PanelPatch: Handling Attrition and Wave Nonresponse

A User Guide developed for use with Early Learning Study at Harvard (ELS@H) Data

June 2023

Submitted to:
Stephanie Jones, Harvard Graduate School of Education
14 Appian Way, Larsen 702
Cambridge, MA 02138

Submitted by:
Abt Associates
6130 Executive Boulevard
Rockville, MD 20852

About This Report

This report was funded through the Saul Zaentz Early Education Initiative at the Harvard Graduate School of Education as part of the Early Learning Study at Harvard (ELS@H).

PanelPatch

A Stata macro designed to prepare data from panel studies for longitudinal analyses involving more than two time points, such as growth curve analysis.

Designed to be used with the original year 1 ELS@H sample. Will not be useful on datasets that include the children added to the sample in year 2.

Authors

David Judkins, Eric Hedberg, Jesse Wood, and Kerry Hofer

Abt Associates | 6130 Executive Boulevard | Rockville, MD 20852

CONTENTS

Using funding from the Early Learning Study at Harvard (ELS@H) project, Abt Associates has developed a comprehensive system of Stata macros to facilitate the preparation of ELS@H data prior to conducting longitudinal analyses such as growth curve analyses. The "PanelPatch" system is designed with two goals in mind. The first goal is to salvage a significant proportion of the partial information from incomplete case histories. The second goal is to enable typical Stata users to achieve the first goal without the assistance of senior statistical consultants. This manual contains five chapters. In the first chapter, we discuss the challenges of incomplete case histories in longitudinal data, our general approach to solving this problem, and the advantages of our approach over other approaches. In the second chapter, we provide guidelines on how to structure a data set to maximize the compatibility with the PanelPatch system. The third chapter is dedicated to providing a detailed explanation of the algorithms utilized within PanelPatch. In the fourth chapter, we provide a practical guide to the PanelPatch system with step-by-step instructions. Lastly, the fifth chapter demonstrates the strengths and weaknesses of the PanelPatch system through a simulation exercise.

1. Theory

In any longitudinal study, it is common for participants to permanently drop out of the study after participating in a certain number of waves. In the survey literature, this phenomenon is referred to as attrition, and the participants whose trajectory matches this pattern are referred to as "**attritors**." Additionally, some participants may temporarily skip a wave but subsequently resume their participation in the study. This phenomenon is referred to as "**wave nonresponse**." Under both scenarios, the participant is missing at least one whole wave of responses. Of course, wave participants may leave some questions unanswered, leading to "**item nonresponse**." Lastly, "unit nonresponse" refers to cases where participation is declined for the entire study.

There are already many existing tools for handling unit nonresponse (in which the study member declines participation at all waves). Moreover, Abt Associates has already prepared weights for ELS@H that correct for differential probabilities of selection and for unit nonresponse. Analyses of wave 1 respondents with these weights yield analyses that are broadly representative of all 3- and 4-years olds in the state of Massachusetts. Given the availability of these weights, PanelPatch assumes that a set of appropriate weights for wave 1 respondents already exists. Nor is our primary focus on item nonresponse. Although PanelPatch deals with item nonresponse, there are other tools that would be easier to use than PanelPatch if the user is interested in analyzing only a single wave of ELS@H. Wave nonresponse and attrition are much more difficult to address well in longitudinal analyses such as growth curve analyses. There is the simple solution of treating them both as unit nonresponse, but this involves discarding all collected data on all cases missing at least one wave. Throwing all those partial cases out has the certain consequence of power loss and the likely consequence of introducing unknown biases.

Another potential, seemingly simple but generally infeasible, solution would be to treat all the items in the missed waves as just another form of item nonresponse. In a typical study, this would mean simultaneously imputing hundreds of variables. This would be a severe challenge for most imputation procedures (due to both memory limitations and required computational time). Although the simple hotdeck procedures pioneered at the Census Bureau can handle very "wide" vectors of variables with item nonresponse, they do a poor job of preserving relationships of the imputed variables to each other.[1]

[1] The original hotdeck procedure from the 1960s could handle an immense set of variables. It simply copied missing items from the last good respondent who matched on a few pre-specified variables known from the sampling frame such as urban/rural

On the other hand, more modern imputation procedures based on Gibbs sampling can preserve relationships among imputed variables but cannot handle such wide vectors.[2]

Given that there are no existing simple tools with desirable properties for dealing with missed waves in a longitudinal study, we developed PanelPatch to meet this challenge. Its general approach is to impute whole waves for study members who only missed one or two waves and to develop weights to compensate for study members who missed more than two waves. The idea for whole wave imputation started with work on the National Medical Care Utilization and Expenditure Survey (NMCUES) (Cox and Bonham, 1983), where it was referred to as "attrition imputation." In PanelPatch, each case that is missing a particular wave is matched to a case (aka a *donor*) that participated in all waves of the study.[3] Then the responses for the missing wave are simply copied from the donor to the wave nonrespondent. The donor matching is conducted in terms of data available from other waves. The transfer of a group of related responses from a single donor ensures that the copied vector is internally consistent within each wave. If the matching is performed well, then longitudinal features such as growth rates and transition probabilities should also be sensible.

Using PanelPatch, a moderately sophisticated Stata user can develop a set of weights and whole-wave imputations that dramatically simplify all subsequent analyses. After running PanelPatch, Stata's full set of analytic procedures (including multi-level modeling procedures) can be utilized without further thought about missing waves. However, use of PanelPatch does require some thinking. It is important for the user to define the research question at hand in considerable specificity in order to get good results. The core challenge is that in a study with thousands of measured inputs and outcomes, the matching of missed waves to donors cannot use all the available data. We highly recommend that the user only involve those variables in the matching that are directly relevant to the specific research question of interest. We refer to this collection of variables as the "**target ensemble**." More information on the required and suggested data set up is found in the following chapter.

While we could have applied PanelPatch to the full set of ELS@H data ourselves and delivered something like a public use file to the Harvard team, we think that a single match cannot serve all purposes well. Decades ago, Judkins (1998, 2000) recommended that a software approach be developed that allows users to narrowly focus imputation on the set of variables needed for a particular analysis. Since then, a variety of software packages have been developed for item imputation that follow this early suggestion, but this is the first package to do so for wave imputation.

Less attention has been paid to the possible benefits of customizing weights to the analysis at hand, but the capability in PanelPatch to do so may be very useful. Most weighting adjustments for missing data model the probability of nonresponse as a propensity score and use (following Little, 1986) either the inverted estimated propensity or a smoothed function of it as the nonresponse adjustment weight. This procedure is known to give the best general-purpose weights but can lead to substantial power losses without offsetting bias reductions.[4] This happens when a baseline variable exists that is strongly predictive of nonresponse but only weakly related to the outcome of substantive interest. In order for a variable to cause serious nonresponse bias, it must be strongly related to both nonresponse propensity and the substantive variables of interest. PanelPatch avoids unnecessary power loss

---

status. This procedure preserves relationship on the imputed variables to that handful of pre-specified variables, but all other relationships fade.

[2] For example, in our experience the FCS method within SAS/MI procedure can handle at most ~50 variables at a time.

[3] Participation here means that the child has some data in the target ensemble at each wave included in the analysis.

[4] Aggressive use of tree-regression to model nonresponse is well known to lead to substantial power losses.

by focusing the nonresponse adjustment on the variables that predict both attrition and the substantive variables of interest.

After running PanelPatch, the resulting dataset can be analyzed with any of Stata's commands that support multiple imputation. Stata has a fairly rich set of analytic commands that meet this requirement, but there are some omissions. In particular, we note that none of Stata's commands support simultaneous use of "replicate" weights and multiple imputations.[5] Other software systems such as SUDAAN and SURVEY (an R package) do offer procedures that support such simultaneous use. This poses a challenge to ELS@H users, in particular, because the statisticians who developed the weights for Wave 1 also developed replicate weights and strongly advocated their use for variance estimation. One solution is to just use the "linearization" approach to variance estimation. Stata SVY commands do support the simultaneous use of linearization and multiple imputations. Another option would be to conduct final analyses in one of the other systems after conducting preliminary analyses in Stata. We return to this issue in Section 4.5.

