Multiple Imputation of Zero-Inflated and Overdispersed Multilevel Count Data

Kristian Kleinke & Jost Reinecke

University of Bielefeld, Faculty of Sociology & Centre for Statistics

Author Note

Dr. Kristian Kleinke

University of Bielefeld

Faculty of Sociology & Centre for Statistics

Postfach 10 01 31

33501 Bielefeld, Germany

kmkleinke@gmail.com

Abstract

Throughout the last couple of years multiple imputation (MI) has become a popular and widely accepted method to address the missing data problem. However, currently existing multiple imputation software has limitations regarding incomplete count data, especially with regard to certain kinds of multilevel count data: We present a multiple imputation solution for ordinary and overdispersed zero-inflated clustered count data based on a two-level hurdle model using a Bayesian regression approach within a chained equations multiple imputation framework (Raghunathan, Lepkowski, van Hoewyk, & Solenberger, 2001; van Buuren, Brand, Groothuis-Oudshoorn, & Rubin, 2006; van Buuren & Groothuis-Oudshoorn, 2011).

Keywords: missing data, multiple imputation, count data, multilevel data, zero-inflation

Multiple Imputation of Zero-Inflated and Overdispersed Multilevel Count Data

1 Introduction

In his textbook, Paul Allison states that the best solution to the missing data problem is prevention (Allison, 2001). This is especially true for complex multilevel data sets, where missing data may occur at different levels and all kinds of variables may be unobserved (cf. van Buuren, 2011): Missing data may occur in the outcome variable, in level-1 predictors, level-2 predictors, or even higher levels and finally in the group identifier. We will see, that missing data solutions for complex multilevel data sets are still in rather early stages of development, and none of them can – at the moment – be deemed optimal.

Missingness at level 1, level 2, or the class variable is a field, where a lot of research still has to be done. Missingness in level-1 predictors is typically dealt with by excluding incomplete cases from the analysis. This is the default of most multilevel software solutions – a wasteful practice, which may lead to biased regression coefficients (van Buuren, 2011). Schafer and Yucel (2002) presented a multiple imputation solution for this problem by "moving" the incomplete variables to the other side of the equation and treating them as outcome variables in a multivariate linear mixed effects model. Their R package `pan` is available from `http://cran.r-project.org/package=pan`. Schafer's and Yucel's solution has also been implemented into the R package `mice` (function `mice.impute.2l.pan()`), which is available from `http://cran.r-project.org/package=mice`. The disadvantage of these functions is that they currently only impute under the normal model, assume homoscedasticity, and do not support data with other distributional assumptions like count data.

Missing data in level-2 predictors are usually handled by excluding the whole group from the analysis. This is again very wasteful and may lead to selection effects at level 2 and to biased estimates of group level effects (van Buuren, 2011). A practicable solution for this problem has been proposed by Gelman and Hill (2007) and Yucel (2008), who use two data sets, one containing level-1 information, one containing level-2 information, and who

impute those data sets separately and iteratively, switching between level 1 and level 2. In `mice`, this approach can be applied by using `mice.impute.2l.pan()` for incomplete level-1 variables and using `mice.impute.2lonly.norm()` or `mice.impute.2lonly.pmm()` for incomplete level-2 variables (for details, refer to the documentation of the respective `mice` functions). Again, the problem is, that this solution is not yet available for count variables.

Studies that address the problem of imputing a missing group identifier (for example when an employee forgets to fill in the department, in which he or she is working) are to our best knowledge not yet available. An incomplete class variable may lead to the exclusion of valuable information.

Only handling missing data in the outcome variable has been researched sufficiently well and can be done for example by using direct maximum likelihood techniques like full information maximum likelihood estimation (FIML) or restricted maximum likelihood estimation (REML) (van Buuren, 2011). However, as we will see, there are some limitations with regard to incomplete count data models. FIML approaches try to use all available information in the data to predict missing information (cf. Enders, 2010; Muthén & Muthén, 2012). REML is a closely related alternative to FIML that is less sensitive to small-sample bias (cf. van Buuren, 2011). The quality of parameter estimates of maximum likelihood approaches depends to a great extent upon the correct specification of the model. Variables that predict missingness should be included into the model. However, using so-called auxiliary variables in the sense of Collins, Schafer, and Kam (2001) is still problematic in currently available FIML software. Auxiliary variables are variables that are of no interest to the data analyst and that therefore would not be included into the analysis model, but that are highly correlated with the incomplete variables and their missing data indicators and thus help to improve imputation quality. M*plus* only supports very basic auxiliary variable models. Count models with auxiliary variables are currently not supported (Muthén & Muthén, 2012). Furthermore, M*plus* is a commercial package – closed source – and it is typically not possible for the end users to implement add-ons like

FIML count data models that allow for auxiliary variables themselves. Users have to wait until the M*plus* developers make such models available. Open source SEM packages like the `sem` package by John Fox (available from `http://cran.r-project.org/package=sem`) or `lavaan` (Rosseel, 2012) are in much earlier stages of development in comparison to M*plus* and would require a lot more work before they are ready to support auxiliary variable count models. `lavaan` at the moment does not even support any kind of count model.

Including auxiliary variables into a multiple imputation model on the other hand is very straightforward. Multiple imputation (MI) is a state-of-the-art technique that – like FIML – can use all available information in the data set to predict missing information (Schafer & Graham, 2002). However, most of the currently available MI approaches do not support multilevel models and imputed values do not reflect the clustered structure of the data. The available multilevel imputation solutions on the other hand support only basic two-level models and currently do not support complex count models.

This paper proposes and evaluates a multiple imputation solution for ordinary and overdispersed zero-inflated clustered count data using a Bayesian regression approach within a chained equations multiple imputation framework (Raghunathan et al., 2001; van Buuren et al., 2006; van Buuren & Groothuis-Oudshoorn, 2011). In sequential regressions MI, each incomplete variable is imputed separately and iteratively based on predictions derived from a plausible regression model. Here, we use a two-level hurdle model to address both zero-inflation and overdispersion. Our proposed solution is part of a comprehensive MI package for incomplete count data called `countimp` (Kleinke & Reinecke, 2013a), which may be obtained from `www.uni-bielefeld.de/soz/kds/software.html`. It also includes functions to impute incomplete two-level Poisson or two-level negative binomial data.

This paper is structured as follows: We first introduce the count data models that our proposed imputation solution uses. We then give a brief introduction to multiple imputation and explain our imputation approach in detail (Section 2). Section 3 describes the setup and the results of our Monte Carlo simulations. We evaluated the quality of the

proposed MI solution for incomplete zero-inflated two-level count data and compared its performance to currently available MI solutions that are used to impute two-level data of any type like predictive mean matching (cf. van Buuren, 2011). We end with a discussion of our findings, give advice for the practitioner, and outline fruitful avenues for future research (Section 4).

2 Theoretical Background

2.1 Count data models

The classical Poisson model

$$P(y) = \frac{\exp(-\mu)\mu^y}{y!}, \mu \in \mathbb{R}_{>0}, y = 0, 1, 2, \ldots, \tag{1}$$

which is often used to model count data, assumes that the variance $\text{VAR}(\mu)$ is equal to the mean μ. When data are overdispersed, meaning that the variance is larger in comparison to the mean, fitting a Negative Binomial (NB) model is usually the better alternative[1].

