Guidelines to Small Area Estimation for Poverty Mapping

Paul Corral, Isabel Molina, Alexandru Cojocaru, and Sandra Segovia

June 16, 2022

Contents

Acknowledgments

"If I have seen further it is by standing in the shoulder of giants." – Sir Isaac Newton

These guidelines were prepared by Paul Corral, Isabel Molina, Alexandru Cojocaru, and Sandra Segovia. The work has been carried out under the general direction of Carolina Sanchez, Benu Bidani, Carlos Rodriguez Castelan and Kristen Himelein. These guidelines build upon the work of people within the World Bank and academia like Chris Elbers, Jean Lanjouw and Pete Lanjouw, who developed the ELL methodology, as well as the work of many other statisticians outside of the World Bank. Special thanks to Qinghua Zhao for his advice and guidance during the creation of the Stata version of PovMap. Also thanks to João Pedro de Azevedo, Minh Cong Nguyen, and Roy Van der Weide who all played crucial roles in the advancement of the methods and tools used for poverty mapping within the World Bank. The team also appreciates the many helpful comments on the guidelines and the background papers from Minh Cong Nguyen, William Seitz, João Pedro de Azevedo, Moritz Meyer, Samuel Freije-Rodriguez, Roy Van der Weide and David Newhouse. Also a special thanks to the reviewers of the final guidelines: Domingo Morales, Harm Jan Boonstra, Roy Van der Weide and Partha Lahiri. Finally, thanks to Kristen Himelein and C. Jason Smith for their excellent editing. The team is also grateful to our many colleagues in national statistical agencies across the globe who have helped us to improve our approach to the work. A special thanks is also given to Mexico's Instituto Nacional de Estadística y Geografia (INEGI) for providing us access to their Intercensal Survey. Without this survey, much of the work would not have been possible.

Chapter 1

Small Area Estimation for Poverty Mapping

The eradication of poverty, which was the first of the Millennium Development Goals (MDG) established by the United Nations and followed by the Sustainable Development Goals (SDG), requires knowing where the poor are located. Traditionally, household surveys are considered the best source of information on the living standards of a country's population. Data from these surveys typically provide a sufficiently accurate direct estimate of household expenditures or income and thus estimates of poverty at the national level and larger international regions. However, when one starts to disaggregate data by local areas or population subgroups, the quality of these direct estimates diminishes. Consequently, National Statistical Offices (NSOs) cannot provide reliable wellbeing statistical figures at a local level. For example, the "Module of Socioeconomic Conditions of the Mexican National Survey of Household Income and Expenditure" (ENIGH in Spanish) is designed to produce estimates of poverty and inequality at the national level and for the 32 federate entities (31 states and Mexico City) with disaggregation by rural and urban zones, every two years, but there is a mandate to produce estimates by municipality every five years, and the ENIGH alone cannot provide estimates for all municipalities with adequate precision. This makes monitoring progress toward the Sustainable Development Goals more difficult.

Beyond the SDGs, poverty estimates at a local level are crucial for poverty reduction. Policymakers interested in poverty reduction and alleviation will often have a limited budget, and information on poverty at a local level can potentially improve targeting. The need for cost-effective disaggregated statistics makes it necessary to use indirect techniques for poverty estimation at a local level. To better understand the spatial distribution of poverty within countries, the World Bank has been producing poverty estimates at increasingly higher resolutions (poverty maps) since before the turn of the 21st century. A poverty map is the illustration of poverty rates on a map and is used to highlight the spatial distribution of poverty and inequality across a given country. Poverty mapping relies on Small Area Estimation (SAE) methods which are statistical tools that can be deployed to produce statistics for small areas with much better quality than those obtained directly from a sample survey.[1]

Small area techniques combine auxiliary information from censuses, registers, or other larger surveys to produce disaggregated statistics of sufficient precision where survey data alone is insufficient. The

[1]There is no universal threshold used to define a small area; every NSO establishes its own limit of what is an acceptable coefficient of variation (CV). Anything above the threshold is unlikely to be published by the NSO. Moreover, there is also no defined sample size that can determine if a group can be considered a small area since this is also dependent on the indicator one wishes to estimate. For poverty, for example, the magnitude of the poverty rate is also related to the necessary sample size for its accurate estimation, with higher proportions requiring smaller sample sizes.

techniques achieve this by specifying some homogeneity relationships across areas; hence, they are based on the idea that "union is strength." Despite small area estimation being useful for many indicators, the focus here is on small area methods used for poverty and related indicators derived from welfare.

Ending extreme poverty is one of the World Bank's twin goals and poverty reduction is how the institution's efforts are often gauged. Accurate estimates of poverty at a granular level are an essential tool for monitoring poverty reduction and spatial inequalities. The World Bank's efforts to develop small area estimates of poverty can be traced back to the late 1990s. An initial effort by the institution was implemented by Hentschel et al. (1998) in an application in Ecuador and is the bedrock to what came after. Much of the research efforts on small area estimation within the World Bank came to a zenith in the early 2000s with the publication of the paper "Micro-level estimation of poverty and inequality" by Elbers, Lanjouw, and Lanjouw (2003). The method came to be known as ELL, and for over a decade-and-a-half, it was the default method used for all poverty mapping conducted by the World Bank. The method's popularity was spurred by the World Bank's development of a software application called `PovMap` (Zhao 2006).[2] The software provided a user-friendly interface for practitioners and did not require knowledge of specialized programming languages. The `PovMap` software was followed nearly a decade after by a Stata version called `sae` (Nguyen et al. 2018).[3]

The Guidelines to Small Area Estimation for Poverty Mapping (hereafter the Guidelines) come after more than two decades of poverty mapping by the World Bank. These guidelines on small area estimation are meant as a practical guide to enable practitioners working on small area estimation of poverty to make informed choices between the available methods and make them aware of the strengths and limitations of the methodologies at hand. The Guidelines are also presented for practitioners who wish to learn about updates to the software and methods used within the World Bank for small area estimation. The Guidelines are not meant to be an exhaustive presentation of the small area estimation literature, instead they focus almost exclusively on poverty and are a practical guide for practitioners to assist them in navigating the multiple different small area techniques available.[4] The methods include, but are not limited to, those proposed by Elbers, Lanjouw, and Lanjouw (2003) that generated a lot of the early poverty mapping work at the World Bank, a number of subsequent refinements and improvements to the original ELL methodology, area-level models, and some ongoing research. The Guidelines advise readers on what may be the best approach for different poverty mapping projects and different sets of data constraints by reviewing the relevant literature and conducting simulation exercises that compare across available methodologies. It is expected that the Guidelines will become a valuable resource to practitioners working in the area of poverty mapping.

Before jumping into the Guidelines, a brief background to small area estimation is provided, as well as a decision tree aimed at practitioners (Figure 1.1) who seek what may be the best approach given their context. After the brief background, the guidelines present direct estimates in Chapter 2. Area-level models are presented in Chapter 3, focusing on Fay-Herriot methods (Fay and Herriot 1979). Unit-level methods are presented in Chapter 4, giving special attention to the approach from Elbers, Lanjouw, and Lanjouw (2003) as well as that of Molina and Rao (2010). Chapter 5, presents proposed alternatives for poverty mapping in off-census years beyond area-level models. Finally, Chapter 6 is devoted to model selection and other considerations. Throughout chapters 2–5 the pros and cons of the chapter's corresponding approaches are noted.

[2] Demombynes (2002) presents earlier work of a module in SAS of the methodology.

[3] Although `sae` was initially created to replicate many of the features and methods implemented in `PovMap`, the package has undergone many updates. Corral, Molina, and Nguyen (2021) detail the main updates and the reasons behind these.

[4] For a more thorough discussion on small area estimation, including applications for other indicators of interest, readers should refer to Rao and Molina (2015).

1.1 A Brief Background to Small Area Estimation

Small area estimates are often used to create a poverty map, but the two terms, "Poverty Mapping" and "Small Area Estimation", should not be used interchangeably. Small area estimation is a set of methods used by National Statistical Offices (NSO) and other organizations to achieve estimates of acceptable quality for publication. As statistical information is disaggregated into smaller population subgroups, it will become noisier and will yield larger Coefficients of Variation (CV), which may exceed the threshold established by the NSO above which they will choose not to publish that specific estimate. Model-based small area estimation techniques incorporate auxiliary information from censuses, registers, geospatial data or other large surveys, to produce estimates of much better quality, even for areas with very small sample sizes.

Model-based techniques are popular because they yield good quality estimates even for areas with very small sample sizes. Of course, to achieve this, one needs to make model assumptions, and these assumptions need to be validated using the available data (Molina, Corral, and Nguyen 2021). Moreover, the quality of the model behind the estimates should be thoroughly evaluated and small area estimates should be accompanied by valid measures of their precision (Rao and Molina 2015).

Among small area estimation methodologies, area-level models use only aggregate auxiliary information at the area level. Aggregated data are more readily available because it is typically not subject to confidentiality. These models have other advantages that come with aggregate data, such as lower sensitivity to unit-level outliers. The disadvantage of the area-level method is less detailed information than unit-level models, specifically at the microdata level. Perhaps the most popular area method is the one proposed by Fay and Herriot (1979) which is discussed in more detail in Chapter 3 of these Guidelines.

Unit-level models take advantage of detailed unit-record information, when such information is available. The models rely on detailed income/expenditure information from household surveys and a set of household-level characteristics in both a household survey and a population census to simulate household expenditures or incomes for each household in the population census data.[5]

The first unit-level model for small area estimation, discussed in more detail in Chapter 4 of these Guidelines, was proposed by Battese, Harter, and Fuller (1988) to estimate county crop areas of corn and soybeans. That model, called hereafter the BHF Model, includes, apart from the usual individual errors, random effects for the areas representing the between-area heterogeneity or idiosyncrasy the available auxiliary variables cannot explain. These models might also include area-level (or contextual) variables and thus may incorporate much more information in the estimation process than area-level models. As noted by Robinson (1991), models with random effects belong to the general class of mixed models and are commonly used in other fields such as Biostatistics, Engineering, Econometrics, and additional Social Sciences.[6]

Model-based small area estimates rely on statistical models to establish a relationship between a dependent variable and a set of covariates. The covariates and the parameters obtained from the fitted model are then used to predict the dependent variable in the auxiliary data. In the context of small area estimates of poverty, under unit-level models, the dependent variable is typically consumption/expenditure or income and rely on properly replicating the welfare distribution. The simulated welfare distribution makes it possible to obtain well-being indicators, such as the FGT class of indicators (Foster, Greer, and

[5] Note that simulations are not required when using analytical formulas to obtain poverty estimates. When assuming normality, the expected value of the poverty headcount and gap can be calculated without resorting to Monte Carlo simulations (Molina 2019).

[6] Early applications were in the area of animal and plant breeding to estimate genetic merits (Robinson 1991).

Thorbecke 1984). In the case of area-level models, the dependent variable is the direct estimator of the same indicator for the area.

The ELL method falls under the class of methods based on unit-level models for small area estimation. The method relies on a BHF model specified at the household level and may also include area or sub-area-level (contextual) variables. At the time of publication, it was one of the first times a model relied on using microdata to simulate the welfare distribution, which allowed the calculation of numerous indicators beyond just area aggregates.[7] The methodology quickly gained traction within the World Bank, as well as among National Statistical Offices around the world, and has been one of the most applied methods to obtain small area estimates. Since ELL's publication, however, there have been considerable advances in the literature, including Molina and Rao's (2010) Empirical Best (EB) method.

When deciding which small area estimation method is preferable, there are many considerations the practitioner should take into account, but ultimately, the method chosen is often driven by the data at hand. When the practitioner has access to census data and a contemporaneous household survey, it is nearly always advisable to apply a unit-level model (Chapter 4), because they often yield the most significant gains in precision. However, the data requirements for these models are considerable, and these data need to be checked to ensure the two data sources are also comparable beyond the temporal aspect. Potential variables need to be defined similarly in both data sources and must have similar distributions, as differences may lead to biased estimators. When choosing a unit-level model-based procedure, the next decision is often the aggregation level at which estimates will be reported. If there is a need for estimates at two different levels, then two-fold nested error models may be used; otherwise one-fold nested error models are adequate.

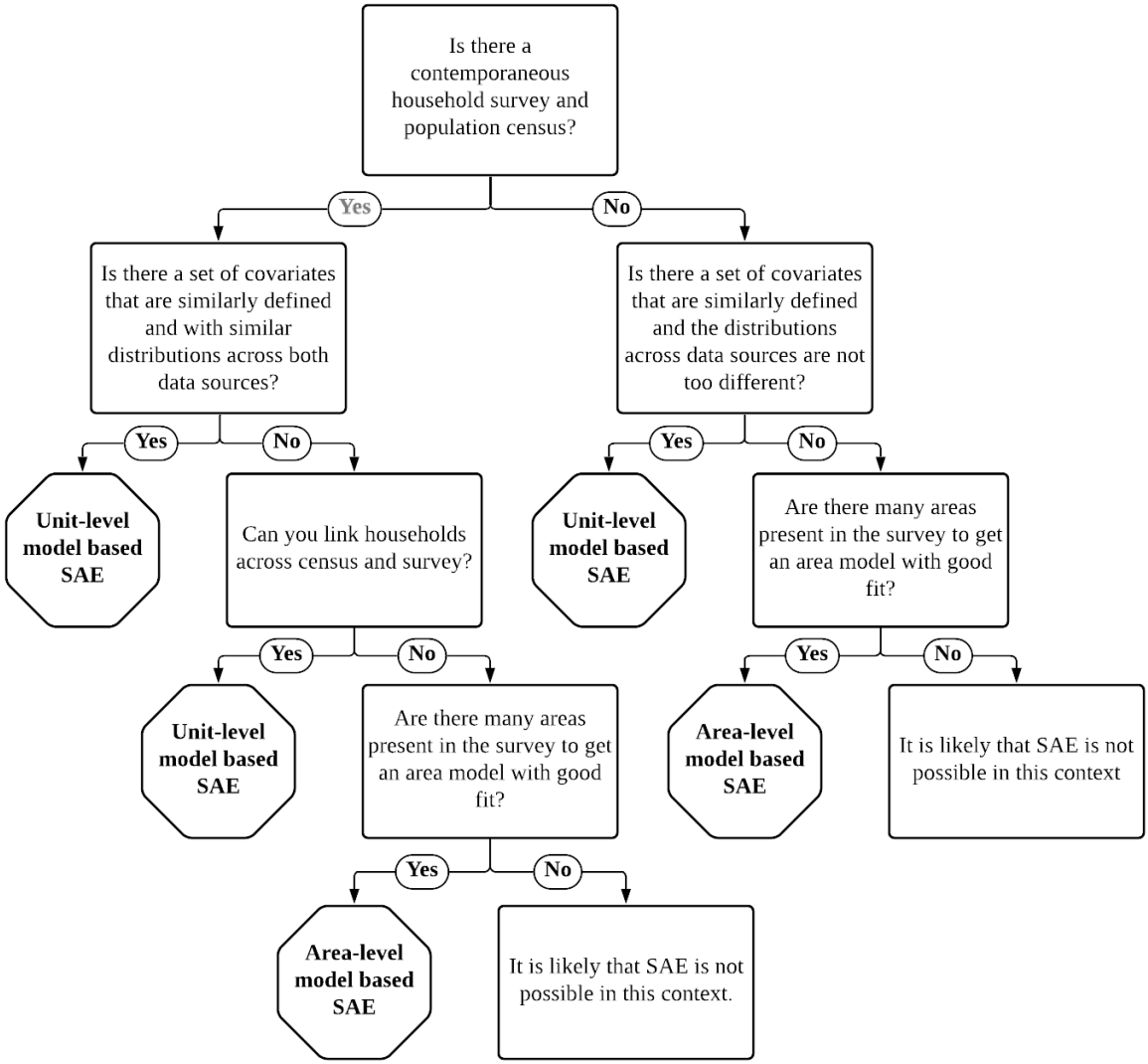
When unit-level models are not possible, area-level models may be a viable alternative (Chapter 3). Area-level models are often easier to implement than unit-level models, since data requirements are minimal and may be used when census data and survey data are not contemporaneous or incongruous. Also, area-level models rely on aggregated data, which is often not subject to confidentiality. In addition, if the most recent census is too old for a unit-level model, the recommended approach is to consider an area-level model. The most popular area-level model for small area estimation is the Fay-Herriot (FH) model, introduced by Fay and Herriot (1979), to estimate per capita income of small areas in the United States. The U.S. Census Bureau has used area-level models within the Small Area Income and Poverty Estimates (SAIPE) project to inform the allocation of federal funds to school districts. The Fay-Herriot method, discussed in the area-level model chapter, will often require a separate model for each indicator, implying that area-level covariates are linearly related to the indicator of interest. If there is data at different aggregation levels, it may be possible to implement a two-fold area-level model introduced by Torabi and Rao (2014), although no software implementation is currently available. Alternatively, it is possible to perform an FH model at the lowest aggregation level, and aggregate these FH estimates to a higher level (see section 3.4.1). Although area-level models predate ELL and in principle, should be simpler to execute, they have not been as commonly implemented by World Bank teams.

In addition to unit- and area-level models, unit-context models have been proposed in the context of SAE for poverty during off-census years (Chapter 5). Unit-context models (section 5.1) attempt to model the population's welfare distribution by using only area-level covariates. This method may be attractive because it can rely on easily accessible aggregate data and may benefit from the larger sample size obtained via household-level welfare distribution and thus appears to combine the best features of both area-level and unit-level methodologies. Nevertheless, simulated data and real-world data have shown that unit-context models yield biased poverty estimates and, therefore, based on the literature

[7] ELL follows and builds upon the work of Hentschel et al. (1998).

to date, are discouraged and not considered preferable over estimators based on FH area-level models. Additionally, unit-context models rely on a parametric bootstrap procedure to estimate Mean Square Error (MSE). Given the assumptions of the unit-context model, that household-level welfare distribution does not depend on household-specific characteristics, it is unlikely that the MSE can be accurately estimated following a parametric bootstrap, making it difficult to evaluate the resulting small area estimates properly.

Figure 1.1: SAE Decision tree



Machine learning methods have also been proposed to obtain poverty maps in off-census years or when census data is unavailable. The methods often combine household survey data with ancillary Geo-coded (publicly available and proprietary) data to present local estimates of welfare. These methods have gained popularity among academia and policy circles since the publication of Jean et al. (2016). For example, during the COVID-19 pandemic, cash transfers in Nigeria were informed by poverty estimates obtained from a machine learning approach by Chi et al. (2021). However, these methods lack an accurate estimate of the method's noise and, to date, have not been submitted to a rigorous statistical validation. The Guidelines present a brief look into how these machine learning methods compare to other available small area estimation approaches. Although much work remains to be done regarding machine learning

methods, the Guidelines posit that these methods are promising (see section 5.2).

As the reader delves into the Guidelines and cited literature, the complexity of which method to choose becomes apparent. Despite this caveat, a simple decision tree on method availability is presented in Figure 1.1. This decision tree can be used to assist practitioners in choosing which model is the best route for their context. The decision tree simplifies the process by directing readers to specific chapters in the Guidelines relevant to their particular situation.

1.2 About the Guidelines

The Guidelines are meant as a practical guide for readers. Stata scripts are provided throughout the chapters. These scripts allow replicating most sections and figures, which hopefully can assist new and more experienced practitioners in their SAE applications. Readers may download the latest Stata (SAE) packages used throughout the Guidelines from:

1. Unit-level models: https://github.com/pcorralrodas/SAE-Stata-Package
2. Area-level models: https://github.com/jpazvd/fhsae

Although R offers a wide variety of packages for poverty-related small area estimation these are described in other sources: Rao and Molina (2015), Pratesi (2016), Molina (2019), and Morales et al. (2021), for example.

To assist readers throughout the text, simulated data (created following an assumed model) as well as real world data are used. These data are used for method comparisons. Simulated data follow an assumed model and data generating process and are used for model-based simulations. Scripts are provided in the case of simulated data which can assist readers to replicate many of the analyses in the text. These data assist illustrating the application of the different methods discussed and are used to make salient the advantages and disadvantages of the methods discussed.[8] In addition to simulated data, the Guidelines use the *Mexican Intracensal survey* for design-based validations where methods are compared using real world data and the real data generating process is unknown.[9]

[8] See Tzavidis et al. (2018) for more details.

[9] The *Mexican Intracensal survey* contains a welfare measure and is modified to mimic a census of 3.9 million households and 1,865 municipalities to allow for a design-based validation. From the created census, 500 survey samples were drawn for validations shown throughout the document (see Corral et al. (2021) for more details on the modifications to the *Mexican Intracensal survey* and the sampling approach taken).

References

Battese, George E., Rachel M. Harter, and Wayne A. Fuller (1988). "An Error-Components Model for Prediction of County Crop Areas Using Survey and Satellite Data". In: *Journal of the American Statistical Association* 83.401, pp. 28–36. ISSN: 01621459. URL: http://www.jstor.org/stable/2288915.

Chi, Guanghua, Han Fang, Sourav Chatterjee, and Joshua E Blumenstock (2021). "Micro-estimates of Wealth for All Low-and Middle-income Countries". In: *arXiv preprint arXiv:2104.07761*.

Corral, Paul, Kristen Himelein, Kevin McGee, and Isabel Molina (2021). "A Map of the Poor or a Poor Map?" In: *Mathematics* 9.21. ISSN: 2227-7390. DOI: 10.3390/math9212780. URL: https://www.mdpi.com/2227-7390/9/21/2780.

Corral, Paul, Isabel Molina, and Minh Cong Nguyen (2021). "Pull Your Small Area Estimates up by the Bootstraps". In: *Journal of Statistical Computation and Simulation* 91.16, pp. 3304–3357. DOI: 10.1080/00949655.2021.1926460. URL: https://www.tandfonline.com/doi/abs/10.1080/00949655.2021.1926460.

Demombynes, Gabriel (2002). "A Manual for the Poverty and Inequality Mapper Module". In: *University of California, Berkeley and Development Research Group, the World Bank*.

Elbers, Chris, Jean O Lanjouw, and Peter Lanjouw (2003). "Micro–level Estimation of Poverty and Inequality". In: *Econometrica* 71.1, pp. 355–364.

Fay, Robert E and Roger A Herriot (1979). "Estimates of Income for Small Places: An Application of James-Stein Procedures to Census Data". In: *Journal of the American Statistical Association* 74.366a, pp. 269–277.

Foster, James, Joel Greer, and Erik Thorbecke (1984). "A Class of Decomposable Poverty Measures". In: *Econometrica: Journal of the Econometric Society* 52, pp. 761–766.

Hentschel, Jesko, Jean Olson Lanjouw, Peter Lanjouw, and Poggi Javier (1998). "Combining Census and Survey Data to Study Spatial Dimensions of Poverty". In: *World Bank Policy Research Working Paper* 1928.

Jean, Neal, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon (2016). "Combining Satellite Imagery and Machine Learning to Predict Poverty". In: *Science* 353.6301, pp. 790–794. URL: https://www.science.org/doi/abs/10.1126/science.aaf7894.

Molina, Isabel (2019). *Desagregación De Datos En Encuestas De Hogares: Metodologías De Estimación En áreas Pequeñas*. CEPAL. URL: https://repositorio.cepal.org/handle/11362/44214.

Molina, Isabel, Paul Corral, and Minh Cong Nguyen (2021). "Model-based Methods for Poverty Mapping: A Review".

Molina, Isabel and JNK Rao (2010). "Small Area Estimation of Poverty Indicators". In: *Canadian Journal of Statistics* 38.3, pp. 369–385.

Morales, Domingo, María Dolores Esteban, Agustín Pérez, and Tomáš Hobza (2021). *A Course on Small Area Estimation and Mixed Models: Methods, Theory and Applications in R*. Springer Nature.

Nguyen, Minh Cong, Paul Corral, João Pedro Azevedo, and Qinghua Zhao (2018). "sae: A Stata Package for Unit Level Small Area Estimation". In: *World Bank Policy Research Working Paper* 8630.

Pratesi, Monica (2016). *Analysis of Poverty Data by Small Area Estimation*. 1st. Wiley Series in Survey Methodology. John Wiley and Sons Ltd. ISBN: 978-1-118-81501-4. URL: https://books.google.com.gh/books?hl=en&lr=&id=TS29BgAAQBAJ&oi=fnd&pg=PR15&dq=Analysis%20of%20poverty%20data%20by%20small%20area%20estimation%C3%A2%C2%80%C2%9D.&ots=cPy8eHP_Xu&sig=Ktc_IhhF1IjHGGdpnDKSA7vivDE&redir_esc=y#v=onepage&q=Analysis%20of%20poverty%20data%20by%20small%20area%20estimation%C3%A2%C2%80%C2%9D.&f=false.

Rao, JNK and Isabel Molina (2015). *Small Area Estimation*. 2nd. John Wiley & Sons.

Robinson, George K (1991). "That BLUP Is a Good Thing: The Estimation of Random Effects". In: *Statistical science* 6.1, pp. 15–32.

Torabi, Mahmoud and JNK Rao (2014). "On Small Area Estimation under a Sub-area Level Model". In: *Journal of Multivariate Analysis* 127, pp. 36–55.

Tzavidis, Nikos, Li-Chun Zhang, Angela Luna, Timo Schmid, and Natalia Rojas-Perilla (2018). "From Start to Finish: A Framework for the Production of Small Area Official Statistics". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 181.4, pp. 927–979.

Zhao, Qinghua (2006). *User Manual for Povmap*. URL: http://siteresources.%20worldbank.%20org/INTPGI/Resources/342674-1092157888460/Zhao_%20ManualPovMap.%20pdf.

Chapter 2

Direct Estimates

To fully benefit from the Guidelines, it is necessary first to understand the background materials, including a description of the design-based setting, followed by typical direct estimators, along with the advantages and disadvantages in their applications. The goal of Chapter 2 is to present background material for the chapters that follow. The chapter begins by describing the design-based setting. Then it gives a description of the typical direct estimators and their advantages and disadvantages in their applications. Finally, some tips to evaluate estimates are provided.

Direct estimates are generated by taking a sample from the target universe of observations, surveying that sample, then using statistical techniques to generalize the results to the entire population of interest. Sample surveys provide reliable estimates of a target population without observing the entire population. Sample survey data is used to produce direct estimates of totals, means, and other characteristics for the whole population and other domains. Here, "direct" means that only the sample units from the area are used (See Lohr (2010) or Cochran (2007) for a thorough treatment of sampling theory and direct estimation).

In areas with sufficiently large sample size, statistical institutes often report direct estimates by default due to their desirable properties with respect to the sampling design (Molina 2019). Though there are some global norms, defining "sufficiently large," usually measured by the coefficient of variation, is done by the data producers. In cases where direct estimates are not sufficiently large, the results may still be reported, but the lack of quality is usually flagged for users. In instances where the desired quality is not met, small area estimates are necessary. Hence, small area estimates may be judged by how much they improve the quality of the direct estimates.

2.1 Design-Based Estimation

The traditional way to provide estimates of finite population parameters based on survey data is to rely on design-based (or repeated sampling) estimation methods. However, using a survey to produce direct estimates of adequate precision for particular domains of the population can be economically prohibitive since it may require considerably large sample sizes (Cochran 2007). Even in the case of limitless resources, huge samples may not be advisable because of a heightened risk of non-sampling or implementation error. The quality of direct design-based estimates is evaluated with respect to their sampling design, that is, with respect to all the possible samples taken from the population under the

proposed sampling design (Molina 2019). In this scenario the values of the variable of interest are considered fixed and they only vary due to the variation induced by the sampling design (*ibid*).[1]

Consider the following notation for the next sections:

- U is the target population, e.g. the set of inhabitants in a country, of size N.
- $y_1, \ldots, y_N$ are measurements of the study variable at the population units (assumed to be fixed).
- Target finite population parameter or quantity of interest: $\delta = h(y_1, \ldots, y_N)$. For example, the population mean $\bar{Y} = \frac{1}{N}\sum_{j=1}^{N} y_j$.
- s is a random sample of size n drawn from the population U according to a given survey design.
- $r = U - s$ is the set of non-sampled units (size $N - n$).
- There are D subpopulations/domains/areas, indexed by $d = 1, \ldots, D$, of sizes $N_1, \ldots, N_D$.
- The mean of a variable Y in a subpopulation/domain/area d is given by $\bar{Y}_d = N_d^{-1} \sum_{i=1}^{N_d} y_{di}$, where y_{di} denotes the value of Y for individual i in area/domain d.
- s_d is the subsample of size n_d (which may be the empty set) from domain/area d, where $\sum_{d=1}^{D} n_d = n$.
- r_d is the set of elements outside the sample of area $d = 1, \ldots, D$.
- π_{di} is the inclusion probability of individual i in the sample from area/domain d.
- $\omega_{di} = \pi_{di}^{-1}$ is the sampling weight of individual i in area/domain d.
- $\pi_{d,ij}$ is the joint inclusion probability of individuals i and j in the sample from area/domain d.

With this notation in mind, the following subsection presents the standard direct estimators, including basic direct estimators such as Horvitz-Thompson (HT), Hájek, and estimators requiring auxiliary data like GREG and calibration estimators.

2.1.1 Basic Direct Estimators

Horvitz-Thompson (HT)

The Horvitz-Thompson (HT) estimator of the **total** of area/domain d, $Y_d = \sum_{i=1}^{N_d} y_{di}$, is $\hat{Y}_d = \sum_{i \in s_d} \omega_{di} y_{di}$. This estimator is unbiased under the sampling design.