[5] "Replicate" weights are used for variance estimation only. They are designed to capture the effects of complex weight-adjustment systems that violate the assumption that samples from different clusters are independent of each other. They are produced by perturbing the main sample weights and then repeating all adjustments separately on each set of perturbed weights. For a gentle introduction to variance estimation based on replicate weights, see Wolter 2007.

2. Data Set-up

We envision that a variety of analysts will use ELS@H data to answer a variety of research questions. We assume that each of these analysts will want to publish a paper on the issue studied. We assume further that the work to prepare a paper will entail a variety of analyses with a fairly narrow subset of all the variables collected across the history of ELS@H. Our idea is that each analyst will need to run PanelPatch once (and only once) to support their work on a particular paper. PanelPatch can take multiple hours of CPU time to run. The user needs to identify the ensemble of variables required for a particular paper. If too wide an ensemble is specified, the required CPU time may become impractical. The set up needs to be carefully considered—wide enough to support all analyses required for a paper but narrow enough to run in a reasonable length of time. Since different papers will involve different ensembles of variables, it will generally not be sensible to share the output of PanelPatch across papers. Each analyst will need to run it themselves. If they choose the ensemble well, they will only need to run it once per paper.

To ensure compatibility with the PanelPatch system, it is important to adhere to a specific structure for the input dataset. This chapter provides guidelines on how users should organize their data and classify the target ensemble variables before utilizing PanelPatch.

2.1 Data Structure & Identification Variables

The dataset must be structured to the child-wave level.[6] In other words, there must be a unique record per wave per child that is identified by the combination of a child ID variable and a wave ID variable. For variance-estimation purposes, each record must also include a sample cluster ID which is constant for each child across waves (the early child care provider for each child in the first wave in the case of ELS@H).[7] In addition, a wave response indicator variable and a base sampling weight for each child (i.e., sampling weight adjusted for nonresponse in the first wave) must exist for each record.[8]

Exhibit 2-1 provides an illustrative example of a data set ready for PanelPatch. Although Child A is represented within the data across three instances, pairing the child ID with the wave ID variable provides three unique child-wave records. For example, only one instance for Child A and Wave 1 exists across the entire data set. The provider ID and base sampling weight should be constant for each child across each wave. The wave response variable indicates that Child A did not respond in Wave 2 (i.e., Wave Response = 0). In this case, all repeating variables (i.e., variables that are updated at each wave) will be missing for this specific record. However, item nonresponse may still occur within the data even if the record indicates a wave response. Item nonresponse should be recorded as system

[6] This report is written for the ELS@H project and refers to participants as "child" to match the study. However, PanelPatch could be used to prepare any panel for longitudinal analyses. If used on another panel, the user may replace "Child ID" with their respective unit identification variable.

[7] The child ID must at least be unique within each provider ID. For variance estimation purposes, it is important that the imputation and weighting adjustments preserve cross-cluster variation. If there exists a stable variable that defines the provider type, and the imputation model includes provider type fixed effects, the provider ID will only be used to provide cluster-corrected standard errors. If no provider ID exists, the user should generate a constant "stand-in" provider ID. Note that the use of provider ID in PanelPatch does not compel the user to use cluster-corrected analysis methods. After running PanelPatch, the user can ignore the provider ID in their analyses if desired.

[8] Sampling weights differ from frequency weights and should denote the inverse of the probability that a child is included in the sample due to the sampling design. Abt Associates provided the ELS@H team with base sampling weights that have not been adjusted for nonresponse.

missing values for the relevant variables (e.g., the repeating variable for Child B and Wave 1). Chapter 3 explains how PanelPatch imputes the missing data for each of these cases.

Exhibit 2-1 Example Data Structure

Child ID	Wave ID	Provider ID	Base Sampling Weight[9]	Wave Response	Repeating Variable
A	1	Provider C	3997.456099	1	0
A	2	Provider C	3997.456099	0	
A	3	Provider C	3997.456099	1	1
B	1	Provider D	4026.466052	1	
B	2	Provider D	4026.466052	1	0
B	3	Provider D	4026.466052	0	

Note to ELS@H Team: Abt Associated provided "Child_Stratumid" and "Child_Cluster" as the strata and unit identification variables, respectively. As a reminder, the child-level sample weight is named "CHILD_WEIGHT_FINAL" and replicate weights beginning with the stem "Child_Weight_Replicate".

2.2 Target Ensemble Variables

The user must define a target ensemble based on their research question. The target ensemble consists of variables of analytic interest *as well as* any relevant auxiliary data (i.e., other variables to be used in identifying donors such as potential confounders, moderators, and mediators). Prior to utilizing PanelPatch, the user should classify all target ensemble variables into one of the following 12 classes:[10]

1. Stable variables (baseline variables that are constant across waves for a child)
    a. Binary
    b. Ordered categorical
    c. Unordered categorical
    d. Interval-valued
2. One-off variables (variables that exist for specific waves but not all waves)
    a. Binary
    b. Ordered categorical
    c. Unordered categorical
    d. Interval-valued
3. Repeating variables (variables that are updated at each wave)
    a. Binary
    b. Ordered categorical

[9] PanelPatch also allows for replicate weights. Each replicate weight will be adjusted for nonresponse. If replicate weights are used, some software other than Stata will be needed analyze the data after PanelPatch.

[10] Variables that change in a deterministic fashion across waves such as current age do not fit neatly in any of these categories, but they probably do not need to be included given that waves are about one year apart in ELS@H. Age at baseline, of course, would be a stable variable.

c. Unordered categorical
d. Interval-valued

Caution: It is recommended that the user backfill in any missing wave 1 stable variables if they are available in subsequent waves. While PanelPatch will impute any missing wave 1 stable variable, it is not guaranteed to match the value found in subsequent waves.

Caution: To ensure optimal performance and reduce computation time, it is strongly advised that users include only variables that are directly relevant to their immediate research question within their target ensemble. Including unnecessary variables can significantly increase the computational workload and potentially slow down the analysis process. We have successfully run jobs with age, 30 stable variables, and 30 repeating variables. These runs took close to three hours on an Apple M1 MacBook Pro running Stata 17 with 4 Computation Cores available, and nearly eight hours on a shared virtual Windows 10 machine also running Stata 17 with multiple computation cores in the cloud.[11] We have not experimented with higher numbers of variables and so do not yet know where the practical limit might be.

Caution: The current version of the PanelPatch system does not provide native support for one-off variables. However, if a one-off variable is necessary, users can manually address this by restructuring the data. This involves replacing any missing values across waves with the one-off value. Once this restructuring is completed, the one-off variable can be treated as a stable variable within the PanelPatch system. After the run is complete, the user can then delete the extra records corresponding to waves where the one-off variable was not asked.

Exhibit 2-2 provides an example of how to prepare stable variables, age, and converting one-off variables into stable variables prior to using PanelPatch. The left table shows a missing stable variable value in wave 1 and a missing age value in wave 3. The right table shows the user inputted data underlined and in bold. The stable and age variables are now ready to be utilized in PanelPatch. Similarly, the left table shows an example of a wave 2 one-off variable, and the right table shows the one-off variable as ready to be utilized as a stable variable.

Exhibit 2-2 Preparing Stable and One-off Variables for PanelPatch

Original Data (not prepared for PanelPatch)				
Child ID	Wave ID	Stable Variable	Age	One-off Variable
A	1		4	
A	2	1	5	19
A	3	1		

Data Prepared for PanelPatch				
Child ID	Wave ID	Stable Variable	Age	One-off Variable
A	1	**1**	4	**19**
A	2	1	5	19
A	3	1	**6**	**19**

[11] These estimates of runtime include the time to run all three of the macros (ItemImpute, WaveImpute, and AttritWeight). The benefit of multiple computation cores in this case is mostly for the matrix computations typical of regression models, but as this set of programs each run several models, the overall benefit is limited as several of the steps are procedural and not independent from preceding or subsequent steps; hence these run times are not likely to be improved by purchasing a more expensive Stata License.