There are different ways to parametrize the NB model (Hilbe, 2011). In Fisher's notation (Fisher, 1941), the Negative Binomial model is written as

$$P(y) = \frac{(k+y-1)!}{y!(k-1)!} \frac{p^y}{(1+p)^{k+y}}, y = 0, 1, \ldots; p, k > 0, \tag{2}$$

where y is a non-negative integer number, and $P(y)$ the probability of observing the respective count y in the given sample. k and $p \in [0, 1]$ are shape and scale parameters in that distribution. The mean of (2) is $\mu = pk$, the variance is $\mu + \frac{\mu^2}{k}$.

A further problem apart from overdispersion that often arises in practise is zero-inflation, meaning that empirical count data exhibit more zero counts than would be predicted by either the Poisson or the NB model. One way to address an excess number of zeros is to fit a hurdle model (Mullahy, 1986). Hurdle models are mixture models and contain two model components: (a) a model for the probability of having a zero vs.

[1] A comprehensive list of factors that "cause" overdispersion and a discussion of various possible solutions is given by Hilbe (2011).

non-zero count, and (b) for the non-zero cases, a model that determines, what non-zero count the observational unit has. Typically, the zero model is a binomial generalized linear model (GLM), and the count model is either a zero-truncated Poisson model

$$P(y) = \frac{1 - \pi_0 e^{-\mu} \mu^y}{(1 - e^{-\mu}) y!}, y = 1, 2, \ldots, \tag{3}$$

or a zero-truncated NB model

$$P(y) = \frac{w^k}{1 - w^k} \frac{(k + y - 1)!}{(k - 1)! y!} \eta^y, y = 1, 2, \ldots, \tag{4}$$

with $w = \frac{1}{1+p}$ and $\eta = 1 - w$. Note, that (4) is derived from (2) by dividing (2) by $(1 - P(0))$, with $P(0) = \frac{1}{(1+p)^k}$ (Sampford, 1955).

Note also, that the zero and the count models can have different predictors, as the process that determines whether or not the observational unit has a non-zero count might be quite different from the process that determines what non-zero count the observational unit has.

2.2 Mixed effects modeling

A basic assumption of the above mentioned models is that observations are independent from one another. When data are clustered (e.g. when students are nested in classes) this assumption is usually violated: Units within the same cluster typically share certain properties, e.g. students going to schools in certain areas might differ quite noticeably from students from other areas. Statistical models need to represent these cluster-specific properties. Using mixed effects models is a feasible approach to address this. The term “mixed effects” means that the model consists of fixed variables, whose values are not supposed to change across clusters, and random variables, which may differ between groups (like for example delinquency rates or levels of job satisfaction).

Generalized linear mixed effects models (GLMM) can be written in the form

$$\begin{aligned} y_{ij} &= \beta_{0j} + \beta_{1j}x_{1ij} + \cdots + e_{ij} \\ \beta_{0j} &= \gamma_{00} + \gamma_{01}z_{1j} + \cdots + u_{0j} \\ \beta_{1j} &= \gamma_{10} + \gamma_{11}z_{1j} + \cdots + u_{1j} \\ &\dots, \end{aligned} \tag{5}$$

where y_{ij} represents the dependent variable, the observation of participant i in group j. β are the regression coefficients and e_{ij} denotes the individual level error term. x_{1ij} is an individual level predictor, z_{1j} a group level predictor. Unlike "standard" regression models, which treat regression coefficients β as constant across clusters or groups, generalized linear mixed effects models estimate "random" parameters separately for each group and provide an estimate of the variation of intercepts and slopes across groups. It is possible to include predictors for differences in intercepts and slopes across clusters (cf. Bryk & Raudenbush, 1992). γ are the group-level regression coefficients and u the corresponding residuals. (5) can be collapsed into a single equation. With one individual and one group level predictor, for example, (5) can be written as

$$\begin{aligned} y_{ij} &= \gamma_{00} + \gamma_{01}z_{1j} + u_{0j} + (\gamma_{10} + \gamma_{11}z_{1j} + u_{1j})x_{1ij} + e_{ij} \\ y_{ij} &= \gamma_{00} + \gamma_{01}z_{1j} + u_{0j} + \gamma_{10}x_{ij} + \gamma_{11}z_{1j}x_{1ij} + u_{1j}x_{1_ij} + e_{ij} \\ y_{ij} &= \underbrace{\gamma_{00} + \gamma_{01}z_{1j} + \gamma_{10}x_{ij} + \gamma_{11}z_{1j}x_{1ij}}_{\text{"fixed" part}} + \underbrace{u_{0j} + u_{1j}x_{1ij}}_{\text{"random" part}} + e_{ij}. \end{aligned} \tag{6}$$

With more predictors, (6) can become confusingly large and is more easily written in matrix notation:

$$y_j = X_j\beta + Z_ju_j + e_j, \tag{7}$$

with y_j being a $n_j \times 1$ vector, the dependent variable with n_j representing the number of observations in the j^{th} group. X_j is a design matrix with n_j rows and p colums (including a column of 1s referring to the intercept term). X_j contains the fixed effects predictors. Z_j is a design matrix with n_j rows and q colums (including a column of 1s referring to the

intercept term), which contains the random effects regressors. β is a $p \times 1$ vector containing all parameters of the "fixed" part of the model, u_j is a $q \times 1$ vector of random effects and e_j is a $n_j \times 1$ vector of individual level errors.

Our proposed missing data procedure for two-level zero-inflated and overdispersed count data imputes missing data on the basis of a two-level hurdle model. The zero model is a binomial generalized linear mixed effects model. The count model is a zero-truncated two-level NB model. Before we elaborate on that procedure, we give a brief introduction to multiple imputation in general and introduce the chained equations MI approach.

2.3 Missing data and multiple imputation

Rubin (1976, 1987) coined the terms *missing completely at random* (MCAR), *missing at random* (MAR) and *not missing at random* (NMAR) to refer to the randomness or non-randomness of missing data processes. MCAR means that missingness depends only on random factors and cannot be predicted by any other variable, whereas under MAR, missingness is allowed to depend on observed information in the data set.

NMAR processes are typically "feared" by practitioners, as they are very hard to handle and a lot of untestable assumptions have to be made, which involves a lot of "guessing". Under NMAR, missingness depends at least to some extent on unobserved information.

The current state-of-the-art procedures were designed to work under MCAR and MAR mechanisms. The NMAR problem is beyond the scope of this paper. A good and practical solution however is to do a NMAR sensitivity analysis to get a rough idea how and to what extent different NMAR processes can affect parameter estimates (van Buuren & Groothuis-Oudshoorn, 2011) and to discuss these effects thoroughly.

The reason why FIML and MI work under MAR is that they can use all available information in the data set to try and predict missing information. Simple imputation solutions like unconditional mean imputation cannot achieve this and produce biased

results under MAR (Little & Rubin, 1987; Schafer & Graham, 2002).

While some more sophisticated single imputation solutions like expectation maximization based approaches for example can also incorporate the information in Y_{obs} – the observed part of the data set Y – about Y_{mis} – the missing part – and consequently produce unbiased parameter estimates under MAR, they typically underestimate standard errors, unless some kind of correction procedure is applied (Graham, Cumsille, & Elek-Fisk, 2003; Schafer & Graham, 2002). The currently most feasible approach to fix this problem is to create multiple imputations following Rubin's theory (Rubin, 1987). By replacing each missing value m times with a different, but equally plausible value, the researcher ends up with m data sets that differ only in the missing part. These m data sets are then analyzed separately (e.g. in our case, the two-level hurdle model is fitted to each of these m data sets) and m statistical results are obtained. These are integrated into a single overall result using Rubin's rules for MI inference (Rubin, 1987). These rules produce a combined parameter estimate (the mean of the m parameter estimates), as well as a combined standard error, a t-ratio to test the null hypothis that the respective parameter is zero, as well as a (usually 95%) confidence interval. The advantage over using single imputation is that the combined standard error consists of a within imputation component and a between imputation part. This combination incorporates the additional estimation uncertainty due to missing data and typically increases standard errors and broadens confidence intervals in an adequate way to reflect this estimation uncertainty.

