e., state in another time-dependent state space) and it can include fixed individual-level information. All the considerations regarding the requirements of a good imputation model in conventional circumstances apply equally here (in particular the imputation model should be "congenial" (Allison, 2009, p.84) with the analysis model), but with the additional requirement that longitudinal information in the imputed state must be included.

4 Demonstrations

To demonstrate how MICT works, and assess its performance, a number of examples will be presented:

1. Real data (school-to-work transitions) with simulated longitudinal missingness, allowing comparison between the true state and the imputations
2. Wholly simulated data using a simple structure, that permits comparison between MICT and MICE
3. Real data (mothers' labour market histories) with real missingness, using a realistic model
4. The same data using an enhanced model that takes account of knowledge about the data generation process to generate superior imputations

4.1 Real data with simulated missing

The first demonstration of the algorithm uses data from McVicar and Anyadike-Danes (2002), which reports six years of the lifecourse of Northern Irish young people, starting at the completion of compulsory schooling, in a state space concerned with education, training and the labour market. 712 individuals are observed over 72 months. To demonstrate and assess the performance of the algorithm, we impose missingness at random on this data, such that each month has a 1.25% chance of being missing, but with a 66% chance if the previous month is missing. This generates a pattern of runs of missingness, which are missing completely at random with respect to the observed data. The simulated data is stored in wide format (one variable per monthly observation: `state1` to `state72`; one row per individual).

4.1.1 Default imputation model

The default (excessively simple) imputation model in `MICT` is as follows, where `_mct_state` is the internal copy of the state variable, and `_mct_last` and `_mct_next` respectively the most recent and nearest future observation:

```
mi impute mlogit _mct_state i._mct_next i._mct_last, add(1) force augment
```

Initial and terminal gaps are imputed using only respectively subsequent and prior information.

We carry out this imputation with the following Stata code:

```
use mvadmar
mict_prep state, id(id)
mict_impute
```

4.1.2 Defining better imputation models

This default imputation model is very simple: each state is predicted only by the prior and next states. In effect, it assumes a zero-order Markov process where the transition rates are constant over time and across individuals. This model is built-in to the ado-files, and should in normal use be over-ridden by a more adequate model, e.g., to relax the zero-order assumption, allow transition rates to change over time, or incorporate other variables.

We can over-ride the built-in models by redefining the programs `mict_model_gap`, `mict_model_initial` and `mict_model_terminal`, as follows:

```
capture program drop mict_model_gap
program define mict_model_gap
mi impute mlogit _mct_state ///
                i._mct_next##c._mct_t i._mct_last##c._mct_t ///
                _mct_before* _mct_after*, ///
           add(1) force augment
end

capture program drop mict_model_initial
program define mict_model_initial
mi impute mlogit _mct_state i._mct_next _mct_after*, add(1) force augment
end

capture program drop mict_model_terminal
program define mict_model_terminal
mi impute mlogit _mct_state i._mct_last _mct_before, add(1) force augment
```

```
end
```

Variable _mct_before1 to _mct_beforeC and _mct_after1 to _mct_afterC (where C is the number of categories) are created by mict_prep, and represent the proportion of time before and after the gap spent in each of the C categories of the state variable, and thus offer a means of including in the model "history" prior to and after the nearest observed time units.

The interactions i._mct_next##c._mct_t i._mct_last##c._mct_t allow the effect of prior and next state to vary in a linear fashion with time, relaxing the assumption that transition rates are constant. Depending on the data, it may be desirable to relax this constraint even further, perhaps with time as a quadratic. Other variables can also be entered, including fixed individual variables and variables indicating time-dependent state in another domain.

The options "add(1) force augment" to mi impute mlogit are required, to make a single imputation, to force imputation even where the predictor variables are not fully observed, and to use augmented multinomial regression if perfect prediction is encountered. Other options may be used as appropriate.

4.1.3 Capturing convergence issues

In practice, more complex models will be more likely to have convergence problems. Sometimes models that fit most of the time will fail to converge a small proportion of the time, depending on the patterns of already imputed values. We can define simpler fallback models as in this example:

```
capture program drop mict_model_gap
program define mict_model_gap
di "Attempt first gap model"
capture mi impute mlogit _mct_state ///
   i._mct_next##c._mct_t i._mct_last##c._mct_t, ///
   add(1) force augment iterate(40)
if (_rc==430) {
  di as error "NO CONVERGENCE, fitting simplest gap model"
  mi impute mlogit _mct_state i._mct_next i._mct_last, ///
  add(1) force augment
}
else if _rc {
  exit _rc
}
end
```

In this case, if the full model fails to converge, a simpler model is fitted instead. If failure to converge is relatively rare, this will not affect the imputations very materially.