2.3 Installing PanelPatch

PanelPatch is not available for download from the typical Stata library repositories. In order to use it within your Stata session, it must in installed in a directory which Stata lists as a source of commands. These paths are displayed in Stata when the use types “adopath.” Placing the unzipped PanelPatch directory in either the path noted as “Personal” or “Plus” will make the command available to users who also have access to that directory.

PanelPatch, like any other Stata user-written command, can also be installed in a directory not listed in those reported when typing “adopath.” For example, if multiple users employ Stata for a project and want to ensure that a common version of PanelPatch is used, the unzipped folder could be placed in a shared directory, say the same directory in which data for analyses are stored. This directory must then be declared in a do-file using the adopath + “path” command, where “path” is replaced with the Windows or Li/Unix or path.

Example

If a shared directory has a path D:\ChildStudy\Analyses\AbtCode

Then you can add PanelPatch to this directory and add

adopath + “D:\ChildStudy\Analyses\AbtCode\”

to your analysis or data preparation do-files as needed.

3. Algorithms

When using the PanelPatch system, three major macros are executed to facilitate the data imputation process: **ItemImpute**, **WaveImpute** and **AttritWeight**. The **ItemImpute** macro addresses the issue of item nonresponse by imputing any skipped items within the target ensemble. The **WaveImpute** macro imputes the target ensemble variables for skipped waves, addressing wave nonresponse. The **AttritWeight** macro develops nonresponse-adjusted sampling weights for use in analysis.

3.1 ItemImpute Macro

ItemImpute incorporates modern methods native to Stata to impute instances of item nonresponse.[12] This macro maintains consistency with the literature and other Stata procedures by automatically applying these methods to each of the target ensemble variables. These methods work best on small ensembles of data. If the number of variables in an ensemble is greater than three or four dozen, run times might be very slow.

These methods fill in initial guesses for all the missing items and then gradually improve those initial guesses by iteratively modeling each target ensemble variable in terms of all the remaining variables and then using these models to make new guesses (i.e., imputations). This type of procedure is often referred to as Gibbs sampling and has its origins in astronomy and remote sensing.

The ItemImpute procedure sets up Stata to use the appropriate model based on the given target ensemble variable's category (e.g., binary, ordered categorical, unordered categorical, or interval-valued). Each model includes the remaining variables plus dummies for the sample cluster variable (early child care provider in the case of ELS@H). For repeating variables, the models also include the values of all the repeating variables from both the prior wave (for waves after wave 1) and from the following wave (up through the second to final wave).

In addition to being sensitive to the number of variables in the target ensemble, run-times are sensitive to the number of "burn-in" cycles that are requested. The burn-in phase helps the imputation process achieve convergence to a steady-state by ignoring the first specified number of iterations. Higher values of burn-in allows the initial imputed data set to be more "stable" for the imputation phase, which leads to better correlations between variables in the final imputed data set. Due to the trade-off between computing time and the number of burn-in iterations, PanelPatch uses a default value of 10 burn-in iterations. However, the user is able to set this to their desired value. It is highly recommended to have at least 5 burn-in cycles to avoid the potential convergence to an unsteady-state. Marginal means tend to converge very quickly, but correlations converge more slowly. If resources allow, specification of 25 burn-in cycles will tend to recover more of the complex structure.

After running ItemImpute, all stable variables will be non-missing with either the original value or the imputed value. Any originally missing repeating and one-off variables will also be imputed for any records with a wave response but will remain missing for nonresponse waves. Instead, these values will be imputed using the WaveImpute procedure described below.

Caution: ItemImpute will not give sensible imputations for variables that are only supposed to be answered by subsets of the data (based on responses to prior questions in the instrument). Imputations may also be problematic for collections of variables with strong nonlinear relationships to each other, depending on the strength of the

[12] PanelPatch uses Stata's native multiple-imputation suite to deal with item nonresponse. Visit https://www.stata.com/manuals/mi.pdf for more information.

nonlinearity. The user can partially compensate for the second issue by including appropriate transformations of the raw variables.

3.2 WaveImpute Macro

The WaveImpute procedure provides a novel approach to handling cases of attrition and wave nonresponse. The core idea of WaveImpute is to match each nonresponding child to a responding child (i.e., donor) and transfer any relevant data from the donor to the nonresponding child. We define a "responding" child as one with at least one response for all available waves. Since WaveImpute follows the ItemImpute procedure, responding children may have imputed values for item nonresponse cases. Alternatively, a "nonresponding" child has at least one instance of wave nonresponse. The default behavior for the WaveImpute procedure is to impute all variables in the target ensemble at each missing wave for each child for whom at least some information (on some variable in the ensemble) at each of at least three waves.[13] The next macro in the system (AttritWeight) will use the short histories (one or two waves) to refine the nonresponse-adjusted weights from wave 1, but, other than that, these cases with short histories cannot contribute much information to a growth curve analysis and are therefore excluded from wave imputation.

To find a good match and transfer the data, WaveImpute executes four steps, each explained in more detail below:

1. Modeling of repeating variables
2. Clustering of children based on their predicted trajectories
3. Matching each nonresponding child with a responding child from the same trajectory cluster
4. Diagnostic reports and data output.

Step 1. Modeling of Repeating Variables

For each repeating variable in the target ensemble, WaveImpute fits a linear or generalized linear growth curve model. The structure of the growth curve model varies according to the user-inputted variable category found in Section 2.2.[14] These models include (as predictors) the average values of the other variables in the ensemble from the nearest responding wave or waves,[15] wave, wave-squared, and fixed effects for the sample cluster variable.[16] WaveImpute uses a separate cross-validated linear LASSO to select a smaller set of these variables for the generalized models in order to improve processing speed and minimize overfit. The model-predicted values of the repeating variables for every child for all waves (including both waves that were and were not skipped) are then used to cluster the sample as described in the next step.

Step 2. Clustering Children by their Predicted Trajectories

WaveImpute uses the predicted values of the variables in the ensemble to cluster the sample. The algorithm tries to minimize the variation of all the variables in the ensemble within clusters and maximize the variation in cluster averages subject to a minimum size constraint in terms of responding children.[17] This cluster size constraint allows

[13] PanelPatch allows the user to adjust the required minimum number of responding waves.
[14] Binary and interval-valued variables utilize an OLS regression framework; ordered categorical variables utilize an ordered logit regression framework; and unordered categorical variables utilize a multinomial logit regression framework.
[15] The first and final waves only include average values for the leads and lags, respectively, for the remaining variables. Every other wave includes both average values for the leads and lags.
[16] In the case of ELS@H, these would be the daycare providers from which the child were initially sampled.
[17] The algorithm is based on the *k*-means algorithm provided by Stata but has a special wrapper to enforce the size constraints. The wrapper works somewhat like tree regression, collapsing clusters that are too small and attempting to split large clusters. The default minimum size constraint is at least five responding children per wave.

for meaningful multiple imputation as will be explained more fully in the next step. Given the mix of variables measured on different scales, we used the Gower distance measure (Gower, 1971) to define the distance between pairs of children.[18] Dummy variables are used for the levels of unordered categorical variables, unless the level is rare, in which case, it is ignored.

Note: These clusters have nothing to do with geography. They instead group together sample members with similar trajectories. In the cases of ELS@H this might mean children who received similar services during their preschool year and have similar outcomes during their early school years.

Step 3. Matching Nonresponding Children to Responding Children

For each nonresponding child, WaveImpute selects a simple random sample with replacement of responding children from the same cluster. These responding children are referred to as donors because their values for the repeating variables are used to fill in the missing gaps in the histories of the nonresponding child. The user must specify the number of multiple imputations they require using the *add* option.

Step 4. Diagnostic Reports and Data Output

The output of WaveImpute includes:

- For each wave, the number of study members for whom all the repeating variables were imputed because of wave nonresponse;
- For each wave, the weighted mean values of binary and interval-valued repeating variables variable (with the associated cluster-corrected standard errors) before and after imputation;
- For each wave, the weighted frequencies of categorical repeating variables (with the associated cluster-corrected standard errors) before and after imputation;
- The cross-wave linear and quadratic trends in each binary and interval-valued variable (with the associated cluster-corrected standard error) before and after imputation;
- The gamma statistic for the association of each ordered categorical repeating variable with wave before and after imputation;
- The cross-wave linear trend in each level of each unordered categorical variable (with the associated cluster-corrected standard error) before and after imputation;
- The correlation of wave 1 with wave 5 for each interval-valued repeating variable before and after imputation; and
- The R-squared for a linear model of each binary, ordered categorical, and interval-valued variable at wave 5 in terms of the complete set of stable variables before and after imputation.