Thus, if done properly, using MI yields widely unbiased parameter estimates and measures of uncertainty in a wide range of scenarios. As violated assumptions of the imputation model (e.g. violations of the MAR assumption) affect only the imputed part of the data set, MI can be assumed to be quite robust to model misspecifications – depending on the percentage of missing data, of course. The risk of introducing significant bias due to model misspecifications and violated assumptions increases with an increasing amount of values to be imputed (cf. Schafer & Graham, 2002).

There are two established and widely used MI aproaches: joint modeling (Schafer, 1997a, 1997b) and conditional modeling, which is also known under the name `mice` – multiple imputation by chained equations or sequential regressions multiple imputation (Raghunathan et al., 2001; van Buuren et al., 2006; van Buuren & Groothuis-Oudshoorn, 2011). The former approach requires the specification of a joint model for all the variables in the data set, whereas the latter one tackles the missing data problem on a variable to variable basis and imputes each incomplete variable separately based on a certain statistical model – typically some kind of regression model with a specified set of predictors. The advantage of using conditional modeling over joint modeling is that the former approach is much more flexible when it comes to imputing data sets with variables of different types such as continuous, categorical, or semi-continuous, censored or truncated variables. Software that uses the conditional modeling MI framework is for example IVEware (imputation and variance estimation software) (Raghunathan, Solenberger, & Van Hoewyk, 2002) or `mice`, (van Buuren & Groothuis-Oudshoorn, 2011), which we use as basis for our proposed MI solution.

2.4 Multiple imputation of multilevel zero-inflated and overdispersed data

Our proposed multiple imputation procedure follows the conditional modeling approach and works as an add-on function to the popular and powerful R package `mice` (van Buuren & Groothuis-Oudshoorn, 2011). The m imputations are generated by the `mice()` function, which automatically calls our functions. This has to be set up via the `method` argument, as explained in the `mice` user's manual (van Buuren & Groothuis-Oudshoorn, 2011). The imputation model for each incomplete variable is specified via the `predictorMatrix` argument of the `mice()` function. For a general introduction to `mice`, see van Buuren and Groothuis-Oudshoorn (2011). For a detailed example regarding how to use the proposed imputation method, see Kleinke and Reinecke (2013a).

We now describe the functions to multiply impute incomplete zero-inflated (and overdispersed) clustered or panel count data in detail. The functions are part of the `countimp` R package (Kleinke & Reinecke, 2013a), and are also available from `https://github.com/kkleinke`:

```
mice.impute.2l.zihnb(y,ry,x,type,intercept.c=TRUE,intercept.z=TRUE)
mice.impute.2l.zihnb.noint.zero(y,ry,x,type,intercept.c=TRUE,intercept.z=FALSE)
mice.impute.2l.zihnb.noint.count(y,ry,x,type,intercept.c=FALSE,intercept.z=TRUE)
mice.impute.2l.zihnb.noint.both(y,ry,x,type,intercept.c=FALSE,intercept.z=FALSE)
```

The name "`2l.zihnb`" stands for "multiple imputation of two-level (`2l`) zero-inflated count data based on a hurdle negative binomial model".

The "`.noint`" variants treat the intercept only as a fixed, but not as a random effect. It may be specified, if the intercept is treated only as a fixed effect in the zero model ("`.noint.zero`"), the count model ("`.noint.count`"), or both models ("`.noint.both`"). Note that the `noint` models *do* estimate an intercept term, but treat the intercept only as a fixed effect in the respective model. A better label would probably have been `no.random.int`, but we continued to use the name `noint` for consistency with other `mice` functions.

All functions fit a binomial GLMM with a logit link as the zero model. This model determines if the respective observational unit has a zero or non-zero value. The count model on the other hand is a zero-truncated mixed effects NB model, determining, what kind of non-zero value the observational unit has. It is possible to specify different sets of predictors for the zero and the count model. This is done via the `predictorMatrix`, as explained below.

The functions receive the following input from the main `mice()` function. `y` is a numeric vector with incomplete data in long format – the variable to be imputed. Please note that `mice()` does not automatically transform the variable into long format (meaning

that the data of the different groups are stacked upon each other). If the data are in wide (i.e. multivariate) format, the user has to bring them into long format for example by using the `reshape()` function from R package **stats**, before calling our functions. `ry` is the response indicator of `y`, with `ry=TRUE` indicating that the respective value in `y` has been observed. `x` is a matrix of complete covariates (also in long format), containing the variables that will be used to predict missing information in `y`. The information stored in `type` (a vector of length equal to the number of columns in `x`) determines the imputation model. `type` is extracted from the respective row of the `predictorMatrix` slot of the `mice()` call: Allowed entries in the `predictorMatrix` are $\{-2, 0, 1, 2, 3, 4, 5, 6\}$: Codes $\{-2, 0, 1, 2\}$ are used in the same way as in other two-level `mice` imputation functions (e.g. `mice.impute.2l.norm()`). '-2' identifies the class variable (please note that the current version allows only one class variable). '0' means that the variable will not be included in the imputation model and thus not be part of `x`. '1' denotes a variable that will be included as a fixed effect in both the zero and the count model. '2' means that the variable will be included as a fixed and random effect both in the zero and count model. '3' indicates a variable to be included only as a fixed effect and only in the count model. '4' means the variable will be included as a fixed and random effect, but only in the count model. '5' stands for variables that will be included as a fixed effect only in the zero model. Finally, '6' denotes a variable to be included as a fixed and random effect in the zero model only. An example regarding how to set up the `predictorMatrix` properly is given in Kleinke and Reinecke (2013a). The functions furthermore use the following arguments: `intercept.c` can be either `TRUE` or `FALSE`. `TRUE` means, that the model will include the intercept as a random effect in the count model, `FALSE` means that the model will not use the intercept as a random effect. `intercept.z` works analogously for the zero model.

Between imputation variability is introduced by Bayesian regression (cf. Rubin, 1987, pp. 169–170):

1. The zero model is fitted to the data – a two-level binomial generalized linear mixed

effects model using the `glmmPQL` function from package `MASS`, and we compute $\hat{\theta}_z$, the posterior mean, and VAR($\hat{\theta}_z$), the posterior variance of model parameters θ_z.

2. We draw new parameters θ_z^* from $N(\hat{\theta}_z, \text{VAR}(\hat{\theta}_z))$.
3. Predicted probabilities for having a zero vs. non-zero count are computed, using θ_z^*.
4. We draw imputations from a Binomial distribution and "remember" cases, who are supposed to get a non-zero count later on.
5. The count model is fitted – a zero-truncated two-level NB model using the `glmmadmb` function from package `glmmADMB` and the `truncnbinom` family.
6. We get $\hat{\theta}_c$,, the posterior mean, and VAR($\hat{\theta}_c$),the posterior variance of model parameters θ_z.
7. We draw θ_c^* from $N(\hat{\theta}_c, \text{VAR}(\hat{\theta}_c))$.
8. Predicted counts are computed using θ_c^*.
9. We finally draw non-zero imputations (see step 4) from a zero-truncated NB distribution.