4.1.4 Simulated missing on MVAD data: some results

Using this model we generate ten imputations. Figure 2 shows four typical cases, with the fully observed data, the data with random runs of missingness imposed, and the ten imputations, shown as horizontal lines.[2] These display features that are typical of the full data set: first, short gaps with the same state before and after tend to get filled in with that state, which is often correct (e.g., case 33 and the first two gaps in case 62). When longer, such gaps show a

[2]The figure, an "indexplot", is created using the sqindexplot command from the SQ package (Kohler and Brzinsky-Fay, 2005; Brzinsky-Fay et al., 2006).

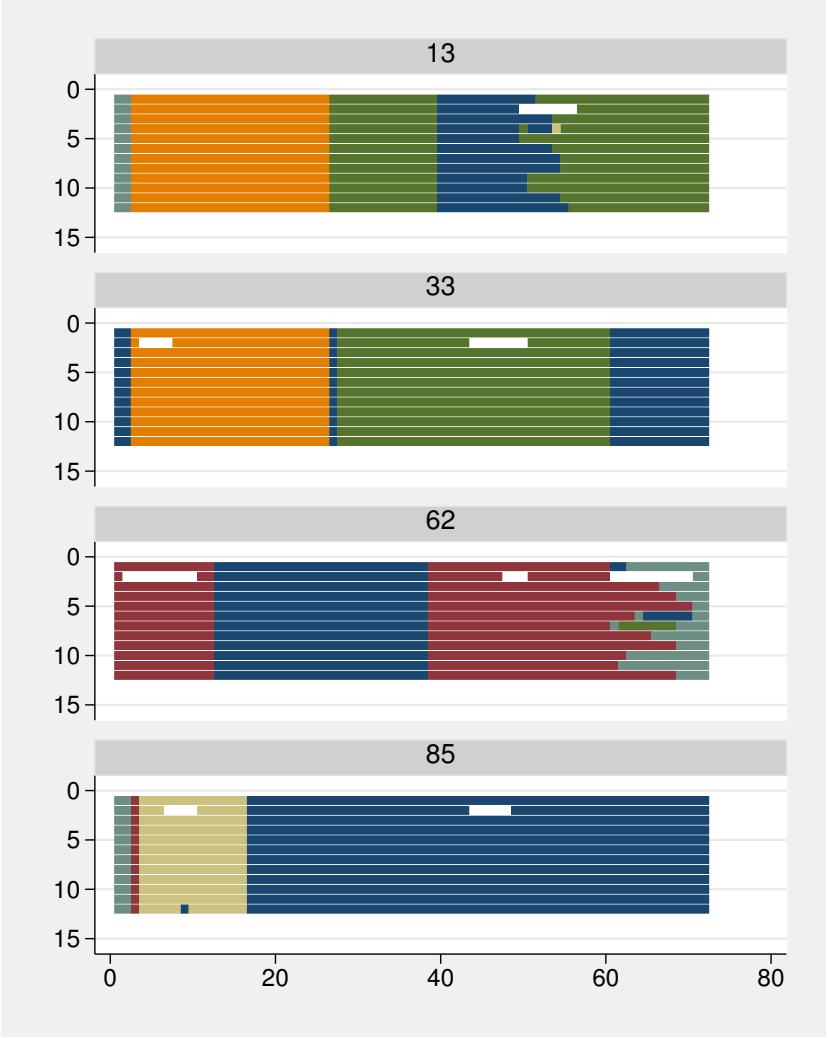

Figure 2: Sample imputations. For three typical cases, the fully observed data (first line), data with imposed missings (second line, missing is white), and ten imputations are shown.

quite small tendency to have other states interpolated (imputation 10 for case 85 shows an example). Where a gap has two different states as neighbours, the majority of imputations show a single transition from one to the other, where the timing of transition is randomly distributed in the gap (case 13). Such gaps show a larger tendency to interpolate one or more other states, particularly as the gaps get longer (more so than gaps bracketed by a single state). However, if the run of missingness completely obliterates a spell in the original data (which is something we can only know because we have the fully observed data; see the third gap in case 62), the imputations are very unlikely to pick up that state (though in this case one of the imputations does create such a state, if somewhat different in timing and duration).

Thus we see how to use MICT to program imputations using a simple but more-or-less realistic predictive model. We also see that in general, the imputations approximate the true data quite well, because there is a lot of redundancy in data like this. However, it should be noted that where a gap envelopes a complete spell, the redundancy is markedly less.

4.2 Simulated data with simulated missingness

We now move on to the second demonstration.