To estimate the **mean** of area/domain d, $\bar{Y}_d = N_d^{-1} \sum_{i=1}^{N_d} y_{di}$, the HT estimator is $\hat{\bar{Y}}_d = N_d^{-1} \sum_{i \in s_d} \omega_{di} y_{di}$. Note that here it is assumed that the true area's population N_d, is known.

Assuming positive inclusion probabilities π_{di} and $\pi_{d,ij}$ for every pair of units i, j in the area, an unbiased estimate of the variance is given by:

$$\widehat{\text{var}}_\pi(\hat{\bar{Y}}_d) = N_d^{-2} \left[\sum_{i \in s_d} \frac{y_{di}^2}{\pi_{di}^2}(1 - \pi_{di}) + 2 \sum_{i \in s_d} \sum_{j \in s_d, j > i} \frac{y_{di} y_{dj}}{\pi_{di}\pi_{dj}} \left(\frac{\pi_{d,ij} - \pi_{di}\pi_{dj}}{\pi_{d,ij}} \right) \right]$$

[1] Rao and Molina (2015) note that, it is possible to model population values as random instead of fixed to obtain model-dependent direct estimates. For such estimates, their inferences are based on the probability distribution induced by the model; and will typically not depend on survey weights.

One typical challenge in applications is that there is not enough information about the sampling design. Without the second-order inclusion probabilities $\pi_{d,ij}$, the formula above cannot be applied. Nevertheless, for certain sampling designs, where $\pi_{dij} \approx \pi_{di}\pi_{dj}$, for $j \neq i$, the second term of the formula approximates zero. Therefore, a simpler variance estimator, which does not depend on the second-order inclusion probabilities $\pi_{d,ij}$, would be: $\widehat{\text{var}}_\pi(\hat{\bar{Y}}_d) = N_d^{-2} \sum_{i \in s_d} \omega_{di}(w_{di} - 1)y_{di}^2$, where $\omega_{di} = \pi_{di}^{-1}$ is the sampling weight of unit i in area/domain d.

Hájek

Despite the HT estimator being unbiased under the sampling design, its variance under the design may be quite large. An alternative, although slightly biased, is Hájek's estimator. This estimator is obtained just by using the sampling weights for the considered area to estimate N_d; consequently, for estimation of $\bar{Y}_d$, it does not require knowing the population size of area d. The Hájek estimator uses $\hat{N}_d = \sum_{i \in s_d} \omega_{di}$ instead of N_d to estimate the area mean $\bar{Y}_d$. However, an unbiased estimator of its variance is obtained replacing y_{di} by $e_{di} = y_{di} - \hat{\bar{Y}}_d^{HA}$ in the variance estimator of the HT estimator.

Both HT and Hájek use only area-specific sample data, that is to say, they only use the portion of the survey data for a given area d.

Variance Estimation in Applications

Another challenge that may be encountered in applications is that, in many domains/areas d, there is only a single primary sampling unit (PSU) present in the survey data; hence, the habitual variance estimator cannot be calculated and some approximation of the variance is required.

Alternatives to approximate the variance of the estimator that depend on the sampling design and on the information available to the practitioner include linearization, a balanced repeated replication BRR method with Fay's method,[2] as well as jackknife and bootstrap techniques. Rao (1988) and Rust and Rao (1996) provide a deeper discussion on variance estimation in sample surveys and application examples. Additionally, Bruch, Münnich, and Zins (2011) provide an overview of variance estimation in the presence of complex survey designs.

2.1.2 GREG and Calibration Estimators

Unlike the basic direct estimators presented above, the Generalized Regression (GREG) and, more generally, calibration estimators, require auxiliary information, such as the population totals or means of auxiliary variables in d, $\bar{X}_d = N_d^{-1} \sum_{i=1}^{N_d} y_{di}$. If $\hat{\bar{X}}_d$ is the HT estimator of $\bar{X}_d$, then the GREG estimator of $\bar{Y}_d$ is given by: $\hat{\bar{Y}}_d + (\hat{\bar{X}}_d - \bar{X}_d,)'\hat{\boldsymbol{B}}_d$, where $\hat{\boldsymbol{B}}_d$ is the weighted least squares estimator of the coefficient vector $\boldsymbol{\beta}_d$ in linear regression of y_{di} in terms of the vector x_{di} in area d. For an exhaustive presentation of GREG estimators, see Rao and Molina (2015).

2.1.3 Pros and Cons of Direct Estimators

Data requirements:

- Sampling weights ω_{di} for survey units.
- True population size N_d of area d for HT estimator of $\bar{Y}_d$.

[2] Section 4 of Dippo, Fay, and Morganstein (1984) details the application of the method at the U.S. Census Bureau.

Some advantages and disadvantages of direct estimates are the following:

Pros:

- They make no model assumptions (non-parametric).
- HT is exactly unbiased under the sampling design for large areas. Hájek and GREG estimators are approximately unbiased under the sampling design for large areas.
- **It is an unbiased estimator under the sampling design** if the expected value (mean) of the estimator across all the possible samples s_d drawn from area d with the given design is equal to the population parameter. In other words, if its design bias is equal to zero for all the values of the parameter. The estimator $\hat{\theta}_d$ of parameter θ is unbiased if and only if $E_\pi(\hat{\theta}_d) - \theta_d = 0$.
- **Consistency:** as the sample size increases, the probability that the estimator $\hat{\theta}$ differs from the true value θ by more than ε approximates 0, for every $\varepsilon > 0$.
- **Additivity (Benchmarking property):** $\sum_{d=1}^{D} \hat{Y}_d = \hat{Y}$, that is, the direct estimate of the total for a larger area covering several areas coincides with the aggregation of the estimates of the totals for the areas within the larger area.

Cons:

- Inefficient for small areas. For an area d with small n_d, traditional area-specific direct estimators do not provide adequate precision.
- Direct estimates cannot be calculated for non-sampled domains.

2.1.4 Example: Direct Estimates of Poverty Indicators

Example based on Molina and Rao (2010) and Molina (2019).

- Target population: N households in a country.
- There are C different areas/clusters $c = 1, \ldots, C$ of sizes $N_1, \ldots, N_C$ respectively.
- Let y_{ch} be total income for each household $h = 1, \ldots, N_c$ in area/cluster $c = 1, \ldots, C$.
- Let z be a fixed poverty line.
- FGT indicators are defined as $F_{\alpha ch} = \left(\frac{z - y_{ch}}{z}\right)^{\alpha} I(y_{ch} < z)$, $h = 1, \ldots, N_c$, $\alpha = 0, 1, 2$, where the indicator function $I(\cdot)$ is 1 if household h is under the poverty line z, and 0 otherwise (Foster, Greer, and Thorbecke 1984).
- Then, for every cluster/area c, the FGT poverty indicator of order α for area c is $F_{\alpha c} = \frac{1}{N_c} \sum_{h=1}^{N_c} F_{\alpha ch}$.
- The goal is to estimate the poverty indicator $F_{\alpha c}$ for every cluster/area c; take a random sample s_c of size n_c (which might be equal to zero) from cluster/area c.
- Horvitz-Thompson estimator: $\hat{F}_{\alpha c} = \frac{1}{N_c} \sum_{h \in s_c} \omega_{ch} F_{\alpha ch}$.
- Hájek estimator: $\hat{F}_{\alpha c} = \frac{1}{\hat{N}_c} \sum_{h \in s_c} \omega_{ch} F_{\alpha ch}$, where $\hat{N}_c = \sum_{h \in s_c} \omega_{ch}$.
- Here, $\omega_{ch} = \pi_{ch}^{-1}$, where π_{ch} is the inclusion probability of household h in the sample of cluster/area c.

2.2 Evaluation of Estimates

Comparison of estimates is based on statistical measures of precision and accuracy. It is stated that estimate $\hat{\theta}$ has more **precision** than estimate $\hat{\theta}'$ if the (replicable) values of the estimate $\hat{\theta}$ are closer to each other than the ones of $\hat{\theta}'$, that is to say, if the realizations of $\hat{\theta}$ have less dispersion. On the other hand, **accuracy** measures the closeness of estimate $\hat{\theta}$, i.e. $E(\hat{\theta})$, to the true value of the population parameter θ.

For example, the estimated mean squared error (MSE) is a measure of precision (measured by the variance of the estimate $\hat{\theta}$) and accuracy (measured by the square bias of the estimate $\hat{\theta}$).

The particular error measures and "acceptable" error levels depend on National Statistical Office's guidelines or practitioners/researchers. The most common error measures of an estimator are: standard error, variance, coefficient of variation (CV), and the mean squared error (MSE). The smaller the value for all these measures, the better is the estimator.

The sections above reviewed direct estimators for means and totals. Direct estimators are obtained similarly for any other target parameter that is additive in the individual observations. Direct estimators are recommended at the national level and for disaggregated levels where direct estimators have a coefficient of variation (CV) below an established threshold for every area (Molina 2019). For disaggregations where direct estimators have a CV above the CV threshold or absolute relative bias (ARB) above a given ARB threshold, direct estimates are not recommended (see Chapters 3 and 4 for the application of indirect estimators). If there are areas where not even the indirect estimators satisfy the former requirements, it is recommended to properly highlight these estimates, indicating that these have low quality.[3]

In order to obtain a clear assessment of the precision and accuracy gains of model-based SAE methods over direct estimation, it is helpful to present the following information when reporting poverty estimates:[4]

- Area code d
- True population size of area d, N_d
- Sample size of area d, n_d
- Direct estimate of poverty for area d, $\hat{F}_{\alpha d}$
- Small area estimate of poverty for area d, for example EB $\hat{F}^{EB}_{\alpha d}$
- Estimated variance of area d, $\hat{F}_{\alpha d}$
- Estimated mean squared error (MSE) of $\hat{F}^{EB}_{\alpha d}$ in area d **or**
- Coefficient of variation (CV) of both $\hat{F}_{\alpha d}$ and $\hat{F}^{EB}_{\alpha d}$ in area d

Direct estimators have the advantage that they are (approximately) unbiased under the sampling design. When comparing two unbiased estimators, choosing the one with the smallest variance is recommended.[5] If one of them is not unbiased, it is recommended to select the one with the smallest MSE (Molina and

[3]The CV threshold can vary from scenario to scenario; some practitioners and national statistical offices consider that estimates with a CV above 20% are not reliable.

[4]See Molina and Rao (2010) for a clear example of poverty indicators report and assessment.

[5]Since $MSE_\pi(\hat{F}_{\alpha d}) = Bias^2_\pi(\hat{F}_{\alpha d}) + Var_\pi(\hat{F}_{\alpha d})$ and direct estimators are (approximately) unbiased, both MSE and variance are approximately equal. This holds for areas with large sample size n_d, for areas with small sample size, the Hájek estimator is biased.

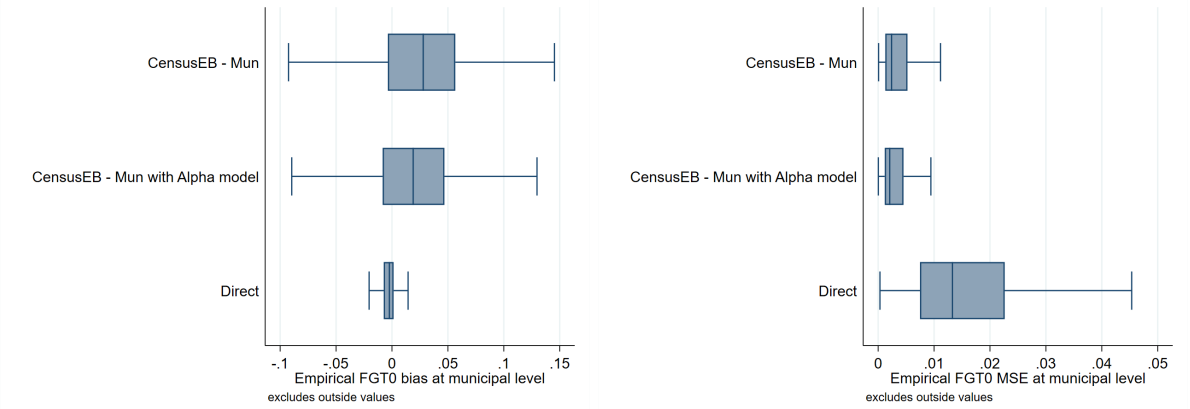
García-Portugues 2021), as it is preferable to tolerate small amounts of bias if there are large gains in precision (the bias-variance tradeoff). The model-based SAE methods presented in the subsequent chapters introduce some bias under the design while decreasing the MSE. There is, however, a limit to the amount of bias that can be tolerated before the estimators lose utility.

Why use the coefficient of variation (CV)? The CV of $\hat{\theta}$, $cv(\hat{\theta}) = \sqrt{\text{var}(\hat{\theta})}/\theta$, is a relative measure of error, which makes it unitless and easier to interpret than the MSE. Nevertheless, for the CV to have a correct interpretation, the value of the estimate must be greater than zero. Moreover, when estimating a proportion P (poverty rates, for example), larger sample sizes are often necessary to keep the error measure below a certain limit than the sample size required when estimating totals (Molina 2019).[6] Since the CV increases as the proportion P decreases, the CV is no longer a measure of the relative error but a measure of how small the proportion P is. Therefore, calculating the MSE (or its root) for proportions P is a better option.

2.2.1 Example: Bias vs MSE

Figure 2.1 shows the design bias (left) and design MSE (right) of direct estimates of FGT0 indicators (poverty rates) compared to unit-level model-based estimators, both presented at the municipal level. It is essential to notice that even though the direct estimate's bias is approximately equal to 0, its variance (or MSE) in many areas is much greater than the one from the model-based estimators.

Figure 2.1: Bias and MSE of direct estimates of the FGT0 indicators at the municipal level



Source: Based on 500 samples drawn from the *Mexican Intracensal survey* which is used as a census of 3.9 million households (see Corral et al. (2021)). Figure shows the comparison between design bias, where Hájek direct estimates are approximately unbiased, but have a large MSE. On the other hand, indirect methods such as unit-level SAE, Census EB, sacrifice design bias in pursuit of improved precision.

[6] The CV will tend to be large when the poverty rate for a given area is low since even a small variance will yield a relatively large CV. The only way to remedy such a result is by increasing the sample size.

2.3 Appendix

2.3.1 Definitions

Definitions are based on Rao and Molina (2015) and Cochran (2007).

- **Unit of analysis:** the level at which a measurement is taken. For example, persons, households, farms, firms.
- **Target population or universe:** The population of interest from which the relevant information is desired. For example, the country's population in the case where national living standards are the objective.
- **Domain/area:** domain d may be defined by geographic areas (state, county, school district, health service area) or socio-demographic groups or both (specific age-sex-race group within a large geographic area) or other subpopulations (e.g., set of firms belonging to a census division by industry group).
- **Finite population parameter θ:** quantity computed from the N measurements in the units of analysis in the population. It is often a descriptive measure of a population, such as the mean, variance, rate or proportion.
- **Estimator $\hat{\theta}$:** function of the sample data that takes values "close" to θ.
- **Direct estimator $\hat{\theta}_d$ of θ_d:** estimator based only on the domain-specific sample data. An estimator of an indicator for area d, θ_d, is **direct** if it is calculated only using data from that domain/area d, without using data from any other area. These estimators have the advantage of being (at least approximately) unbiased but tend to have low precision in areas with a small sample size.
- **Unbiased estimator under the sampling design:** If the estimator's expected value (mean) across all the possible samples is equal to the population parameter. If its design bias is equal to zero for all the values of the parameter. The estimator $\hat{\theta}$ of parameter θ is design-unbiased if and only if $E_\pi(\hat{\theta}) - \theta = 0$.
- **Estimation error:** $\hat{\theta} - \theta$, note that the estimation error for a given sample is typically different from zero even if the estimator is unbiased. The design bias is the mean estimation error across all the possible samples: $Bias_\pi(\hat{\theta}) = E_\pi(\hat{\theta}) - \theta = E_\pi(\hat{\theta} - \theta)$.
- **Mean squared (estimation) error (MSE):** $MSE(\hat{\theta}) = E[(\hat{\theta} - \theta)^2]$, note that $MSE(\hat{\theta}) = Bias^2_\pi(\hat{\theta}) + Var_\pi(\hat{\theta})$ in the design-based setup, where θ is not random. In the model-based setup, it is $MSE(\hat{\theta}) = Bias^2(\hat{\theta}) + Var(\hat{\theta} - \theta)$.
- **Standard error (SE) of $\hat{\theta}$:** is the standard deviation of the sampling distribution of the estimator $\hat{\theta}$ of parameter θ.
- **Coefficient of variation (CV) of $\hat{\theta}$:** $cv(\hat{\theta}) = \sqrt{var(\hat{\theta})}/E(\hat{\theta})$ is the associated standard error of the estimate over the expected value of the estimate ($E(\hat{\theta}) = \theta$ when the estimator is unbiased. The CV is also known as the relative standard deviation (RSD).
- **Absolute Relative Bias (ARB) of $\hat{\theta}$:** $ARB(\hat{\theta}) = |(E(\hat{\theta}) - \theta)/E(\hat{\theta})|$.

- **Consistency of $\hat{\theta}$:** as the sample size increases, the probability that the estimator $\hat{\theta}$ differs from the true value θ in more than ε approximates 0, for every $\varepsilon > 0$.
- **Simple random sampling (SRS):** Sampling approach where a sample is chosen from a larger population at random. First-order inclusion probabilities are equal across elements (Särndal, Swensson, and Wretman 2003).
  - With replacement: Elements within the population may be randomly sampled multiple times. In an ordered design sample containing information on the drawing order and the number of times an element is drawn, every ordered sample has the same selection probability (*ibid*).
  - Without replacement: Elements are selected from a population without replacement. Every sample of a fixed size has an equal probability of being selected (*ibid*).
- **Sampling proportional to size:** Method of sampling from a finite population where size measures for each unit are available. The probability of selecting a unit is proportional to its size. It is often used for multistage sampling (Skinner 2014).
- **Informative sampling:** If the sample selection probabilities are related to the outcome values (Rao and Molina 2015). This results in a set of base weights reflecting the unequal probability of being sampled. Under informative sampling, in the absence of weights, the values of the outcome variable are not representative of the population (Pfeffermann and Sverchkov 2009).
- **Benchmarking:** A practice where the direct estimator is assumed to be reliable and adjustments to the small area estimates are necessary to ensure agreement with the reliable estimator (Rao and Molina 2015). For example, the direct estimator could be the national level poverty rate and small area estimates for the areas within the country are adjusted to ensure agreement between the aggregate national poverty rate and the direct estimate of national poverty.

References

Bruch, Christian, Ralf Münnich, and Stefan Zins (2011). "Variance Estimation for Complex Surveys". In: *Deliverable D3* 1.

Cochran, William G (2007). *Sampling Techniques*. John Wiley & Sons.

Corral, Paul, Kristen Himelein, Kevin McGee, and Isabel Molina (2021). "A Map of the Poor or a Poor Map?" In: *Mathematics* 9.21. ISSN: 2227-7390. DOI: 10.3390/math9212780. URL: https://www.mdpi.com/2227-7390/9/21/2780.

Dippo, Cathryn S, Robert E Fay, and David H Morganstein (1984). "Computing variances from complex samples with replicate weights". In: *Proceedings of the Survey Research Methods Section*. American Statistical Association Alexandria, VA, pp. 489–494.

Foster, James, Joel Greer, and Erik Thorbecke (1984). "A Class of Decomposable Poverty Measures". In: *Econometrica: Journal of the Econometric Society* 52, pp. 761–766.

Lohr, Sharon L (2010). *Sampling: Design and Analysis (advanced Series)*. 2nd. Cengage Learning. ISBN: 0495105279. URL: http://dni.dali.dartmouth.edu/8vxeqmju4q43/12-bailee-klein-1/read-0495105279-sampling-design-and-analysis-advanced-series.pdf.

Molina, Isabel (2019). *Desagregación De Datos En Encuestas De Hogares: Metodologías De Estimación En áreas Pequeñas*. CEPAL. URL: https://repositorio.cepal.org/handle/11362/44214.

Molina, Isabel and Eduardo García-Portugues (2021). *A First Course on Statistical Inference*. Accessed: 2010-06-22. Bookdown.org. URL: https://bookdown.org/egarpor/inference/.

Molina, Isabel and JNK Rao (2010). "Small Area Estimation of Poverty Indicators". In: *Canadian Journal of Statistics* 38.3, pp. 369–385.

Pfeffermann, Danny and Michail Sverchkov (2009). "Inference under Informative Sampling". In: *Handbook of statistics*. Vol. 29. Elsevier, pp. 455–487.

Rao, JNK (1988). "Variance Estimation in Sample Surveys". In: *Handbook of Statistics*. Ed. by PK Krishnaiah and CR Rao. Vol. 6. Elsevier B.V. Chap. 17, pp. 427–447. URL: https://doi.org/10.1016/S0169-7161(88)06019-5.

Rao, JNK and Isabel Molina (2015). *Small Area Estimation*. 2nd. John Wiley & Sons.

Rust, Keith F and JNK Rao (1996). "Variance Estimation for Complex Surveys Using Replication Techniques". In: *Statistical methods in medical research* 5.3, pp. 283–310.

Särndal, Carl-Erik, Bengt Swensson, and Jan Wretman (2003). *Model Assisted Survey Sampling*. Springer Science & Business Media. ISBN: 0-387-40620-4. URL: https://books.google.com.gh/books?hl=en&lr=&id=ufdONK3E1TcC&oi=fnd&pg=PR5&dq=Model%20assisted%20survey%20sampling&ots=7hYW4A7HRH&sig=qkwyn4uwUZjs-RvboBiQt5RzW04&redir_esc=y#v=onepage&q=Model%20assisted%20survey%20sampling&f=false.

Skinner, Chris J (2014). "Probability Proportional to Size (PPS) Sampling". In: *Wiley StatsRef: Statistics Reference Online*, pp. 1–5.

Chapter 3

Area-level Models for Small Area Estimation

This chapter focuses on the application of the basic area-level models for small area estimation, using Fay and Herriot's (1979) model as an example. Following a brief introduction, the next section provides an example of an application of the Fay-Herriot Model for small area estimation. Then the advantages and disadvantages of the method are discussed. Finally, a technical appendix for the methodology is provided. This section draws heavily from Molina (2019) and Molina, Corral, and Nguyen (2021), and readers are advised to consult these texts for further information.

3.1 Overview

In the context of poverty, methods based on area-level models typically rely on estimating the distribution of the target area-level poverty indicator, given a set of area-level covariates based on two models – a sampling model for direct area estimates of poverty and a linking model where the true indicator is linearly related to a vector of auxiliary variables across all areas (see Technical appendix in this chapter).

In contrast to unit-level methods, these methods do not require unit record (most commonly household level) census data since they use only auxiliary information aggregated to the area level. Not typically subject to confidentiality concerns, aggregated data are usually more available. Such is often the case with population census data, where aggregated estimates for localities may be readily available while the detailed household-level data are restricted. Beyond censuses, data from other sources, such as administrative records, remote sensing data, and previous census aggregates, may also be available at the level of the areas or can be readily aggregated to that level and may be included as covariates in an area-level model. Aggregated data are also less sensitive to unit-level outliers. However, if there is high heterogeneity within the geographic units at which the area models are estimated, these benefits are balanced by a substantial loss of information when unit record level data are aggregated. Ultimately, whether unit record-level models offer a greater degree of precision depends on the explanatory power of the covariates that can be included in the area-level and unit-level models.

The most popular area-level model is the Fay-Herriot (FH) model, introduced by Fay and Herriot (1979) to estimate mean per capita income in small places in the United States of America (USA). The U.S. Census Bureau has regularly used it within the Small Area Income and Poverty Estimates (SAIPE) project to produce estimates for counties and states of the total number of people in poverty by age

groups, household median income, and mean per capita income (see Bell 1997). The U.S. Department of Education uses these estimates to determine annual allocations of federal funds given to school districts. Several other studies have employed area-level models to obtain poverty estimates small areas around the world. Molina and Morales (2009) used these models to obtain estimates for Spanish provinces, Corral and Cojocaru (2019) in Moldova, Casas-Cordero Valencia, Encina, and Lahiri (2016) in Chile, as well as Seitz (2019) which applies the FH model to obtain district-level estimates of poverty in Central Asian countries.

Extensions of the basic FH model to account for temporal and spatial correlation have also been considered in the poverty mapping context, although not explored in these Guidelines. Interested readers may refer to Esteban et al. (2012a; 2012b), who used the FH model with temporal correlation first proposed by Rao and Yu (1994) to estimate small area poverty indicators. Giusti, Masserini, and Pratesi (2017) used a spatial FH model to estimate mean income and poverty rates for the 57 Labor Local Systems of the Tuscany region in Italy for the year 2011. Marhuenda, Molina, and Morales (2013) considered a spatio-temporal model to estimate poverty indicators for Spanish provinces in 2008, making use of survey data from years 2004-2008.

3.2 Conducting Your First SAE Application with the Fay-Herriot Model

The first step towards obtaining small area estimates based on the FH model is to compute direct estimates. For the purpose of this exercise, a 1 percent SRS sample within municipalities was drawn from the *Mexican intracensal survey* of 2015 (Encuesta Intracensal). To obtain direct estimates with correctly calculated standard errors from a complex sample, it is necessary to input the structure by specifying the stratification and clustering used. Complex survey designs usually include a minimum of two clusters per stratum to permit the estimation of sampling variance using standard procedures, but as SAE reports results for areas below which the sample was originally designed, there may be areas represented in the sample by just one PSU; in such cases, the usual variance estimates cannot be computed and it may be necessary to obtain an estimate of the sampling variance via alternative methods. This is a clear limitation of the FH method.[1] Another limitation of the method is that, in locations with a small sample size, everyone may be poor or no one is poor, which leads to an estimated variance of 0. In this case, the FH model is not directly applicable in those locations unless another method is used to predict those variances. The Stata script below describes how to obtain direct estimates of poverty rates $\hat{\tau}_d^{DIR}$ and their sampling variances ψ_d for each location d in the sample:

```
clear all
set more off

/*==============================================================================
Do-file prepared for SAE Guidelines
- Real world data application
- authors Paul Corral
*==============================================================================*/

global main     "C:\Users\\`c(username)´\OneDrive\SAE Guidelines 2021\"
global section  "$main\2_area_level\"
global mdata    "$section\1_data\"
global survey   "$mdata\Survey_1p_mun.dta"
global povline = 715
```

[1] See Section 3.4.

```
version     15
set matsize 8000
set seed    648743

*===============================================================================
// End of preamble
*===============================================================================
//load in survey data
use "$survey", clear
        //Population weights and survey setup
        gen popw = Whh*hhsize

        //FGT
        forval a=0/2{
            gen fgt`a´ = (e_y<${povline})*(1-e_y/(${povline}))^`a´
        }

        //SVY set, SRS
        svyset [pw=popw], strata(HID_mun)
        //Get direct estimates
        qui:svy: proportion fgt0, over(HID_mun)
        //Extract poverty rates and  (1,865 Municipalities) - ordered from
        //smallest HID_mun to largest HID_mun
        mata: fgt0     = st_matrix("e(b)")
        mata: fgt0     = fgt0[1866..2*1865]´
        mata: fgt0_var = st_matrix("e(V)")
        mata: fgt0_var = diagonal(fgt0_var)[1866..2*1865]
        //Our survey already has the area level means of interest obtained from
        // the census. Leave data at the municipality level.
        gen num_obs = 1
        groupfunction [aw=popw], rawsum(num_obs) mean(poor fgt0) first(mun_*) by(HID_mun)
        sort HID_mun //ordered to match proportion output

        //Pull proportion´s results
        getmata dir_fgt0 = fgt0 dir_fgt0_var = fgt0_var
        replace dir_fgt0_var = . if dir_fgt0_var==0
        replace dir_fgt0     = . if missing(dir_fgt0_var)

save "$mdata\direct.dta", replace
```

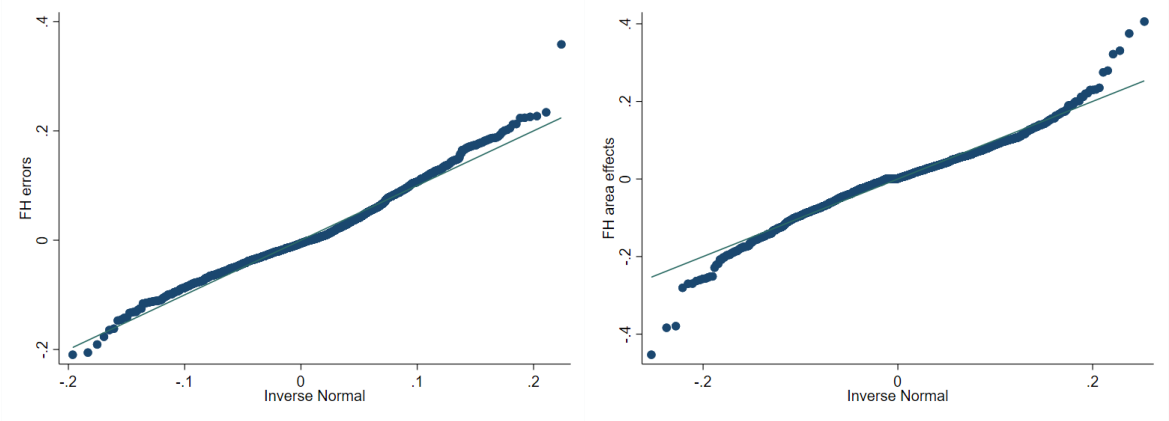
With the direct survey estimates $\hat{\tau}_d^{DIR}$ of the indicators τ_d in hand, it is possible to produce new estimates of τ_d based on the fitted model. This model takes as an input, apart from the direct estimates for each location, $\hat{\tau}_d^{DIR}$, their corresponding (heteroskedastic) sampling variances ψ_d, which are assumed known, and the aggregated values of the covariates $\mathbf{x}_d$. This information is used to estimate the FH regression coefficients β and the variance of the area effects u_d, σ_u^2. There are multiple approaches to predict the unknown random effects of the model, u_d, in Eq. 3.1. Stata's `fhsae` package (Corral et al. 2018)[2] obtains estimates similar to those provided from the `eblupFH()` function of Molina and Marhuenda's (2015) R `sae` package, while the `fayherriot` package in Stata by Halbmeier et al. (2019) is an excellent alternative since it offers transformations to the data to ensure the assumptions of the model are met. The available model-fitting methods in the `fhsae` package are maximum likelihood (ML), restricted maximum likelihood (REML), and an approach presented by Fay and Herriot (1979) based on the method of moments. As noted by Molina (2019), REML corrects the ML estimate of σ_u^2 by the degrees of freedom due to estimating the regression coefficients, β, and yields a less biased estimator in finite samples; thus REML is used in the example below.