All functions return a numeric vector with imputations of length equal to the number of unobserved data points in `y` to the main `mice()` function.

3 Monte Carlo simulations

3.1 Overview of the simulations and hypotheses

To test the quality of the proposed imputation solution, we ran two Monte Carlo simulations. In the first one, we simulated two-level data sets with MAR missingness in the zero-inflated count variable. Missingness depended on an observed continuous individual level predictor. Missing data were imputed using our proposed procedure, and also by some

other procedures: two-level Poisson imputation (POI), two-level NB imputation (NB), and predictive mean matching (pmm).

Predictive mean matching Little (1988); Rubin (1986) has been recommended for imputation of variables of *any* kind, including discrete and semi-continuous data (van Buuren, 2013, May; Vink, Frank, Pannekoek, & van Buuren, 2013). The function fits a linear regression model and selects one observed value from a pool of values, whose fitted value is closest to the value predicted by the regression model. By imputing an actual observed value, pmm can buffer some of the effects regarding the misspecified regression model (here the underlying normality assumption). The question is, where this robustness ends. We wanted to test, if the recommedation that pmm may be used for variables of any kind can also be extended to the rather complex data structure of two-level zero-inflated and overdispersed count data. Following research by van Buuren (2011), we tested two pmm strategies: the first one simply applied the pmm algorithm as it is implemented in `mice` (labeled PMM subsequently). The second strategy included a cluster dummy among the predictors (labeled PMMG) – thus estimating an intercept term per group. This strategy is supposed to work for random intercept models, but is usually not able to cater for random slopes. We did not have any specific hypotheses regarding the pmm approach and our research interest was rather exploratory. However, as our data generation model also included a random slope, we did not expect the second pmm strategy to work very well.

Two-level Poisson imputation and two-level negative binomial imputation are strategies offered by the `countimp` package (Kleinke & Reinecke, 2013a). Based on findings by Kleinke and Reinecke (2013b), who demonstrated that the count data imputation model has to fit the data well to be able to obtain unbiased statistical estimates, we hypothesized that both strategies would produce suboptimal results, when the multilevel data are zero-inflated *and* overdispersed: Two-level NB imputation would not be able to estimate the correct percentage of zero counts. Additionally, two-level Poisson imputation would

also underestimate the true level of dispersion in the data.

We supposed that only an imputation procedure that is specially tailored to the problem at hand will be able to produce unbiased estimates.

In the second simulation, we again simulated missingness in the dependend count variable. Our model included a continuous individual level predictor and a continuous group level predictor, which was the "cause" of missingness here. The second simulation only tested the proposed two-level approach for zero-inflated and overdispersed count data. The purpose of this simulation was to demonstrate that the proposed procedure is able to yield unbiased parameter estimates and reasonable measures of uncertainty in the given scenario.

3.2 Quality criteria

To evaluate the quality of the respective missing data methods, we relied on quantities that are well established in the missing data literature (Demirtas, 2009; Demirtas & Hedeker, 2008; Schafer & Graham, 2002): the average parameter estimate across the replications, its standard deviation, bias, coverage rate, and the average confidence interval width. Let Q be the population parameter of interest and $\widehat{Q}$ the average estimate of Q across the replicated samples, based on the respective sample and the applied missing-data procedure. Its standard deviation is $SD_{\widehat{Q}}$. Bias in parameter estimation is defined as $\text{BIAS} = Q - \widehat{Q}$ and measures estimation *accuracy*. Relative bias is defined as $\frac{\text{BIAS}}{Q} * 100\%$ (cf. Muthén & Muthén, 2012). An accurate missing data procedure produces near-zero bias. Furthermore, a good missing data procedure yields *consistently* accurate estimates across the replicated samples, thus $SD_{\widehat{Q}}$ is supposed to be small and shall be similar to the "true" population standard error. An estimate of this standard error can be obtained by computing $SD_{\widehat{Q}}$ based on the complete data before the introduction of missing data (not to be confused with complete case analysis). Demirtas and Hedeker (2008) and Demirtas (2009) use the term estimation *precision* to refer to the adequate size

of standard errors and consequently the widths of the respective confidence intervals. Coverage rate (CR) is a hybrid measure that reflects both bias of parameter estimates and bias of standard errors. CR is defined as the percentage of 95% confidence intervals that cover the true parameter, and shall obviously be close to 95%. Schafer and Graham (2002) define CR $< 90\%$ as undercoverage. Undercoverage may be caused by low accuracy, which means that the confidence interval is too far to the left or to the right to cover the true parameter, by too narrow intervals, or by a combination of both factors.

In summary, a good missing data procedure manages to produce narrow intervals in combination with near-zero bias and high coverage.

3.3 Simulation 1

We generated two-level count that were zero-inflated and overdispersed. We ran 200 replications. Each data set consisted of $g = 50$ groups with sample size $n_j = 100$, $j = 1, \ldots, g$, which we simulated separately and which were then stacked upon each other to obtain two-level data sets in "long format". The total sample size was $N = \Sigma_{j=1}^{g} n_j = 5000$. To introduce an excess number of zero counts, we needed to specify two models, one model determining if the observational unit had a zero or non-zero count (the zero model), and the count model, which determined what non-zero count the observational unit had. The zero model was a binomial GLMM, the count model was a zero-truncated NB model. In the first simulation, we used a model with only one individual level predictor x_1.

3.3.1 Data generation. The data generation process for each group j worked in the following way: Firstly, we simulated $x_{1j} \sim \mathcal{N}(0, 1)$, $u_{0zj} \sim \mathcal{N}(0, .5)$, $u_{1zj} \sim \mathcal{N}(0, .3)$, $u_{0cj} \sim \mathcal{N}(0, .5)$, and $u_{1cj} \sim \mathcal{N}(0, .3)$, where $u_{...}$ denote the random effects, with subscripts z and c referring to the zero model and count model respectively, and 0 and 1 referring to the intercept term and slope respectively. We then obtained the parameters for the zero model

by using

$$\beta_{zj} = \gamma_z + u_{zj},$$

where

$$\beta_{zj} = \begin{bmatrix} \beta_{0zj} \\ \beta_{1zj} \end{bmatrix}, \gamma_z = \begin{bmatrix} \gamma_{00z} \\ \gamma_{10z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}, \text{and } u_{zj} = \begin{bmatrix} u_{0zj} \\ u_{1zj} \end{bmatrix}.$$

The following process determined, if y_{ij} had a zero or non-zero value:

$$y_{ij} = \begin{cases} 0 & \text{if} \quad r_{ij} \sim \mathcal{U}(0,1) < \text{invlogit}(X_j\beta_{zj}) \\ > 0 & \text{if} \quad r_{ij} \sim \mathcal{U}(0,1) >= \text{invlogit}(X_j\beta_{zj}) \end{cases}.$$

X_j is a design matrix containing a column of 1s referring to the intercept term and predictor x_{1j}. Non-zero entries in y_{ij}, labeled $y_{>0ij}$ were drawn from a zero-truncated NB distribution with size $\theta = 2$, and means $\mu_j = \exp(X_{>0j}\beta_{cj})$, where

$$\beta_{cj} = \gamma_c + u_{cj},$$

with

$$\beta_{cj} = \begin{bmatrix} \beta_{0cj} \\ \beta_{1cj} \end{bmatrix}, \gamma_c = \begin{bmatrix} \gamma_{00c} \\ \gamma_{10c} \end{bmatrix} = \begin{bmatrix} 1 \\ 0.75 \end{bmatrix}, \text{and } u_{cj} = \begin{bmatrix} u_{0cj} \\ u_{1cj} \end{bmatrix}.$$

$X_{>0j}$ refers to the subset of X_j with corresponding non-zero y_{ij} values.