It is difficult to compare the performance of the gap-filling approach with multiple imputation with chained equations, via either the official `mi impute` or Royston's `ice`, because it is very difficult to specify analogous models. To facilitate comparison I present a simulation that permits much simpler models. A zero-order Markov process with time-constant transition rates is used to create 36-element long, 4-state sequences, with random gaps imposed. Since the generating process is zero-order, only adjacent last and next observations carry information with which to impute. Thus for the MICT approach the only meaningful imputation model uses just `_mct_last` and `_mct_next` as predictors, while for MICE, only the immediately adjacent states, s_{t-1} and s_{t+1}, are used. 2,000 sequences are generated, and for each method, MICT, `mi impute chained` and `ice`, 10 imputations are made.

In what follows I use both the `mi impute chained` and `ice` implementations of MICE. In this more conventional framework, chained imputation takes care of the fact that, given the data in a wide format, missingness is non-monotonic (i.e., that predictors of missing values may well themselves be missing).

For `ice` the models are defined as follows:

```
ice m.m1 m.m2 m.m3 m.m4 m.m5 m.m6 m.m7 m.m8 m.m9 m.m10 ///
    m.m11 m.m12 m.m13 m.m14 m.m15 m.m16 m.m17 m.m18 m.m19 m.m20 ///
    m.m21 m.m22 m.m23 m.m24 m.m25 m.m26 m.m27 m.m28 m.m29 m.m30 ///
    m.m31 m.m32 m.m33 m.m34 m.m35 m.m36, ///
  saving(puresim_ice_cycles, replace) persist m(10) cycles(10) ///
  eq(m1:    i.m2      , ///
     m36:  i.m35      , ///
     m2:    i.m1  i.m3,  ///
     m3:    i.m2  i.m4,  ///
 [ code omitted ]
     m35:  i.m34 i.m36)
```

For `mi impute chained` the following, rather verbose, code is used (note elisions):

```
mi set flong
```

```
mi register imputed m*
mi impute chained ///
  (mlogit, omit(          i.m3 i.m4  [ ... ]  i.m34 i.m35 i.m36 ))  m1   ///
  (mlogit, omit(               i.m4  [ ... ]  i.m34 i.m35 i.m36 ))  m2   ///
  (mlogit, omit(i.m1                 [ ... ]  i.m34 i.m35 i.m36 ))  m3   ///
  (mlogit, omit(i.m1 i.m2            [ ... ]  i.m34 i.m35 i.m36 ))  m4   ///
  (mlogit, omit(i.m1 i.m2 i.m3       [ ... ]  i.m34 i.m35 i.m36 ))  m5   ///
 [ ... ]
  (mlogit, omit(i.m1 i.m2 i.m3 i.m4  [ ... ]                    ))  m35  ///
  (mlogit, omit(i.m1 i.m2 i.m3 i.m4  [ ... ]  i.m34             ))  m36, ///
  add(10) force augment
```

Given the way the data is simulated, each of the three imputations has the best possible model in its framework.

Figure 3 shows some example imputations for a handful of cases across the three strategies. As can be seen, `mi impute` and `ice` show somewhat more transitions in the imputed sections. To test whether this is a systematic feature, I compare the number of spells in the imputed sequences with the true number (in the simulated data, before imposition of missingness). We can use `mi estimate` to test the null that the difference is zero, by running a null regression and looking at the t-statistic for `_cons` (this is a way of getting `mi estimate` to run a t-test).

Method	_cons	Std. Err.	t	p
MICT	.01025	.0253	0.40	0.687
`mi impute`	.2878	.0437	6.59	0.000
`ice`	.31645	.0347	9.12	0.000

Over the ten imputations of 2000 sequences, MICT does not signficantly increase the mean number of spells, while `mi impute` and `ice` do (by 0.29 and 0.32 respectively). Thus in this simple comparison, MICT retains longitudinal consistency whereas MICE does not.[3]

4.3 Mothers' labour market histories

Let us now consider how MICT performs with real missingness. Using data drawn from the British Household Panel Survey (BHPS), I created six-year monthly employment status histories, for women who have a birth at the end of the second year (Taylor et al., 2010; Halpin, 1998). This data set contains 706 fully observed sequences, another 190 with gaps of up to 12 months, and 425 with longer gaps. I choose to impute gaps of up to 12 months, but use data from sequences with longer gaps to provide information for the imputation models. See Figure 4, which shows the overall picture of a retreat from paid work as the birth approaches, and a qualified return afterwards, predominantly to part-time.

We use the following predictive model for imputing the internal gaps (analogous models are used for initial and terminal gaps):

```
capture program drop mict_model_gap
program define mict_model_gap
mi impute mlogit _mct_state ///
     i._mct_next##c._mct_t##c._mct_t i._mct_last##c._mct_t##c._mct_t ///
     _mct_before* _mct_after*
end
```

[3]Experiments with increasing the number of cycles in the MICE chains were attempted, to see if greater consistency could be achieved with a longer burn-in, but there was no tendency to a systematic improvement.