[18] The Gower distance between two cases based on a vector of measurements is not scale sensitive. (However, the Gower distance is sensitive to including multiple albeit slightly varying measurements of the same concept.) This distance is defined in two steps. In the first step, a partial distance between the two cases is calculated separately on each variable in a scale-free manner. In the second step, these partial distances are then averaged across the full set of variables, leading to a distance that is always bounded by 0 and 1. The partial difference between two interval-valued variables is the absolute difference in the values of the variable between the two cases, divided by the range of the variable across the entire dataset. The partial difference between two categorical variables is 1 if they disagree and 0 if they agree. If all the variables are interval-valued, then the Gower distance is equivalent to the range-normalized L1 distance. If all the variables are categorical, then the Gower distance is merely the proportion of the variables that disagree between the two cases.

3.3 AttritWeight Macro

AttritWeight creates a nonresponse-adjusted weight for each wave after wave 1, which already has a base weight.[19] The weight for wave *w* is appropriate for analyzing either wave *w* by itself or analyzing cross-wave patterns from wave 1 up through wave *w*. The weight for the final wave is appropriate for any longitudinal analysis involving all waves. The macro first computes weights for wave 2, then for waves 3 through 5 in order. AttritWeight is activated after whole wave imputation if the *weightvar* option is used.

Similar to WaveImpute, this macro has four major steps, but, unlike WaveImpute, these steps are wave-specific:

1. Modeling of response propensity and repeating variables for the wave
2. Clustering of children based on their predicted response propensity and wave-specific outcomes
3. Construction of the adjustment factor for each cluster
4. Diagnostic reports and data output.

Step 1. Modeling of Repeating Variables

Unlike the modeling in WaveImpute, the models in AttritWeight are all linear. Moreover, there are no separate runs to thin out the predictor set. Instead, predictions are made directly from cross-validated linear lasso models for response status (respondent vs nonrespondent) and for each repeating variable. The predictor variables in each model include age, the stable variables, and the values of the repeating variables from the prior wave.

Step 2. Clustering Children by their Predicted Trajectories

As mentioned in Chapter 1, if a variable from a prior wave is associated with both attrition and substantive variables of interest for the current wave, then a failure to adjust the weights for the influence of this prior variable will result in nonresponse bias. If either the response propensity model or the outcome model is true, then there will be no nonresponse bias. Methods that use both types of models in adjusting for nonresponse are known in the survey literature as "doubly robust." AttritWeight strives to deliver this property.

AttritWeight uses the predicted values from step 1 to cluster the sample. This is done separately for each wave after wave 1. The size constraint is much larger for this clustering than the clustering in WaveImpute. For AttritWeight, the minimum permissible sample size is 25 respondents. This is because the adjustment factor is the inverse response rate for the cluster, and inverse ratios based on small sample sizes are unstable. In addition to the minimum size constraint, the algorithm requires an unweighted wave response rate of at least 50 percent. This constraint prevents adjustment factors larger than two, a common rule-of-thumb in nonresponse adjustments. (Adjustment factors larger than two will definitely increase variances, while their potential to reduce nonresponse bias is uncertain.)

The dual constraints on minimal sample sizes and response rates are achieved by first forming a large number of clusters and then collapsing each deficient cluster with its nearest neighboring cluster. The initial cluster solution is obtained with a k-means algorithm, as in WaveImpute, but the distance measure is Euclidean distance rather than Gower's distance. Because Eulicidean distance is sensitive to the scale of each variable, all the predictions except the predicted response propensity are standardized to have a mean of zero and a variance of one. The predicted response propensities are scaled to have a variance equal to the number of repeating variables. Further testing might indicate a different scale factor as optimal. Our idea was that if there is a large number of repeating variables, then

[19] PanelPatch also allows for replicate weights. Each replicate weight will be adjusted for nonresponse. If replicate weights are used, some other statistical software system will need to be used to analyze the data after PanelPatch.

the clusters might not differ much at all in their adjustment factors and therefore have little ability to remove nonresponse bias.

Step 3. Matching Nonresponding Children to Responding Children

The nonresponse-adjusted weights for a given wave are calculated by multiplying the weight for the prior wave with an adjustment factor. The adjustment factor for a particular wave is equal to the ratio of the sample weighted respondents for that wave over the sum of the sample weighted respondents and nonrespondents. For wave 2, the base sample weights are used in calculating the sample weighted sums. Subsequent waves use the last wave's nonresponse-adjusted sample weights to calculate the sample weight sums of wave respondents and nonrespondents. For example, the adjustment factor for wave 3 uses wave 2's nonresponse-adjusted sampling weights.

Step 4. Diagnostic Reports and Data Output

The output of AttritWeight includes:

- The "design effect" induced by the differential weighting;
- For each wave, the weighted mean values of binary and interval-valued repeating variables variable (with the associated cluster-corrected standard errors) before and after weight adjustment; and
- For each wave, the weighted frequencies of categorical repeating variables (with the associated cluster-corrected standard errors) before and after weight adjustment

3.4 Weighting versus Imputing

Deciding to whether use weights versus sole-imputation depends on several factors. Exhibit 3-1 serves as a guideline for determining whether to use "AttritWeight" in addition to "WaveImpute" in the context of a five-wave longitudinal study for illustrative purposes. (Note, however, that PanelPatch can handle more than five waves and that it automatically detects the number of waves.) The exhibit presents patterns that indicate wave response with a "1" and wave nonresponse with "0" for each respective wave based on the position of the value. Specifically, the first value pertains to Wave 1, the second value corresponds to Wave 2, and so on.

Among children with missing waves, some of the nonresponse patterns are "nested," while others are not. A pattern is nested if no missing waves occur between nonmissing waves in the pattern. For example, the pattern 11000 (attritor after two waves) is nested while the pattern 11010 is *not* nested. From our experience, weights are most often used to compensate for nested patterns and imputation is most often used for nonnested patterns. However, for late attritors, it might make more sense to use imputation since there is so much information that can be preserved over weights. Also, there are some nonnested patterns with such sparse information that weights might be the more sensible tool. The user may decide which approach is most appropriate for each pattern setting the required minimum number of responding waves in order for imputation to occur using the *minwave* option. Additionally, the user may induce weighting by changing the wave status from respondent to nonrespondent.

Exhibit 3-1 Recommendations for Weighting versus Whole Wave Imputation

Pattern	Recommended Approach	Notes
10000	Weight	
10100	Weight	Discards wave 3 data, but hard to do a good imputation job with just two waves
10010	Weight	Discards wave 4 data, but hard to do a good imputation job with just two waves

Pattern	Recommended Approach	Notes
10001	Weight	Discards wave 5 data, but hard to do a good imputation job with just two waves
11000	Weight	
11100	Impute waves 4 and 5	Weighting is also reasonable
11110	Impute wave 5	
10110	Impute waves 2 and 5	
10101	Impute waves 2 and 4	
10111	Impute wave 2	
10011	Impute waves 2 and 3	
11111	Perfect as is	
11010	Impute waves 3 and 5	Weighting is also reasonable, but would require discarding wave 4
11011	Impute wave 3	
11101	Impute wave 4	
11001	Impute wave 3 and 4	

4. Instructions & Stata Example

This chapter serves as a comprehensive guide to the PanelPatch system, providing step-by-step instructions by walking through a basic example within Stata. The example showcases the usage of the system and covers various aspects such as handling item nonresponse, imputing wave nonresponse, and calculating nonresponse-adjusted sample weights.

4.1 Example Data Set

Prior to using PanelPatch, the user should ensure the data is structured at the child-wave level (each child has one row per wave) and that variables which provide child identification, wave identification, provider identification, and base sampling weight variables are non-missing. It is also suggested that the user fill in any missing stable variables for the first wave, fill in any missing ages, and restructure any required one-off variables. Refer to Chapter 2 for more details on the proper data structure for PanelPatch.