[2] Downloadable from https://github.com/pcorralrodas/fhsae.

The final estimate of the target indicator τ_d is the empirical best linear unbiased (EBLUP) predictor of τ_d under the FH model. This estimator uses the estimated value $\hat{\sigma}_u^2$ of σ_u^2 and can be expressed as a weighted average between the direct estimator and the regression-synthetic estimator $\mathbf{x}_d'\hat{\beta}$ as noted by Rao and Molina (2015). The final EBLUP referred to as FH estimator, is given by $\hat{\tau}_d^{FH} = \hat{\gamma}_d\hat{\tau}_d^{DIR} + \left(1 - \hat{\gamma}_d\mathbf{x}_d'\hat{\beta}\right)$. In this estimator, the weight attached to the direct estimator $\hat{\tau}_d^{DIR}$ is given by $\hat{\gamma}_d = \frac{\hat{\sigma}_u^2}{\hat{\sigma}_u^2+\psi_d}$ which lies between 0 and 1 and decreases in the error variance, ψ_d. In locations d with small sample size and consequently where the direct estimator has a large sampling variance, ψ_d, or when the auxiliary information in the model yields good explanatory power, a higher weight is assigned to the corresponding synthetic estimator.

The script illustrating model selection and small area estimation based on FH is provided below. Although there are options such as stepwise or lasso regression, it is preferable to use methods that consider the random effects (Lahiri and Suntornchost 2015). The method for variable selection used in the script considers the random effects and is similar in spirit to a stepwise regression. Initially, the random effect model is fit including all the covariates. Due to lower computational requirements, the process employs the feasible generalized least squares (FGLS) method from Chandra, Sud, and Gupta (2013). Non-significant covariates are removed from the model sequentially, starting with those with the largest p-values.[3] Then, from the remaining covariates, those with a variance inflation factor (VIF) above five are similarly removed. Finally, it is also essential to check that the model's assumptions hold. In this case, the model residuals are relatively well aligned to the theoretical quantiles of a normal distribution, although some outliers are present (Figure 3.1).

Figure 3.1: Fay-Herriot model residuals (left) and predicted area effects (right)



Source: Based on a 1% SRS sample by municipality of Mexican Intracensal survey of 2015. The figure on the left presents a normal q-q plot where the empirical quantiles of the standardized residuals from the FH model are plotted against the theoretical quantiles of the normal distribution. The figure on the right plots the empirical quantiles of the predicted area effects against the theoretical quantiles of the normal distribution.

```
clear all
set more off

/*================================================================================
Do-file prepared for SAE Guidelines
- Real world data application
- authors Paul Corral
*================================================================================*/
```

[3]The removal of non-significant covariates is done using FGLS to ensure that the method's covariance matrix is considered.

```
global main     "C:\Users\\`c(username)´\OneDrive\SAE Guidelines 2021\"
global section  "$main\2_area_level\"
global mdata    "$section\1_data\"
global survey "$mdata\Survey_1p_mun.dta"
global povline = 715
//Global with eligible variables
global thevar mun_hhsize mun_age_hh mun_male_hh mun_piped_water ///
mun_no_piped_water mun_no_sewage mun_sewage_pub mun_sewage_priv ///
mun_electricity mun_telephone mun_cellphone mun_internet ///
mun_computer mun_washmachine mun_fridge mun_television mun_share_under15 ///
mun_share_elderly mun_share_adult mun_max_tertiary mun_max_secondary ///
mun_share_female

version     15
set matsize 8000
set seed    648743

*===============================================================================
// End of preamble
*===============================================================================
use "$mdata\direct.dta", clear

        //Fit full model
        fhsae dir_fgt0 $thevar, revar(dir_fgt0_var) method(fh)

        local hhvars $thevar

        //Removal of non-significant variables
        forval z= 0.8(-0.05)0.0001{
                qui:fhsae dir_fgt0 `hhvars´, revar(dir_fgt0_var) method(fh)
                mata: bb=st_matrix("e(b)")
                mata: se=sqrt(diagonal(st_matrix("e(V)")))
                mata: zvals = bb´:/se
                mata: st_matrix("min",min(abs(zvals)))
                local zv = (-min[1,1])
                if (2*normal(`zv´)<`z´) exit
                foreach x of varlist `hhvars´{
                        local hhvars1
                        qui: fhsae dir_fgt0 `hhvars´, revar(dir_fgt0_var) method(fh)
                        qui: test `x´
                        if (r(p)>`z´){
                                local hhvars1
                                foreach yy of local hhvars{
                                        if ("`yy´"=="`x´") dis ""
                                        else local hhvars1 `hhvars1´ `yy´
                                }
                        }
                        else local hhvars1 `hhvars´
                        local hhvars `hhvars1´
                }
        }

        //Global with non-significant variables removed
        global postsign `hhvars´
        //Final model without non-significant variables
        fhsae dir_fgt0 $postsign, revar(dir_fgt0_var) method(fh)
        //Check VIF
        reg dir_fgt0 $postsign, r
        gen touse = e(sample)
        gen weight = 1
        mata: ds = _f_stepvif("$postsign","weight",5,"touse")
        global postvif `vifvar´

        local hhvars $postvif
```

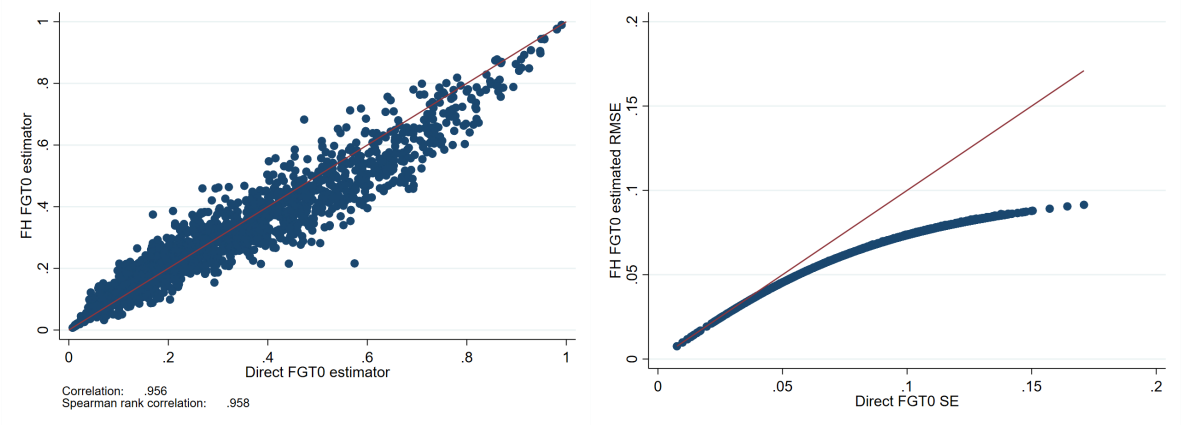
```
        //One final removal of non-significant covariates
        forval z= 0.8(-0.05)0.0001{
                qui:fhsae dir_fgt0 `hhvars´, revar(dir_fgt0_var) method(fh)
                mata: bb=st_matrix("e(b)")
                mata: se=sqrt(diagonal(st_matrix("e(V)")))
                mata: zvals = bb´:/se
                mata: st_matrix("min",min(abs(zvals)))
                local zv = (-min[1,1])
                if (2*normal(`zv´)<`z´) exit
                foreach x of varlist `hhvars´{
                        local hhvars1
                        qui: fhsae dir_fgt0 `hhvars´, revar(dir_fgt0_var) method(fh)
                        qui: test `x´
                        if (r(p)>`z´){
                                local hhvars1
                                foreach yy of local hhvars{
                                        if ("`yy´"=="`x´") dis ""
                                        else local hhvars1 `hhvars1´ `yy´
                                }
                        }
                        else local hhvars1 `hhvars´
                        local hhvars `hhvars1´
                }
        }

        global last `hhvars´

        //Obtain SAE-FH-estimates
        fhsae dir_fgt0 $last, revar(dir_fgt0_var) method(reml) fh(fh_fgt0) ///
        fhse(fh_fgt0_se) fhcv(fh_fgt0_cv) gamma(fh_fgt0_gamma) out

        //Check normal errors
        predict xb
        gen u_d = dir_fgt0 - xb

        histogram u_d

        gen e_d = dir_fgt0 - fh_fgt0

        histogram e_d

save "$mdata\direct_and_fh.dta", replace
```

If no transformation is applied in the model the FH estimates of the poverty rates may be negative for some areas. For those areas, the recommendation is to not use those negative FH estimates.[4]

As can be seen from Figure 3.2, FH estimates are aligned to direct estimates. This figure also illustrates the larger adjustment made to direct estimates in locations where the sampling variance is larger. The resulting FH estimates also have a lower estimated root mean squared error (RMSE) than that of the direct estimates, suggesting an improved efficiency of the model-based estimates (Figure 3.2, right).

[4]This decision must be made with care and usually in consultation with colleagues from the relevant statistical agency.

Figure 3.2: FH poverty estimates compared to direct estimates



Source: Based on a 1% SRS sample by municipality of Mexican Intracensal survey of 2015. The figure on the left presents a scatter plot of direct estimates on the X-axis and FH estimates on the Y-axis. The FH approach will often yield estimates that are highly aligned to the direct estimates.
The figure on the right illustrates the gains in terms of root mean squared error for the FH estimates (Y-axis) versus the standard error of the direct estimates (X-axis). Gains are often larger for the areas with smaller sample sizes (larger sampling errors of direct estimates), and though most if not all areas typically see some gains in precision.

3.3 Pros and Cons of Fay-Herriot Models

The information in this section borrows heavily from Molina (2019) while adding more information on relevant aspects.

Model requirements:

1. Direct estimates of indicators of interest and its sampling variance for the areas considered, $\hat{\tau}_d^{DIR}$ and $\hat{\psi}_d$ respectively (from the survey).

2. Aggregate data at area level of all necessary covariates for the model for every area considered, $\mathbf{x}_d$, $d = 1, \ldots, D$.

Pros:

1. Tend to improve the direct estimates in terms of efficiency (MSE).

2. The regression model incorporates unexplained heterogeneity between areas through the area effects.

3. The FH estimator is a weighted average of the direct estimator and the regression synthetic estimator (i.e., obtained by a linear regression). For areas where the sample size is small or where direct estimators have considerable sampling variance, the synthetic estimator will be given a higher weight. Accordingly, the FH estimate will approximate the direct estimate in areas with a sufficient sample .

4. Since the FH estimates make use of the direct estimates, which consider sampling weights with γ_d tending to one as the area sample size increases, it is consistent under the sampling design. Hence, provided that the sampling weights are accurate, FH estimates will be less affected by informative sampling (i.e., when the selection of the units for the sample depends on the outcome values - Rao and Molina 2015).

5. Since area-level aggregates are used, the final estimates are less affected by outliers.
6. Can provide estimates for non-sampled areas based on the regression-synthetic component.
7. The Prasad-Rao MSE estimates (Prasad and Rao 1990) are approximately unbiased. This holds as long as a sufficient number of areas are present and model assumptions are met, with normality for random effects and model errors.
8. It is easy and computationally fast to implement; considerably faster than unit-level models because the number of areas is typically much smaller than the actual sample size in typical unit-level models, and Monte Carlo or bootstrap methods are not necessary.
9. Does not require comparison of auxiliary variables between the different data sources to consider their use in the model.
10. Because aggregate data are used, confidentiality restrictions do not hinder the data acquisition process.

Cons:

1. Estimates are based on a model; thus, the model needs to be checked. For non-linear parameters (like poverty), the linear assumption of the model can be problematic.
2. The sampling variances of the direct estimates are assumed to be known, and the error of estimating these is not usually incorporated in the final estimates' estimated MSE.
3. The Prasad-Rao MSE estimator (Prasad and Rao 1990) is approximately unbiased under the model with normality but is not design unbiased for a given area.
4. The model is fit only on sampled areas, which can be a very small number out of the total number of areas using only one observation by area (the direct estimator). As a consequence, model parameter estimators will be much less efficient than those obtained under unit-level models, and hence gains in precision are expected to be lower than those of estimators based on unit-level models.
5. Every indicator requires its own model.
6. Once estimates are obtained, these cannot be further disaggregated unless a new model at that level is constructed.
7. May require benchmarking adjustment to match direct estimates at higher aggregation levels. Although sub-area-level models by Torabi and Rao (2014) could be used to obtain estimates that naturally match at two different aggregation levels (different from the national level), user-friendly software is still not available.

3.3.1 Alternative Software Packages for Area-Level Models and Extensions

In this section, the focus has been on Stata's `fhsae` package. However, there is a wide variety of packages available in R for small area estimation based on the basic FH area-level model and on some extensions of this model proposed in the literature. Below is a list of some of these extensions of the FH model and the corresponding software implementation:[5]

[5]The discussion in this section borrows from Molina et al. (2021), as well as from Harmening et al. (2021).

1. The extension of the FH model proposed by Rao and Yu (1994) that incorporates temporal correlation is implemented in the `saery` package in R by Lefler, Gonzalez, and Martin (2014). R's `sae` package by Molina and Marhuenda (2015) includes a FH model with temporal and spatial correlation. Spatial correlation is also included in R's `emdi` package (Kreutzmann et al. 2019).

   (a) Robust alternatives to the basic FH model, as well as its extensions, including spatial and temporal correlation, are detailed in Warnholz (2016) and implemented in R's `saeRobust` package (Warnholz 2018). The robust spatial correlation model is also available in the `emdi` package in R (Kreutzmann et al. 2019).

2. In applications with several dependent target indicators, multivariate versions of the FH model may help provide even more efficient estimators. A multivariate version of the FH model was originally proposed by Fay (1987), and has been used to improve the estimates of median income of four-person families by using direct estimators of median income for three and five-person families by Datta, Fay, and Ghosh (1991) and Datta et al. (1996). Multivariate FH models can be fitted with the R package `msae` (Permatasari and Ubaidillah 2020).

3. The basic FH model assumes that covariates are population aggregates measured without error. For cases where auxiliary variables are measured with error, Ybarra and Lohr (2008) offer an extension to the basic FH model to account for the measurement error. Measurement error models can be applied with the R package `saeME` (Mubarak and Ubaidillah 2020). This extension is also available in the R package `emdi` (Kreutzmann et al. 2019).

4. Hierarchical Bayesian methods have been applied in the `hbsae` R package by Boonstra (2015). The `BayesSAE` R package (Shi 2018) also includes Bayesian extensions.

5. Halbmeier et al. (2019) introduced the `fayherriot` Stata command, which allows for adjusting non-positive random effect variance estimates, and allows for transformation of the variable of interest to deal with violations of the model's assumptions. Transformation is also available in the `emdi` package in R (Kreutzmann et al. 2019).

3.4 Technical Annex

The FH model is defined in two stages.[6] In the first stage, the true values of the indicators τ_d for all the areas $d = 1, \ldots, D$, are linked by establishing a common linear regression model, called the *linking model*, where the true indicator is linearly related to a vector of area-level covariates $\mathbf{x}_d$,

$$\tau_d = \mathbf{x}_d'\beta + u_d. \tag{3.1}$$

The random errors in this regression model, u_d, are called area effects, because they represent the unexplained between-area heterogeneity. They are assumed to satisfy usual regression assumptions, such as having zero mean and constant variance σ_u^2, which is typically unknown. Note that the linear regression model (3.1) cannot be fitted to the data, because true values of the indicators τ_d are not observed. Direct estimators $\hat{\tau}_d^{DIR}$ of the indicators are available from the survey microdata but are, obviously, subject to survey error, since they are calculated only with the area-specific survey data, which is of small size. Hence, in the second stage, this survey error is modeled by assuming that the direct estimators are centered around the true values of the indicators, as follows:

$$\hat{\tau}_d^{DIR} = \tau_d + e_d, \tag{3.2}$$

with heteroscedastic errors, where error variances are $\text{var}\,(e_d|\tau_d) = \psi_d$, $d = 1, \ldots, D$, which is called the *sampling model*. Note that the D error variances ψ_d, $d = 1, \ldots, D$, need to be assumed as known even if they are not; otherwise, there would be more unknown parameters than the available observations: $\hat{\tau}_d^{DIR}$, $d = 1, \ldots, D$. These error variances are customarily estimated based on the survey microdata from the area of interest and perhaps smoothed afterward since these estimates are highly unstable due to the small area sample sizes.

The usual small area estimator obtained from this model is based on the *best linear unbiased predictor* (BLUP), which is defined as the linear function $\hat{\tau}_d = \alpha'\mathbf{y}$ of the data $\mathbf{y} = (\hat{\tau}_1^{DIR}, \ldots, \hat{\tau}_D^{DIR})'$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_D)'$, which is unbiased under the model, that is, $E(\hat{\tau}_d - \tau_d) = 0$, and is optimal (or *best*) in the sense of minimizing the mean squared error (MSE). The resulting BLUP $\hat{\tau}_d$ of τ_d can be expressed as a weighted average of the direct estimator $\hat{\tau}_d^{DIR}$ and the regression-synthetic estimator, $\mathbf{x}_d'\hat{\beta}$, where the weight $\gamma_d = \sigma_u^2/(\sigma_u^2 + \psi_d)$ attached to the direct estimator grows as the error variances ψ_d decrease (or the area sample size increases), or the unexplained variation σ_u^2 grows. This means that, in areas with a large sample size or when the auxiliary information is not very powerful, the FH estimates come close to the direct estimators; in the opposite case, they come close to the regression-synthetic estimators. This leads to the BLUP automatically borrowing strength only when it is actually needed and approximating to the conventional survey direct estimators when the area sample size is large or the model is not useful.

The BLUP does not require normality assumptions on the model errors or area effects, and it is unbiased under the model. The *empirical* BLUP (EBLUP) is obtained replacing the unknown area effects' variance σ_u^2 by an estimator $\hat{\sigma}_u^2$ with good properties (i.e. consistent). The EBLUP preserves unbiasedness under symmetric distributions and under certain regularity assumptions on the considered estimator $\hat{\sigma}_u^2$ of σ_u^2 (and a translation invariant function of the sample data $\mathbf{y}$), which are satisfied by the usual estimators obtained by ML, REML, or Fay-Herriot's method of moments. However, their exact mean squared error (MSE) does not have a closed form. For this reason, analytical approximations with good properties for **large number of areas** D have been obtained under normality. Similarly, an analytical estimator of the

[6] This section draws heavily from Molina, Corral and Nguyen (2021) as well as Rao and Molina (2015)

true MSE that is nearly unbiased for large D has been obtained under normality. This MSE estimator depends on the actual estimator of σ_u^2 that is applied; for the exact formulas, see Rao and Molina (2015), Section 6.2.1. For areas with moderate or large sample sizes and indicators (and corresponding direct estimators) defined in terms of sums over the area units, the normality assumption is minimally ensured by the Central Limit Theorem. However, for areas with small sample sizes or when target indicators are not defined in terms of sums over the area units, the normality assumption should be validated if these customary MSE estimates are considered; otherwise, they should be interpreted with care.

A problem that may occur when applying FH models is that the estimated model variance σ_u^2 may be non-positive, and then it is customary to set it to zero. In these instances, the weight γ_d is equal to 0 for all areas, implying a weight of zero to direct estimates regardless of the area's sample size. Hence, the EBLUP reduces to the synthetic estimator, which may be considerably different to the direct estimator (Rao and Molina 2015). Li and Lahiri (2010) and Yoshimori and Lahiri (2014) propose adjustments to the maximum likelihood approach to obtain strictly positive estimates. These adjustments are incorporated into the `emdi` package in R (Kreutzmann et al. 2019).

A clear drawback of the FH model is that the sampling variances of direct estimators, ψ_d, $d = 1, \ldots, D$, are assumed to be known even if they are actually estimated,[7] and usual MSE estimators do not account for the uncertainty due to estimation of these variances. Note also that, if one wishes to estimate several indicators, a different model needs to be constructed for each indicator, even if they are all based on the same welfare variable. This means that one needs to find area-level covariates that are linearly related to each indicator. It seems more reasonable to fit a model for the considered welfare variable at the unit-level and then estimate all the target monetary indicators based on that same model, as is done in the procedures based on unit-level models described in Chapter 4.

Moreover, the area-level auxiliary information commonly has the shape of area means or proportions. For indicators with a more complex shape than the simple mean welfare, it might be difficult to find area-level covariates that are linearly related with the complex function of welfare for the area; consequently, the linearity assumption in the FH model might fail, and finding an adequate transformation for linearity to hold might not be easy in some applications. In any case, even if the gains obtained using the FH model depend on the explanatory power of the available covariates, Molina and Morales (2009) observed very mild gains of the EBLUPs of poverty rates and gaps based on the FH model in comparison to those obtained by Molina and Rao (2010) using the same data sources, but relying on unit-level models (and using the microdata). When considering indicators, such as the FGT poverty indicators, an additional drawback of the FH model is that due to the small area sample sizes, direct estimators that are equal to zero or one are possible (when no one or everyone is found to be below the poverty line), even if the area sample size is not zero. In those areas, the sampling variances of the direct estimators are also zero and unless a different approach is applied to obtain strictly positive variances, the FH Model is not applicable for those areas. A common approach to correct for this is to remove those areas in the modeling phase and then use a synthetic estimator for that area instead.

3.4.1 Aggregation to Higher Geographic Areas

Even if the area-level model may be applied to produce poverty estimates at a given level of aggregation, one may also wish to aggregate these estimates to higher aggregation levels (let us call these larger areas "regions"). While the estimate of the poverty rate for a region is simply the population weighted average of the area-level poverty rates contained within the region, obtaining a correct MSE estimate for

[7] See Chapter 2 for how these can be estimated

the estimator of that region requires aggregating the estimates of the MSEs and mean crossed product errors (MCPEs) at the area level, d, up to the regional level, r. For a region, r, containing D_r areas, the poverty rate is then $\tau_r = \frac{1}{N_r}\sum_{d=1}^{D_r} N_d\tau_d$, where τ_d is the poverty rate in area d, from region r, N_d is the population of domain d in region r, and $N_r = \sum_{d=1}^{D_r} N_d$ the population of the region. Then a model-based estimate of the poverty rate for region r is $\hat{\tau}_r = \frac{1}{N_r}\sum_{d=1}^{D_r} N_d\hat{\tau}_d$, and the MSE of $\hat{\tau}_r$ is given by:

$$MSE(\hat{\tau}_r) = \frac{1}{N_r^2}\sum_{d=1}^{D_r}\sum_{l=1}^{D_r} N_d N_l MCPE(\hat{\tau}_d,\ \hat{\tau}_l)$$

where $MCPE\left(\hat{\tau}_d,\ \hat{\tau}_l\right) = E\left[\left(\hat{\tau}_d - \tau_d\right)\left(\hat{\tau}_l - \tau_l\right)\right]$ (Rao and Molina 2015, p144). The `fhsae` package in Stata has the option to apply such an approach via its `aggarea()` option. The method is also available in the `hbsae` R package by Boonstra (2015).

References

Bell, W (1997). "Models for County and State Poverty Estimates". In: *Preprint, Statistical Research Division, US Census Bureau*.

Boonstra, Harm Jan (2015). "Package 'hbsae'". In: *R Package Version* 1.

Casas-Cordero Valencia, Carolina, Jenny Encina, and Partha Lahiri (2016). "Poverty Mapping for the Chilean Comunas". In: *Analysis of Poverty Data by Small Area Estimation*, pp. 379–404.

Chandra, Hukum, UC Sud, and VK Gupta (2013). "Small Area Estimation under Area Level Model Using R Software". In: *New Delhi: Indian Agricultural Statistics Research Institute*.

Corral, Paul and Alexandru Cojocaru (2019). "Moldova Poverty Map: An Application of Small Area Estimation."

Corral, Paul, William Seitz, João Pedro Azevedo, and Minh Cong Nguyen (2018). *fhsae: Stata module to fit an area level Fay-Herriot model. Statistical Software Components*. URL: https://econpapers.repec.org/software/bocbocode/s458495.htm.

Datta, Gauri S, RE Fay, and M Ghosh (1991). "Hierarchical and Empirical Multivariate Bayes Analysis in Small Area Estimation". In: *Proceedings of bureau of the census 1991 annual research conference, US Department of Commerce, Bureau of the Census*. Vol. 7, pp. 63–79.

Datta, GS, Malay Ghosh, Narinder Nangia, and K Natarajan (1996). "Estimation of Median Income of Four-person Families: A Bayesian Approach". In: *Bayesian analysis in statistics and econometrics: essays in honor of Arnold Zellner*. Ed. by Donald A Berry Kathryn Chaloner John and Geweke Arnold Zellner. John Wiley & Sons, pp. 129–140. ISBN: 0-471-11856-7. URL: https://dialnet.unirioja.es/servlet/articulo?codigo=3418625.

Esteban, María Dolores, Domingo Morales, A Pérez, and Laureano Santamaría (2012a). "Small Area Estimation of Poverty Proportions under Area-level Time Models". In: *Computational Statistics & Data Analysis* 56.10, pp. 2840–2855. URL: https://www.sciencedirect.com/science/article/abs/pii/S0167947311003720.

Esteban, MD, D Morales, A Pérez, and L Santamaría (2012b). "Two Area-level Time Models for Estimating Small Area Poverty Indicators". In: *Journal of the Indian Society of Agricultural Statistics* 66.1, pp. 75–89.

Fay, Robert E and Roger A Herriot (1979). "Estimates of Income for Small Places: An Application of James-Stein Procedures to Census Data". In: *Journal of the American Statistical Association* 74.366a, pp. 269–277.

Fay, Robert E. (1987). "Application of Multivariate Regression to Small Domain Estimation". In: *Small area statistics: An international symposium*. Ed. by Platek, Richard and Rao, Jon N. K. and Singh, M. P. and Särndal, Carl-Erik and others. Vol. 213. John Wiley & Sons Incorporated, pp. 91–102.

Giusti, Caterina, Lucio Masserini, and Monica Pratesi (2017). "Local Comparisons of Small Area Estimates of Poverty: An Application within the Tuscany Region in Italy". In: *Social Indicators Research* 131.1, pp. 235–254.

Halbmeier, Christoph, Ann-Kristin Kreutzmann, Timo Schmid, and Carsten Schröder (2019). "The fayherriot Command for Estimating Small-area Indicators". In: *The Stata Journal* 19.3, pp. 626–644.

Harmening, Sylvia, Ann-Kristin Kreutzmann, Sören Pannier, Nicola Salvati, and Timo Schmid (2021). *A Framework for Producing Small Area Estimates Based on Area-Level Models in R*. R Package emdi. URL: https://mirror.las.iastate.edu/CRAN/web/packages/emdi/vignettes/vignette_fh.pdf.

Kreutzmann, Ann-Kristin, Sören Pannier, Natalia Rojas-Perilla, Timo Schmid, Matthias Templ, and Nikos Tzavidis (2019). "The R Package emdi for Estimating and Mapping Regionally Disaggregated Indicators". In: *Journal of Statistical Software* 91.7, pp. 1–33. DOI: 10.18637/jss.v091.i07. URL: https://www.jstatsoft.org/index.php/jss/article/view/v091i07.

Lahiri, Partha and Jiraphan Suntornchost (2015). "Variable Selection for Linear Mixed Models with Applications in Small Area Estimation". In: *Sankhya B* 77.2, pp. 312–320.

Lefler, M, D Gonzalez, and A Martin (2014). "saery: Small Area Estimation for Rao and Yu Model". In: *R package version* 1.

Li, Huilin and Partha Lahiri (2010). "An Adjusted Maximum Likelihood Method for Solving Small Area Estimation Problems". In: *Journal of multivariate analysis* 101.4, pp. 882–892.

Marhuenda, Yolanda, Isabel Molina, and Domingo Morales (2013). "Small area estimation with spatio-temporal Fay–Herriot models". In: *Computational Statistics & Data Analysis* 58, pp. 308–325.

Molina, Isabel (2019). *Desagregación De Datos En Encuestas De Hogares: Metodologías De Estimación En áreas Pequeñas*. CEPAL. URL: https://repositorio.cepal.org/handle/11362/44214.

Molina, Isabel, Paul Corral, and Minh Cong Nguyen (2021). "Model-based Methods for Poverty Mapping: A Review".