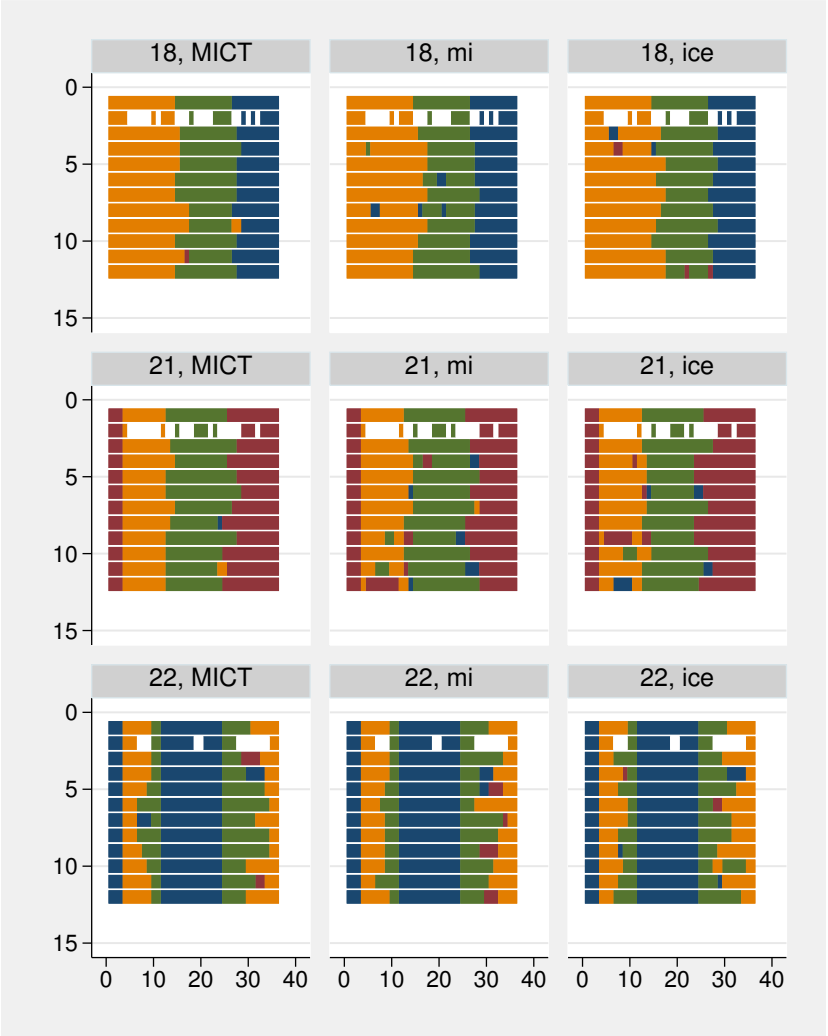

Figure 3: Imputations by MICT, `mi impute chained` and `ice`, three example sequences

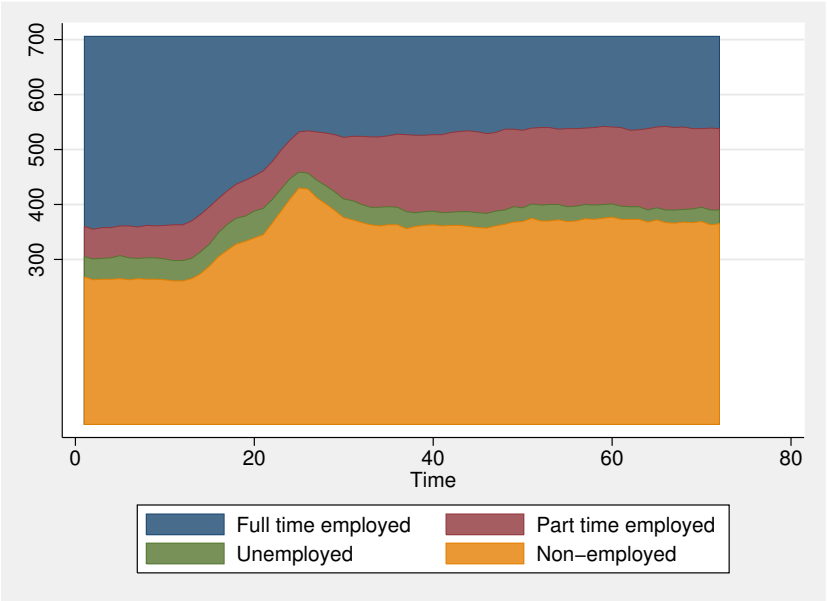

Figure 4: State distribution of fully observed mothers' labour market sequences

This is a relatively simple model, implying that transition patterns vary in a non-linear pattern, and using the prior and subsequent cumulative distributions of states. That is, the effects of the last and next observed states vary in interaction with a quadratic time term, and the `_mct_before*` and `_mct_after*` terms represent the proportion of time spent in the various states before and after the gap.