Repeating the process g times and stacking the data upon each other, we ended up with a data set in long format, containing the following variables: y the zero-inflated count variable, x_1 the individual level predictor, and the group identifier grp.

Missing data were finally introduced according to the following rule:

$$y_{ij} = \begin{cases} \text{NA} & \text{if} \quad r_{ij} \sim \mathcal{U}(0,1) < \text{invlogit}(-1 + x_{1ij}) \\ y_{ij} & \text{if} \quad r_{ij} \sim \mathcal{U}(0,1) >= \text{invlogit}(-1 + x_{1ij}) \end{cases},$$

where NA indicates a missing value. This generated an average of 30.3% of missing data across the 200 replications. Missingness in y was MAR in the sense of Rubin (Rubin, 1987) and depended on x_1.

3.3.2 Missing data imputation. Missing data were imputed with the R package `mice` (van Buuren & Groothuis-Oudshoorn, 2011), using the following imputation functions: (a) `mice.impute.2l.zihnb()`, our proposed procedure specially taillored to zero-inflated and overdispersed count data, (b) `mice.impute.2l.poisson()`, two-level Poisson imputation, (c) `mice.impute.2l.nb2()`, two-level negative binomial imputation, based on a NB2 model (Hilbe, 2011) and (d) `mice.impute.pmm()`, classical "flat file" predictive mean matching. Here, we tested two variants – the first one by using the pmm function as it is, and the second one by additionally including a cluster dummy among the predictors. `mice.impute.pmm` already comes with the standard `mice` installation. The other functions are available from the `countimp` package and are described in detail in the `countimp` user's manual (Kleinke & Reinecke, 2013a). The imputation models of all two-level imputation functions estimated the intercept as well as the slope of x_1 as random factors. The two-level hurdle model imputation function estimated random intercepts and slopes for the zero and the count model. The first pmm strategy (labeled PMM) ignored the multilevel structure of the data set and imputed missing data in the incomplete count variable y in long format on the basis of the linear regression model $y_i = \beta_0 + \beta_1 x_{1i} + \varepsilon_i$. The second pmm strategy additionally included the group identifier as a categorical variable among the predictors.

3.3.3 Data analysis. The data sets were then analyzed by a two-level hurdle NB model. The zero model was a binomial GLMM, with x_1 as a fixed and random factor. The model was estimated using the function `glmmPQL()` from R package `MASS` (Venables & Ripley, 2002). The count model – a zero-truncated two-level NB model – also included x_1 among the fixed and random effects. The model was estimated by the `glmmadmb()` function from package `glmmADMB` using the `truncnbinom` family.

3.3.4 Results. We first compared, how well each missing data procedure was able to preserve the distribution of the original data. We compared estimates of the mean, standard deviation and the percentages of the respective counts from 0 to 10. Results are

displayed in Table 1.

As can be seen in that table, two-level hurdle NB imputation (HNB) and PMM seemed to be able to preserve the original distribution of the incomplete count variable quite well, especially the percentage of zero counts had been estimated well. The pmm strategy with the cluster dummy (PMMG) on the other hand seemed to slightly underestimate the percentage of zeros, as did the Poisson and the negative binomial approches: Poisson imputation underestimated the percentage of zero counts by 11.7%. The ones on the other hand were overestimated by 4.5%. The standard deviation was on average 1.46 points lower in comparison to the complete data. Also, NB imputation underestimated the standard deviation by 1.09. The percentage of zero counts was 3.7% lower in comparison to the original data and ones were overestimated by 2.5%. Based on these findings, we might already expect that regression coefficients of the pmm cluster dummy variant, two-level Poisson imputation and NB2 imputation will not be very accurate.

Results regarding the model coefficients may be found in Tables 2 and 3. Table 2 gives a comparison of the complete data estimates and the two-level hurdle NB results. As can be seen in that table, bias was rather small in most cases. The highest absolute bias was -.08 (i.e. 4% of the original parameter) for the overdispersion parameter θ of the count model. The highest relative bias was 13.3% (i.e. an absolute bias of .04) for the random slope in the zero model. Coverage rates were well above the 90% threshold. The average confidence interval widths were slightly higher in comparison to the complete data estimates. Note again, that MI adds a little bit of extra conservativeness to the standard error estimates by combining within and between imputation variation to reflect the uncertainty in parameter estimates due to missing data. Note also that the multilevel functions in R typically do not estimate standard errors of the random effects (for reasons, see for example: Jul 15, 2006 posting by D Bates to the R-help mailing list; `https://stat.ethz.ch/pipermail/r-help/2006-July/109308.html`). We therefore did

not compute confidence intervals and coverage rates for the random parts of the models. All in all, results of the proposed hurdle NB approach can be regarded as sufficiently good. Results of the other aproaches are presented in Table 3. We can see that two-level Poisson imputation produced fixed effects estimates that went far astray. Biases were huge, effects of the count model were severely underestimated, and parameter estimates of the zero model even went in the wrong direction. Coverage was close to zero. Two-level NB imputation also produced highly unsatisfactory results. Coverage was far from the acceptable 90% threshold. All fixed effects were underestimated quite noticeably. Even the overdispersion parameter was not estimated correctly. We now turn to the two pmm approaches. The ordinary pmm solution produced rather small relative biases in the fixed effects estimates of between 0% and 8%, with corresponding coverage rates of between 79.9% and 93.47%. This is actually not too bad, however on the other hand also not sufficiently good. Biases in the random effects estimates were all larger in comparison to our proposed hurdle NB procedure and pmm also did not estimate the overdispersion parameter correctly. The pmm cluster dummy variant produced noticeably worse results than the other pmm solution. It is a practicable solution for random intercept models, but obviously cannot cater for random slopes. Note again that both pmm approaches imputed an actual observed value on the basis of a standard linear regression model assuming homoscedasticity and normal errors. By imputing an observed value, pmm can buffer violated model assumptions to *some* extent. However, this simulation showed, where the limits of this robustness lie. Neither of the currently existing approaches could produce results that were as good as those obtained by our proposed procedure.

3.4 Simulation 2

The second simulation worked in the same way as Simulation 1, with the following exceptions: Here, we used two predictors, an individual level predictor x_1 and a group level predictor, labeled z_1. Again, for each group j,we simulated the following quantities:

$x_{1j} \sim \mathcal{N}(0,1)$, $z_{1j} \sim \mathcal{N}(0,1)$, $u_{0zj} \sim \mathcal{N}(0,.5)$, $u_{1zj} \sim \mathcal{N}(0,.3)$, $u_{0cj} \sim \mathcal{N}(0,.5)$, and $u_{1cj} \sim \mathcal{N}(0,.3)$.