Molina, Isabel and Yolanda Marhuenda (2015). "Sae: An R Package for Small Area Estimation". In: *The R Journal* 7.1, pp. 81–98.

Molina, Isabel and Domingo Morales (2009). "Small Area Estimation of Poverty Indicators". In: *Estadistica e Investigacion Operativa (SEIO)* 25.3, pp. 218–225. URL: https://emis.dsd.sztaki.hu/journals/BEIO/files/BEIOVol25Num3_IMolina+DMorales.pdf.

Molina, Isabel and JNK Rao (2010). "Small Area Estimation of Poverty Indicators". In: *Canadian Journal of Statistics* 38.3, pp. 369–385.

Mubarak, M and A Ubaidillah (2020). "saeME: Small Area Estimation with Measurement Error". In: *R package version 1.2.4*. URL: https://CRAN.R-project.org/package=saeME.

Permatasari, N and A Ubaidillah (2020). "msae: Multivariate Fay–Herriot Models for Small Area Estimation". In: *R package version 0.1* 1.

Prasad, N. G. N. and J. N. K. Rao (1990). "The Estimation of the Mean Squared Error of Small-area Estimators". In: *Journal of the American Statistical Association* 85.409, pp. 163–171. ISSN: 01621459. URL: http://www.jstor.org/stable/2289539.

Rao, JNK and Isabel Molina (2015). *Small Area Estimation*. 2nd. John Wiley & Sons.

Rao, Jon NK and Mingyu Yu (1994). "Small-area Estimation by Combining Time-series and Cross-sectional Data". In: *Canadian Journal of Statistics* 22.4, pp. 511–528.

Seitz, William Hutchins (2019). "Where They Live: District-level Measures of Poverty, Average Consumption, and the Middle Class in Central Asia". In: *World Bank Policy Research Working Paper* 8940.

Shi, C (2018). "BayesSAE: Bayesian Analysis of Small Area Estimation". In: *R package version 1.0-2, URL https://CRAN.R-project.org/package=BayesSAE*.

Torabi, Mahmoud and JNK Rao (2014). "On Small Area Estimation under a Sub-area Level Model". In: *Journal of Multivariate Analysis* 127, pp. 36–55.

Warnholz, S (2018). *saeRobust: Robust Small Area Estimation*. URL: https://CRAN.%20R-project.%20org/package=%20saeRobust.%20R%20package%20version%200.2.%200.

Warnholz, Sebastian (2016). "Small Area Estimation Using Robust Extensions to Area Level Models: Theory, Implementation and Simulation Studies". PhD thesis. Freie Universität Berlin. URL: http://dx.doi.org/10.17169/refubium-13904.

Ybarra, Lynn MR and Sharon L Lohr (2008). "Small Area Estimation When Auxiliary Information Is Measured with Error". In: *Biometrika* 95.4, pp. 919–931.

Yoshimori, Masayo and Partha Lahiri (2014). "A New Adjusted Maximum Likelihood Method for the Fay–herriot Small Area Model". In: *Journal of Multivariate Analysis* 124, pp. 281–294.

Chapter 4

Unit-level Models for Small Area Estimation

This chapter focuses on unit-level models for small area estimation of poverty, which typically rely on estimating the distribution of the household's welfare, given a set of auxiliary variables or correlates.[1] The methods presented in this chapter use model-based techniques that "borrow strength" from other areas by using a larger data set with auxiliary information – usually the population census. The resulting indirect estimators sacrifice design bias, but often yield more precise estimates in terms of mean-squared errors (MSE). In software implementations of the method described in this chapter, model parameters are used to simulate multiple welfare vectors from the fitted distribution for all households in the census. This is done because the census often lacks a welfare measure for poverty measurement (Elbers et al. 2007). From the simulated census vectors, it is possible to obtain poverty rates or any other welfare indicator, for every area, including the non-sampled ones. This chapter serves as a guide for the production of unit-level small area estimates of poverty indicators, focusing on two of the most popular approaches – the one proposed by Elbers, Lanjouw, and Lanjouw (2003, ELL), and the Empirical Best (EB) approach by Molina and Rao (2010, MR).[2] It presents evidence on certain aspects of SAE and recommendations on what may be the preferred approach in some scenarios and offers advice for aspects where perhaps more research is needed.

The chapter begins by presenting some of the prerequisites that should be satisfied before commencing a SAE exercise. It provides insights that may help practitioners choose a unit-level SAE method that is adequate for their needs. It then presents considerations that practitioners may keep in mind when conducting SAE, such as data transformation and why ELL has fallen out of favor with many practitioners. Additionally, it presents the usual steps towards selecting a model and the production of small area estimates. The sections rely on real world and simulated data, and offer codes which readers may use to replicate some of the analysis; in other instances it pulls evidence from recent research.[3]

[1] This section draws from the background papers for these guidelines. See Corral, Molina, and Nguyen (2021) and Corral et al. (2021).

[2] A factor that likely contributed to the expansion of these methods is the availability of software, which can be easily used to apply these approaches; `PovMap` (Zhao 2006) for ELL and R's `sae` package (Molina and Marhuenda 2015). While other methods exist, for the purposes of this section, attention is given to the methods noted.

[3] It is highly recommended that novice practitioners start by reading section 4.4.1.

4.1 Preparing for Unit-Level Small Area Estimation

The basic ingredients for model-based unit-level small area estimation for poverty are a contemporaneous survey and census – ideally from the same year. Small area estimates rely on household-level data to obtain the conditional distribution of welfare given the observed correlates. This information is then used to generate welfare for all the census units. Thus, if characteristics in the two datasets differ considerably, or if the structure of the conditional distribution has changed between the two data sources, then the estimates could be biased and of little use.

Obtaining a pool of candidate variables for the analysis requires a thorough review of the survey and census data to ensure comparability. In the census and survey questionnaires, questions should be asked in a similarly, ideally worded in the same manner. Checks should then be conducted on the survey weights to understand how these are constructed. For example, in some countries where there is heavy migratory work, household expansion factors (usually the number of household members) may be adjusted by the number of days the individual is present in the dwelling. Such adjustments should also be possible in the census data; otherwise, the poverty estimates obtained from SAE methods will not be comparable to those obtained directly from the survey. Even if questions are asked in a similar fashion and weights are replicable, differences may still arise. Teams are strongly recommended to compare the mean and distribution the potential pool of covariates across the survey and the census – including whether it is possible to reproduce the direct estimates of poverty produced by the National Statistical Office. Ideally, the mean and distribution of the covariates should be comparable, at least at the aggregation level at which the survey is designed to be representative.

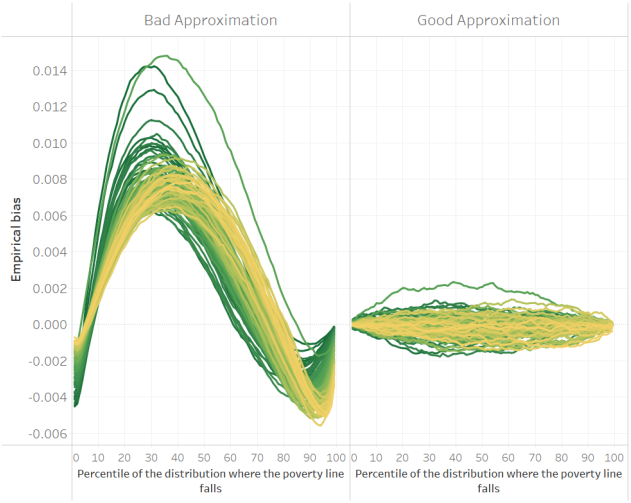
One of the first decisions that must be made is the level at which estimates must be produced. As noted afterward, this determines the level at which location effects are specified in the model in unit-level models. For reasons that will be apparent in the following sections, the locations in the survey must be matched to the locations in the census. Matching locations will also ensure that when using information from external sources, for example, administrative data or satellite imagery-derived data, these correspond to the correct location. Additionally, the use of hierarchical identifiers for the location variable is recommended (Zhao 2006). Under Stata's `sae` package, these identifiers must be numeric and no longer than 16 digits,[4] and should have an equal number of digits for every area. For example, in the hierarchical identifier SSSMMMMPPP, the first 3 digits represent states (S), the next 4 digits represent municipalities (M), and the final 3 digits represent PSUs (P). The use of hierarchical identifiers is necessary for two-fold nested error models. In the code syntax, the user needs to indicate the number of digits to remove (from the right) to arrive at the larger area's identifier. For example, to go from PSU to municipality level, it is necessary to remove 3 digits, yielding SSSMMMM.

4.1.1 The Assumed Model

The model based small area estimation methods described in this chapter are dependent on an assumed model. The nested error model used for small area estimation by ELL (2003) and MR (2010) was originally proposed by Battese, Harter and Fuller (1988) to produce county-level corn and soybean crop area estimates for the American state of Iowa. For the estimation of poverty and welfare, the ELL and MR methods assume that the transformed welfare y_{ch} for each household h within each location c in the population is linearly related to a $1 \times K$ vector of characteristics (or correlates) x_{ch} for that household,

[4] Stata's double-precision only goes up to 16 digits.

Figure 4.1: Empirical bias of two different models for unit-level small area estimation



Source: Simulation based on 1,000 populations generated as described in this section 5.5.1.1. Each line corresponds to one of the 100 areas of the population. The x-axis represents the percentile on which the poverty line falls on, and the y-axis is the empirical bias. The do-file to replicate these results can be found in 5.5.2.

according to the nested error model:

$$y_{ch} = x_{ch}\beta + \eta_c + e_{ch},\ h = 1, \ldots, N_c,\ c = 1, \ldots, C, \tag{4.1}$$

where η_c and e_{ch} are respectively location and household-specific idiosyncratic errors, assumed to be independent from each other, following:

$$\eta_c \overset{iid}{\sim} N\left(0, \sigma_\eta^2\right),\ e_{ch} \overset{iid}{\sim} N\left(0, \sigma_e^2\right)$$

where the variances σ_η^2 and σ_e^2 are unknown. Here, C is the number of locations in which the population is divided and N_c is the number of households in location c, for $c = 1, \ldots, C$. Finally, β is the $K \times 1$ vector of regression coefficients.[5]

One of the main assumptions is that errors are normally distributed. The assumption does not necessarily imply that the transformed welfare (the dependent variable) is normally distributed but instead implies that conditional on the observed characteristics the residuals are normally distributed. Under the assumed model, variation in welfare across the population is determined by three components: the variation in household characteristics, the variation in household-specific unobservables, and the variation in location-specific effects. Within any given area, the welfare distribution is determined by the variation in household specific characteristics and household specific errors.

It is important to note that while the method's goal may be to produce headcount poverty rates or other welfare indicators, these are not explicitly modeled in the sense that the dependent variable is welfare (or transformed welfare) and not the desired indicator (e.g. headcount poverty). The methodology relies on approximating as closely as possible the welfare distribution in each area. A model that approximates the welfare distribution poorly may yield a decent estimate for a given poverty line, yet when judged at a different poverty line the method may completely fail. This can be seen in Figure 4.1, where the headcount poverty estimates from the model on the left may work relatively well if the poverty threshold were set at the 70th percentile and yield completely biased estimates at most other thresholds. On the

[5] Additional information is provided in the technical annex 4.4.

other hand, the model on the right approximates the welfare distribution well in all areas and yields poverty estimates with a small bias across all poverty thresholds and areas.

4.2 Conducting Your First SAE Application with a Unit-Level Model

The focus of this section is on the process of producing small area estimates of monetary indicators using a census and a survey. For the exercise conducted in the following sections it is assumed that the data has already been prepared and all of the checks noted in section 4.1 have been conducted. The section provides step-by-step processes for producing small area estimates of poverty using census and survey data. It also draws from existing evidence to make recommendations based on current best practices for conducting SAE applications.

4.2.1 Model Choice Considerations

Ideally, the particular model to be used should be aligned to the specific SAE goals. One of the main considerations regarding the model to be used is the level at which poverty estimates will be reported. This will determine the level at which the random location effect should be specified (see section 4.4 for details). Specifying the random effects at a lower level of aggregation than the level at which estimation is desired (e.g., for clusters nested within the areas of interest) will lead to noisier estimates (Marhuenda et al. 2017), albeit with little effect on bias (Corral et al. 2021). The magnitude by which the MSE of the estimates based on a model with cluster effects instead of area effects will increase depends on the ratio between the variances of the random effects associated to the different locations. If the variance of the random effects of the clusters within areas, σ^2_{ac}, is larger than that of the area's random effects, σ^2_a, then the MSE may not increase by much. On the other hand, when the variance of the area's random effects is larger, the MSE of the estimates based on a model with only cluster effects worsens considerably. Thus, the random location effect should be at the same level for which estimates will be reported (see Figure 4.2).

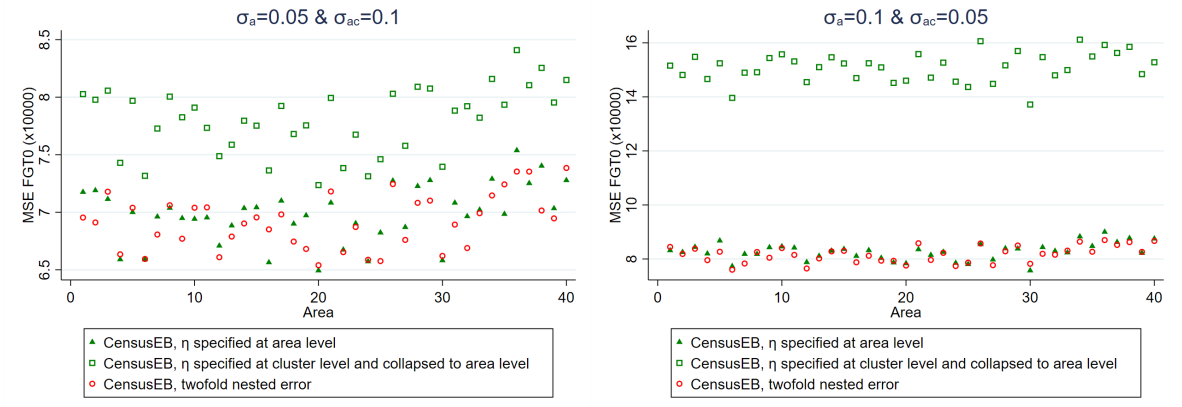
Additionally, under the latest edition of the Stata `sae` package, practitioners can apply two-fold nested error models (Marhuenda et al. 2017), besides the traditional one-fold nested error model. The drawback of the implemented two-fold nested error model in the Stata `sae` package is that it does not consider survey weights, nor does it consider heteroskedasticity. The benefit from applying a two-fold nested error model is that the resulting estimates are optimal at two levels of aggregation, because the MSEs of the estimates at both levels tend to be smaller under the assumed model than when using one-fold nested error models with only cluster or area effects (see Figure 4.2).

Two model fitting approaches are available under Stata's `sae` package: restricted maximum likelihood (REML) and Generalized Least Squares (GLS), where the variance parameters are estimated using Henderson's method III (Henderson 1953, H3). Both approaches yield similar estimated regression coefficients when heteroskedasticity and survey weights are not considered (Corral, Molina, and Nguyen (2021) – Figure A4). Given the different available models implemented in Stata's `sae` package, it is important to understand which options are available under each fitting method.

1. Model fit by REML does not incorporate survey weights nor heteroskedasticity. Uses Stata's `xtmixed` command in the background to fit the model.[6]

[6] Interested readers can refer to Stata's documentation to learn more about mixed models.

Figure 4.2: Empirical MSE comparison considering only cluster effects and when aggregating the cluster level results to the area level versus specifying the random effects at the reporting level, area.



Source: Simulation based on populations generated as described in Corral et al. (2021) section 3. Notice how the empirical MSE of CensusEB one-fold nested error models, where η is specified at the area level, are closely aligned to those under the two-fold nested error model with cluster and area effects. On the other hand, CensusEB estimators based on one-fold nested error models, where η is specified at the cluster level, have a considerably larger empirical MSE.

(a) When choosing this fitting method, the one-fold nested error model considered is the same one used in Molina and Marhuenda's (2015) `sae` package in R. Stata's `sae` package implements a similar REML fitting approach to Molina and Marhuenda's (2015) `sae` R package.[7]

```
sae model reml depvar indepvars [if] [in], area(varname)
```

Beyond the model fitting method, it is also possible to approximate the original Molina and Marhuenda's (2015) EB estimates by Monte Carlo simulation (replace `model` by `sim`) by adding the option `appendsvy`. However, the default approach obtains CensusEB[8] estimates instead of the original EB ones.

(b) Stata's `sae` package also implements the two-fold nested error model with cluster effects nested within area effects considered by Marhuenda et al. (2017). This model is appropriate when point estimates at two different levels of aggregation are desired. Section 4.2.4 presents the hierarchical identifiers needed for specifying two-fold nested error models.[9]

```
sae model reml2 depvar indepvars [if] [in], area(#) subarea(varname)
```

Here the `subarea` option requires an identifier for sub-areas of an equal number of digits for every area. Under the `area` option, the user should specify the number of digits of the hierarchical ID to remove from the right to the left to arrive at an identifier for the areas. These identifiers are commonly referred to as hierarchical identifiers and are explained in Nguyen et al. (2018) and Zhao (2006) in greater detail. Through Monte Carlo simulation (replace `model` by `sim`) the approach yields CensusEB estimates based on the two-fold nested error model.

2. Model fit with GLS, where variance parameters, estimated under Henderson's method III (1953), do allow for the inclusion of heteroskedasticity and survey weights according to Van der Weide (2014).[10] However, this method has only been implemented for one-fold nested error models.[11]

[7] To see the help file in Stata type: `help sae_ebp` for the one-fold nested error model.

[8] CensusEB estimates are similar to EB estimates, but do not include survey observations when calculating small area estimates. See Molina (2019) or Corral, Molina, and Nguyen (2021) for more details.

[9] To see help file in Stata type: `help sae_ebp2` for the two-fold nested error model.

[10] Methods that do not consider the survey sampling weights may in turn include covariates that capture the sampling/response mechanism, trying to correctly model their relationship to the target variable.

[11] To see help file in Stata type: `help sae_mc_bs` for the one-fold nested error model fit with Henderson's method III heteroskedasticity and survey weights (see Van der Weide 2014).

`sae model h3 depvar indepvars [if] [in] [aw], area(varname) [zvar(varnames) yhat(varnames) yhat2(varnames) alfatest(new varname)]`

The method will obtain CensusEB estimates for areas present in the survey and the census. When weights and heteroskedasticity are not specified, results will be very close to those from `sae model reml` (Corral, Molina, and Nguyen 2021).

The `alfatest` option allows users to obtain the dependent variable for the alpha model, a model for heteroskedasticity introduced in Elbers, Lanjouw, and Lanjouw (2002).[12] The resulting dependent variable for the alpha model can facilitate the selection of a suitable model. The covariates for the alpha model are specified under `zvar` and, when interactions with the main model's linear fit $X\hat{\beta}$ are desired, then users may specify covariates under the `yhat` and `yhat2` options for the interaction of covariates with $X\hat{\beta}$ and $(X\hat{\beta})^2$, respectively.

3. An updated GLS method with an adaptation of Henderson's method III (1953) that incorporates the error decomposition to estimate σ^2_η and σ^2_e as described in Elbers, Lanjouw, and Lanjouw (2002) and Nguyen et al. (2018) is available.[13] When users select this fitting method it is not possible to obtain EB or CensusEB estimates and does not require linking between survey and census areas, and hence not recommended.[14] It is still available in Stata's `sae` package to allow for comparisons and replicability.[15]

   `sae model ell depvar indepvars [if] [in] [aw], area(varname) [zvar(varnames) yhat(varnames) yhat2(varnames) alfatest(new varname)]`

   The `alfatest, zvar, yhat,` and `yhat2` options are similar to those described under H3 fitting method.

The most crucial difference between ELL and EB methods is that ELL estimators are obtained from the distribution of welfare without conditioning on the variable survey sample data, though other differences between the original ELL (2003) and MR (2010) exist. For example, though the underlying assumed model is the same, different model-fitting approaches are applied between ELL and MR approaches. Noise estimates of the small area estimators are also obtained differently. The implementation of ELL in `PovMap` draws from the multiple imputation literature, where minimizing the MSE is not the primary goal, while MR's method relies on the assumed model's data generating process to estimate the noise of the small area estimators with a parametric bootstrap introduced by González-Manteiga et al. (2008).[16]

Regardless of the model's structure (one-fold or two-fold) and the model-fitting approach used (REML or H3 in Stata's `sae` package), users are advised to obtain Empirical Best (EB) or CensusEB estimates rather than ELL estimates, because EB will yield more accurate and efficient estimates since EB conditions on the survey sample data and makes more efficient use of the information at hand - see Corral, Molina, and Nguyen (2021) for a detailed comparison, also see section 4.4.1.[17] The gains from EB can be quite considerable when the assumed model holds (Figure 4.3, left), but simulations based on real-world data,

[12] Readers may refer to Elbers, Lanjouw, and Lanjouw (2002) or Nguyen et al. (2018) for a full exposition of the alpha model for heteroskedasticity.

[13] The adaptation was made in light of criticism regarding the fitting method described in ELL (2002). See Nguyen et al. (2018) for more details.

[14] Ensuring the use of same area identifiers is still recommended for the inclusion of area-level variables.

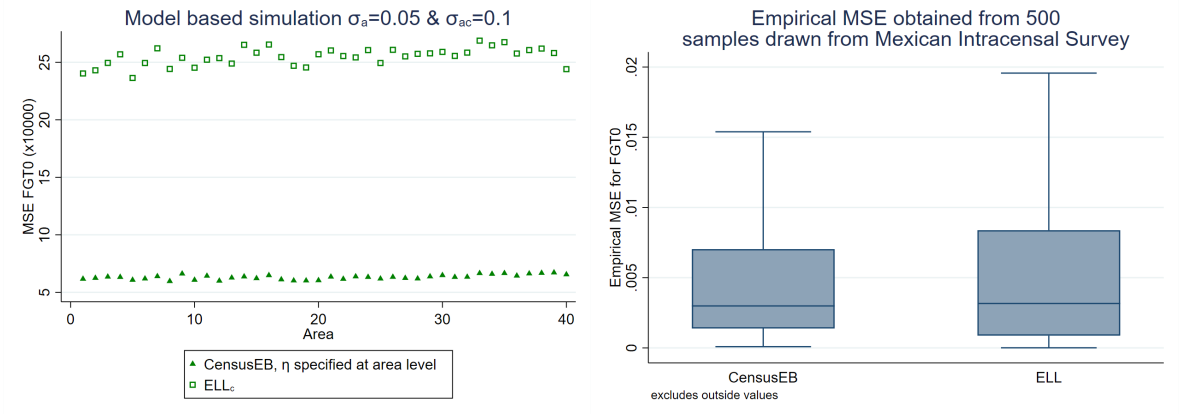
[15] To see help file in Stata type: `help sae_ell` for the one-fold nested error model fit with heteroskedasticity and survey weights (see Van der Weide 2014).

[16] Corral, Molina, and Nguyen (2021) discuss the differences between the methods used for noise estimation and how the methods fare when applied to simulated data.

[17] Section 4.4 gives a more detailed explanation on the differences.

where the validity of the assumed model is in doubt, also show gains (Figure 4.3, right).[18] Thus, despite ELL being one of the optional approaches in the Stata `sae` package, it is not recommended because EB estimates will yield more accurate and efficient estimates, see Corral, Molina, and Nguyen (2021) and section 4.4.1.[19]

Figure 4.3: Empirical MSE comparison between ELL and CensusEB



Source: Simulation based on populations generated as described in Corral et al. (2021), section 3 and 4. The simulations illustrate how the empirical MSE of ELL estimates is considerably larger than that of CensusEB under simulated data (left) and also under real world data (right).

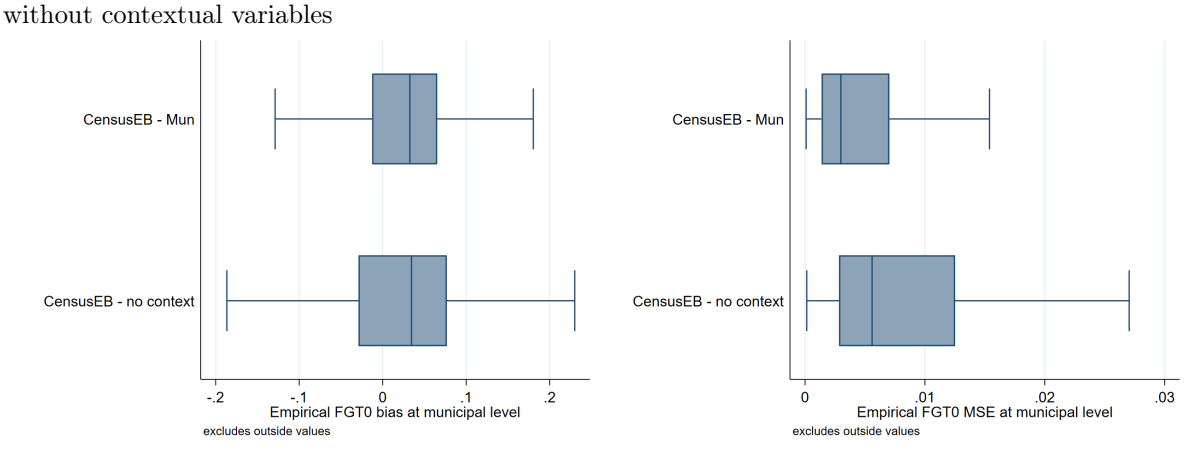
The use of contextual variables in the unit-level model is also recommended as long as their regression coefficients are significant. ELL (2002) noted the importance of explaining the variation in welfare due to location and recommended the inclusion of location means of household-level characteristics, as well as covariates derived from Geographic Information Systems (GIS). Contextual level variables were also noted by Haslett (2016) as being essential in ELL to reduce area-specific bias and standard errors. Since contextual variables can help explain between-area variations, these reduce the portion of unexplained between-area variation and complement household-level information to produce more precise estimates. An example of the value that contextual variables may provide in SAE can be seen in Figure 4.4, obtained in simulations using real world data. Poverty estimates obtained without contextual variables present more bias and a considerably larger MSE, though specific results will depend on the chosen variables and the data at hand.

As noted, the Stata `sae` package offers practitioners the option to fit ELL, EB or CensusEB estimates. CensusEB estimates are similar to EB, except that households are not linked across Census and survey data. In practice, the sample size of a given area in the survey is typically much smaller than the actual population size. In these cases, the difference between CensusEB and EB is negligible. In Corral, Molina, and Nguyen (2021) simulation experiments, where the sample by area is 4 percent of the total population, the difference between EB and CensusEB's empirical MSE is already indiscernible, see Figure 4.5.

[18] Empirical MSE are obtained from multiple samples. Under model-based simulations, the sample is kept fixed, and across populations the only difference is in the errors. Under design-based simulations, multiple samples are taken from a single (fixed) population.

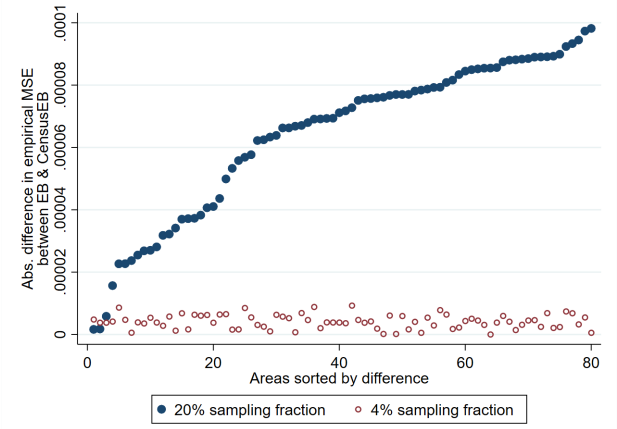
[19] In areas with no information (areas not sampled), EB and ELL estimates are similar.

Figure 4.4: Empirical bias and MSE of Census EB estimators based on a nested error model with and without contextual variables



Source: Simulation based on populations generated as described in Corral et al. (2021) section 3 and 4. These design-based simulations provide an example of the possible gains in terms of bias and MSE, obtained from adding contextual variables to the nested error model.

Figure 4.5: Empirical MSE difference between EB and CensusEB depending on sampling fraction



Source: Corral, Molina, and Nguyen (2021). Figure illustrates that the difference in empirical MSE of CensusEB and EB will be close to zero as the sample fraction of the population decreases.