The following code runs the whole example, with the `maxgap(12)` and `maxitgap(6)` options to `mict_impute` limiting the imputations to cases with maximum internal gap lengths of 12 and initial/terminal gaps of 6 months. The `nimp(10)` option causes it to generate 10 imputations.

```
mict_prep state, id(pid)

capture program drop mict_model_gap
program define mict_model_gap
mi impute mlogit _mct_state i._mct_next##c._mct_t##c._mct_t ///
                            i._mct_last##c._mct_t##c._mct_t ///
                            _mct_before* _mct_after*, ///
          add(1) force augment noisily iterate(40)
end

capture program drop mict_model_initial
program define mict_model_initial
mi impute mlogit _mct_state i._mct_next##c._mct_t _mct_after*, ///
          add(1) force augment iterate(40)
end

capture program drop mict_model_terminal
program define mict_model_terminal
mi impute mlogit _mct_state i._mct_last##c._mct_t _mct_before*, ///
          add(1) force augment iterate(40)
end
```

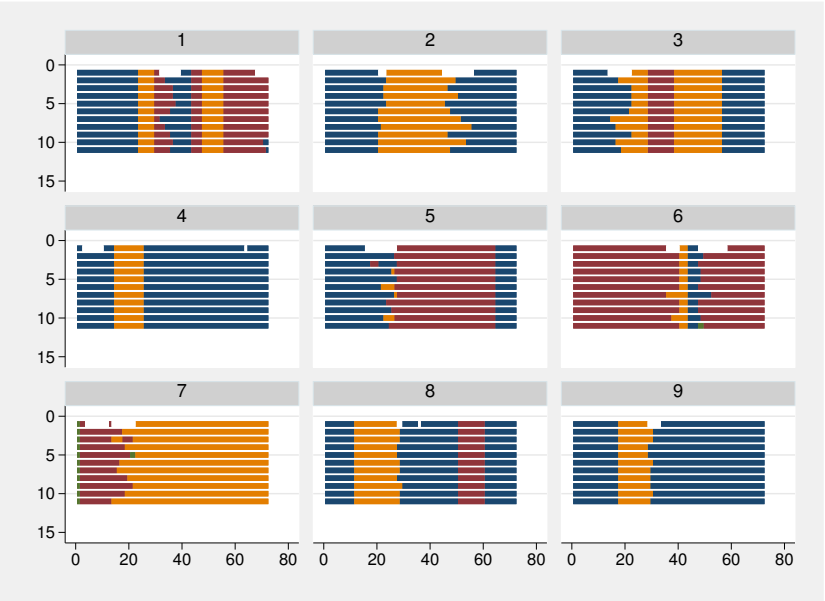

Figure 5: Imputations for selected cases, mothers' labour market history, first model

```
mict_impute, maxgap(12) maxitgap(6) nimp(10)
```

We can examine the performance of the imputation model in Figure 5. In outline the performance is similar to the simulations: gaps bracketed by a single state tend to be filled in by that state, with a certain amount of interpolation of other states, that increases as the gap-length increases; gaps between different states are often filled by the two states with a distribution of transition points, but are also quite likely to feature extra transitions and third states, again more so for longer gaps.

4.3.1 Which sequences get imputed?

In real life data, there tends to be a strong relationship between the nature of a lifecourse sequence and its probability of having gaps. In particular, volatile lifecourses will be more likely to be subject to interruptions in data collection, and in being complex are more vulnerable to missed measurement opportunities. We can see this by looking at indicators of complexity such as the number of spells, or the Shannon entropy.[4]

```
. table gap, c(n ent mean ent mean nsp) format(%5.2f)
```

gap	N(ent)	mean(ent)	mean(nsp)
0	7,060	0.53	2.39
1	1,900	1.03	3.80

[4]Utilities to calculate the number of spells, and the Shannon Entropy, are available in the SADI package (Halpin, 2014)

Fully observed sequences have a mean of 2.39 spells, compared to 3.80 for sequences that have been imputed. Similarly, fully observed sequences have a mean entropy of 0.53 compared to 1.03 for imputed. The number of spells tracks how transition-prone sequences are, and the entropy additionally indicates how diverse the states visited are. Clearly, on average gappy sequences that are imputed have more transitions and are more diverse.[5] This is important, in that complex sequences are often substantively interesting, and their exclusion reduces the useful information in the data set, as well as potentially introducing bias.