We then obtained the parameters for the zero model by using

$$\beta_{zj} = \Gamma_z z_j + u_{zj},$$

where

$$\beta_{zj} = \begin{bmatrix} \beta_{0zj} \\ \beta_{1zj} \end{bmatrix}, \Gamma_z = \begin{bmatrix} \gamma_{00z} & \gamma_{01z} \\ \gamma_{10z} & \gamma_{11z} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0.5 & 0 \end{bmatrix}, z_j = \begin{bmatrix} 1 \\ z_{1j} \end{bmatrix}, \text{and } u_{zj} = \begin{bmatrix} u_{0zj} \\ u_{1zj} \end{bmatrix}.$$

The following process determined, if y_{ij} had a zero or non-zero value:

$$y_{ij} = \begin{cases} 0 & \text{if} \quad r_{ij} \sim \mathcal{U}(0,1) < \text{invlogit}(X_j \boldsymbol{\beta}_{zj}) \\ > 0 & \text{if} \quad r_{ij} \sim \mathcal{U}(0,1) >= \text{invlogit}(X_j \boldsymbol{\beta}_{zj}) \end{cases}$$

Non-zero entries in y_{ij}, labeled $y_{>0ij}$ were drawn from a zero-truncated NB distribution with size $\theta = 1$, and means $\mu_j = \exp(X_{>0j}\beta_{cj})$, where

$$\beta_{cj} = \Gamma_c z_j + u_{cj},$$

with

$$\beta_{cj} = \begin{bmatrix} \beta_{0cj} \\ \beta_{1cj} \end{bmatrix}, \Gamma_c = \begin{bmatrix} \gamma_{00c} & \gamma_{01c} \\ \gamma_{10c} & \gamma_{11c} \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0.75 & 0 \end{bmatrix}, z_j = \begin{bmatrix} 1 \\ z_{1j} \end{bmatrix}, \text{and } u_{cj} = \begin{bmatrix} u_{0cj} \\ u_{1cj} \end{bmatrix}.$$

$X_{>0j}$ refers to the subset of X_j with corresponding non-zero y_{ij} values.

Again, by repeating the process g times and stacking the data upon each other, we got a data set in long format, containing the following variables: y the zero-inflated count variable, x_1 the individual level predictor, z_1, the group level predictor, and the group identifier grp.

Missing data were finally introduced according to the following rule:

$$y_{ij} = \begin{cases} \text{NA} & \text{if} \quad r_{ij} \sim \mathcal{U}(0,1) < \text{invlogit}(-1 + \boldsymbol{z}_{1j}) \\ y_{ij} & \text{if} \quad r_{ij} \sim \mathcal{U}(0,1) >= \text{invlogit}(-1 + \boldsymbol{z}_{1j}) \end{cases},$$

where NA indicates a missing value. This generated an average of 30.1% of missing data across the 200 replications. In this simulation, missingness depended on group level variable $\boldsymbol{z}_1$.

3.4.1 Results. Results are presented in Table 4. Bias in parameter estimation was negligibly small. The highest relative bias was 3% (corresponding to an absolute bias of .01) for σ_{11c}. Confidence interval widths were comparable to the complete data results, coverage was acceptably large, indicating that both parameter and corresponding standard error estimates were reasonable.

4 Discussion

We have proposed a multiple imputation procedure for incomplete two-level count data that are zero-inflated or zero-inflated and additionally overdispersed. The procedure is based on the multiple imputation by chained equations approach and works as an add-on to the `mice` software in R (van Buuren & Groothuis-Oudshoorn, 2011). It imputes missing data on the basis of a two-level hurdle negative binomial model. Hurdle models are mixture models and consist of a zero model (a binomial GLMM) and a count model (a zero-truncated NB model). The zero model determines, if the observational unit has a zero vs. non-zero value. The count model determines, what non-zero value the observational unit has.

We presented two Monte Carlo simulations in which we evaluated the quality of the proposed approach. In the first simulation we compared the performance of our procedure to other currently available solutions for two-level count data: two-level Poisson imputation, two-level NB imputation and flat file predictive mean matching. Here we compared a simple predictive mean matching approach and a pmm variant that included a cluster dummy among the predictors. This strategy was supposed to yield better estimates of random intercepts (van Buuren, 2011). We found that only the two-level hurdle NB approach yielded acceptable results. Poisson imputation severely underestimated the percentage of zero counts, NB imputation also underestimated the zero counts. With about 30% of the values in y to be filled in, we see that neither the Poisson model nor the NB model are good imputation models in the sense that they are able to preserve both the

structure of the data and the relationships within the data set. We further see that filling in about one third of the data in y with inadequate information can actually do a lot of damage.

The cluster dummy pmm variant produced inacceptable results, as well. This was in accordance with our hypothesis that the cluster dummy variant is only an option for random intercept models, but not for random slope models. The simple pmm variant did astonishingly well. Though the underlying linear regression model was severely violated (e.g. it ignored the hierarchical structure of the data and assumed a normal model), pmm could buffer these violations quite well by imputing an actual observed value. However, results were still suboptimal and our solution that was specially tailored to the problem at hand produced arguably better results.

Our results corroborate our earlier stated notion that for more complex problems (where the robustness of currently available solutions like pmm fails), we need missing data procedures that are specially tailored to the problem at hand.

The second simulation used a model that was a little bit more complex than the one from Simulation 1 – with an individual level predictor and a group level predictor, which was the "cause" of missingness. Again, the proposed procedure produced reasonable estimates.

Limitations of the present study can be stated as follows: Using artificial data always has advantages and disadvantages. On the one hand it is good to be able to control as many parameters as possible in an artificial Monte Carlo simulation. It is then possible to demonstrate that the procedures work in the scenarios they were designed for and that the results are a consequence of the manipulations made in the study. On the other hand, real life scenarios are often different from scenarios used in Monte Carlo studies. Monte Carlo simulations often test the limits of certain procedures and missing data mechanisms and percentages are often less severe in empirical data. Assumed violations of the statistical model are often not so extreme than the scenarios used in Monte Carlo simulations.

Kleinke, Stemmler, Reinecke, and Lösel (2011) for example have shown that pmm worked well for empirical two-level data that were approximately multivariate normal. We have tried to be "realistic" in our simulations in a way that we used maximum missing data percentages of about 30% that may be typically found in empirical longitudinal data (cf. Lally, Mangione, & Honig, 1988; McCord, 1978). Future research should test pmm using empirical multilevel count data to see if pmm can cope better in these scenarios.

A second disadvantage of using simulated data is that the distribution of empirical data furthermore deviates at least to some extent from the convenient statistical models used for data imputation and data analysis. Future research needs to look more deeply into how well our procedure copes with varying degrees of model fit.

Future studies also need to address various kinds of model misspecifications: Currently, our procedure assumes homoscedasticity – an assumption that is sometimes problematic when analyzing empirical data. It will be a fruitful avenue for future software development to allow for a heteroscedastic imputation model. Sometimes, empirical data also violate other parametric assumptions of the imputation and analysis models. The `gamlss` package in R (Stasinopoulos & Rigby, 2007) allows the estimation of semiparametric models that might be used in these cases. de Jong, van Buuren, and Spiess (2013) already have proposed an imputation procedure for continuous data based on that idea, which may be generalized to two-level count models. Future research needs to establish practical guidelines, which procedures work best in certain given scenarios, i.e. when to use parametric approaches, and when to use non- or semiparametric approaches.

Finally, two hierarchical levels are sometimes not sufficient (e.g. students nested in classes nested in schools). It will be a necessary avenue for future program development to support more than two levels.