4.2.2 Data Transformation

Since the EB method of MR (2010) assumes that the errors are normally distributed, data transformation to get a distribution close to normal can lead to less biased and less noisy estimates (see Rao and Molina 2015; Rojas-Perilla et al. 2020; Corral et al. 2021; and Tzavidis et al. 2018).[20] More generally, variable transformation to make the assumptions underlying the model hold is one of the main aspects of model-based small area estimation (see Technical Annex of this chapter, section 4.4.1). Specifically, since the SAE models used often assume normality in the errors, the goal in data transformation is to approximate

[20]The implementation of the ELL method in PovMap allowed practitioners to draw errors from the empirical distribution, which in principle may benefit the method when normality does not hold. Nevertheless, when comparing CensusEB method (assuming normal errors) and ELL method (drawing errors from the empirical distribution) in a model-based simulation experiment, where the model errors used to generate the populations were drawn from a Student's t-distribution, CensusEB outperformed ELL by a considerable margin (see Corral et al. 2021).

normally distributed residuals.[21]

Most software packages for small area estimation offer the possibility of data transformation. Perhaps the most popular transformation is the natural logarithm, although it is not always ideal as it can produce left skewed distributions for small values of welfare (see Figure 4.7). Other options include transformation from the Box-Cox family (Box and Cox 1964) and the log-shift transformations, which have the advantage that the transformation parameters may be driven by the specific data at hand (Tzavidis et al. 2018).The Box-Cox family's transformation parameter, which is the power to be taken, is denoted by λ. When $\lambda = 0$, it yields the natural logarithm; otherwise, λ is typically data-driven and chosen to minimize skewness in the dependent variable., which is the power to be taken, is denoted by λ. When $\lambda = 0$, it yields the natural logarithm; otherwise, λ is typically data-driven and chosen to minimize skewness in the dependent variable.[22] A log-shift transformation adds a positive shift to the observed welfare values before taking the log to avoid the left skewness caused by taking the log of very small welfare values. Other options, such as the ordered quantile normalization (Peterson and Cavanaugh 2019), may generate reliable estimates of headcount poverty, but these transformations are not reversible and therefore they cannot be used to obtain other indicators such as mean welfare, poverty gap, poverty severity among others (Masaki et al. 2020; Corral et al. 2021), limiting their applicability. Regardless of the transformation chosen, it is always recommended to check the residuals to determine if these follow the underlying model assumptions.

When working with real-world data, assumptions are often only approximately met (Marhuenda et al. 2017) and effort must be made to find one that approximates the normal distribution best. In simulation studies that seek to validate small area estimation methods based on actual data, data-driven transformations may reduce bias and the noise due to departures from normality (see Corral et al. 2021 and Tzavidis et al. 2018 among others). Using the *Mexican Intracensal survey of 2015* as a 3.9 million household census data, Corral et al. (2021) note that the gains from a transformation of the dependent variable that minimizes skewness can be quite considerable. In the validation exercises conducted in that paper, bias and MSEs, are considerably reduced when using a Box-Cox or a log-shift transformation (see Figure 4.6).

Implementation of a suitable data transformation in Stata is straightforward, since Stata offers commands for the Box-Cox and the zero-skewness log (log-shift) transformation.[23] The Stata `sae` package (Nguyen et al. 2018) also includes these transformations, in addition to the natural logarithm, as options and will apply the transformation for the model. The package will also reverse the transformation to calculate the indicators of interest in each Monte Carlo simulation. Prior to estimating the indicators, practitioners may wish to do model selection (e.g., by lasso or stepwise regression) using an already transformed dependent variable. Once the transformed dependent variable is obtained, practitioners can then conduct histograms and visually inspect the shape of the distribution of the dependent variable (see Figure 4.7).

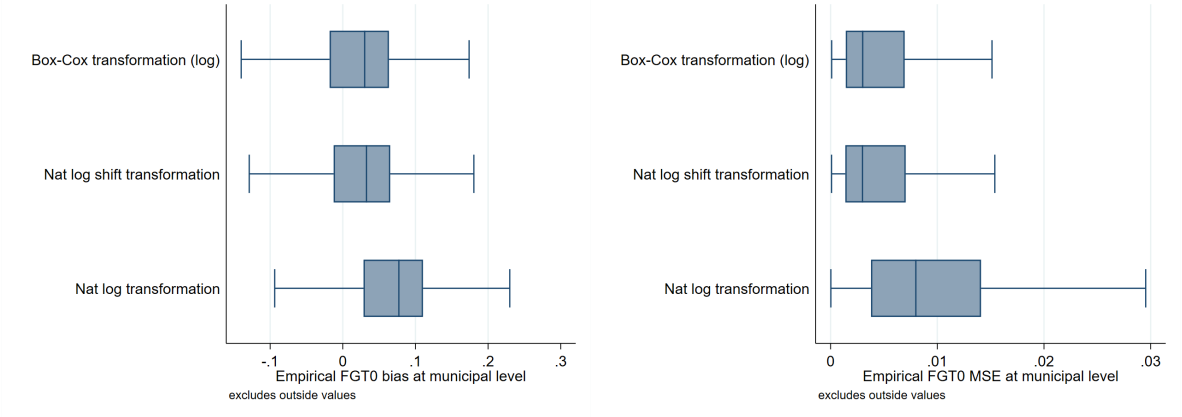
Once a satisfactory transformation is obtained, the process of model selection of the covariates can occur. The model selection process consists of selecting the response variable and the most powerful set of covariates such that the assumptions underlying the chosen SAE method hold – primarily linearity and normality of the residuals. The consequences of deviations from the underlying model's assumptions could lead to invalid inferences (Rao and Molina 2015) or not too noisy but biased estimates (Banerjee et al. 2006). The model selection procedure is further explored in chapter 6 in section 6.2.

[21] In cases where, even after transformation, the deviation is apparent only in isolated observations or isolated areas, models based on mixtures of normal distributions may be used. To estimate poverty and inequality indicators, a mixture may be specified as done by Elbers and Van der Weide (2014). Bikauskaite et al. (2020) extend the EB procedure to a multivariate mixture model that incorporates heterogeneity in the regression coefficients apart from the variance components. The proposed model was used to estimate FGT0 and FGT1 by gender in West Bank and Gaza.

[22] R packages `sae` (Molina and Marhuenda 2015) and `emdi` (Kreutzmann et al. 2019) also incorporate transformations.

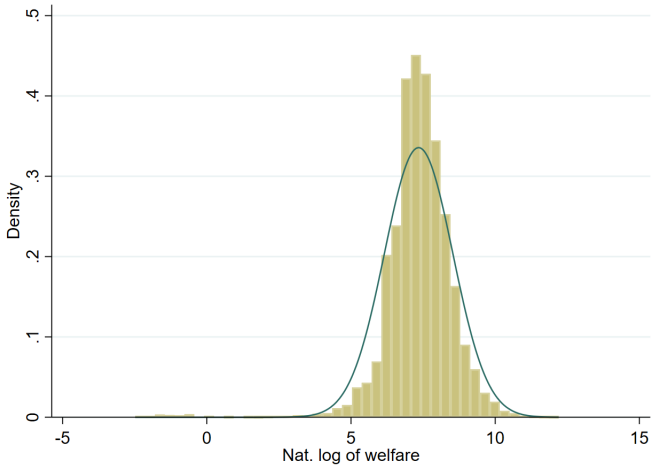
[23] Refer to the help files for `bcskew0` and `lnskew0` to learn more about the transformation.

Figure 4.6: Empirical Bias and MSE for CensusEB FGT0 estimates under different transformations



Source: Based on samples generated as described in Corral et al. (2021) section 4. The figure illustrates the potential gains from a correct data transformation. Taking the natural logarithm of the dependent variable in this case is not enough and can lead to considerable deviations from normality and thus may yield larger empirical MSEs and biased predictors.

Figure 4.7: Natural logarithm of welfare



Source: Corral et al. (2021) . The figure illustrates how, in this case, taking the natural logarithm of welfare leaves a left skewed distribution which may yield biased and noisier estimators.

4.2.3 The Alpha Model

The alpha model proposed by ELL (2002) is a preliminary model for the variance of the idiosyncratic errors used for heteroskedasticity that resembles the one presented by Harvey (1976).[24] Only limited research has been conducted on the impact of the alpha model for heteroskedasticity on the quality of the final small area estimates. In most applications, the adjusted R^2 of the alpha model is relatively small, rarely above 0.05.

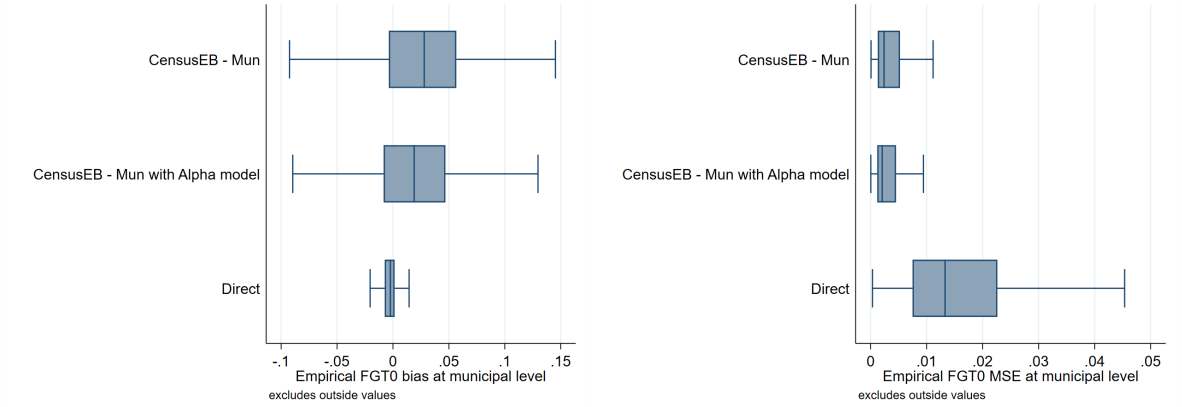
Nevertheless, the alpha model may play a considerable role in the bias and noise of the estimates. A simulation experiment based on the same data used for the validations conducted by Corral et al. (2021) shows that applying an alpha model to obtain CensusEB estimates might yield considerably less noisy and less biased estimates (see Figure 4.8).[25] This result may not translate to other data sets, which

[24] An alternative approach to modeling heteroskedasticity is to consider a random coefficient model similar to the one studied in Hobza and Morales (2013) and applied to the estimation of mean income. However, the method has not been implemented in any software.

[25] The example shows lower bias and a smaller MSE for a particular data set and could not translate equally to other data sets.

might not suffer from heteroskedasticity, and thus, using the alpha model should not be considered a universal recommendation. Practitioners should carefully inspect their model's residuals for signs of heteroskedasticity before deciding to use the alpha model.[26]

Figure 4.8: Empirical Bias and MSE for CensusEB FGT0 estimates with and without alpha model



Source: Based on samples of populations generated as described in Corral et al. (2021) section 4. The alpha model as specified by ELL (2002) can be applied using the sae Package in Stata. Under the Mexican data used by Corral et al. (2021) the alpha model appears to yield positive results in terms of bias and MSE. This should be interpreted with caution as it may not be the case in other scenarios.

In practice, the implementation of the alpha model may limit modeling options for practitioners since, though available in the `PovMap` software, it is not included in many of the newer software packages for SAE. For example, heteroskadasticity is also considered by Molina and Rao (2010), as well as Marhuenda et al. (2017). However, they do not specify the alpha model, and no software package implementation of their methods allows users to specify heteroskedasticity. The Stata `sae` package allows users to model heteroskedasticity for SAE, but only does so when using specific model fitting procedures such as the Henderson method III extension proposed by Van der Weide (2014) or the traditional ELL fitting method.[27] When choosing to apply the alpha model, the Stata `sae` package has an option that automatically constructs the dependent variable. This dependent variable may be used to facilitate the selection of a suitable model for heteroskedasticity. A removal process of non-significant covariates may also be implemented for the alpha model, as shown in the script of section 4.5.1.

4.2.4 Producing Small Area Estimates

Once the model selection stage is completed, the next step is to obtain estimates for small areas. Model selection and simulation stages may be separated in different scripts to avoid re-running unnecessary processes in case of errors in only one of the stages.[28] Model selection (see Stata script in section 4.5.1) is discussed in greater detail in Chapter 6 where attention is given to the different steps involved.

The Monte Carlo simulation process can be seen in the Stata script of section 4.5.2 and follows the model selection stage. The production of small area estimates requires first importing the census data. In Stata, it is imported into a Mata (Stata's matrix programming language) data file, which will allow users to work with larger data sets than what is usually possible.[29] The user must specify the vectors

[26] Heteroskedasticity may be checked by plotting residuals against predicted values.

[27] In terms of alternatives, the R package `mcmcsae` (Boonstra and Baltissen 2021) can also model heteroskedasticity in terms of covariates.

[28] It is advisable to split the processes across multiple do files, because model selection, as well as simulating welfare in the cesus and obtaining small area estimates for all areas is not computationally negligible.

[29] Note Stata cannot open this data file as a regular dta file.

she wishes to import, including household expansion factors (typically household size) within the census data. The census data should have variable names equal to those in the survey. It is recommended to import only the necessary variables from the census to avoid creating larger data files than needed for the process.

The SAE process can be broken up into two steps. The first step entails producing the point estimates through Monte Carlo simulation. This step is relatively fast, and the recommended amount of simulations is at least 100. The estimation of the MSE is considerably more time-consuming and will usually take several hours, depending on the size of the census. Thus, it is recommended to first obtain small area estimates without specifying bootstrap simulations for MSE estimation. Once users have done their checks and determined the quality of their point estimates, the MSE can be estimated. It is recommended to use at least 200 bootstrap replications to estimate MSEs.

4.3 Pros and Cons of Unit-Level Models

This section presents a convenient list of pros and cons for each method. It also notes the needs for each of the methods. The section borrows heavily and adds to the work presented in Molina (2019).

4.3.1 ELL

Model requirements:

1. Microdata from a household survey and census at the household level, with variables measured and questions asked similarly.
2. A set of variables related to the welfare variable of interest. These variables should have similar distributions between the two data sources.
3. Areas in the survey and the census should have identifiers that can be linked across each other.

Pros:

1. The model is based on household-level data, which gives more detailed information than area-level data, and have a considerably larger sample size.
2. Can provide estimates for non-sampled areas.
3. Estimates from any indicator that is a function of welfare can be obtained without relying on different models.
4. The estimates are unbiased under the model if the model is true and the model parameters are known.
5. It is possible to get estimates at any level of aggregation.
6. Offers a software-implemented model for heteroskedasticity.

Cons:

1. The method, as is currently implemented, may yield rather noisy (inefficient) estimates and may even be worse than direct estimates (obtained using only the area-specific information from the survey), if heterogeneity between areas is not explained adequately.
   (a) The noise of the traditional ELL estimates is large but it is also underestimated under the ELL bootstrap procedure (see Corral, Molina, and Nguyen (2021) for details).
   (b) To reduce the above noise to the largest possible extent, it is always recommendable to include all the potentially significant covariates that may affect welfare, including contextual covariates, which may be taken from data sources different from the census.
2. Estimates are the outcome of a model and thus, the model needs to be appropriately checked.
3. The traditional ELL model specifies the location effects for the clusters (PSUs), typically nested within the areas where the estimation is desired. Estimations at higher aggregation levels than those at which the location effects are specified can underestimate noise if the area-level variables included in the model fail to explain the between-area heterogeneity.

4.3.2 Empirical Best and CensusEB

Model requirements:

1. Microdata from a household survey and census at the household level with variables measured and questions asked in a similar manner.
2. A set of variables related to the welfare variable of interest. These variables should have similar distributions between the two data sources.
3. Areas in the survey and the census should have identifiers that can be linked across each other.

Pros:

1. Based on household-level data which provide more detailed information than area-level data and have a considerably larger sample size.
2. Can provide estimates for non-sampled areas.
3. Estimates from any indicator that is a function of welfare can be obtained without relying on different models.
4. Unbiased under the model, if the model is true and model parameters are known.
5. Optimal in the sense that they minimize the MSE under the model.
6. Considerably less noisy than ELL when unexplained heterogeneity across areas is considerable. For non-sampled areas, ELL and EB estimators are similar.
7. Offers a software-implemented model for heteroskedasticity (choosing H3 CensusEB in the Stata `sae` package).

Cons:

1. They are the outcome of a model and thus the model needs to be properly checked.
2. It may be affected by deviations from the model's assumptions. For example, deviation from normality or outliers can have detrimental effects on SAE - isolated unit-level outliers may not have much impact if the sample size is large.
3. Original EB estimates do not consider the sampling design and are not design-unbiased and hence may be affected by informative sampling designs (i.e., when the sample selection probabilities depend on the outcome values - Rao and Molina 2015).
   (a) In the Stata `sae` package, the REML option obtains the original EB, but H3 CensusEB incorporates the sampling weights and should not be sensibly affected by informative sampling.
4. The bootstrap method for MSE estimation is computationally intensive.

4.3.3 Alternative Software Packages for Unit-Level Models and Extensions

- Packages that implement the one-fold nested error model EB approach from Molina and Rao (2010):
  - The `sae` package in R (Molina and Marhuenda 2015).
  - After the updates presented in Corral, Molina, and Nguyen (2021) the `sae` package in Stata (Nguyen et al. 2018).
  - The `emdi` package in R (Kreutzmann et al. 2019).
- Empirical best prediction that incorporates the survey weights (Pseudo EB):
  - The `emdi` package in R (Kreutzmann et al. 2019) has been extended to include the Pseudo EB method presented in Guadarrama, Molina, and Rao (2018).
  - After the updates presented in Corral, Molina, and Nguyen (2021) the `sae` package in Stata (Nguyen et al. 2018) includes the model-fitting method that incorporates the survey weights presented in Van der Weide (2014).
    * The implemented method only obtains CensusEB estimates.
- Two-fold nested error model EB approach as presented in Marhuenda et al (2017):
  - After the updates presented in Corral, Molina, and Nguyen (2021), the `sae` package in Stata (Nguyen et al. 2018).
    * The implemented method only obtains CensusEB estimates.
- Estimation of area means (without transformation of welfare) based on Hierarchical Bayes unit-level models:
  - The `hbsae` package in R (Boonstra 2015).
- ELL approach:
  - The `sae` package in Stata (Nguyen et al. 2018).
  - `PovMap`, which is a free stand-alone software (Zhao 2006).

4.4 Unit-Level Models – Technical Annex

4.4.1 The Assumed Model

The nested error model used for small area estimation by ELL (2003) and Molina and Rao (2010) was originally proposed by Battese, Harter and Fuller (1988) to produce county-level corn and soybean crop area estimates for the American state of Iowa. For the estimation of poverty and welfare, the ELL and MR methods assume the transformed welfare y_{ch} for each household h in the population within each location c is linearly related to a $1 \times K$ vector of characteristics (or correlates) x_{ch} for that household, according to the nested error model:

$$y_{ch} = x_{ch}\beta + \eta_c + e_{ch},\ h = 1, \ldots, N_c,\ c = 1, \ldots, C. \tag{4.2}$$

Here η_c and e_{ch} are respectively location and household-specific idiosyncratic errors, assumed to be independent from each other, satisfying:

$$\eta_c \overset{iid}{\sim} N\left(0, \sigma_\eta^2\right),\ e_{ch} \overset{iid}{\sim} N\left(0, \sigma_e^2\right),$$

where the variances σ_η^2 and σ_e^2 are unknown. Here, C is the number of locations in which the population is divided and N_c is the number of households in location c, for $c = 1, \ldots, C$. Finally, β is the $K \times 1$ vector of coefficients.

To illustrate the differences between the methods, understanding the model is essential. First, note that the random location effect is the same for all households within a given area. Second, note that the random location effect is not necessarily 0 for a given location, despite the random location effect being drawn from a normal distribution with mean 0. Consequently, for any given realization of the population, the random location effect is unlikely to be equal to 0, but its expected value is 0 across all the possible realizations. To see this, a simulation where the random location effects for 80 areas are drawn from a normal distribution $\eta_c \overset{iid}{\sim} N\left(0, 0.15^2\right)$, is shown in the Stata code snippet below. This is repeated across 10,000 simulated populations.

```
. //Do file simulates random location effects for 80 areas
. set more off

. clear all

. version 14

. set maxvar 10100


.
. //Seed for replicability
. set seed 232989

. //Necessary macros
. local sigmaeta = 0.15     //Sigma eta

. local Npop     = 10000    //number of populations

. local Narea    = 80

. //Create simulated data
. set obs `Narea´
number of observations (_N) was 0, now 80

. //Simulate random location effects for 80 areas, 10000 populations
. forval z=1/`Npop´{
```

```
  2.         gen double eta_`z´ = rnormal(0,`sigmaeta´)
  3. }

. //Look at values for 1 population
. sum eta_1, d //very close to 0

                          eta_1
-------------------------------------------------------------

      Percentiles      Smallest
 1%    -.3601142      -.3601142
 5%    -.2164503      -.3536255
10%    -.1782243       -.235858       Obs                  80
25%    -.1081445      -.2219594       Sum of Wgt.          80

50%     .0134193                      Mean           .0047588
                       Largest       Std. Dev.      .1484272
75%     .1098011        .249745
90%     .1700735       .2672989       Variance       .0220306
95%     .2327892       .3286754       Skewness        .007834
99%     .4155954       .4155954       Kurtosis       2.991039

. list eta_1 if _n==23 //not 0

     +-----------+
     |     eta_1 |
     |-----------|
23.  | -.22195943 |
     +-----------+

. //The expected value of the random location effect across populations
. egen double E_eta = rmean(eta_*)

. sum E_eta,d

                          E_eta
-------------------------------------------------------------

      Percentiles      Smallest
 1%     -.003775       -.003775
 5%    -.0018793      -.0031299
10%    -.0016342      -.0023112       Obs                  80
25%    -.0008481      -.0019355       Sum of Wgt.          80

50%     .0004301                      Mean           .0002009
                       Largest       Std. Dev.      .0014468
75%     .0010856       .0025296
90%     .0023057       .0026368       Variance       2.09e-06
95%        .0025       .0029182       Skewness      -.1558987
99%     .0033486       .0033486       Kurtosis       2.744393

. //The expected value for one area across all populations
. list E_eta if _n==23 //approximating 0

     +-----------+
     |     E_eta |
     |-----------|
23.  | .00242892 |
     +-----------+

.
end of do-file
```

Note how in the results, the value of the random location effect for a given area and population is not necessarily equal to 0. However, across all the generated populations, the mean value of the random location effect for all areas is very close to 0.

This aspect is important, because it highlights the key difference between the methodology from ELL (2003) and the EB approach from MR (2010). Under ELL, when simulating welfare for the census, the

location effect is drawn exactly as assumed under the model, that is as, $\eta_c \overset{iid}{\sim} N\left(0, \sigma_\eta^2\right)$. In essence, under ELL, for any given area present in the sample, the ELL estimator of the census area mean $\bar{y}_c$ is obtained by averaging the actual area means $\bar{y}_c^{*(m)} = \bar{X}_c'\beta + \eta_c^{*(m)} + \bar{e}_c^{*(m)}$, $m = 1, \ldots, M$, across M simulated populations, that is, the ELL estimator is $\frac{1}{M}\sum_{m=1}^{M} \bar{y}_c^{*(m)}$, which approximates $E\left(\bar{y}_c\right)$. But note that $E[\eta_c] = 0$ and $E[e_{ch}] = 0$. Hence, the ELL estimator reduces to the regression-synthetic estimator, $\bar{X}_c'\beta$ (MR, 2010). On the other hand, under MR (2010), conditioning on the survey sample ensures that the estimator includes the random location effect, since $E[\bar{y}_c|\eta_c] = \bar{X}_c'\beta + \eta_c$. Haslett (2016) suggests that the inclusion of contextual variables somewhat attenuates the issue. Nevertheless, unless the contextual variables together with the household-level ones fully explain the variation in welfare across areas, there will always be gains from using EB compared to ELL. Actually, the EB approach from Molina and Rao (2010) gives approximately the *best* estimator in the sense that it yields the estimator with the minimum mean squared error (MSE) for target areas. Consequently, EB estimators are considerably more efficient than ELL (see Molina and Rao 2010; Corral, Molina, and Nguyen 2021; and Corral et al. 2021).

A census data set of $N = 20,000$ observations is created, similar to MR (2010), to observe how conditioning on the survey sample affects the resulting small area estimates. Under the created data, all observations are uniformly spread among $C = 80$ areas, labeled from 1 to 80. This means that every area consists of $N_c = 250$ observations. The location effects are simulated as $\eta_c \overset{iid}{\sim} N\left(0, 0.15^2\right)$; note that every observation within a given area will have the same simulated effect. Then, values of two right-hand side binary variables are simulated. The first one, x_1, takes a value of 1 if a generated random uniform value between 0 and 1 is less than or equal to $0.3 + 0.5\frac{c}{80}$. This means that observations in areas with a higher label are more likely to get a value of 1. The next one, x_2, is not tied to the area's label. This variable takes the value 1 if a simulated random uniform value between 0 and 1 is less than or equal to 0.2. The census welfare vectors $y_c = (y_{c,1}, \ldots, y_{c,N_c})^T$ for each area $c = 1, \ldots, C$, are then created from the model as follows:

$$\ln(y_{ch}) = 3 + 0.03x_{1,ch} - 0.04x_{2,ch} + \eta_c + e_{ch},$$

where household-level errors are generated under the homoskedastic setup, as $e_{ch} \overset{iid}{\sim} N\left(0, 0.5^2\right)$. The poverty line is fixed at $z = 12$, corresponding to roughly 60 percent of the median welfare of a generated population. The steps to creating such a population in Stata are shown below:

```
. set more off

. clear all

. version 14

. set seed 648743

.
. local obsnum    = 20000  //Number of observations in our "census"

. local areasize  = 250    //Number of observations by area

. local outsample = 20     //Sample size from each area (%)

. local sigmaeta = 0.15     //Sigma eta

. local sigmaeps = 0.5      //Sigma eps

.
.
. // Create area random effects
. set obs `=`obsnum´/`areasize´´
number of observations (_N) was 0, now 80

. gen area = _n   // label areas 1 to C

. gen eta  = rnormal(0,`sigmaeta´)  //Generate random location effects
```

```
. expand `areasize´ //leaves us with 250 observations per area
(19,920 observations created)

. sort area //To ensure everything is ordered - this matters for replicability

. //Household identifier
. gen hhid = _n

. //Household expansion factors - assume all 1
. gen hhsize = 1

. //Household specific residual
. gen e = rnormal(0,`sigmaeps´)

. //Covariates, some are corrlated to the area´s label
. gen x1=runiform()<=(0.3+.5*area/`=`obsnum´/`areasize´´)

. gen x2=runiform()<=(0.2)

. //Welfare vector
. gen Y_B = 3+ .03* x1-.04* x2 + eta + e

.
. preserve

.         sort hhid

.         sample 20, by(area)
(16,000 observations deleted)

.         keep Y_B x1 x2 area hhid

.         save "$mdata\sample.dta", replace
file C:\Users\Paul Corral\OneDrive\SAE Guidelines 2021\\3_Unit_level\\1_data\\s
> ample.dta saved

. restore

.
. save "$mdata\thepopulation.dta", replace
file C:\Users\Paul Corral\OneDrive\SAE Guidelines 2021\\3_Unit_level\\1_data\\t
> hepopulation.dta saved

.
end of do-file
```

The nested error model is then fit to the population data using restricted maximum likelihood:

```
. mixed Y_B x1 x2 || area:, reml

Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood = -14736.236
Iteration 1:   log restricted-likelihood = -14736.236

Computing standard errors:

Mixed-effects REML regression                   Number of obs     =     20,000
Group variable: area                            Number of groups  =         80

                                                Obs per group:
                                                              min =        250
                                                              avg =      250.0
                                                              max =        250

                                                Wald chi2(2)      =      44.41
Log restricted-likelihood = -14736.236          Prob > chi2       =     0.0000

------------------------------------------------------------------------------
         Y_B |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
```

x1	.0252974	.0074975	3.37	0.001	.0106026	.0399922
x2	-.0505327	.0088161	-5.73	0.000	-.067812	-.0332534
_cons	3.014335	.0201408	149.66	0.000	2.97486	3.05381

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
area: Identity				
var(_cons)	.0298249	.0049059	.0216055	.0411712
var(Residual)	.2518635	.0025238	.2469652	.256859

LR test vs. linear model: chibar2(01) = 1940.36 Prob >= chibar2 = 0.0000

Linear predictions, $X\hat{\beta}$, can also be obtained:

```
. predict double xb, xb
```

Predicted random location effects, $\hat{\eta}_c$, can also be obtained:

```
. predict double eta_pred, reffects
. sum eta_pred
```

Variable	Obs	Mean	Std. Dev.	Min	Max
eta_pred	20,000	-2.17e-13	.1687893	-.3371651	.3667746

Stata can also produce the linear prediction, $X\hat{\beta}$, plus the estimated location effects, $\hat{\eta}_c$:

```
. predict double xb_eta, fitted
```

The expected values of the linear predictor $X\hat{\beta}$, and the linear predictor that includes the predicted location effects, $X\hat{\beta} + \hat{\eta}_c$, are the same. The difference between these is that $X\hat{\beta} + \hat{\eta}_c$ includes the estimated location effect, and thus, a larger share of the variance is explained, which leads to minimized estimation errors for the areas.

```
. sum Y_B xb xb_eta
```

Variable	Obs	Mean	Std. Dev.	Min	Max
Y_B	20,000	3.017888	.5308399	1.027403	5.197237
xb	20,000	3.017888	.0240033	2.963802	3.039632
xb_eta	20,000	3.017888	.1702655	2.626637	3.406407

At the same time, the area specific predictions are considerably closer to the observed values:

```
. collapse Y_B xb xb_eta, by(area)
. list Y_B xb xb_eta in 1/10
```

	Y_B	xb	xb_eta
1.	2.693685	3.011817	2.7040804
2.	3.334479	3.0108076	3.3239027
3.	2.683372	3.0086839	2.6940016

4.	3.117312	3.0112101	3.1138448
5.	2.670211	3.0146496	2.6814654
6.	3.049783	3.0128294	3.0485756
7.	2.945565	3.0121218	2.9477392
8.	3.180324	3.0099986	3.1747587
9.	3.139087	3.0139408	3.1349977
10.	2.891956	3.01273	2.8959019

4.4.2 Monte Carlo Simulation and Bootstrap Procedures Under CensusEB

The best predictor is defined as a conditional expectation. For indicators with a complex shape, the conditional expectation defining the best predictor may not have a closed form. Regardless of the shape, the best predictor can be approximated by Monte Carlo simulation (Molina 2019). ELL and MR's EB approach clearly differ in the computational procedures used to estimate the indicators of interest and their noise. On the other hand, the original implementation of EB in `PovMap` and the `sae` Stata command used a similar computational approach to ELL. Under ELL and the original EB implementation in `PovMap`, noise and indicator estimates are all obtained with a single computational procedure. Corral, Molina, and Nguyen (2021) show that the noise estimates of the original ELL (referred to as variance by the authors) underestimate the true MSE of the indicators, despite ELL estimates being much noisier. On the other hand, the parametric bootstrap procedure from González-Manteiga et al. (2008) seems to estimate the noise (MSE) of EB estimators correctly.