4.4 Refining models

The model used above is relatively simple. Various ways to improve its quality as an imputation model are available, most readily the inclusion of fixed individual level variables, and the incorporation of more interactions. Fully observed time-dependent variables in another domain can also be included, such as residential or marital histories. Where that variable is also subject to missing, a simple imputation strategy such as carry forward may be adequate.[6]

To add a completely observed (or filled-in) time-series in another domain, add that variable name (as a `reshape`-style stub) to the `mict_prep` statement. If the variable to be imputed is `state1` to `stateN` and the second variable is `resid1` to `residN`, the command would take this form:

```
mict_prep state resid, id(pid)
```

The `resid` variable is then available to be used in the model statement. More than one extra time-series variable can be added in this fashion.

4.4.1 Picking up the missingness process

In what follows we demonstrate the inclusion of an extra time-dependent variable, and address a substantive issue raised above, by the simulations. It was observed that in general there tends to be a high degree of redundancy in gappy lifecourse data like this, such that imputations resemble the observed data well, except where the gap overlaps a complete spell. If this happens at random it is not a serious problem for the imputation/estimation process, but if the occurrence of gaps is somehow associated with the spell structure of the history represented by the data, then the imputations will understate the true variability. Depending on the domain, and on how data is collected, this is quite likely to occur.

The mothers' labour market histories are drawn from the BHPS, whose data collection is annual, with retrospective accounts covering the period between the interview and the start-date of the last year's fieldwork (Halpin, 1998). This assures continuity if no interviews are missed, and the retrospective accounts

[5] This is not an artifact of the imputation process: if instead we mechanically carry forward the previous state to fill gaps, thus not increasing the number of spells or the diversity, we get very similar results.

[6] In the typical case where data in the other domain is missing at the same time as the main variable, MICT could in principle draw information in that domain from the prior and subsequent time points determined by the current gap-length, but that would require updating (imputing) the observations for the other domain as the process fills in the gaps. This would add another layer of complexity to the package, so simpler imputations are preferred.

are without measurement error (assumptions that are frequently violated). In the BHPS worklife histories, gaps may occur because of omitted spells, of spells whose start- or end-date is mis-reported, and of missed data collection points. In the first case the gap may be exactly coterminous with the spell, in the second, the gap will start at the beginning of the spell if the misreported date is late, and in the third the gap will begin just after the previous data collection point, and end at the start of the account at the next data collection point. This means that the process of missingness is quite structured. However, we have data relating to this process, on which we can draw. If the previously observed state was reported explicitly as a spell-end in the retrospective account, we know that there is a much greater probability that the current state is different, since a transition is indicated (it could however be a transition to the same state). It is the same case if the next observed state is reported as a start-of-spell. If the previous observation was the date of an interview, it is logically the case that the current state is more likely to be the same as the state at the interview, since no transition is reported. However, in the observed data there is a distinct pattern of seam effects, where the account from the following interview clashes with the prior data, often by backdating a spell start such that it precedes the earlier interview. In the data used, this is represented as a transition immediately after the interview, on the logic that the current state reported at time t-1 is more authoratitative than the retrospective report from time t (Halpin, 1998): thus whether months of interview tend to be followed by elevated transition risk is an empirical question.

To account for these effects, a variable is created which distinguishes between "neutral" months, explicitly reported spell starts from the inter-wave job history (where a start may be a reported start from a spell with missing state information but valid dates, or one month after the end of a fully reported spell), explicitly reported spell starts of the spell current at interview, and months in which the annual interview fell. Where no information is available, it is allowed default to neutral. This variable, `obstype`, is incorporated in the imputation model as follows:

```
program define mict_model_gap
capture drop _mct_n2 _mct_l2
recode _mct_last 3=2, generate(_mct_l2)
recode _mct_next 3=2, generate(_mct_n2)
mi impute mlogit _mct_state i._mct_next##c._mct_t##c._mct_t ///
                            i._mct_last##c._mct_t##c._mct_t ///
                            _mct_before* _mct_after* ///
                            i.obstype##i._mct_n2 ///
                            i.obstype##i._mct_l2, ///
   add(1) force augment
end
```

It is modelled in interaction with the next and prior states, since its effect is via the transition pattern. However, in this data there are relatively few transitions to and from unemployment, which causes problems in estimation, so `obstype` is interacted with recoded versions of these variables, which are recreated each iteration.[7]

Sample imputations are shown for the models with and without the observation structure variable in Figure 6. In each panel the first row indicates the

[7] In general, creation of any sort of transformed variable can be carried out within the `mict_model_gap`, `mict_model_initial` and `mict_model_terminal` programs.