Next, the user should identify and categorize the target ensemble variables. These should include any analytic variables and any variables to be used in identifying donors (refer to Chapter 2.2 for more details).

The example data set (a file between 5 and 6 mb) referred to throughout this chapter has already completed these steps. The example stata data set (between 5 and 6 mb) is available in the code package.

Exhibit 4-1 lists the required variables and target ensemble variables, the PanelPatch variable category, and a description or possible values found within the example data set.

Exhibit 4-1 Example Required and Target Ensemble Variable Descriptions

Example Variable	Category	Description/Values
ChildID	Required	• Identifies a child across waves
WaveID	Required	• Links child's responses to specific wave
ProviderID	Required	• Identifies child's provider
BaseWeight	Required	• Unadjusted sampling weight
WaveResponse	Required	0. Nonresponse 1. Response 2. Not Applicable
SexAtBirth	Stable: binary	0. Male 1. Female
Discipline	Stable: ordered categorical	• Number of Disciplinary Infractions
ProviderType	Stable: unordered categorical	1. Public school 2. Head Start 3. Another licensed provider 4. Provider not listed
BaselineHouseholdIncome	Stable: interval-valued	• Interval-valued
SNAPEligible	Repeated: binary	0. No 1. Yes
MotherEducation	Repeated: ordered categorical	1. Did not complete high school 2. High school degree or equivalent 3. Bachelor's degree 4. Master's degree or higher

Example Variable	Category	Description/Values
ProviderDelivery	Repeated: unordered categorical	1. In-person 2. Remote 3. Hybrid
AssessmentScore	Repeated: interval-valued	• Interval-valued
Age	Age (repeated: interval-valued)	• Interval-valued

4.2 ItemImpute Example

Exhibit 4-2 illustrates several item nonresponse cases that may be found within a data set. Each child-wave record is indicated as a wave responder (WaveResponse = 1), but there are missing values for the stable binary variable "SexAtBirth," the repeating unordered categorical variable "ProviderDelivery," and the repeating interval-valued variable "AssessmentScore." Using the method described in Chapter 3.1, the ItemImpute macro imputes values for these missing cases. It is important to note that the user does not need to specify ItemImpute to run. The PanelPatch system will automatically identify these cases and execute the ItemImpute procedure.

Exhibit 4-2 Item Nonresponse Example

ChildID	WaveID	WaveResponse	SexAtBirth	ProviderDelivery	AssessmentScore
A	1	1	1	Remote	94
A	2	1			88
A	3	1	1	Hybrid	

Using the provided example data set in Chapter 4.1, we can identify cases of item nonresponse cases by conditioning on instances of wave response. Since stables variables are defined to be constant across waves, we only check the item nonresponse cases in wave 1. SexAtBirth is missing for 125 children within the wave 1 among wave responders.

```
missings report SexAtBirth Discipline ProgramType BaselineHouseholdIncome if
WaveResponse == 1 & WaveID == 1, percent

----------------------------------------
                         |     #       %
-------------------------+--------------
              SexAtBirth |   125    6.07
              Discipline |   103    5.00
             ProgramType |   130    6.31
 BaselineHouseholdIncome |   114    5.53
----------------------------------------
```

Similarly, we can check the item nonresponse cases for repeating variables. SNAPEligible is missing for 592 child-wave observations among wave responders. Notably, Age will not have any missing values because they have been manually filled in.

```
missings report SexAtBirth Discipline ProgramType BaselineHouseholdIncome if
WaveResponse == 1 & WaveID == 1, percent

----------------------------------------
```

```
                          |      #       %
--------------------------+----------------
              SexAtBirth  |    125    6.07
              Discipline  |    103    5.00
             ProgramType  |    130    6.31
 BaselineHouseholdIncome  |    114    5.53
-----------------------------------------
```

We will impute these item nonresponses using the -PanelPatch- command followed by all of the target ensemble variables. PanelPatch also requires the user to specify which variables are associated with child identification (*i*), program identification (*j*), wave identification (*wave*), and wave response (*waveresponseflag*). The user must also properly specify the variable category type for each target ensemble variable. Stable binary and interval-valued variables are listed within the *snumericvars* option; stable unordered categorical variables are listed in the *sunorderedvars* option; and stable ordered categorical variables are listed in the *sorderedvars* option. Repeating binary, repeating interval-valued, and age are listed within the *vnumbericvars* option; repeating unordered categorical variables are listed within the *vunorderedvars* option; and repeating ordered categorical variables are listed within the *vorderedvars* option.[20] The final required option for PanelPatch is the *add* option, which identifies the number of imputed datasets the user wishes to create. This example will create 5 imputed datasets, which when combined with the nonresponse-adjusted sampling weights created by PanelPatch, can be utilized by the Stata user for analysis. Optionally, users can specify the base sampling weight (*weightvar*) to trigger the weight-adjustment macro.

```
PanelPatch SexAtBirth Discipline ProgramType BaselineHouseholdIncome
SNAPEligible MotherEducation ProgramDelivery AssessmentScore Age, i(ChildID)
j(ProgramID) wave(WaveID) waveresponseflag(WaveResponse)
weightvar(BaseWeight)  snumericvars(SexAtBirth BaselineHouseholdIncome)
sunorderedvars(ProgramType) sorderedvars(Discipline)
vnumericvars(SNAPEligible AssessmentScore Age)
vunorderedvars(ProgramDelivery) vorderedvars(MotherEducation) add(5)
```

Once PanelPatch has completed, the user should view the diagnostic output to ensure the imputed data makes sense. The diagnostic output can be found as a text document saved in the set working directory under the name PanelPatch_DiagnosticTables.txt.

4.3 WaveImpute Example

Wave nonresponse can be identified when a child-wave record has missing values for all repeating. If a wave response variable does not exist within the data set, the user should create this variable. The example data set has 3,904 child-wave records with wave nonresponse.

```
tabulate WaveResponse

WaveResponse |      Freq.     Percent        Cum.
-------------+-----------------------------------
           0 |      3,904       37.90       37.90
           1 |      6,396       62.10      100.00
```

[20] Note that all variables listed before the options must be accounted for in the set of variables included in the following options (if used): *snumericvars*, *sunorderedvars, sorderedvars, vnumbericvars, vunorderedvars* and *vorderedvars.* The program will check for this and report an error as appropriate.

```
-------------+-----------------------------------
       Total |     10,300      100.00
```

PanelPatch outputs several diagnostic tables as a text document within the current working directory. The relevant diagnostic output for wave nonresponse may be found in Step 4 in Chapter 3.2. In our example, PanelPatch salvaged 799 child-wave records through imputation.

```
Table 1. Study Members with Imputed Waves due to Wave Nonresponse
     WaveID |      Freq.     Percent        Cum.
------------+-----------------------------------
          2 |         87       10.89       10.89
          3 |         83       10.39       21.28
          4 |        244       30.54       51.81
          5 |        385       48.19      100.00
------------+-----------------------------------
      Total |        799      100.00
```

We also show the changes in weighted means (with their respective cluster-corrected standard errors) for binary and interval-valued repeating variables, and the changes to the weighted frequencies for each categorical variable. This table includes any imputations made by ItemImpute and WaveImpute and uses the sampling weights generated by AttritWeight. Refer to Section 3.2 for the full list of diagnostic tables available for WaveImpute.

```
Table 2. Weighted Mean Values for Binary and Interval-Valued Repeating Variables Pre- and Post-
Imputation
------------------------------------------------------------------------------
                     |    Wave 1    Wave 2    Wave 3    Wave 4    Wave 5
---------------------+--------------------------------------------------------
SNAPEligible(pre)    |
  Weighted Mean      |   .4797492   .4618855   .4427991   .4093863   .4089593
  Clustered SE       |   .0115957   .0138244   .0153632   .0167722   .0180735
SNAPEligible(post)   |
  Weighted Mean      |   .4742103   .4560015   .4419622   .4024178   .4051662
  Clustered SE       |   .0122784   .0144276   .0179383   .0158436   .0173831
AssessmentScore(pre) |
  Weighted Mean      |   .0025766  -.1635991  -.2653479  -.4452715  -.5071647
  Clustered SE       |   .0332143     .04446   .0514824     .05811   .0742696
AssessmentScore(post)|
  Weighted Mean      |  -.0056851  -.2049475   -.303469  -.4830232  -.5859706
  Clustered SE       |   .0434143   .0537196   .0604014     .05581   .0688125
```