For an in-depth discussion on how ELL and EB computational procedures differ, see Corral, Molina, and Nguyen (2021). For easy interpretation, the expositions presented here do not consider survey weights or heteroskedasticity.[30]

4.4.2.1 Molina and Rao's (2010) Monte Carlo Simulation Procedure for Point Estimates

The EB method from MR (2010) conditions on the survey sample data and thus makes more efficient use of the survey data, which contains the only available (and hence precious) information on the actual welfare. Conditioning on the survey data requires matching households across the survey and census. Matching households was required in the original EB approach introduced in MR (2010). However, the census and sample households can not be matched in practice, except in some countries. When linking the two data sources is not possible, this sample may be appended to the population for areas where there is a sample. This is the approach taken in the original EB implementation in the `sae` R package by Molina and Marhuenda (2015). In order to obtain a Monte Carlo approximation to the EB estimates using this software, estimates are obtained as a weighted average between a simulated census poverty rate for the area and the direct estimate of poverty obtained using the area's sample. When the sample size relative to the population size for a given area is considerable, for example, 20%, using the sample observations of welfare may yield to a considerable gain in MSE. However, as noted in Figure 4.5 in section 4.2.1 the gain in MSE of EB over CensusEB approximates 0 as the sample size by area shrinks. When the sample is 4% of the population per area, an already quite large sample rarely encountered in real-world scenarios, the difference between the approaches is nearly zero.

To illustrate how CensusEB estimates are implemented, a 20% sample by area of the data created in section 4.4.1 is used as a survey to which the model 4.2 is fit. To obtain point estimates under Molina

[30] For the full exposition, see Corral, Molina and Nguyen (2021).

and Rao (2010), the authors assume that the considered unit-level model is the one that generates the data. Thus, the estimates for β, σ^2_η, and σ^2_e obtained from the fitted model are kept fixed in the Monte Carlo simulation procedure used to obtain EB point estimates.[31] Additionally, the predicted random location effects, $\hat{\eta}_c$, are also kept fixed as a result of conditioning on the sample observations of welfare.

The first step consists in fitting the model and obtaining the parameter estimates, $\hat{\theta}_0 = \left(\hat{\beta}_0, \hat{\sigma}^2_{\eta 0}, \hat{\sigma}^2_{e0}\right)$ which are later used to simulate vectors of welfare for all the population of households (a census of welfare).

```
. //MonteCarlo simulation to obtain CensusEB estimates
. use "$mdata\sample.dta", clear

.
.         //fit model on the sample
.         mixed Y_B x1 x2 || area:, reml

Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood = -3009.9023
Iteration 1:   log restricted-likelihood = -3009.9023

Computing standard errors:

Mixed-effects REML regression                   Number of obs     =      4,000
Group variable: area                            Number of groups  =         80

                                                Obs per group:
                                                              min =         50
                                                              avg =       50.0
                                                              max =         50

                                                Wald chi2(2)      =       7.31
Log restricted-likelihood = -3009.9023          Prob > chi2       =     0.0258
```

Y_B	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
x1	.028721	.0167756	1.71	0.087	-.0041585	.0616005
x2	-.0423625	.0201979	-2.10	0.036	-.0819496	-.0027753
_cons	3.010993	.0241635	124.61	0.000	2.963633	3.058353

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
area: Identity				
var(_cons)	.0339989	.0062132	.0237635	.048643
var(Residual)	.2521756	.0056975	.2412523	.2635934

```
LR test vs. linear model: chibar2(01) = 338.23        Prob >= chibar2 = 0.0000

.
.         //Obtain the necessary parameters
.                 local sigma_eta2 =  (exp([lns1_1_1]_cons))^2

.                 local sigma_e2   =  (exp([lnsig_e]_cons))^2
```

With these estimates in hand, it is possible to predict the random location effects, η_c, as well as the shrinkage parameter, $\gamma_c = \sigma^2_\eta \left(\sigma^2_\eta + \sigma^2_e/n_c\right)^{-1}$ and the variance of the random location effect, $\sigma^2_\eta(1-\gamma_c)$. Notice that γ_c will be between 0 and 1, hence the variance of the location effect will be smaller than σ^2_η,

[31] Note that this is considerably different from the approach that was implemented in `PovMap` (Corral, Molina, and Nguyen 2021).

which is the variance of the location effect under ELL. These model parameter estimates will then be used as true values to simulate the welfare vectors for the whole population of households.

```
.         //Let´s calculate the linear fit
.         predict xb, xb

.
.         //Get the residual to calculate the random location effect
.         gen residual = Y_B - xb

.
.         //Number of observations by area
.         gen n_area = 1 //we´ll add it later

.
. groupfunction, mean(residual) sum(n_area) by(area)

.         //Gamma or adjustment factor ()
.         gen double gamma  = `sigma_eta2´/(`sigma_eta2´+`sigma_e2´/n_area)

.         //Produce eta
.         gen double eta = gamma*residual

.         //variance of random location effects
.         gen double var_eta = `sigma_eta2´*(1-gamma)

.
.         keep area eta var_eta

.
. tempfile mylocs

. save `mylocs´
file C:\Users\PAULCO~1\AppData\Local\Temp\ST_2bd8_000003.tmp saved
```

The next step consists in applying the estimated parameters, $\hat{\theta}_0$, to generate the population vectors of welfare. Note how the same $\hat{\beta}_0$ are used to produce the linear fit in every Monte Carlo simulation. Notice that the location effects are predicted from the sample from the corresponding area; thus, it is necessary to match areas across the sample and population. The next step consists in generating 100 vectors of welfare for the population units (census of welfare). Notice that here, the natural logarithm of welfare is reversed. Additionally, welfare for each household, h, is drawn from $\ln(y_{ch}) \sim N\left(x_{ch}\hat{\beta}_0 + \hat{\eta}_{c0},\ \hat{\sigma}^2_{\eta 0}(1-\hat{\gamma}_c) + \hat{\sigma}^2_{e0}\right)$.

```
. //Set seed for replicability
. set seed 9374

.
. //Bring in the population
. use x1 x2 area hhid using "$mdata\thepopulation.dta", clear

.         //obtain linear fit - e(b) is still in memory
.         predict xb, xb

.
.         //Include eta and var_eta
.         merge m:1 area using `mylocs´
(note: variable area was float, now double to accommodate using data´s values)

    Result                           # of obs.
    -----------------------------------------
    not matched                             0
    matched                            20,000  (_merge==3)
    -----------------------------------------

.                drop if _m==2
(0 observations deleted)

.                drop _m
```

```
.         //to ensure replicability
.         sort hhid


. //generate 100 vectors of welfare in the population
. forval z=1/100{
  2.     //Take the exponential to obtain welfare
.     gen double Y_`z´ = exp(rnormal(xb + eta, sqrt(`sigma_e2´+var_eta)))
  3. }
```

Finally, FGT indicators (Foster, Greer, and Thorbecke 1984) are calculated for each area from the simulated census – this is done for the 100 simulated censuses. Then, the CensusEB estimate is obtained by averaging across the 100 FGT indicators obtained for the area.

```
.         //Indicate poverty line
.         gen povline = 12

.         //Obtain FGTs for each area under every simulated vector
.         sp_groupfunction, poverty(Y_*) povertyline(povline) by(area)

.         //Average over simulations to obtain the EB estimate
.         groupfunction, mean(value) by(measure area)

.         //Reshape to obtain wide data at the area level
.         qui:reshape wide value, i(area) j(measure) string

. //Save CensusEB
. save "$mdata\CensusEBfgt.dta", replace
file C:\Users\Paul Corral\OneDrive\SAE Guidelines 2021\\3_Unit_level\\1_data\\CensusEBfgt.dta saved
```

Note that this is quite different from the method used for ELL, as implemented in `PovMap` and the Stata `sae` package.[32] The steps detailed above are aligned to those applied by the command `sae sim reml` to obtain CensusEB estimates and are very similar to the approach under `sae sim h3`.

4.4.2.2 Parametric Bootstrap

The parametric bootstrap used to estimate the MSE of the EB estimates was introduced by González-Manteiga et al. (2008). It is the approach used by Molina and Rao (2010) and the one used in the `sae` R package from Molina and Marhuenda (2015), as well as in the updated Stata `sae` package.[33] The procedure is computationally more demanding than the bootstrap procedure used in `PovMap` which was inspired by the MI literature. However, the parametric bootstrap procedure from González-Manteiga et al. (2008) tracks the real MSE values.

The process consists in, first, creating a population vector of welfare (i.e. census) using $\hat{\theta}_0 = \left(\hat{\beta}_0,\ \hat{\sigma}^2_{\eta 0},\ \hat{\sigma}^2_{e0}\right)$. Note that $\hat{\theta}_0 = \left(\hat{\beta}_0,\ \hat{\sigma}^2_{\eta 0},\ \hat{\sigma}^2_{e0}\right)$ are obtained from the original sample and match those from the ones used in the previous step for computing CensusEB point estimates.

```
. //Macors used below
. global povline = 12

. global bs = 1 //Bootstrap replicates

.
.
```

[32] See Corral, Molina, and Nguyen (2021) for more details.
[33] See Corral et al. (2021) for more details.

```
. //Gonzales-Manteiga et al. (2008) Bootstrap for estimating MSE
. use "$mdata\sample.dta", clear

.
.         //fit model on the sample
.         mixed Y_B x1 x2 || area:, reml

Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood = -3009.9023
Iteration 1:   log restricted-likelihood = -3009.9023

Computing standard errors:

Mixed-effects REML regression                   Number of obs     =      4,000
Group variable: area                            Number of groups  =         80

                                                Obs per group:
                                                              min =         50
                                                              avg =       50.0
                                                              max =         50

                                                Wald chi2(2)      =       7.31
Log restricted-likelihood = -3009.9023          Prob > chi2       =     0.0258
```

Y_B	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
x1	.028721	.0167756	1.71	0.087	-.0041585	.0616005
x2	-.0423625	.0201979	-2.10	0.036	-.0819496	-.0027753
_cons	3.010993	.0241635	124.61	0.000	2.963633	3.058353

Random-effects Parameters	Estimate	Std. Err.	[95% Conf.	Interval]
area: Identity				
var(_cons)	.0339989	.0062132	.0237635	.048643
var(Residual)	.2521756	.0056975	.2412523	.2635934

```
LR test vs. linear model: chibar2(01) = 338.23        Prob >= chibar2 = 0.0000

.
.         //Obtain the necessary parameters
.                 local sigma_eta2 =  (exp([lns1_1_1]_cons))^2

.                 local sigma_e2   =  (exp([lnsig_e]_cons))^2

.         //Let´s calculate the linear fit
.         predict xb, xb

.         dis sqrt(`sigma_eta2´)
.18438803

.         dis sqrt(`sigma_e2´)
.50217086

.
. tempfile sample

. save `sample´
file C:\Users\PAULCO~1\AppData\Local\Temp\ST_1690_000003.tmp saved
```

Using $\hat{\theta}_0 = \left(\hat{\beta}_0, \hat{\sigma}^2_{\eta 0}, \hat{\sigma}^2_{e0}\right)$ as the vector of true values, B population vectors of welfare are created following the DGP from Eq. 4.2. Each area's desired set of indicators is obtained from each population vector. These indicators are regarded as the true values in each bootstrap replicate and the benchmark to which CensusEB estimates for that generated population will be compared to. The estimate of the

MSE is the mean across the B replicates of the squared differences between the bootstrap replicate's estimate and the bootstrap replicate's "true" value. In the code below, the steps for the first bootstrap replicate, $b = 1$, in this parametric bootstrap procedure are shown.

```
. *================================================================================
. //The second stage consists in "extracting the sample" and obtaining estimates
. // via MonteCarlo simulation. For each bootstrap replicate...this is b=1
. *================================================================================
.         use x1 x2 xb hhid area using `sample´, clear

.                 //Include the etas
.                 //note that these are the same used in the population
.                 merge m:1 area using `etas´

    Result                           # of obs.
    -----------------------------------------
    not matched                             0
    matched                             4,000  (_merge==3)
    -----------------------------------------

.                         drop _m

.                 sort hhid

.                 //Generate the welfare vector, in exactly the same manner as in the
.                 //population
.                 gen lny = rnormal(xb + eta_1,sqrt(`sigma_e2´))

.
.         local seedstage  `c(rngstate)´

.
.         // Now with the new welfare vector we can obtain CensusEB estimates
.         //for the bootstrap replicate using sae
.         sae sim reml lny x1 x2,  area(area)  mcrep(50) bsrep(0) lny ///
>         seed(`seedstage´) pwcensus(hhsize) indicators(FGT0 FGT1 FGT2) ///
>         aggids(0) uniq(hhid) plines($povline) matin("$mdata\mypop_mata")

Performing EM optimization:

Performing gradient-based optimization:

Iteration 0:   log restricted-likelihood = -3004.5851
Iteration 1:   log restricted-likelihood = -3004.5851

Computing standard errors:

Mixed-effects REML regression                   Number of obs     =      4,000
Group variable: area                            Number of groups  =         80

                                                Obs per group:
                                                              min =         50
                                                              avg =       50.0
                                                              max =         50

                                                Wald chi2(2)      =      17.06
Log restricted-likelihood = -3004.5851          Prob > chi2       =     0.0002

------------------------------------------------------------------------------
         lny |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          x1 |   .0484545   .0167536     2.89   0.004      .015618     .081291
          x2 |  -.0596082   .0201734    -2.95   0.003    -.0991472   -.0200691
       _cons |   2.954358   .0239952   123.12   0.000     2.907328    3.001387
------------------------------------------------------------------------------

------------------------------------------------------------------------------
  Random-effects Parameters |   Estimate   Std. Err.     [95% Conf. Interval]
----------------------------+-------------------------------------------------
```

```
area: Identity
                    sd(_cons)     .182708   .0167331      .1526867    .2186321
-----------------------------+------------------------------------------------
                sd(Residual)     .5015719   .0056661      .4905885    .5128011
------------------------------------------------------------------------------
LR test vs. linear model: chibar2(01) = 331.29        Prob >= chibar2 = 0.0000
file C:\Users\PAULCO~1\AppData\Local\Temp\ST_2bd8_000006.tmp saved
Number of observations in target dataset:

  20000

Number of clusters in target dataset:

  80

Number of simulations: 50
Each dot (.) represents 1 simulation(s).
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
..................................................    50

Number of bootstraps: 0
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
.
.         //MSE is calculated as squared difference between the true and CensusEB.
.         //Bring in the true poverty for this bootstrap replicate
.         rename Unit area

.         merge 1:1 area using `true´

    Result                           # of obs.
    -----------------------------------------
    not matched                             0
    matched                                80  (_merge==3)
    -----------------------------------------

.                 drop _m

.         //Get squared difference, after B bootstrap replicates this is our
.         //MSE estimate
.         forval a = 0/2{
  2.              gen sq_diff_fgt`a´ = ((avg_fgt`a´ - valuefgt`a´)^2)/$bs
  3.          }
```

4.5 Appendix

4.5.1 Model Selection Do-File

This do-file provides an example of the steps that may be followed to select a model for SAE, according to the discussion from section 4.2. The steps for model checking follow those illustrated by UCLA's Statistical Consulting Group (2022).

```
clear all
set more off

/*==============================================================================
Do-file prepared for SAE Guidelines
- Real world data application
- authors Paul Corral & Minh Nguyen
*==============================================================================*/

global main     "C:\Users\\`c(username)´\OneDrive\SAE Guidelines 2021\"
global section  "$main\3_Unit_level\"
global mdata    "$section\1_data\"
global survey "$mdata\survey_public.dta"
global census "$mdata\census_public.dta"

//global with candidate variables.
global myvar rural  lnhhsize age_hh male_hh  piped_water no_piped_water ///
no_sewage sewage_pub sewage_priv electricity telephone cellphone internet ///
computer washmachine fridge television share_under15 share_elderly ///
share_adult max_tertiary max_secondary HID_* mun_* state_*

version 15
set seed 648743

*==============================================================================
// End of preamble
*==============================================================================
//load in survey data
use "$survey", clear
        //Remove small incomes affecting model
        drop if e_y<1
        //Log shift transformation to approximate normality
        lnskew0 double bcy = exp(lny)
        //removes skeweness from distribution
        sum lny, d
        sum bcy, d

        //Data has already been cleaned and prepared. Data preparation and the creation
        // of eligible covariates is of extreme importance.
        // In this instance, we skip these comparison steps because the sample is
        // literally a subsample of the census.
        codebook HID //10 digits, every single one
        codebook HID_mun //7 digits every single one

        //We rename HID_mun
        rename HID_mun MUN
        //Drop automobile, it is missing
        drop *automobile* //all these are missing

        //Check to see if lassoregress is installed, if not install
        cap which lassoregress
        if (_rc) ssc install elasticregress

        //Model selection - with Lasso
```

```
gen lnhhsize = ln(hhsize)
lassoregress bcy  $myvar [aw=Whh], lambda1se epsilon(1e-10) numfolds(10)
local hhvars = e(varlist_nonzero)
global postlasso  `hhvars´

//Try Henderson III GLS
sae model h3 bcy $postlasso [aw=Whh], area(MUN)

//Rename HID_mun
rename MUN HID_mun

//Loop designed to remove non-significant covariates sequentially
forval z= 0.5(-0.05)0.05{
        qui:sae model h3 bcy `hhvars´ [aw=Whh], area(HID_mun)
        mata: bb=st_matrix("e(b_gls)")
        mata: se=sqrt(diagonal(st_matrix("e(V_gls)")))
        mata: zvals = bb´:/se
        mata: st_matrix("min",min(abs(zvals)))
        local zv = (-min[1,1])
        if (2*normal(`zv´)<`z´) exit

        foreach x of varlist `hhvars´{
                local hhvars1
                qui: sae model h3 bcy `hhvars´ [aw=Whh], area(HID_mun)
                qui: test `x´
                if (r(p)>`z´){
                        local hhvars1
                        foreach yy of local hhvars{
                                if ("`yy´"=="`x´") dis ""
                                else local hhvars1 `hhvars1´ `yy´
                        }
                }
                else local hhvars1 `hhvars´
                local hhvars `hhvars1´
        }
}

global postsign `hhvars´

//Henderson III GLS - model post removal of non-significant
sae model h3 bcy $postsign [aw=Whh], area(HID_mun)

//Check for multicollinearity, and remove highly collinear (VIF>3)
reg bcy $postsign [aw=Whh],r
cap drop touse  //remove vector if it is present to avoid error in next step
gen touse = e(sample)           //Indicates the observations used
vif                                             //Variance inflation factor
local hhvars $postsign
//Remove covariates with VIF greater than 3
mata: ds = _f_stepvif("`hhvars´","Whh",3,"touse")
global postvif `vifvar´

//VIF check
reg bcy $postvif [aw=Whh], r
vif

//Henderson III GLS - model post removal of non-significant
sae model h3 bcy $postvif [aw=Whh], area(HID_mun)
```

```
*================================================================================
// 2.5 Model checks
*================================================================================
```

```
        reg bcy $postvif
        predict cdist, cooksd
        predict rstud, rstudent

        reg bcy $postvif [aw=Whh]
        local KK = e(df_m)
        predict lev, leverage
        predict eps, resid
        predict bc_yhat, xb

        //Let´s take a look at our residuals
        //Notice there is a downward sloping line,which seems to be the smallest eps for that xb
        scatter eps bc_yhat
        //so we can see what the figure looks like
        sleep 15000
        // so there´s a bunch of small incomes that may be affecting our model!
        scatter eps bc_yhat if exp(lny)>1

/* https://stats.idre.ucla.edu/stata/dae/robust-regression/
Residual:  The difference between the predicted value (based on the regression ///
equation) and the actual, observed value.

Outlier:  In linear regression, an outlier is an observation with large ///
residual.  In other words, it is an observation whose dependent-variable ///
value is unusual given its value on the predictor variables.  An outlier may ///
indicate a sample peculiarity or may indicate a data entry error or ///
other problem.

Leverage:  An observation with an extreme value on a predictor variable is a ///
point with high leverage.  Leverage is a measure of how far an independent ///
variable deviates from its mean.  High leverage points can have a great ///
amount of effect on the estimate of regression coefficients.

Influence:  An observation is said to be influential if removing the ///
substantially changes the estimate of the regression coefficients.  ///
Influence can be thought of as the product of leverage and outlierness.

Cook´s distance (or Cook´s D): A measure that combines the information of ///
leverage and residual of the observation
*/

/* Rules of thumb:
Cooks -> >4/N, also according to "Regression Diagnostics: An Expository ///
Treatment of Outliers and Influential Cases, values over 1...

Abs(rstu) -> >2 We should pay attention to studentized residuals that exceed ///
+2 or -2, and get even more concerned about residuals that exceed +2.5 or ///
-2.5 and even yet more concerned about residuals that exceed +3 or -3.  ///

leverage ->     >(2k+2)/n
*/
        hist cdist, name(diag_cooksd, replace)
        hist lev, name(diag_leverage, replace)
        hist rstud, name(diag_rstudent, replace)
        twoway scatter cdist lev, name(diag_cooksd_lev, replace)

        lvr2plot, name(lvr2)
        rvfplot, name(rvf)

        sum cdist, d
        local max = r(max)
        local p99 = r(p99)
```

```
        reg lny $postvif [aw=Whh]
        local myN=e(N)
        local myK=e(rank)

        //We have influential data points...
        reg lny $postvif if cdist<4/`myN´ [aw=Whh]
        reg lny $postvif if cdist<`max´   [aw=Whh]
        reg lny $postvif if cdist<`p99´   [aw=Whh]
        gen nogo = abs(rstud)>2 & cdist>4/`myN´ & lev>(2*`myK´+2)/`myN´

*================================================================================
// Selecting the Alpha model
*================================================================================
        //Rename HID_mun
        cap rename HID_mun MUN
        //Henderson III GLS - add alfa model
        sae model h3 bcy $postvif if nogo==0 [aw=Whh], area(MUN) ///
        alfatest(residual) zvar(hhsize)

        des residual_alfa //The dependent variable for the alfa model

        // Macro holding all eligible vars
        unab allvars : $myvar
        //Macro with current variables
        local nogo $postvif

        //We want to only use variables not used
        foreach x of local allvars{
                local in = 0
                foreach y of local nogo{
                        if ("`x´"=="`y´")       local in=1
                }
                if (`in´==0) local A `A´ `x´
        }

        global A `A´ //macro holding eligible variables for alpha model

        lassoregress residual_alfa `A´ if nogo==0 [aw=Whh]

        local alfa = e(varlist_nonzero)
        global alfa `alfa´

        reg residual_alfa $alfa if nogo==0 [aw=Whh],r
        gen tousealfa = e(sample)

        //Remove vif vars
        mata: ds = _f_stepvif("$alfa","Whh",5,"tousealfa")

        global alfa `vifvar´

        //Alfa vars before removal of non-significant vars
        global beforealfa `alfa´

        local hhvars $alfa

        forval z= 0.9(-0.1)0.1{
                foreach x of varlist `hhvars´{
                        local hhvars1
                        qui: reg residual_alfa `hhvars´ [aw=Whh], r
                        qui: test `x´
```

```
                        if (r(p)>`z´){
                                local hhvars1
                                foreach yy of local hhvars{
                                        if ("`yy´"=="`x´") dis ""
                                        else local hhvars1 `hhvars1´ `yy´
                                }
                        }
                        else local hhvars1 `hhvars´
                        local hhvars `hhvars1´

                }
        }
        global alfavars `hhvars´

        //Henderson III Model with alpha model
        sae model h3 bcy $postvif if nogo==0 [aw=Whh], area(MUN) zvar($alfavars)

*================================================================================
// GLS model, one final removal of non-significant variables
*================================================================================
        //Loop designed to remove non-significant covariates sequentially
        local hhvars $postvif
        forval z= 0.5(-0.05)0.05{
                qui:sae model h3 bcy `hhvars´ if nogo==0 [aw=Whh], area(MUN) ///
                zvar($alfavars)
                mata: bb=st_matrix("e(b_gls)")
                mata: se=sqrt(diagonal(st_matrix("e(V_gls)")))
                mata: zvals = bb´:/se
                mata: st_matrix("min",min(abs(zvals)))
                local zv = (-min[1,1])
                if (2*normal(`zv´)<`z´) exit

                foreach x of varlist `hhvars´{
                        local hhvars1
                        qui:sae model h3 bcy `hhvars´ if nogo==0 [aw=Whh], area(MUN) ///
                        zvar($alfavars)
                        qui: test `x´
                        if (r(p)>`z´){
                                local hhvars1
                                foreach yy of local hhvars{
                                        if ("`yy´"=="`x´") dis ""
                                        else local hhvars1 `hhvars1´ `yy´
                                }
                        }
                        else local hhvars1 `hhvars´
                        local hhvars `hhvars1´
                }
        }

        global postalfa `hhvars´

*================================================================================
// SAVE the data with the pertinent covariates and other info
*================================================================================
sae model h3 bcy $postalfa if nogo==0 [aw=Whh], area(MUN) zvar($alfavars)

char _dta[rhs]   $postalfa
char _dta[alpha] $alfavars
char _dta[sel]   nogo

save "$mdata\mysvy.dta", replace
```

4.5.2 SAE Simulation Stage

This do-file provides an example of the steps commonly followed to produce the final CensusEB small area estimates. It follows the discussion from section 4.2.4.

```
clear all
set more off

/*==============================================================================
Do-file prepared for SAE Guidelines
- Real world data application
- authors Paul Corral & Minh Nguyen
*==============================================================================*/

global main     "C:\Users\\`c(username)´\OneDrive\SAE Guidelines 2021\"
global section  "$main\3_Unit_level\"
global mdata    "$section\1_data\"
global survey "$mdata\survey_public.dta"
global census "$mdata\census_public.dta"

version 15
local seed 648743

*==============================================================================
// End of preamble
*==============================================================================
use "$mdata\mysvy.dta", clear
        char list

        global hhmodel : char _dta[rhs]
        global alpha   : char _dta[alpha]
        global sel     : char _dta[sel]

//Add lnhhsize
use "$census"
gen lnhhsize = ln(hhsize)

tempfile census1
save `census1´

// Create data ready for SAE - optimized dataset
sae data import, datain(`census1´) varlist($hhmodel $alpha hhsize) ///
area(HID_mun) uniqid(hhid) dataout("$mdata\census_mata")

*==============================================================================
// Simulation -> Obtain point estimates
*==============================================================================
use "$mdata\mysvy.dta", clear
        drop if e_y<1
        drop if $sel==1
        rename MUN HID_mun
sae sim h3 e_y $hhmodel, area(HID_mun) zvar($alpha) mcrep(100) bsrep(0) ///
lnskew matin("$mdata\census_mata") seed(`seed´) pwcensus(hhsize) ///
indicators(fgt0 fgt1 fgt2) aggids(0 4) uniqid(hhid) plines(715)

*==============================================================================
// Simulation -> Obtain MSE estimates
*==============================================================================

use "$mdata\mysvy.dta", clear
        drop if e_y<1
        drop if $sel==1
        rename MUN HID_mun
sae sim h3 e_y $hhmodel, area(HID_mun) zvar($alpha) mcrep(100) bsrep(200) ///
lnskew matin("$mdata\census_mata") seed(`seed´) pwcensus(hhsize) ///
indicators(fgt0 fgt1 fgt2) aggids(0 4) uniqid(hhid) plines(715)
```

```
save "$mdata\mySAE.dta", replace
```

References

Banerjee, Abhijit V, Angus Deaton, Nora Lustig, Kenneth Rogoff, and Edward Hsu (2006). "An Evaluation of World Bank Research, 1998-2005". In: *Available at SSRN 2950327*.

Battese, George E., Rachel M. Harter, and Wayne A. Fuller (1988). "An Error-Components Model for Prediction of County Crop Areas Using Survey and Satellite Data". In: *Journal of the American Statistical Association* 83.401, pp. 28–36. ISSN: 01621459. URL: http://www.jstor.org/stable/2288915.

Bikauskaite, A, I Molina, and D Morales (2020). "Multivariate Mixture Model for Small Area Estimation of Poverty Indicators".

Boonstra, Harm Jan (2015). "Package 'hbsae'". In: *R Package Version* 1.

Boonstra, Harm Jan and Grzegorz Baltissen (2021). "mcmcsae: Markov Chain Monte Carlo Small Area Estimation". In: *R package version 0.7.0*. URL: https:https://CRAN.R-project.org/package=mcmcsae..

Box, GEP and DR Cox (1964). "An Analysis of Transformations". In: *Journal of the Royal Statistical Society, Series B* 26, pp. 211–252.

Corral, Paul, Kristen Himelein, Kevin McGee, and Isabel Molina (2021). "A Map of the Poor or a Poor Map?" In: *Mathematics* 9.21. ISSN: 2227-7390. DOI: 10.3390/math9212780. URL: https://www.mdpi.com/2227-7390/9/21/2780.