Computational details

The results in this paper were obtained using R 3.0.0 with packages `aster` 0.8-23, `countimp 1.0`, `glmmADMB` 0.7.4, `MASS` 7.3-26, and `mice` 2.17. R itself and the packages `aster`, `MASS` and `mice` are available from `http://CRAN.R-project.org/`. The `countimp` package is available from `www.uni-bielefeld.de/soz/kds/software.html`. `glmmADMB` is available from `http://glmmadmb.r-forge.r-project.org`.

5 References

Allison, P. D. (2001). *Missing data.* Thousand Oaks, CA: Sage.

Bryk, A. S., & Raudenbush, S. W. (1992). *Hierarchical linear models.* Newbury Park, CA: Sage.

Collins, L. M., Schafer, J. L., & Kam, C. M. (2001). A comparison of inclusive and restrictive missing-data strategies in modern missing-data procedures. *Psychological Methods*, *6*, 330–351.

de Jong, R., van Buuren, S., & Spiess, M. (2013). *Multiple imputation of predictor variables using generalized additive models.* (Manuscript submitted for publication)

Demirtas, H. (2009). Multiple imputation under the generalized lambda distribution. *Journal of Biopharmaceutical Statistics*, *19*(1), 77–89.

Demirtas, H., & Hedeker, D. (2008). Multiple imputation under power polynomials. *Communications in Statistics – Simulation and Computation*, *37*(8), 1682–1695.

Enders, C. (2010). *Applied missing data analysis.* New York, NY: Guilford.

Fisher, R. A. (1941). The negative binomial distribution. *Annals of Eugenics*, *11*(1), 182–187.

Gelman, A., & Hill, J. (2007). *Data analysis using multilevel / hierarchical models.* Cambridge: Cambridge University Press.

Graham, J. W., Cumsille, P. E., & Elek-Fisk, E. (2003). Methods for handling missing data. In J. A. Schinka & W. F. Velicer (Eds.), *Handbook of psychology: Volume 2. Research methods in psychology* (pp. 87–114). Hoboken, NJ: Wiley & Sons.

Hilbe, J. M. (2011). *Negative binomial regression* (2^{nd} ed.). Cambridge: Cambridge University Press.

Kleinke, K., & Reinecke, J. (2013a). *countimp 1.0 – A multiple imputation package for incomplete count data* (Technical Report). Bielefeld: University of Bielefeld. Retrieved from www.uni-bielefeld.de/soz/kds/pdf/countimp.pdf

Kleinke, K., & Reinecke, J. (2013b). Multiple imputation of incomplete zero-inflated count

data. *Statistica Neerlandica*, *67*(3), 311–336. doi: 10.1111/stan.12009

Kleinke, K., Stemmler, M., Reinecke, J., & Lösel, F. (2011). Efficient ways to impute incomplete panel data. *Advances in Statistical Analysis*, *95*(4), 351–373.

Lally, J. R., Mangione, P. L., & Honig, A. S. (1988). The Syracuse University Family Development Research Program: Long-range impact of an early intervention with low-income children and their families. In D. R. Powell (Ed.), *Parent education as early childhood intervention: emerging directions in theory, research and practice* (pp. 79–104). Norwood, NJ: Ablex.

Little, R. J. A. (1988). Missing-data adjustments in large surveys. *Journal of Business & Economic Statistics*, *6*(3), 287–296.

Little, R. J. A., & Rubin, D. B. (1987). *Statistical analysis with missing data.* New York: Wiley.

McCord, J. (1978). A thirty-year follow-up of treatment effects. *American Psychologist*, *33*, 284–289.

Mullahy, J. (1986). Specification and testing of some modified count data models. *Journal of Econometrics*, *33*, 341–365.

Muthén, L. K., & Muthén, B. O. (2012). *Mplus user's guide* (7^{th} ed.). Los Angeles, CA: Muthén & Muthén.

Raghunathan, T. E., Lepkowski, J. M., van Hoewyk, J., & Solenberger, P. (2001). A multivariate technique for multiply imputing missing values using a sequence of regression models. *Survey Methodology*, *27*(1), 85–96.

Raghunathan, T. E., Solenberger, P. W., & Van Hoewyk, J. (2002). *IVEware: imputation and variance estimation software.* Michigan. Retrieved September 18, 2013, from ftp://ftp.isr.umich.edu/pub/src/smp/ive/ive_user.pdf

Rosseel, Y. (2012). lavaan: An R package for structural equation modeling. *Journal of Statistical Software*, *48*(1), 1–36.

Rubin, D. B. (1976). Inference and missing data. *Biometrika*, *63*, 581–592.

Rubin, D. B. (1986). Statistical matching using file concatenation with adjusted weights and multiple imputations. *Journal of Business & Economic Statistics*, *4*(1), 87–94.

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. New York: Wiley.

Sampford, M. (1955). The truncated negative binomial distribution. *Biometrika*, *42*(1/2), 58–69.

Schafer, J. L. (1997a). *Analysis of incomplete multivariate data*. London: Chapman & Hall.

Schafer, J. L. (1997b). *Imputation of missing covariates under a general linear mixed model* (Technical Report 97-10). University Park: Pennsylvania State University, The Methodology Center.

Schafer, J. L., & Graham, J. W. (2002). Missing data: Our view of the state of the art. *Psychological Methods*, *7*, 147–177.

Schafer, J. L., & Yucel, R. M. (2002). Computational strategies for multivariate linear mixed-effects models with missing values. *Journal of Computational and Graphical Statistics*, *11*(2), 437–457.

Stasinopoulos, D. M., & Rigby, R. A. (2007). Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software*, *23*(7), 1–46. Retrieved from http://www.jstatsoft.org/v23/i07

van Buuren, S. (2011). Multiple imputation of multilevel data. In J. J. Hox & J. K. Roberts (Eds.), *Handbook of advanced multilevel analysis* (pp. 173–196). New York (NY): Taylor & Francis.

van Buuren, S. (2013, May). *Introduction to MICE and multilevel imputation*. Retrieved September 18, 2013, from http://www.stefvanbuuren.nl/mi/docs/Utrecht-15MayCourse%20handout.pdf (Paper presented at the "Advanced Multiple Imputation" Workshop, Utrecht, NL)

van Buuren, S., Brand, J. P. L., Groothuis-Oudshoorn, C. G. M., & Rubin, D. B. (2006). Fully conditional specification in multivariate imputation. *Journal of Statistical*

Computation and Simulation, *76*(12), 1049–1064.

van Buuren, S., & Groothuis-Oudshoorn, K. (2011). MICE: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, *45*(3), 1–67.

Venables, W. N., & Ripley, B. D. (2002). *Modern applied statistics with S*. New York: Springer.

Vink, G., Frank, L. E., Pannekoek, J., & van Buuren, S. (2013). *Predictive mean matching imputation of semicontinuous variables.* (Manuscript submitted for publication)

Yucel, R. M. (2008). Multiple imputation inference for multivariate multilevel continuous data with ignorable non-response. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1874), 2389–2403.