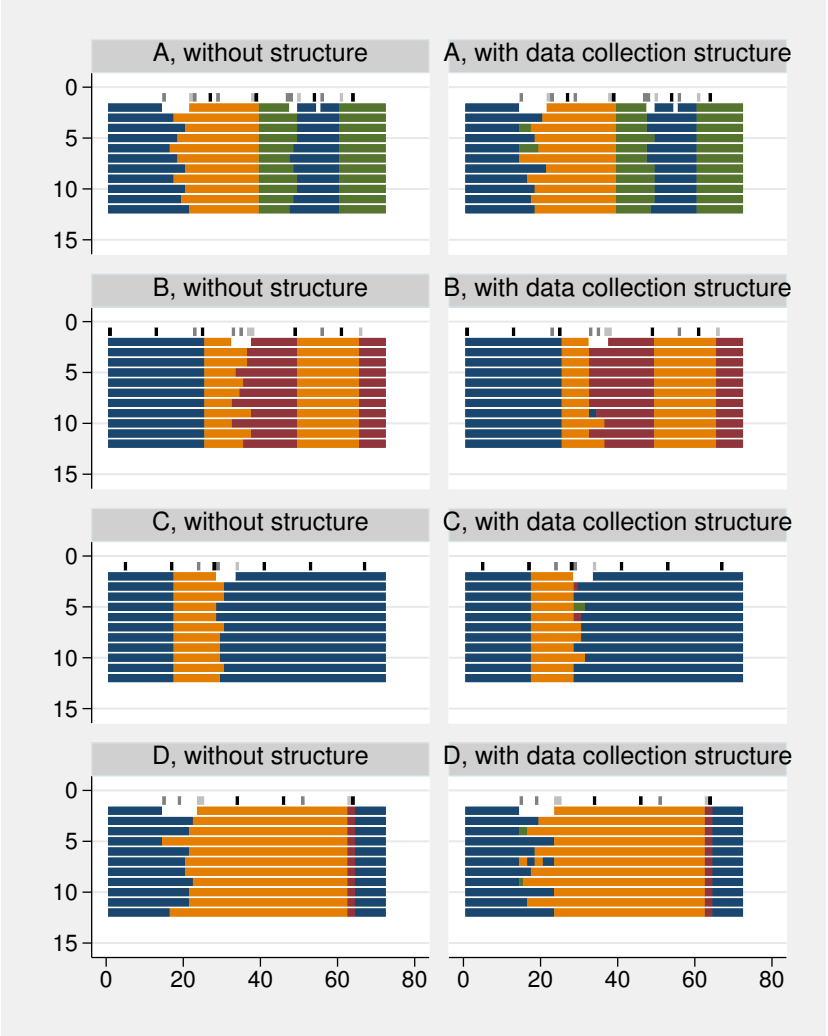

Figure 6: Selected imputations without (L) and without data collection info (R)

sturcture (black is the month of interview, dark grey a start of spell in the inter-wave job history, and light grey the start of a spell current at the interview). As can be seen, taking account of the meta-information regarding spell structure has a systematic effect: first, the imputations are more likely to contain transitions at the key points, and second, they are more likely to interpolate spells in third states. Both of these are desirable since the model without the meta-information understates not only the transition rates around these key points, but also the variability of the imputations.

This illustration is guided by the particular data collection structure of the BHPS, but it is likely that many longitudinal data sets will contain meta-information that could inform the imputation model in an analogous way.

5 Conclusion

`MICT` offers a flexible and longitudinally consistent means of multiply imputing categorical time-series data, particularly when it is characterised by relatively long spells in states, and consecutive runs of missingness, as is typically the case in lifecourse data. As the second simulation (section 4.2) shows, it produces imputations with largely plausible patterns of transitions, while MICE (as either `mi impute chained` or `ice`) generates unrealistically elevated rates of transition.

The key advantage is that MICT structures the imputations as a series of monotonic imputations, focusing on filling gaps. This allows a good deal of flexibility, including the use of binary, multinomial or ordinal logistic regression models as appropriate. It offers a reasonably user-friendly interface to defining models for `mi impute`, such that it is relatively easy to define good imputation models, incorporating fixed and time-dependent effects. It produces imputations that can be used by the `mi impute` post-imputation infrastructure.

It has a number of disadvantages, not least that it presents yet another interface to imputation, a compatible but separate solution to `mi impute`. It can deal only with a single target variable for imputation. It does not update created variables such as `_mct_before*`, nor does it update other time-series variables that may be used in the model (thus effectively falling back on carry-forward imputation for these variables). Extending the package to cope with these would be relatively complicated. However, it presents a relatively lightweight and effective solution for imputing single categorical time-series variables.