Corral, Paul, Isabel Molina, and Minh Cong Nguyen (2021). "Pull Your Small Area Estimates up by the Bootstraps". In: *Journal of Statistical Computation and Simulation* 91.16, pp. 3304–3357. DOI: 10.1080/00949655.2021.1926460. URL: https://www.tandfonline.com/doi/abs/10.1080/00949655.2021.1926460.

Elbers, C and R van der Weide (2014). "Estimation of Normal Mixtures in a Nested Error Model with an Application to Small Area Estimation of Poverty and Inequality". In: *World Bank Policy Research Working Papers* 6962.

Elbers, Chris, Tomoki Fujii, Peter Lanjouw, Berk Ozler, Wesley Yin, et al. (2007). "Poverty Alleviation through Geographic Targeting: How Much Does Disaggregation Help?" In: *Journal of Development Economics* 83.1, pp. 198–213.

Elbers, Chris, Jean O Lanjouw, and Peter Lanjouw (2003). "Micro–level Estimation of Poverty and Inequality". In: *Econometrica* 71.1, pp. 355–364.

Elbers, Chris, Jean Olson Lanjouw, and Peter Lanjouw (2002). "Micro-level Estimation of Welfare". In: *World Bank Policy Research Working Paper* 2911.

Foster, James, Joel Greer, and Erik Thorbecke (1984). "A Class of Decomposable Poverty Measures". In: *Econometrica: Journal of the Econometric Society* 52, pp. 761–766.

González-Manteiga, Wenceslao, Maria J Lombardía, Isabel Molina, Domingo Morales, and Laureano Santamaría (2008). "Bootstrap Mean Squared Error of a Small-area EBLUP". In: *Journal of Statistical Computation and Simulation* 78.5, pp. 443–462.

Guadarrama, María, Isabel Molina, and JNK Rao (2018). "Small Area Estimation of General Parameters under Complex Sampling Designs". In: *Computational Statistics & Data Analysis* 121, pp. 20–40.

Harvey, Andrew C (1976). "Estimating Regression Models with Multiplicative Heteroscedasticity". In: *Econometrica: Journal of the Econometric Society* 44, pp. 461–465. URL: https://www.jstor.org/stable/1913974?casa_token=rLCctNWylroAAAAA:3-S22Hv84lze8TUU-3nyFgDGZsrxkAoCZHwUVKO99Qynn_zXmr8-3kmdqFxdLxRBJT8KiE-xYwfoI5-gzFA9RF3N-KHS3g7na1Soi8WbYp_QBcPfjU7m&seq=1.

Haslett, Stephen J. (2016). "Small Area Estimation Using Both Survey and Census Unit Record Data". In: *Analysis of Poverty Data by Small Area Estimation*. John Wiley & Sons, Ltd. Chap. 18, pp. 325–348. ISBN: 9781118814963. DOI: https://doi.org/10.1002/9781118814963.ch18. eprint: https://

onlinelibrary.wiley.com/doi/pdf/10.1002/9781118814963.ch18. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118814963.ch18.

Henderson, Charles R (1953). "Estimation of Variance and Covariance Components". In: *Biometrics* 9.2, pp. 226–252.

Hobza, Tomáš and Domingo Morales (2013). "Small Area Estimation under Random Regression Coefficient Models". In: *Journal of Statistical Computation and Simulation* 83.11, pp. 2160–2177.

Kreutzmann, Ann-Kristin, Sören Pannier, Natalia Rojas-Perilla, Timo Schmid, Matthias Templ, and Nikos Tzavidis (2019). "The R Package emdi for Estimating and Mapping Regionally Disaggregated Indicators". In: *Journal of Statistical Software* 91.7, pp. 1–33. DOI: 10.18637/jss.v091.i07. URL: https://www.jstatsoft.org/index.php/jss/article/view/v091i07.

Marhuenda, Yolanda, Isabel Molina, Domingo Morales, and JNK Rao (2017). "Poverty Mapping in Small Areas under a Twofold Nested Error Regression Model". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 180.4, pp. 1111–1136. DOI: 10.1111/rssa.12306.

Masaki, Takaaki, David Newhouse, Ani Rudra Silwal, Adane Bedada, and Ryan Engstrom (2020). "Small Area Estimation of Non-monetary Poverty with Geospatial Data". In: *World Bank Policy Research Working Paper* 9383.

Molina, Isabel (2019). *Desagregación De Datos En Encuestas De Hogares: Metodologías De Estimación En áreas Pequeñas*. CEPAL. URL: https://repositorio.cepal.org/handle/11362/44214.

Molina, Isabel and Yolanda Marhuenda (2015). "Sae: An R Package for Small Area Estimation". In: *The R Journal* 7.1, pp. 81–98.

Molina, Isabel and JNK Rao (2010). "Small Area Estimation of Poverty Indicators". In: *Canadian Journal of Statistics* 38.3, pp. 369–385.

Nguyen, Minh Cong, Paul Corral, João Pedro Azevedo, and Qinghua Zhao (2018). "sae: A Stata Package for Unit Level Small Area Estimation". In: *World Bank Policy Research Working Paper* 8630.

Peterson, Ryan A and Joseph E Cavanaugh (2019). "Ordered Quantile Normalization: A Semiparametric Transformation Built for the Cross-validation Era". In: *Journal of Applied Statistics* 47.13-15, pp. 2312–2327. DOI: 10.1080/02664763.2019.1630372. URL: https://www.tandfonline.com/action/showCitFormats?doi=10.1080/02664763.2019.1630372.

Rao, JNK and Isabel Molina (2015). *Small Area Estimation*. 2nd. John Wiley & Sons.

Rojas-Perilla, Natalia, Sören Pannier, Timo Schmid, and Nikos Tzavidis (2020). "Data-driven Transformations in Small Area Estimation". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 183.1, pp. 121–148.

Tzavidis, Nikos, Li-Chun Zhang, Angela Luna, Timo Schmid, and Natalia Rojas-Perilla (2018). "From Start to Finish: A Framework for the Production of Small Area Official Statistics". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 181.4, pp. 927–979.

UCLA: Statistical Consulting Group (2022). *Robust Regression: Stata Data Analysis Examples*. URL: https://stats.oarc.ucla.edu/stata/webbooks/reg/chapter2/stata-webbooksregressionwith-statachapter-2-regression-diagnostics/.

Van der Weide, Roy (2014). "GLS Estimation and Empirical Bayes Prediction for Linear Mixed Models with Heteroskedasticity and Sampling Weights: A Background Study for the POVMAP Project". In: *World Bank Policy Research Working Paper* 7028. URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2495175.

Zhao, Qinghua (2006). *User Manual for Povmap*. URL: http://siteresources.%20worldbank.%20org/INTPGI/Resources/342674-1092157888460/Zhao_%20ManualPovMap.%20pdf.

Chapter 5

Poverty Mapping in Off-Census Years

Considerable attention has been given to produce reliable poverty maps in off-census years. An updated poverty map is increasingly becoming an essential resource to improve the targeting of many social protection programs around the world, which was underscored by the rapid onset and widespread impact of the COVID-19 crisis. Even in the best-case scenarios, poverty maps that rely on unit-level small area estimation techniques combining a census and survey can only be obtained once a decade. In off-census years, the typical small area approach applied has been an area-level model, such as a Fay-Herriot or a sub-area-level model, such as the one proposed by Torabi and Rao (2014). Nevertheless, since the perceived gains in precision from area-level models are less than stellar, methods that combine unit- and area-level models have been proposed (see Nguyen 2012; Lange, Pape, and Pütz 2018; Masaki et al. 2020). These models are here called unit-context models and, although appealing, seem to yield considerably biased estimates.

Beyond the small area estimation literature, the machine learning literature has made several contributions to poverty mapping. Recent research in this area include Chi et al. (2021) and Jean et al. (2016). The authors of these two papers created highly disaggregated poverty maps by modeling the direct estimates of an asset index at a very low geographical level (e.g. villages or enumeration areas) using satellite-derived covariates. The authors of those papers rely on machine learning approaches, such as gradient boosting and ridge regression, to obtain estimates for small areas. These models provide point estimates of poverty at a low geographical level, although they do not necessarily provide an adequate estimate of the method's noise. The methods are attractive since they present the possibility of producing a poverty map even when a contemporaneous or reliable census does not exist.

5.1 Unit-Context Models

Unit-context models attempt to model the population's welfare distribution using only area-level covariates. More specifically, unit-context models combine unit and area-level information to model the transformed household-level welfare (unit) using only area-level covariates (context). Since unit-context models do not require census microdata, they have been proposed as an alternative approach for the case when the available census microdata is too outdated to be considered for use under the conventional model-based methods that include unit-level covariates.[1]

[1] Another approach for cases where the census is outdated is to fit a unit-level model considering only the covariates with low (or even null) variability along time. This approach reduces (or may even solve) the problem of using an outdated census.

Previous applications of unit-context models for small area estimation were proposed by Arora, Lahiri, and Mukherjee (1997), who studied the number of trips home students have taken, and by Efron and Morris (1975), who looked at batting averages and toxoplasmosis cases. In these applications, the method appears to work well, although in both studies, the model with aggregate covariates is used to produce estimates of the area means of the dependent variable in the model (no transformation is considered). In the context of poverty, the target poverty indicators are typically complex nonlinear functions of the dependent variables in the model. Hence, properly replicating the full welfare distribution is essential as noted in Figure 4.1. At the area level, this is complicated since household characteristics are not used in the model. Thus very little, if any, of the variation in welfare across households in the area is explained. If simple area means of the welfare variable of interest are the target, then, due to the assumptions embedded into the nested error models used in chapter 4, a transformation (such as log or log-shift) of the welfare variable is used as the dependent variable in the model. Consequently, the area means of the untransformed welfare variable are desired, which are then means of exponentials of the dependent variable. As is illustrated in the next section, when estimating indicators that are nonlinear functions of the dependent variables in the model, unit-context models will likely produce small area estimators of poverty with substantial bias.

Nguyen (2012) first considered unit-context models for poverty estimation in an application for Vietnam. In this application, the dependent variable was the household-level logarithm of per capita expenditure from the Vietnam Household Living Standard Survey from 2006, whereas all covariates are commune-level means obtained from a dated (1999) census. Nguyen (2012) obtains ELL estimates of poverty for small areas under that model and compares the performance with typical ELL poverty estimates obtained using unit-level covariates from the Vietnam Household Living Standard Survey from 2006 and the 2006 Rural Agriculture and Fishery Census. The author finds that provinces and districts hovering around the middle of the distribution suffered considerable re-rankings across methods. However, those at the top and the bottom were relatively stable.

A similar approach to the one from Nguyen (2012) was presented by Lange, Pape, and Pütz (2018) as an alternative in cases when census and survey data are not from similar periods. However, the same inefficiency issues noted in Chapter 4 regarding ELL estimates would likely persist when considering a model using only area-level covariates. Improvements to the approach were seemingly made by Masaki et al. (2020) by taking measures to address some of the shortcomings of a standard ELL approach and to obtain EB estimators from Molina and Rao (2010). The authors conduct a design-based validation study using census data for Sri Lanka and Tanzania for a wealth index constructed by principal component analysis and suggest that the use of EB improves precision over ELL when implementing unit-context models.

Although the unit-context approach is attractive in that it does not require a contemporaneous census and can readily accommodate variables extracted from the emerging fields related to geospatial analysis, there are serious concerns about bias in unit-context estimators, as noted in Corral et al. (2021) as well as the concerns raised in the following section. The MSE from unit-context models is also likely to be incorrectly estimated since the required parametric bootstrap procedure assumes the underlying model using only area-level characteristics to be correct. In other words, the unit-context model's assumptions require household-level welfare to not depend on household-specific characteristics, which is unlikely to be the case. Incorrect MSE estimates risk presenting a map with considerable bias as being overly precise. Therefore, based on the currently available evidence, area-level models, like Fay-Herriot (Ch. 3), are generally preferred over unit-context models (see the following section for more details).

In cases where neither area- nor unit-level models are advised due to data limitations, no clear consensus

has emerged on the best path forward or if one even exists. In evaluating alternatives, practitioners should choose methods which rely on assumptions that are realistic to the circumstances in which the model will be employed, which are approximately unbiased (or its bias does not exceed a certain limit), and for which an accurate method exists to measure the small area estimators' MSE. In cases where the MSE cannot be adequately estimated, then at least it should be known in which (realistic) scenarios the approach has limited bias. If these conditions cannot be reasonably met, it is preferable to not produce a map than to produce one with potentially biased estimates, or one in which precision is overestimated, or most worrisome, both. In the next section, the limitations of unit-context models are discussed.

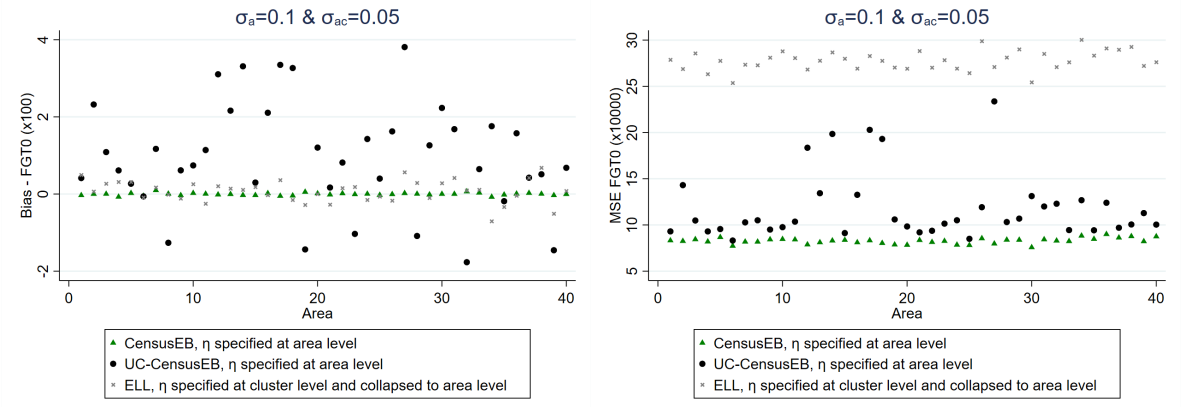
5.1.1 Limitations of Unit-Context Models

Based on results from a validation study using model- and design-based simulations, Corral et al. (2021) conclude unit-context models, like those presented in Masaki et al. (2020), Lange, Pape, and Pütz (2018) and Nguyen (2012) are not recommended except under exceptional cases due to the high likelihood of bias in estimates.[2]

Nguyen's (2012) application of a unit-context model to estimate poverty in small areas in Vietnam already hints toward potential problems of the unit-context approach. The author compares the results from unit-context models to those obtained via a standard unit-level method, ELL, and finds considerable changes in rankings. Nguyen (2012) finds that differences in rankings are largest for locations around the middle of the distribution.

Despite the use of EB, the unit-context application by Masaki et al. (2020), also provides hints of potential methodological problems with the approach. A ratio procedure was used to benchmark the values to ensure alignment between direct estimates at the level of representativeness and the estimates obtained by the approach. The need to benchmark indicates considerable discrepancies between the sum of estimated totals at the lower level and the estimated total at the higher level. The need to benchmark also suggests that the model's assumptions are not satisfied.

Figure 5.1: Empirical Bias and MSE for CensusEB based on a unit-level model and CensusEB based on a unit-context model (UC-CensusEB) and ELL FGT0 estimates from model based simulations



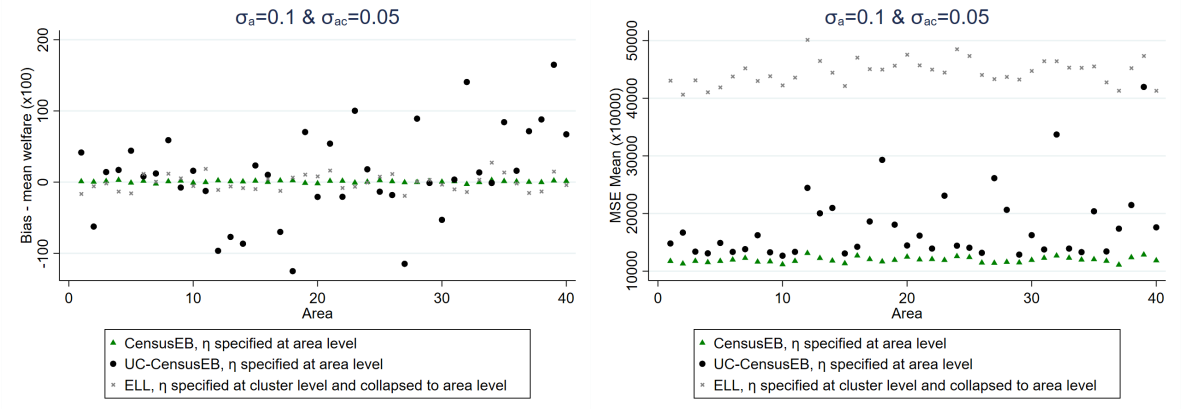
Source: Corral et al. (2021). The figure is obtained from simulations based on 10,000 populations and samples as specified in Corral et al. (2021). The simulations illustrate that unit context (UC) model may yield FGT0 estimates that are upward biased and with MSEs that could be several orders of magnitude above those of CensusEB estimates, based on the analogous unit-level model, and for some areas may be almost as inefficient as ELL.

[2]The method presents advantages over the traditional Fay-Herriot (Fay and Herriot 1979) models: 1) it may be an alternative when there are multiple locations with very small samples, for which the sampling variance of the direct estimator (used on the left-hand side of the Fay-Herriot model) becomes 0, and 2) it may be used to obtain multiple indicators from a single model under reversible transformations.

Unit-context models appear to yield upward biased FGT0 estimates in model-based simulations, as presented in Figure 5.1 (most areas show bias above 0). Since unit-context models are special cases of the models used in ELL and EB procedures (see section 4.4.1), but without household-level characteristics, the between-household variation of welfare is not adequately explained by the model. Corral et al. (2021) suggest that part of the observed bias comes from this misspecification, with effects similar to omitted variable bias (OVB) (see the appendix in Corral et al. 2021). Despite the bias, the empirical MSE of unit-context models seems to outperform that of ELL estimates (Figure 5.1).

Like traditional unit-level models, unit-context models also assume normally distributed errors and departures from normality may also produce bias (which might offset or compound the previous bias). This is why in section (4.2.2), a considerable emphasis is placed on data transformation, all with the aim of approximating the normality assumption. Because of the poor model fit and potential deviations from normality, unit-context models also display considerable bias when estimating mean welfare (see Figure 5.2).

Figure 5.2: Empirical Bias and MSE for CensusEB based on a unit-level model and CensusEB based on a unit-context model (UC-CensusEB) and ELL mean welfare estimates from model based simulations



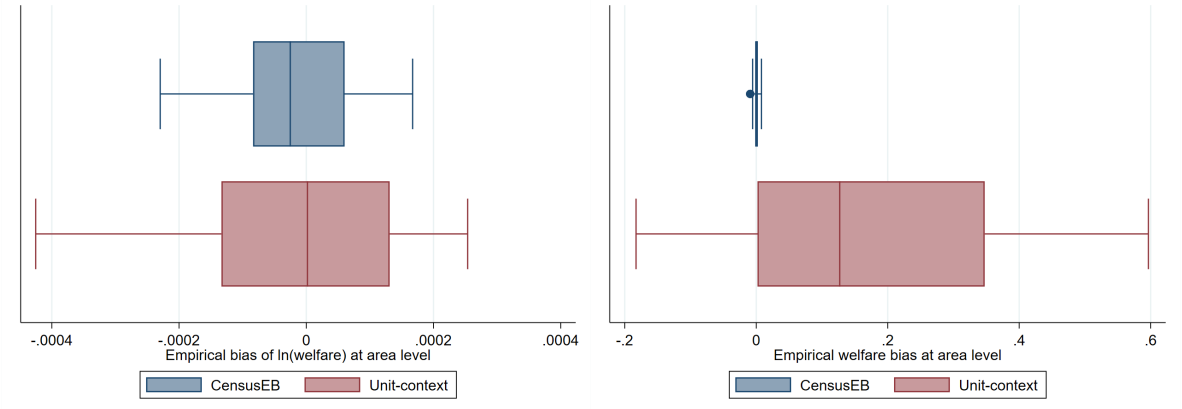
Source: Corral et al. (2021). The figure is obtained from simulations based on 10,000 populations and samples as specified in Corral et al. (2021). The simulations illustrate that unit-context models may yield mean welfare estimates that are considerably biased and with MSEs that could be several orders of magnitude above those of CensusEB estimates based on the analogous unit-level model and for some areas may be as inefficient as ELL.

In the simulation results displayed in Figure 5.2, the dependent variable is the natural log of welfare. Nevertheless, what practitioners are often interested in is welfare, and hence need to reverse the log transformation. Since $\exp\left[E\left(\ln y\right)\right] \neq E\left(y\right)$. The log linear nested error model $\left(\ln y_{ch} = x_{ch}\beta + \eta_c + e_{ch}\right)$ implies that $y_{ch} = \exp\left(x_{ch}\beta\right)\exp\left(\eta_c\right)\exp\left(e_{ch}\right)$, then $E\left(y_{ch}|x_{ch}\right) = \exp\left(x_{ch}\beta\right)E\left[\exp\left(\eta_c\right)\right]E\left[\exp\left(e_{ch}\right)\right]$. Since it is assumed that $\eta_c \sim N\left(0,\sigma_\eta^2\right)$ and $e_{ch} \sim N\left(0,\sigma_e^2\right)$, then $E\left[\exp\left(\eta_c\right)\right] = \exp\left(0.5\sigma_\eta^2\right)$ and $E\left[\exp\left(e_{ch}\right)\right] = \exp\left(0.5\sigma_e^2\right)$. Because unit-context models do not include household-level covariates, the resulting estimate of σ_e^2 is likely much greater than the true σ_e^2. This implies that unit-context models may yield an upward biased prediction of mean household welfare for the area despite the use of EB methods since unbiasedness is assured for the area means of $\ln y$ but not for those of y following back transformation.

To illustrate this last point, a simulation where the modeling and estimates are obtained using the same source is conducted (see Appendix 5.5.1.2). This is done to remove the potential source of bias noted in the Appendix of Corral et al. (2021). The empirical bias of the estimators in each of the 40 areas in the census data set are represented in box-plots. First, note that, when estimating the area means of the dependent variable (similar to when no transformation is taken), the biases of the CensusEB estimators based on the unit-context model are not that large (Figure 5.3, left). However, suppose one estimates the

means of the untransformed welfare variable (exponential of the model's dependent variable). In that case, the incorrect estimate of $\sigma_\eta^2 + \sigma_e^2$ plays a considerable role, and thus the estimates of mean welfare based on unit-context models are upward biased despite the use of EB methods (Figure 5.3, right). As shown in Figure 5.3 (right), the average bias of CensusEB estimators based on unit-context models is considerable. For some areas, the bias may be over 800 times the bias of the CensusEB estimators based on the analogous unit-level models.

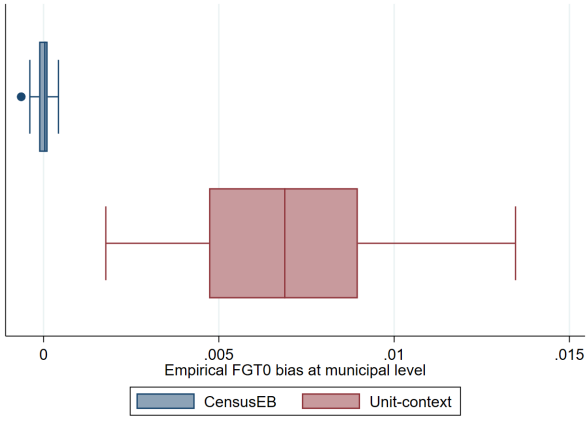
Figure 5.3: Empirical Bias of CensusEB prediction of the area means of $\ln(y_{ais})$, left, and y_{ais}, right, obtained from a unit-level model and from a unit-context model



Note: Simulation based on 5,000 populations generated as described in section 5.5.1. The model is fit to the whole population and then welfare is simulated for the same population to yield 5,000 EB predictors for each area. Box-plots represent the empirical bias for the 40 areas in the census data. Do-file to replicate simulation can be seen in section 5.5.1.1.

The source of bias discussed, i.e. due to biased estimators of the variance components under unit-context models, is in addition to the bias due to the difference between the sample and census means of the covariates noted in Corral et al. (2021). The latter source of bias is not relevant here as it is related to sampling, and in this case, the entire census is taken as the sample. The bias due to variance components will affect the estimation of mean welfare and headcount poverty. Consequently, estimates of headcount poverty under unit-context models will be biased also because of poor model fit. As shown in Figure 5.4, the estimates from the unit-context models are upward biased for all areas, which occurs because the full distribution of the population is not accurately replicated under unit-context models.

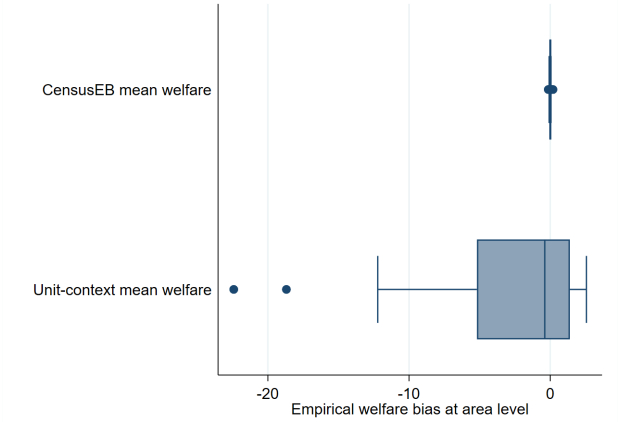
Figure 5.4: Empirical Bias of CensusEB predictors of FGT0 under unit-level model and under a unit-context model



Note: Simulation based on 5,000 populations generated as shown in section . The model is fit on the entire population and then welfare is simulated on the same population to yield 5,000 EB predictors for each area. Box-plot represents the empirical bias for the 40 areas in the census data. Do-file to replicate simulation can be seen in section 5.5.1.1.

A potentially significant difference between the results presented in Figures 5.3 and 5.4, and the applications of Masaki et al. (2020) and Lange, Pape, and Pütz (2018) is the considerably lower R^2 of the unit-context models presented here. To address the low R^2, the simulation experiment is repeated using a slightly modified data generating process leading to an increase of the model's R^2 (Technical Annex 5.5.1). This modification leads to an R^2 of the unit-context model between 0.15 and 0.20, while for the unit-level model the R^2 exceeds 0.60. This adjustment in the data generating process reduces the bias in the estimators of mean welfare based on the unit-context model (Figure 5.5), but the direction of the bias is different from that of the original simulation (Figure 5.2). This change of direction of the bias also occurs when estimating poverty rates, and the direction of the bias seems to change with the poverty threshold. Estimators based on unit-context models are upward biased for poverty rates under a threshold of 13. If this threshold is increased to 28, the unit-context estimators of the poverty rate become downward biased for every area (Figure 5.6). These results illustrate a crucial shortcoming of unit-context models: in a given application, a practitioner cannot know which will be the direction and the magnitude of the bias. Unit-context models seldom reproduce the true distribution accurately, and it is difficult to know in which cases they work properly.

Figure 5.5: Empirical Bias of CensusEB predictors under unit-level model and under a unit-context model

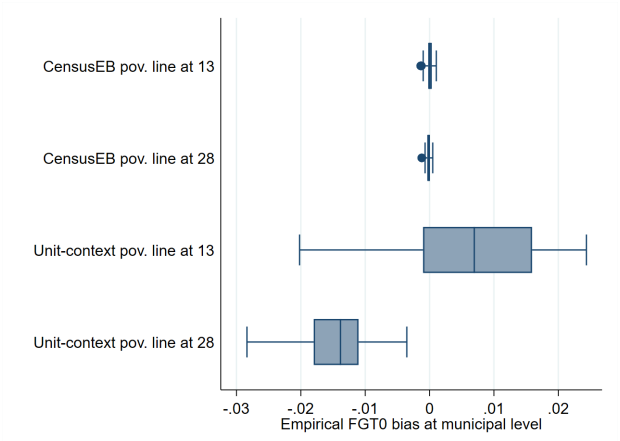


Note: Simulation based on 5,000 populations generated as shown in section 5.5.1. The model is fit on the entire population and then welfare is simulated on the same population to yield 5,000 EB predictors for each area. Box-plot represents the empirical bias for the 40 areas in the census data. Do-file to replicate simulation can be seen in section 5.5.1.2.

Building upon the previous simulation, a final simulation is conducted to observe how unit-context models perform under 99 different poverty thresholds. The thresholds correspond to the 99 percentiles of the first population generated in the simulation (detailed in Appendix 5.5.2). In contrast to the previous simulations, a sample from the population is taken here. This is done to compare unit-context estimators to estimators based on an FH model (Ch. 3).

As expected, the unit-level CensusEB estimator yields the smallest bias across all areas and poverty lines (Figure 5.7). The shape of the bias for unit-context models suggests that it may perform well for some poverty lines, while it will perform poorly for other poverty lines. This agrees with what is noted in figure 5.6, where the method could work for a given poverty line but show substantial bias at a different poverty threshold. Moreover, for many areas and poverty lines, unit-context models seem to perform worse than FH models, not just with respect to bias but also in terms of empirical MSE (Figure 5.8).