4.4 AttritWeight Example

Based on the wave nonresponse pattern and the user-defined minimum number of wave nonresponses required for imputation, the PanelPatch system creates nonresponse-adjusted sampling weights for either unimputed data following nested nonresponse pattern or imputed data. The diagnostic tables for AttritWeight can be found in the same diagnostic output file as for WaveImpute. Table 10 is similar to diagnostic Table 2 used for WaveImpute, but the pre-imputation values use the original base weight variable for sampling weights while the post-imputation values use the sampling weights generated in AttritWeight. Refer to Section 3.3 for the full list of diagnostic tables available for WaveImpute.

```
Table 10. Weighted Means for Repeating Numeric Variables Pre- and Post-Weight Adjustment

------------------------------------------------------------------------------

                     |    Wave 1    Wave 2   Wave 3    Wave 4    Wave 5

---------------------+--------------------------------------------------------
```

```
SNAPEligible(pre)     |
  Weighted Mean       |  .4742103  .4560015 .4419622  .4024178  .4051662
  Clustered SE        |  .0122784  .0144276 .0179383  .0158436  .0173831
SNAPEligible(post)    |
  Weighted Mean       |  .4742103  .4560015 .4419622  .4024178  .4051662
  Clustered SE        |  .0122784  .0144276 .0179383  .0158436  .0173831
AssessmentScore(pre)  |
  Weighted Mean       | -.0056851 -.2049475 -.303469 -.4830232 -.5859706
  Clustered SE        |  .0434143  .0537196 .0604014    .05581  .0688125
AssessmentScore(post) |
  Weighted Mean       | -.0056851 -.2049475 -.303469 -.4830232 -.5859706
  Clustered SE        |  .0434143  .0537196 .0604014    .05581  .0688125
----------------------------------------------------------------------------
```

4.5 Post-PanelPatch Optional Operations

One-off Variable Imputations

After running PanelPatch, the user may utilize Stata's -mi impute- command procedures to further impute any one-off variables. To ensure valid nonresponse adjusted weighting (created via PanelPatch), the user should restrict any one-off variable imputations to the imputed and complete observations. These observations are identified by the "PanelPatchImputed" variable created after running the PanelPatch system. Below, we show an example of imputing a one-off variable from Wave 3 ("oneOFFwave3").

```
mi register imputed oneOFFwave3

mi impute reg oneOFFwave3 = binDYNAMIC contDYNAMIC if PanelPatchImputed==1 &
WaveID == 3, replace
```

Setting Stata for Complex Survey Analysis

Variance estimation with the nonresponse adjusted weights may be conducted by setting the survey design within Stata prior to analysis.

```
mi svyset ProviderID [pweight = BaseWeight_wgtadj], strata(strataID)
vce(linearized)singleunit(scaled)
```

Note to ELS@H Team: For child-level analysis, Abt Associated provided "Child_Stratumid" and "Child_Cluster" as the strata and unit identification variables, respectively. As a reminder, the child-level sample weight is named "CHILD_WEIGHT_FINAL" and replicate weights beginning with the stem "Child_Weight_Replicate". Prior to analysis, if replicate weights were chosen to be adjusted for nonresponse, the user should export their data to a statistical software system that supports the simultaneous use of replicate weights such as SUDAAN or SURVEY. If SURVEY is chosen, the data can be export to R via `mi export ice`.

5. Simulation Study

Our aim was to create a challenging test bed for validating PanelPatch. Before describing the challenge level we built into the simulation study, we need to review some standard terminology. In the survey literature, nonresponse is classified as **missing completely at random** (MCAR), **missing at random** (MAR), or **not missing at random** (NMAR). Nonresponse is MCAR if it is independent of all the outcomes measured in the survey. This is the best type of nonresponse to have since observations with missing data can simply be dropped from any analysis of interest without incurring any nonresponse bias. Nonresponse is MAR if it is conditionally independent of all the outcomes measured in the survey given the available information about the survey sample members. This type of nonresponse requires more work, but with appropriate analysis techniques unbiased estimates can still be wrung from the data. Nonresponse is NMAR if it is dependent on the outcomes of interest even after conditioning on all available data. This is the worst type of nonresponse to have. Techniques have been developed to apply to NMAR data, but they require very strong untestable assumptions and specialized software. Unfortunately, there are no empirical tests to distinguish MAR from NMAR nonresponse. Most statisticians assume MAR nonresponse, develop appropriate adjustments (such as weighting, imputation, or full likelihood estimation) for that assumption, and hope for the best.

Given that we designed PanelPatch to compensate for MAR nonresponse, this is the challenge level that we built into the simulation study. While we could easily have simulated NMAR nonresponse, we don't think the results would have been informative. We already know that PanelPatch will fail to adjust adequately for NMAR nonresponse, and we felt that the degree of failure would not be easily generalizable. Instead, we decided to simulate nonresponse of the type that the PanelPatch was designed to address. We still created a test bed with a high challenge level via complex interrelationships of a large collection of variables. Performing well on this test bed could not be taken for granted.

5.1 Structure

The test dataset is meant to simulate a sample of 200 preschools with an average of 10 children per preschool. Child sample sizes vary across preschools according to a gamma distribution with a standard deviation across schools of children. There are a total of five waves of simulated interviews. The attrition rate for each wave is an average of 2 percent but varies normally on the logit scale across preschools. Also, at each wave, the wave nonresponse rate is an average of 6 percent but also varies normally on the logit scale across schools. The resulting wave response patterns are shown in Exhibit 5-1. For individual items, the item nonresponse rate is 5 percent. The attrition and wave nonresponse are MAR while the item nonresponse is MCAR. We decided not to simulate any NMAR data since any favorable properties of the procedures would be purely coincidental.

Exhibit 5-1 Patterns of Attrition and Wave Skipping

Wave Pattern	Count	Percent	Treatment
10000	92	4.47	Weight
10001	5	0.24	Weight
10010	7	0.34	Weight
10011	23	1.12	Wave Impute
10100	9	0.44	Weight
10101	18	0.87	Wave Impute
10110	28	1.36	Wave Impute
10111	93	4.51	Wave Impute
11000	84	4.08	Weight
11001	14	0.68	Wave Impute
11010	34	1.65	Wave Impute
11011	99	4.81	Wave Impute
11100	106	5.15	Wave Impute
11101	109	5.29	Wave Impute
11110	180	8.74	Wave Impute
11111	1159	56.26	N/A

The test dataset has 30 stable variables (variables that are fixed at baseline and never change such as race and gender) and 30 repeating variables (variables that repeat at each wave and are expected to change over time). Each set of 30 variables is comprised of 5 variables of 6 different measurement types: 5 binary variables, 5 ordered categorical variables, 5 unordered categorical variables, five normal variables, five log-normal variables, and five truncated normal variables. All 60 variables have positive intraclass correlation of 0.15 (meaning that variation is within schools is 85 percent of total variation). Local-level means of all 60 variables are assumed to be normally distributed across preschools.

Since we are starting from the wave 1 respondent sample this means that all nonresponse to the static variables is item nonresponse, which as mentioned above is MCAR. However, the dynamic variables are child to MAR nonresponse.

The example assumes that there are five independent latent traits of interest and that the six measurement types are all merely different ways of trying to measure those latent traits.

Within a preschool, the latent traits are assumed to be normally distributed across children at wave 1 with mean 0 and variance 1. Let θ_{kji} be the value of the *k*-th latent trait (*k*=1 to 5) for the *i*-th child at the *j*-th preschool. Let Y^f_{kjiw} be the measurement of this latent trait with functional form *f* (*f*=1 to 6) at wave *w* (*w*=0 to 5, where *w*=0 corresponds to the static variables). Let $B^f_{ji} \sim N(0,1)$ be random biases associated with measurement type *f*, also with an intraclass correlation of 0.15.