Table 1

Simulation 1 – descriptive statistics

	COM	HNB	PMM	PMMG	POI	NB
M	2.13	2.09	2.13	2.09	2.11	2.04
SD	5.46	5.13	5.29	4.72	4.00	4.37
%0	50.16	50.23	50.20	47.49	38.40	46.46
%1	15.98	15.93	15.98	17.37	20.51	18.46
%2	9.85	9.88	9.88	10.61	13.90	10.85
%3	6.41	6.41	6.39	6.79	8.92	6.83
%4	4.27	4.30	4.28	4.50	5.57	4.50
%5	3.01	3.00	2.96	3.10	3.56	3.07
%6	2.13	2.16	2.15	2.23	2.35	2.18
%7	1.59	1.58	1.57	1.62	1.60	1.59
%8	1.21	1.21	1.21	1.23	1.14	1.19
%9	0.90	0.90	0.89	0.90	0.81	0.89
%10	0.71	0.71	0.70	0.70	0.60	0.69

Note. The table displays, how well different MI procedures were able to preserve the distribution of the original variable. COM are the simulated complete data, HNB is two-level imputation for zero-inflated and overdispersed count data, based on a hurdle NB model, PMM is predictive mean matching, PMMG denotes the predictive mean matching variant with an additional dummy indicating group membership, POI is two-level Poisson imputation and NB is two-level imputation based on a NB2 model. M is the mean, SD the standard deviation, and %0–%10 the observed relative frequency of the respective count (averaged across the 200 replications and – if applicable – the $m = 5$ imputations).

Table 2

Simulation 1 – two-level hurdle NB imputation results (HNB)

		Sim. Complete Data				MI Estimates				
	Q	$\widehat{Q}$	$SD_{\widehat{Q}}$	CR	WID	$\widehat{Q}$	$SD_{\widehat{Q}}$	BIAS	CR	WID
β_{0z}	0.00	0.01	0.06	100.00	0.25	0.01	0.07	-0.01	96.50	0.32
β_{1z}	0.50	0.49	0.06	100.00	0.20	0.49	0.07	0.01	93.50	0.25
β_{0c}	1.00	1.01	0.07	100.00	0.32	1.00	0.11	0.00	94.00	0.31
β_{1c}	0.75	0.75	0.05	100.00	0.17	0.74	0.06	0.01	92.00	0.22
σ_{00z}	0.50	0.49	0.06			0.48	0.06	0.02		
σ_{11z}	0.30	0.28	0.05			0.26	0.06	0.04		
σ_{00c}	0.50	0.49	0.06			0.49	0.06	0.01		
σ_{11c}	0.30	0.29	0.04			0.27	0.05	0.03		
θ	2.00	2.02	0.15			2.08	0.21	-0.08		

Note. Subscrips c and z denote the count and zero model respectively. β are the fixed effects, σ the random effects standard deviations. Q is the simulated population parameter, $\widehat{Q}$ the average parameter estimate across the 200 replications, $SD_{\widehat{Q}}$ its standard deviation. CR is the 95% coverage rate. BIAS is the defined as $Q - \widehat{Q}$. WID is the average confidence interval width. The left part of the table displays the results based on the simulated complete data, the right hand side the results of the multiply imputed incomplete data.

Table 3

Simulation 1 – results of various proxy imputation procedures

	two-level Poisson imputation					two-level NB2 imputation				
	$\widehat{Q}$	$SD_{\widehat{Q}}$	BIAS	CR	WID	$\widehat{Q}$	$SD_{\widehat{Q}}$	BIAS	CR	WID
β_{0z}	-0.49	0.07	0.49	0.00	0.30	-0.15	0.07	0.15	42.64	0.28
β_{1z}	-0.00	0.06	0.50	0.00	0.21	0.22	0.06	0.28	0.51	0.20
β_{0c}	0.69	0.22	0.31	6.28	0.37	0.72	0.09	0.28	7.11	0.34
β_{1c}	0.41	0.26	0.34	0.00	0.20	0.54	0.06	0.21	4.57	0.20
σ_{00z}	0.45	0.05	0.05			0.41	0.05	0.09		
σ_{11z}	0.24	0.04	0.06			0.23	0.04	0.07		
σ_{00c}	0.48	0.06	0.02			0.48	0.06	0.02		
σ_{11c}	0.28	0.04	0.02			0.25	0.04	0.05		
θ	1.91	0.27	0.09			0.99	0.08	1.01		
	pmm					pmm with cluster dummy				
	$\widehat{Q}$	$SD_{\widehat{Q}}$	BIAS	CR	WID	$\widehat{Q}$	$SD_{\widehat{Q}}$	BIAS	CR	WID
β_{0z}	0.00	0.06	-0.00	93.47	0.25	-0.11	0.06	0.11	56.50	0.23
β_{1z}	0.47	0.07	0.03	79.90	0.22	0.28	0.04	0.22	0.50	0.18
β_{0c}	0.99	0.09	0.01	85.93	0.28	0.84	0.09	0.16	53.00	0.34
β_{1c}	0.81	0.08	-0.06	83.92	0.29	0.60	0.07	0.15	21.50	0.19
σ_{00z}	0.33	0.04	0.17			0.34	0.05	0.16		
σ_{11z}	0.20	0.03	0.10			0.19	0.03	0.11		
σ_{00c}	0.35	0.05	0.15			0.52	0.05	-0.02		
σ_{11c}	0.23	0.04	0.07			0.21	0.03	0.09		
θ	1.25	0.18	0.75			1.27	0.16	0.73		

Note. Subscrips c and z denote the count and zero model respectively. β are the fixed effects, σ the random effects standard deviations. Q is the simulated population parameter, $\widehat{Q}$ the average parameter estimate across the 200 replications, $SD_{\widehat{Q}}$ its standard deviation. CR is the 95% coverage rate. BIAS is the defined as $Q - \widehat{Q}$. WID is the average confidence interval width.

Table 4

Simulation 2 – two-level hurdle NB imputation results

		Sim. Complete Data				MI Estimates				
	Q	$\widehat{Q}$	$SD_{\widehat{Q}}$	CR	WID	$\widehat{Q}$	$SD_{\widehat{Q}}$	BIAS	CR	WID
β_{0z}	0.00	-0.00	0.08	100.00	0.31	0.00	0.08	-0.00	92.50	0.31
β_{1z}	0.50	0.49	0.05	100.00	0.20	0.49	0.05	0.00	93.00	0.19
β_{2z}	0.00	-0.00	0.08	100.00	0.30	0.00	0.08	-0.00	95.00	0.32
β_{0c}	1.00	1.00	0.07	100.00	0.31	0.99	0.07	0.00	96.00	0.30
β_{1c}	0.75	0.75	0.05	100.00	0.16	0.74	0.06	0.00	91.50	0.19
β_{2c}	0.50	0.49	0.08	100.00	0.32	0.48	0.08	0.01	95.00	0.31
σ_{00z}	0.50	0.49	0.06			0.48	0.06	0.01		
σ_{11z}	0.30	0.29	0.04			0.28	0.04	0.01		
σ_{00c}	0.50	0.48	0.05			0.48	0.05	0.01		
σ_{11c}	0.30	0.29	0.04			0.29	0.04	0.01		
θ	1.00	1.00	0.05			1.01	0.07	-0.01		

Note. Subscrips c and z stand for the count and zero model respectively. β are the fixed effects (where $\beta_{1.}$ is the coefficient of $\boldsymbol{x}_1$, and $\beta_{2.}$ is the coefficient of $\boldsymbol{z}_1$), σ denote the random effects standard deviations. Q is the simulated population parameter, $\widehat{Q}$ the average parameter estimate across the 200 replications, $SD_{\widehat{Q}}$ its standard deviation. CR is the 95% coverage rate. BIAS is the defined as $Q - \widehat{Q}$. WID is the average confidence interval width. The left side of the table displays the results based on the simulated complete data, the right hand side the results of the multiply imputed incomplete data.