5.1 Existing applications

This approach has already been used in practice (based on the functionally equivalent but less user-friendly approach described in Halpin (2012, 2013)). Fuller and Stecy-Hildebrandt (2015) apply it to the Canadian Survey of Labour and Income Dynamics, and McMunn et al. (2015) and Lacey et al. (2015) apply it to the British cohort study data sets (the MRC National Survey of Health and Development 1946 birth cohort, the National Child Development Study 1958 birth cohort, and the British Cohort Study 1970 birth cohort).

References

Allison, P. D. (2005) Imputation of categorical variables with PROC MI, *SUGI 30 Proceedings 2005*, 1–14.

Allison, P. D. (2009) Missing data, in R. E. Millsap and A. Maydeu-Olivares (eds), *The Sage handbook of quantitative methods in psychology*, Sage, Thousand Oaks, California.

Brzinsky-Fay, C., Kohler, U. and Luniak, M. (2006) Sequence analysis with Stata, *Stata Journal*, **6**(4), 435–460.

Fuller, S. and Stecy-Hildebrandt, N. (2015) Career pathways for temporary workers: exploring heterogeneous mobility dynamics with sequence analysis, *Social Science Research*, **50**(0), 76–99.

Halpin, B. (1998) Unified BHPS work-life histories: combining multiple sources into a user-friendly format, *Bulletin de Méthodologie Sociologique*, (60).

Halpin, B. (2012) Multiple imputation for lifecourse sequence data, *Working Paper WP2012-01*, Dept of Sociology, University of Limerick, Ireland.
URL: *http://www.ul.ie/sociology/pubs/wp2012-01.pdf*

Halpin, B. (2013) Imputing sequence data: extensions to initial and terminal gaps, stata's `mi`, *Working Paper WP2013-01*, Dept of Sociology, University of Limerick, Ireland.
URL: *http://www.ul.ie/sociology/pubs/wp2013-01.pdf*

Halpin, B. (2014) SADI: Sequence analysis tools for Stata, *Working Paper WP2014-03*, Dept of Sociology, University of Limerick, Ireland.

Honaker, J. and King, G. (2010) What to do about missing values in time-series cross-section data, *American Journal of Political Science*, **54**(2), 561581.

Kohler, U. and Brzinsky-Fay, C. (2005) Stata tip 25: Sequence index plots, *Stata Journal*, **5**(4), 601–2.

Lacey, R., Stafford, M., Sacker, A. and McMunn, A. (2015) Work-family life courses and subjective wellbeing in the MRC National Survey of Health and Development (the 1946 British birth cohort study), *Journal of Population Ageing*, 1–21.
URL: *http://dx.doi.org/10.1007/s12062-015-9126-y*

McMunn, A., Lacey, R., Worts, D., McDonough, P., Stafford, M., Booker, C., Kumari, M. and Sacker, A. (2015) De-standardization and gender convergence in workfamily life courses in great britain: A multi-channel sequence analysis, *Advances in Life Course Research*, (0), –.
URL: *http://www.sciencedirect.com/science/article/pii/S1040260815000313*

McVicar, D. and Anyadike-Danes, M. (2002) Predicting successful and unsuccessful transitions from school to work using sequence methods, *Journal of the Royal Statistical Society (Series A)*, **165**, 317–334.

Royston, P. (2004) Multiple imputation of missing values, *The Stata Journal*, **4**(3), 227–241.

Royston, P. (2009) Multiple imputation of missing values: further update of ice, with an emphasis on categorical variables, *Stata Journal*, **9**(3), 466–477(12).

Rubin, D. (1987) *Multiple imputation for non-response in surveys*. New York, John Wiley and Sons.

Taylor, M., Brice, J., Buck, N. and Prentice-Lane, E. (2010) British Household Panel Survey User Manual, *User documentation*, Institute for Social and Economic Research, University of Essex, Colchester.

van Buuren, S. (2007) Multiple imputation of discrete and continuous data by fully conditional specification, *Statistical Methods in Medical Research*, **16**(3), 219–242.

van Buuren, S., Boshuizen, H. and Knook, D. (1999) Multiple imputation of missing blood pressure covariates in survival analysis, *Statistics in Medicine*, **18**, 681–694.

van Buuren, S. and Groothuis-Oudshoorn, K. (2011) mice: Multivariate imputation by chained equations in R, *Journal of Statistical Software*, **45**(3), 1–67. **URL:** *http://www.jstatsoft.org/v45/i03*

Brendan Halpin is Senior Lecturer and Head of the Department of Sociology at the University of Limerick, and has a longstanding interest in longitudinal social science data.