To instill intraclass correlation, the latent traits and random biases were generated as:

$$\theta_{kji} \sim N(0,1) + \sqrt{\frac{0.15}{1.15}}\mu_{kj},$$

$$B^f_{ji} \sim N(0,1) + \sqrt{\frac{0.15}{1.15}}\beta^f_j,$$

where

$\mu_{kj} \sim N(0,1)$ and $\beta^f_j \sim N(0,1)$.

The binary measurements were then generated as:

$$Y^1_{kjiw} = B\left(1, \frac{\log\left(\left(\theta_{kji} + B^1_{ji} + (0.4)Y^4_{kji,w-1}\right)/3\right)}{1+\log\left(\left(\theta_{kji} + B^1_{ji} + (0.4)Y^4_{kji,w-1}\right)/3\right)}(1.1)^{w-1}\right) \text{ for } w = 1,\ldots,5 .$$

The ordered categorical measurements were generated as:

$$Y^2_{kjiw} = \begin{cases} 1 \text{ if } Z^2_{kjiw} < .5 \\ 2 \text{ if } .5 \le Z^2_{kjiw} < 1.66 \\ 3 \text{ if } 1.66 \le Z^2_{kjiw} < 1.96 \\ 4 \text{ if } 1.96 \le Z^2_{kjiw}, \end{cases}$$

where

$$Z^2_{kjiw} \sim N\left(\frac{\log\left(\left(\theta_{kji} + B^2_{ji} + (0.4)Y^4_{kji,w-1}\right)/3\right)}{1+\log\left(\left(\theta_{kji} + B^2_{ji} + (0.4)Y^4_{kji,w-1}\right)/3\right)}(1.1)^{w-1} - 0.5, 1\right) \text{ for } w = 1,..,5$$

The unordered categorical measurements were generated as:

$$Y^3_{kjiw} = \begin{cases} 1 \text{ if } -0.6 < Z^3_{kjiw} < 0 \\ 2 \text{ if } Z^3_{kjiw} > 1.645 \\ 3 \text{ if } Z^3_{kjiw} \le -0.6 \\ 4 \text{ if } 0 \le Z^3_{kjiw} \le 1.645, \end{cases}$$

where

$$Z^3_{kjiw} \sim \mathrm{N}\left(\frac{\log\left(\left(\theta_{kji} + B^3_{ji} + (0.4)Y^4_{kji,w-1}\right)/3\right)}{1+\log\left(\left(\theta_{kji} + B^3_{ji} + (0.4)Y^4_{kji,w-1}\right)/3\right)} (1.1)^{w-1} - 0.5, 1 \right) \text{ for } w = 2,...5$$

The normal measurements were generated as:

$$Y^4_{kjiw} \sim \mathrm{N}\left(\frac{\theta_{jiw} + B^4_{ji} + (0.4)Y^4_{kji,w-1}}{3} (1.1)^{w-1}, 1 \right) \text{ for } w = 2,...,5 .$$

The log-normal measurements were generated as:

$$Y^5_{kjiw} \sim \mathrm{N}\left(\exp\left[\frac{\theta_{kji} + B^5_{ji} + (0.4)Y^4_{kji,w-1}}{3} (1.1)^{w-1} \right], 1 \right) \text{ for } w = 2,...,5 .$$

The truncated normal measurements were generated as:

$$Y^6_{kjiw} \sim \max\left\{ 0, \mathrm{N}\left(\frac{\theta_{kji} + B^6_{ji} + (0.4)Y^4_{kji,w-1}}{3} (1.1)^{w}, 1 \right) \right\} \text{ for } w = 1,..,5 .$$

Note that Y^f_{kjiw} is correlated with $Y^f_{k'jiw}$ for $k \neq k'$ by virtue of the shared influence of B^f_{ji}. Furthermore Y^f_{kjiw} is correlated with $Y^{f'}_{kjiw}$ for $f \neq f'$ by virtue of the shared influence of θ_{kji}. Finally, Y^f_{kjiw} is correlated with $Y^f_{kjiw'}$ for $w \neq w'$ by virtue of the influence of $Y^f_{kji,w-1}$ on Y^f_{kjiw} and the aforementioned correlations across functional forms. This means that all the variables are correlated with each other, both within and across preschools. We created this complex correlation structure to provide a challenging test bed for validation of PanelPatch.

As mentioned above, attrition was made MAR rather than MCAR. We made this choice because it means that a well-done analysis that exploits the relevant observed data can be unbiased, but simple analyses that ignore those data will be biased.

The probability of attrition at wave *w* for *w*>1 was:

$$\lambda_{jiw} = \frac{\xi_j + \pi_{ji}}{1 + \xi_j + \pi_{ji}}$$

where

$$\xi_j \sim N\left(\frac{.02}{1.02}, 0.2\right)$$

and π_{ji} is calculated by summing the row products in this table:

Attribute	Coefficient
Y^4_{1ji0}	0.5
Y^4_{2ji0}	-0.3
Y^4_{3ji0}	0.1
Y^4_{4ji0}	0.05
Y^4_{5ji0}	-0.03
$Y^4_{1ji,w-1}$	1.5
$Y^4_{2ji,w-1}$	1.5
$Y^4_{3ji,w-1}$	1.5
$Y^4_{4ji,w-1}$	-2.0
$Y^4_{5ji,w-1}$	-2.8

Note that because $Y^4_{kji,w-1}$ figures prominently in both Y^4_{kjiw} and λ_{jiw}, nonresponse is highly non-ignorable but still missing at random. Therefore, it is at least theoretically possible to remove any bias caused by this nonresponse through appropriate conditioning. (In this case, it is critical to condition on $Y^4_{kji,w-1}$.)

Similarly, the probability of skipping wave *w* for *w*>1 was:

$$\pi_{jiw} = \frac{\zeta_j - (0.2)Y^4_{1ji0} - (0.03)Y^4_{2ji0} - (0.1)Y^4_{3ji0} + (0.15)Y^4_{4ji0} - (0.07)Y^4_{5ji0} + (0.8)Y^4_{5ji,w-1}}{1 + \zeta_j - (0.2)Y^4_{1ji0} - (0.03)Y^4_{2ji0} - (0.1)Y^4_{3ji0} + (0.15)Y^4_{4ji0} - (0.07)Y^4_{5ji0} + (0.8)Y^4_{5ji,w-1}}$$

where

$$\zeta_j \sim N\left(\frac{0.06}{1.06}, 0.25\right).$$

We constructed the baseline weights to be informative:

$$\omega_{ji} \sim N\left(4000 + Y^5_{5ji0}, 10\right).$$

Since *f*=5 corresponds to lognormal outcomes, these weights vary substantially, and since all the variables are connected to each other, the weights are highly informative.

5.2 Results

As a challenging and relevant validation of PanelPatch, we focused on growth curve modeling for each measurement function. However, mindful that the results might also be used for cross-sectional analyses, we also looked at the quality of marginal means for repeating variables at wave 5.

Growth Curve Models

For the binary outcomes, we fit growth curve models of the form:

$$\frac{\Pr\left(Y^1_{kjiw}=1\right)}{1+\Pr\left(Y^1_{kjiw}=1\right)} = \eta^1_k + \sum_{k,f}\alpha^{1f}_k Y^f_{kji0} + \gamma^1_k w \text{ for } k=1 \text{ to } 5,$$

where the categorical variables were treated as class variables, so there was a separate coefficient for each level. This resulted in a total of 52 coefficients.

For the ordered categorical variables, we fit growth curve models of the form:

$$\frac{\Pr\left(Y^2_{kjiw}\le a\right)}{1+\Pr\left(Y^2_{kjiw}\le a\right)} = \eta^{2a}_k + \sum_{k,f}\alpha^{2f}_k Y^f_{kji0} + \gamma^2_k w \text{ for } a=1 \text{ to } 3 \text{ and } k=1 \text{ to } 5.$$

These models have 54 coefficients for every level of *k*.

For the unordered categorical variables, we fit growth curve models of the form:

$$\frac{\Pr\left(Y^3_{kjiw}=a\right)}{1+\Pr\left(Y^2_{kjiw}=4\right)} = \eta^{3a}_k + \sum_{k,f}\alpha^{3af}_k Y^f_{kji0} + \gamma^{3a}_k w \text{ for } a=1 \text{ to } 3 \text{ and } k=1 \text{ to } 5.$$

These models have 156 coefficients for every level of *k*.

For the normal variables, we fit growth curve models of the form:

$$EY^g_{kjiw} = \eta^g_k + \sum_{k,f}\alpha^{gf}_k Y^f_{kji0} + \gamma^{4g}_k w \text{ for } g=4 \text{ to } 6 \text{ and } k=1 \text{ to } 5.$$

These models each have 52 coefficients for every level of *k*.

We fit all these models on four versions of the dataset:

1. **Uncensored with baseweights;** A version with no nonresponse of any type;
2. **Censored with baseweights:** A version with no imputation or weight adjustment;
3. **Imputed with baseweights:** A version with both item and wave imputation but no weight adjustment; and
4. **Imputed with adjusted weights:** A version with item imputation, wave imputation, and weight adjustment.

In all cases, we used survey-sensitive versions of software to fit these generalized linear models. With no imputation, the modeling software automatically drops all records that are missing the outcome or any of the covariates. This is referred to as a complete-case analysis, but that is a bit of a misnomer here because all the covariates are from the baseline. We did use the weights adjusted for probability of selection and baseline nonresponse. These models treat the preschools as clusters and utilize the weights. We took the model coefficients obtained on the uncensored file to be the "truth" and compared the other two versions to this truth.[21]

Across the 30 models, there are a total of 2,090 coefficients fit from each version of the dataset. We measured model quality in five ways:

1. The first measure of model closeness is the Pearson correlation between the estimated coefficients on the censored dataset and those on the true dataset. The sample size for this correlation is 2,090.
2. The second measure of model closeness is the count of associated hypothesis tests that changed from statistically significant to not statistically significant or vice versa.
3. The third measure of model closeness is the mean absolute standardized error, where the standard error of the coefficient on the uncensored dataset is used to standardize the errors. (Since the coefficients are on very different scales given the inclusion of log normal variables, the mean absolute error would be unduly sensitive to performance on those models.)
4. The fourth measure of model closeness is the root mean squared standardized error, where the standardization of errors is the same as in the prior measure.
5. The fifth measure of quality is the standard error on the estimated parameter.

Exhibit 5-2 displays the results. Interpreting first the "censored" column, we see that the correlation is weak at 0.40, meaning that most of the coefficients are poorly estimated. This is reinforced by the counts of lost and false findings, (189 and 186, respectively). The median absolute standardized error is 1.72, meaning that errors are generally twice times as large as the standard errors of the coefficient on the uncensored dataset. Sometimes they are much worse. One absolute standardized error was 46 times as large as the standard error on the parameter. The root median squared standardized error is much smaller than the median absolute standardized error at 0.10, indicating that many parameters are well estimated, but some are very poorly estimated. Standard errors are much larger on the censored data than on the uncensored because a complete cases analysis discards so much data. In summary, on this dataset, a complete cases analysis frequently goes badly astray even when baseline weights are used.

After applying ItemImpute and WaveImpute (but not AttritWeight), we see dramatic improvements in most of these quality measures (third data column of Exhibit 5-2). The correlation of estimated parameters with their counterparts from uncensored data jumps from 0.40 to 0.90, the number of false findings drops from 186 to 36, the maximum errors decrease dramatically, and the standard errors fall back to levels close to what would be achieved with uncensored data. The number of lost findings, however, stays high, perhaps an unavoidable consequence of information loss.

In this example, applying AttritWeight after WaveImpute (fourth data column) did not yield much benefit. Most of the quality measures worsened slightly. The only improvement was a very slight decrease in the maximum absolute standardized error.

[21] We were not trying to create perfect models for the outcomes. We just wanted the censored analysis to give results similar to the uncensored analysis.

Exhibit 5-2 Quality Diagnostics for Growth Curve Analyses

Quality Measure	Uncensored with baseweights	Censored with baseweights (with no adjustments)	Imputed with baseweights	Imputed with adjusted weights
Correlation with uncensored		0.40	0.90	0.89
Lost significant results		189	154	160
False findings		186	36	48
Absolute standardized error size				
Median		1.72	0.68	0.73
Mean		2.76	0.92	0.97
Max		45.89	10.28	10.19
Squared standardized error				
Root Median		0.10	0.04	0.04
Root Mean		1.63	0.14	0.15
Root Max		37.13	1.47	1.89
Standard Error				
Median	0.06	0.16	0.08	0.09
Mean	0.10	0.22	.013	0.13
Max	0.86	6.06	0.99	1.06

Notes: Error sizes and mean-square errors in this table are standardized with respect to standard errors of the same parameters from growth curve models fit to uncensored data. Statistical significance was defined with alpha equal to 0.05. A "lost" significant result is a parameter that is statistically significant in a model fit to the uncensored data, but no longer significant when fit to the observed data. Conversely, a "false finding" is a parameter that is *not* statistically significant in a model fit to the uncensored data, but is significant when fit to the observed data. Standard errors were obtained with survey-sensitive regression software that corrects for the use of weights and clustered sampling.

Cross-sectional Analyses at Wave 5

Exhibit 5-3 shows parallel results for cross-sectional analyses at wave 5 using a reduced set of diagnostics. Here we simply estimated the mean of each binary and interval-valued variable and the mean of a dummy variable for each level of each categorical variable. For quality measures, we calculated absolute standardized error sizes and effective sample sizes. It is important to note, however, that the standardization is different for these analyses than for the growth curve analyses. For these cross-sectional analyses, we used the standard deviations of the variables rather than the standard errors of means. That is why they are an order of magnitude smaller. The effective sample size was calculated as the ration of the population variance to the variance on the estimated marginal mean.

In general, the simulated censoring did not cause much nonresponse bias in cross-sectional analyses, so it is perhaps not surprising that wave imputation and weighting did not lead to any appreciable decreases in biases for these analyses. The focus of PanelPatch on longitudinal rather than cross-sectional analyses may also be a factor. We do note, however, that the effective sample sizes are clearly better after PanelPatch than before.

Exhibit 5-3 Quality Diagnostics for Cross-sectional Analyses at Wave 5

Quality Measure	Uncensored with baseweights	Censored with baseweights (with no adjustments)	Imputed with baseweights	Imputed with adjusted weights
Absolute standardized error size				
Median		0.00	0.00	0.00
Mean		0.04	0.04	0.04
Max		0.23	0.27	0.27
Effective Sample size				
Median	1856	799	890	874
Mean	1838	1430	1572	1530
Min	911	267	385	399

Notes: Error sizes and mean-square errors in this table are standardized with respect to standard deviations (not standard errors) of repeating variables. Standard errors were obtained with survey-sensitive regression software that corrects for the use of weights and clustered sampling. The effective sample size is the square of the ratio of the standard deviation of the variable (one the uncensored data) to the standard error of the mean (on the observed sample).

References

Cox, Brenda G. and Gordon Scott Bonham (1983). Source and solutions for missing data in the NMCUES. *Proceedings of the Section on Survey Research Methods of the American Statistical Association*, pp. 444-449.

Gower (1971) A general coefficient of similarity and some of its properties. *Biometrics*, 27, 857–874

Judkins, D. R. (1998). Multivariate imputation in practice and a proposal. May Day for Missing Data, a conference on statistical analysis with missing data, sponsored by the Chicago Chapter of the American Statistical Association.

Judkins, D. R. (2000) Discussion of Session 44: New Developments in Imputation of Business Survey Data. *Proceedings of the Second International Conference on Establishment Surveys -Survey Methods for Businesses, Farms, and Institutions - Invited Papers.* pp629-631.

Little, R. J. A. (1986). Survey nonresponse adjustments. *International Statistical Review*, **54**, 139-157.

Wolter, K. M. (2007). *Introduction to variance estimation* (2nd ed.). Springer Science + Business Media.