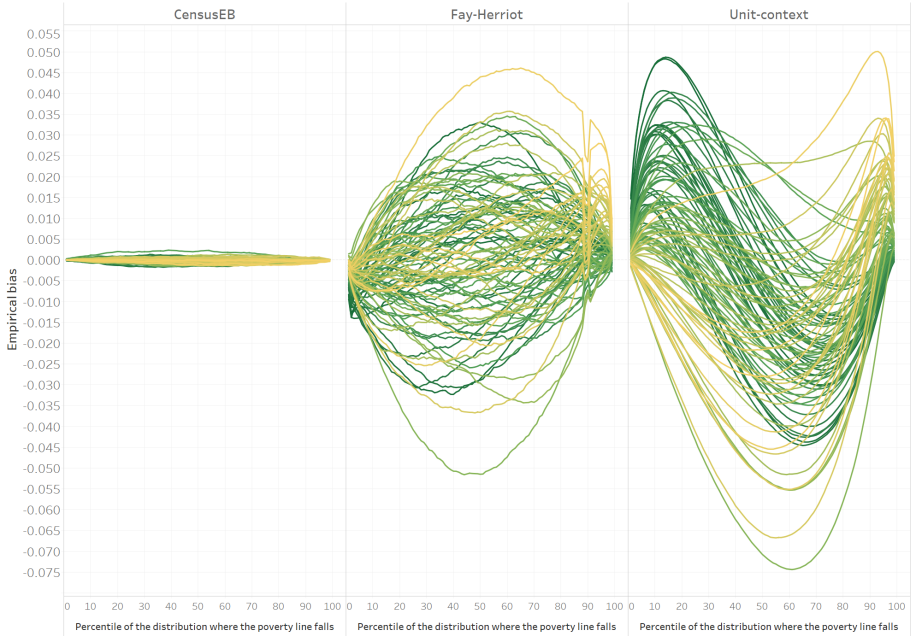
Figure 5.6: Empirical Bias of CensusEB predictors of FGT0 under unit-level model and under a unit-context model



Note: Simulation based on 5,000 populations generated as shown in section 5.5.1. The model is fit on the entire population and then welfare is simulated on the same population to yield 5,000 EB predictors for each area. Box-plot represents the empirical bias for the 40 areas in the census data. Do-file to replicate simulation can be seen in section 5.5.1.2.

The average absolute empirical bias can be used to rank the different methods. Across all 99 poverty lines, unit-context models have the highest average absolute empirical bias (Figure 5.9).[3] Additionally, the average empirical MSE (across the 100 areas) for each poverty line is surprisingly close between Fay-Herriot and unit-context models. However, neither dominates the other (Figure 5.10), supporting the conclusion that unit-context models should be used with caution and only when unit-level and area-level options are not feasible. If unit-context models are used, all caveats should be made clear to the users of the resulting estimates.

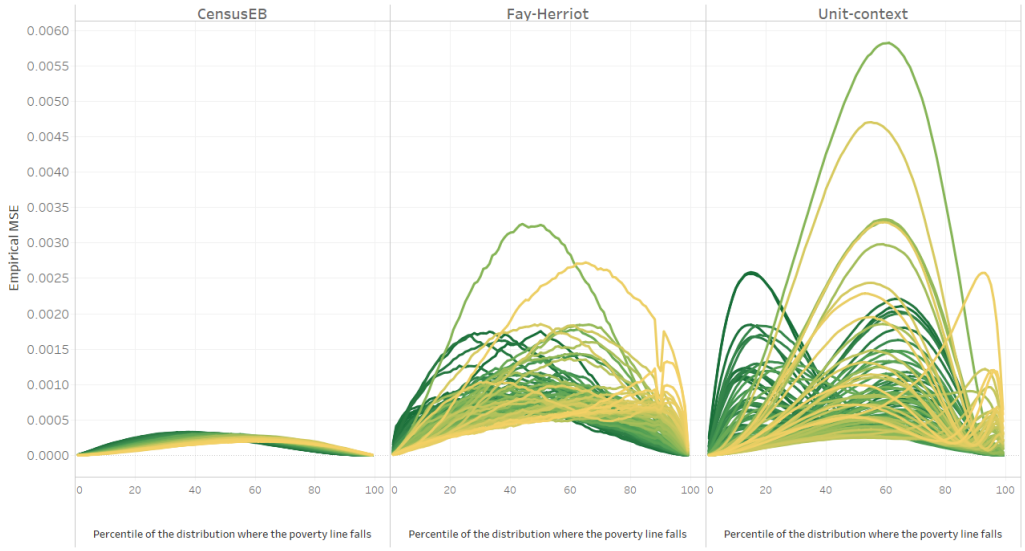
Figure 5.7: Empirical bias of different methods for each area and poverty line



Note: Simulation based on 1,000 populations generated as described in this section. Each line corresponds to one of the 100 areas. The x-axis represents the percentile on which the poverty line falls on, and the y-axis is the empirical bias. In instances where direct estimates have a variance of 0 and hence FH estimates cannot be obtained, the direct estimate is used as the FH estimate for that area. The do-file to replicate these results can be found in 5.5.2.1.

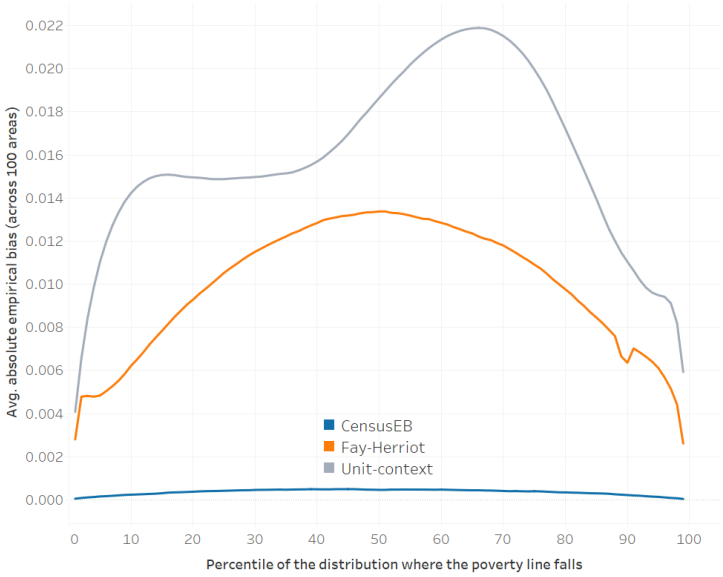
[3]The average absolute empirical bias is the average across areas of the area-specific absolute biases.

Figure 5.8: Empirical MSE of different methods for each area and poverty line



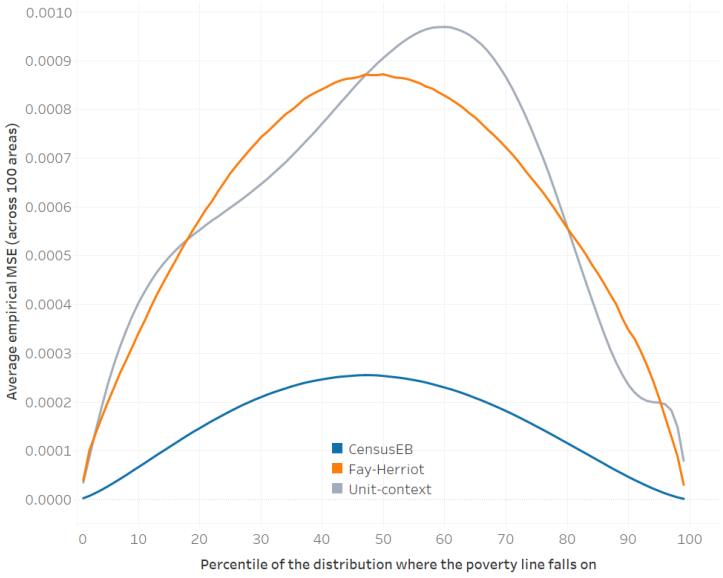
Note: Simulation based on 1,000 populations generated as described in this section. Each line corresponds to one of the 100 areas. The x-axis represents the percentile on which the poverty line falls on, and the y-axis is the empirical MSE. In instances where direct estimates have a variance of 0 and hence FH estimates cannot be obtained, the direct estimate is used as the FH estimate for that area. The do-file to replicate these results can be found in 5.5.2.1.

Figure 5.9: Average absolute empirical bias of different methods for each poverty line



Note: Simulation based on 1,000 populations generated as described in this section. Each line corresponds to the average across 100 areas of the absolute empirical bias for each method. The x-axis represents the percentile on which the poverty line falls on, and the y-axis is the average absolute empirical bias. In instances where direct estimates have a variance of 0 and hence FH estimates cannot be obtained, the direct estimate is used as the FH estimate for that area. The do-file to replicate these results can be found in 5.5.2.1.

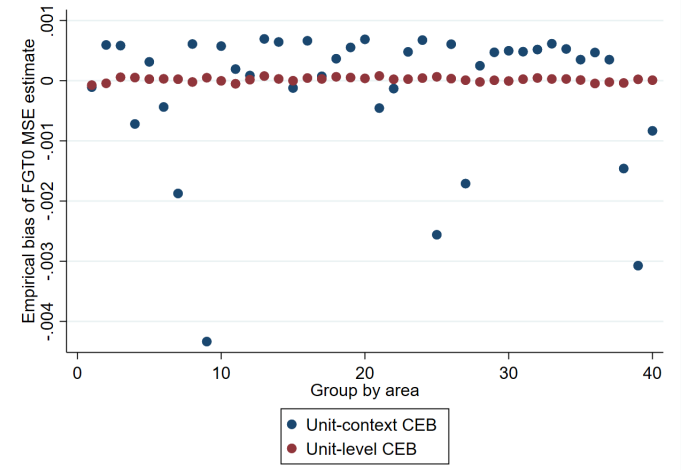
Figure 5.10: Average empirical MSE of different methods for each poverty line



Note: Simulation based on 1,000 populations generated as described in this section. Each line corresponds to the average across 100 areas of the mean squared error for each method. The x-axis represents the percentile on which the poverty line falls on, and the y-axis is the average MSE. In instances where direct estimates have a variance of 0 and hence FH estimates cannot be obtained, the direct estimate is used as the FH estimate for that area. The do-file to replicate these results can be found in 5.5.2.1.

In addition to bias, there are concerns with an overstatement of precision from a unit-context model. If welfare depends on household-level characteristics (which is expected) the MSE estimates of CensusEB estimators based on unit-context models, obtained using the parametric bootstrap procedure of González-Manteiga et al. (2008), would not be appropriate as it may underestimate the method's true MSE by a considerable amount. The parametric bootstrap procedure under the unit-context model is done considering that the unit-context model is the true data generating process and thus generates bootstrap populations from that model. However, this assumption is shown to not hold in the simulation experiment, leading to inaccurate MSE estimates (Figure 5.11). Consequently, under unit-context models, one can produce biased poverty estimates but may be presented as being very precise.

Figure 5.11: Bias of parametric MSE estimate by method



Note: Simulation based on 1,000 populations generated as described in this section. Under each population, MSE estimates are obtained using the parametric bootstrap procedure of González-Manteiga et al. (2008). The empirical MSE is compared to the estimated MSE to obtain the bias of the MSE estimates of each method.

Summarizing concerns with unit-context models, given the limitations of unit-context models, even under an ideal scenario where the model is fit to the whole set of population data and EB estimates are obtained also using the population data, unit-context models should be considered only if unit-level and area-level options have been exhausted. While it is possible that in real world applications, there may be bias from multiple sources which offset each other out for a given poverty threshold, there is no way for an analyst to determine if biases are canceling or compounding each other as the direction of the bias is not known *a priori*. A likely overstatement of precision further complicates this unknown bias. This situation recalls an original concern levied against ELL by Banerjee et al. (2006) where the authors state: "What we are most concerned about is the possibility that the Bank is making very attractive poverty maps, whose precision is not known, but which come with statements of precision that we suspect may be seriously misleading. In the worst case scenario, a map that may be worthless is presented as one that is extremely precise" (Banerjee et al. 2006, pg 61).

5.2 Gradient Boosting for Poverty Mapping

With the advent of increased computing power, machine learning approaches have gained popularity in the literature as well as in policy circles. For example, poverty estimates for small areas derived from a gradient boosting application by Chi et al. (2021) guided the expansion of social protection in Nigeria; specifically, the estimates were used as an input to the Rapid Response Register for the COVID-19 Cash Transfer Project.[4]

Gradient boosting methods rely on, first, creating a linear fit (usually a constant term) to the data at hand and then fitting a new model onto the residuals. In contrast to ensemble techniques, the multiple fits are not averaged. Instead, the predicted residuals (scaled by a learning rate) are added to the previous step's prediction, successively taking small steps using the same or different covariates at each step toward the final prediction. This process repeats until the requested number of predictions are completed, or there is no longer a gain in prediction.[5] A complete exposition of the approach is beyond the scope of the Guidelines, and interested readers can refer to Natekin and Knoll (2013) or Chen and Guestrin (2016) for a more nuanced description of the approach.

A validation approach similar to the one done by Corral et al. (2021) is implemented to test how well gradient boosting works in a poverty mapping context. Specifically, 500 samples taken from the census created from the *Mexican Intracensal Survey* are used to conduct a design-based simulation experiment to validate the method.[6] The *Mexican Intracensal Survey* is uniquely suited for this validation since it includes a measure of income at the household level and is representative at the municipal level and localities with 50,000 or more inhabitants. The survey was modified to obtain a census dataset of 3.9 million households and 1,865 municipalities (Corral et al. 2021). Using a sampling approach similar to the one used for Living Standards Measurement Study (LSMS) surveys, 500 samples are drawn from the resulting census.

In each of the 500 samples, a model is fit. The model's dependent variable is the direct estimator of the headcount poverty rate at the PSU level (or municipality), and PSU aggregates (or municipality) from the census data are used as covariates. The models are fit using the XGBoost algorithm available in Python and R.[7] Introduced by Chen and Guestrin (2016), XGBoost is a scalable machine learning

[4] Blumenstock et al. (2021)https://blogs.worldbank.org/opendata/using-big-data-and-machine-learning-locate-poor-nigeria

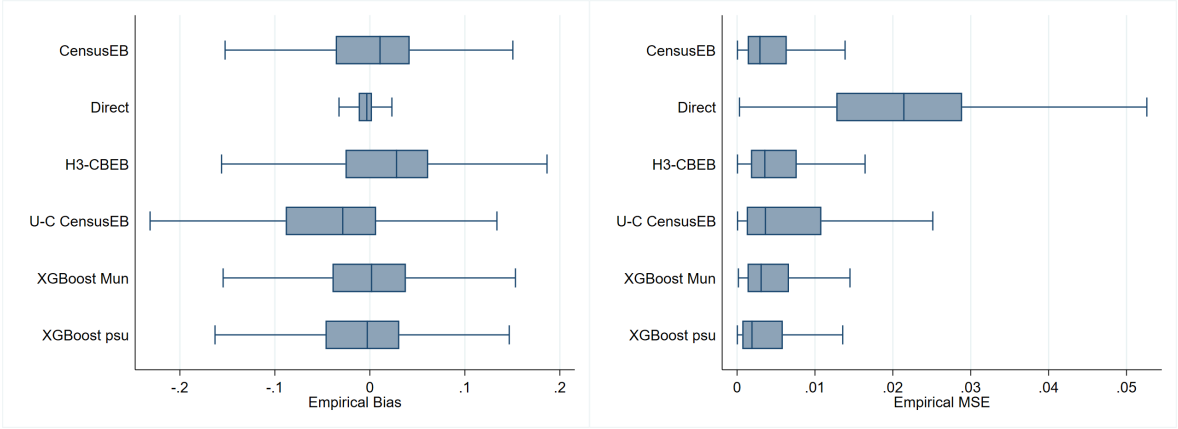
[5] The method relies on a squared-error loss function where the sequential fits are added until there is no improvement in the loss function. For a detailed description of gradient boosting, refer to Natekin and Knoll (2013).

[6] Corral et al. (2021) provides a detailed explanation of how this dataset was created.

[7] The results shown here were obtained from Python.

system for tree boosting and is available as an open-source software library. To make the comparison fair, for each of the 500 samples, a model is selected using lasso regression as detailed in section 6.2 and fit for the considered small area estimation methods. This is different from the approach usually taken under design-based validation, where one model is considered the true model and is used across all samples. The results for the XGBoost and the SAE methods are illustrated in Figure 5.12. The results illustrate that in the case of the Mexican data, XGBoost yields unbiased estimates of poverty. Moreover, the empirical MSE compares favorably with that of CensusEB methods, which are more computational and data-intensive. Additionally, in this Mexican scenario, the gradient boosted estimates are superior to those from the unit-context method discussed in the previous section (labeled U-C CensusEB).

Figure 5.12: Empirical bias and MSE of FGT0 for different methods



Source: Data from Corral et al. (2021). The simulations are based on 500 samples taken from the *Mexican Intracensal survey*, which is treated as a census of 3.9 million households. Under each sample, predictors for each of the methods are obtained and then compared to the true values obtained from the *Mexican Intracensal survey*. The figure illustrates that in the *Mexican Intracensal* case, XGBoost yields FGT0 estimates that are very close in performance, in terms of bias and MSE, to those from CensusEB estimators based on a unit-level model. H3-CBEB is the clustered bootstrap EB which is discussed in detail in Corral et al. (2021), the method was the EB approach implemented in `PovMap` and the original `sae` Stata package.

Despite the gradient boosting method's performance, it also carries some caveats. First, though the method performs well with the Mexican data, there is uncertainty as to the degree to which these results can be extrapolated to other contexts.[8] Second, there are currently no implemented software options for estimating the method's MSE.[9] Finally, despite XGBoost being based on open-source software, it is still somewhat of a black box and not easily interpretable. Therefore, further research is required to properly assess how well the method works, though early results are encouraging.

5.3 Pros and Cons of Methods for Poverty Mapping in Off-Census Years

This section presents a convenient list of pros and cons for each method discussed in this chapter. It also notes the needs for each of the methods. The section is borrows from Molina (2019).

[8]The quality of the covariates and how well these predict poverty at the modeling level determine the overall quality of the estimates obtained.

[9]What is shown in Figure 5.12 is the empirical MSE, not an estimate of the MSE.

5.3.1 Unit-Context Models

Model requirements:

1. Microdata from a household survey (only the model's dependent variable) and administrative, satellite-derived data, or any other area or sub-level data.
   (a) Masaki et al. (2020) recommend using data that is at least one level below the level at which results are to be presented, although the more disaggregated, the better.
2. The population size/count at the area level is needed, at least the number of dwellings in the area.
3. Areas and sub-areas in the survey and the census should have identifiers that can be linked.

Pros:

1. Based on household-level welfare from a survey and area-level data from any point in time. It may be used in off census years, avoiding the use of outdated census data.
2. Unlike Fay-Herriot area-level models (Fay and Herriot 1979), it can be applied for areas with an estimated sampling variance equal to 0.
3. May provide estimates for non-sampled areas.

Cons:

1. The welfare model is presumably incorrectly specified unless household-level welfare is dependent only on area-level characteristics. Hence, estimates are expected to be biased, and the direction and magnitude of the bias are unknown *a priori.*
2. The bootstrap method for MSE estimation is computationally intensive.
3. The parametric bootstrap approach from González-Manteiga et al. (2008) under the unit-context model is likely to yield an inaccurate measure of MSE. In many instances, the MSE may be considerably underestimated.

5.3.2 Gradient Boosting

Model requirements:

1. Direct estimates of indicators of interest for the areas considered, $\hat{\tau}_d^{DIR}$ (from the survey).
2. Aggregate data at the area level of all necessary covariates for the model for every area considered, $\mathbf{x}_d$, $d = 1, \ldots, D$.
3. Areas in the survey and the census should have identifiers that can be linked across each other.

Pros:

1. Based on direct estimates from a survey and area-level data from any point in time. It may be used in off census years, avoiding the use of outdated census data.

2. Unlike Fay-Herriot area-level models (Fay and Herriot 1979), it can be applied for areas with an estimated sampling variance equal to 0.

3. The method's dependent variable and the target indicators are the same. In a design-based simulation using the *Mexican Intracensal Survey* the method yields estimates of comparable quality to CensusEB and with better performance than unit-context models.

4. May provide estimates for non-sampled areas.

Cons:

1. The method requires validation exercises in more scenarios beyond the one conducted in this chapter. There is no guarantee that the method will work as well with covariates with considerably lower predictive power than the ones from the example provided in section 5.2.

2. The method currently lacks an approach for obtaining noise estimates (MSE). Consequently, it is difficult to assess the precision of the final estimates.

5.4 Unit-Context Models – Technical Annex

5.4.1 Producing Estimators Based on Unit-Context Models

The production of estimators based on unit-context models is similar to those using regular unit-level models, except that unit-level covariates are not used. This implies that the share of welfare variation across households explained by the model's covariates is expected to be lower and, within many areas, welfare may be poorly explained. Still, unit-context models may be regarded as an approximation to the true underlying data generating process. Actually, they are particular cases of unit-level models (Eq. 4.2); consequently, normality and linearity assumptions need to be checked similarly with the corresponding covariates. The focus of this section is on the unit-context models as presented in Masaki et al. (2020) and not those from Lange, Pape, and Pütz (2018) and Nguyen (2012). The reason for this choice is that Lange, Pape, and Pütz (2018) and Nguyen's (2012) approach relies on ELL's method, which suffers from the same issues noted by previous work (see Molina and Rao 2010; Corral, Molina, and Nguyen 2021; Corral et al. 2021 among others). In addition, Masaki et al. (2020) tested different methods, including EB, and concluded that EB provides a considerable gain in accuracy and efficiency over other methods.

Unit-context versions (i.e. those with aggregated covariates only) may be specified for either a one-fold nested error model or a two-fold nested error model. A possible unit-context model follows:

$$y_{sach} = z_{sac}\alpha + t_{sa}\omega + g_s\lambda + \eta_{sa} + \varepsilon_{sach}$$

where s is used for an aggregation level that is over the target areas (a super-area), and c is used for subareas. Hence, z_{sac} contains subarea-level characteristics, t_{sa} includes area-level characteristics and g_s is composed of super-area-level characteristics (which may include super-area fixed effects). The regression coefficients across these levels are respectively denoted α, ω and λ. The random effects, η_{sa}, are specified in this model at the area level. Note that, among the set of covariates in this model, none is at the unit level; covariates only vary at the subarea level.

Model selection may be implemented following the approach described in section 6.2, except that only contextual variables will be among the pool of eligible covariates. Data transformation is also important in unit-context models and is emphasized by Masaki et al. (2020). In contrast to the data transformation used in section 6.2, Masaki et al. (2020) recommend transforming the dependent variable with ordered quantile normalization. Nevertheless, this transformation cannot be used for the most common poverty and inequality indicators beyond headcount poverty because the transformation is not reversible. EB point and noise estimates are obtained following a Monte Carlo simulation and parametric bootstrap procedures, respectively, similar to the conventional application of the EB method under a unit-level model and detailed in the technical annex (sections 4.4.2.1 and 4.4.2.2). Finally, Masaki et al. (2020) also recommend adjusting the model-based estimators to match direct estimates, usually to the level where the survey is representative (benchmarking).

Benchmarking is not recommended unless publication requirements include that estimates of totals at a lower aggregation level add up to the estimated total at a higher level (e.g., the national level). The need to benchmark due to substantial discrepancies between the sum of estimated totals at the lower level and the estimated total at the higher level may indicate that the model assumptions are not satisfied. EB estimators based on a correct model are approximately model-unbiased and optimal in terms of minimizing the MSE for a given area; thus, when adjusted afterwards for benchmarking, so that these

match usual estimates at higher aggregation levels, the optimal properties are lost and estimators usually become worse in terms of bias and MSE under the model.[10] When benchmarking adjustments are large, as those likely required by estimators derived from unit-context model variants, it is an indication that the model does not hold for the data. Note that a significant bias in the final estimates may lead to considerable re-ranking of locations in terms of poverty estimates. Consequently, a limit on the acceptable bias should usually be determined according to needs. This is particularly important when determining priorities across areas based on small area estimates. If an area's true poverty rate is 50% and the method yields an estimator of 10% due to an incorrect model, there is a real risk that this area may not be assisted when needed. Molina (2019) suggests 5 or 10 percent of absolute relative bias as an acceptable threshold.

An additional problem for unit-context models in many applications is that it may not be possible to match census and survey PSUs. In some cases, it is due to confidentiality reasons and, in others, it is due to different sampling frames. The latter problem will likely affect applications where the census and survey correspond to different years. Fay-Herriot and other area or subarea models that use the same aggregated variables are an alternative approach to unit-context models for the case where the census is outdated, for which the model is not necessarily in question, since these models may be correctly specified. Of course, model checking is also needed.

[10] Beyond unit-context models, benchmarking in many instances may be necessary to ensure aggregate estimates are aligned to official published estimates.

5.5 Appendix

5.5.1 Simulation Experiment 1 for Unit-Context Models

A simulation experiment is conducted with the purpose of illustrating the inherent bias of the resulting CensusEB estimators based on unit-context models due to biased estimators of the model parameters. To remove a source of bias of estimators based on these models, which is due to differences between the sample and census means of covariates as shown in the Appendix of Corral et al. (2021), the model is fit to the whole population data and small area estimates are also calculated based on the same population data. The simulation is inspired on those conducted by Marhuenda et al. (2017) where the true data generating process is a two-fold nested error model. This model will better accommodate the usual applications of poverty mapping, where household surveys use two-stage sampling. A two-fold structure also allows for the inclusion of contextual variables that are at the cluster level while the random location effect is specified at the area level, similar as in Masaki et al. (2020). The creation of the census data set is similar to the one shown in section 3 of Corral et al. (2021).

A census data set of $N = 20,000$ observations is created, where observations are allocated among 40 areas ($a = 1, \ldots, A$). Within each area, observations are uniformly allocated over 10 clusters ($c = 1, \ldots, C_a$). Each cluster, c, consists of $N_{ac} = 50$ observations, and each cluster is labeled from 1 to 10. The assumed model contains both cluster and area effects. Cluster effects are simulated as $\eta_{ac} \overset{iid}{\sim} N\left(0, 0.05^2\right)$, area effects as $\eta_a \overset{iid}{\sim} N\left(0, 0.1\right)$ and household specific residuals as $e_{ach} \overset{iid}{\sim} N\left(0, 0.5^2\right)$, where $h = 1, \ldots, N_{ac}$; $c = 1, \ldots, C_a$; $a = 1, \ldots, A$. Covariates are simulated as follows:[11]

1. x_1 is a binary variable, taking value 1 when a random uniform number between 0 and 1, at the household-level, is less than or equal to $0.3 + 0.5\frac{a}{40} + 0.2\frac{c}{10}$.
2. x_2 is a binary variable, taking value 1 when a random uniform number between 0 and 1, at the household-level, is less than or equal to 0.2.
3. x_3 is a binary variable, taking value 1 when a random uniform number between 0 and 1, at the household-level, is less than or equal to $0.1 + 0.2\frac{a}{40}$.
4. x_4 is a binary variable, taking value 1 when a random uniform number between 0 and 1, at the household-level, is less than or equal to $0.5 + 0.3\frac{a}{40} + 0.1\frac{c}{10}$
5. x_5 is a discrete variable, simulated as the rounded integer value of the maximum between 1 and a random Poisson variable with mean $\lambda = 3\left(1 - 0.1\frac{a}{40}\right)$.
6. x_6 is a binary variable, taking value 1 when a random uniform value between 0 and 1 is less than or equal to 0.4. Note that the values of x_6 are not related to the area's label.
7. x_7 is a binary variable, taking value 1 when a random uniform number between 0 and 1 is greater than or equal to $0.2 + 0.4\frac{a}{40} + 0.1\frac{c}{10}$

The welfare vector for each household within a cluster within an area is created from the model with these covariates, as follows:

$$y_{ach} = 3 + 0.09x_{1ach} - 0.04x_{2ach} - 0.09x_{3ach} + 0.4x_{4ach} - 0.25x_{5ach} + 0.1x_{6ach} + 0.33x_{7ach} + \eta_a + \eta_{ac} + e_{ach}, \tag{5.1}$$

[11] Covariates are simulated following Corral et al. (2021) who follow the approach from Molina and Rao (2010) and Marhuenda et al. (2017), with slight modifications.

The dependent variable, y_{ach}, is the log of the variable of interest. The poverty line in this scenario is fixed at $z = 12$. This generation process is repeated 5,000 times. This will yield 5,000 true poverty rates for each area.

As already said, to show that estimators based on unit-context models are still biased even if the source of bias noted in Corral et al. (2021) is removed, instead of drawing a sample from the population to fit the models, the models are fit to the whole set of census data. This eliminates the latter source of bias. The unit-context model includes the cluster means of the 7 covariates. In each of the 5,000 simulations, the following quantities are computed for the poverty rates and gaps in each area:

1. True poverty indicators τ_a, using the "census".
2. Census EB estimators $\hat{\tau}_a^{CEB_a}$ presented in Corral, Molina, and Nguyen (2021) based on a nested error model with only **area** random effects and including the unit-level values of the covariates, and obtained using a Monte Carlo approximation with $M = 50$ replicates. The R^2 of this unit-level model is a slightly below 0.5.
3. Unit-context Census EB estimators $\hat{\tau}_a^{UC-CEB_a}$ based on a nested error model with random effects at the **area level** obtained using a Monte Carlo approximation with $M = 50$ replicates. This estimator follows the approach from Masaki et al. (2020) and uses only cluster means for all of the covariates. The R^2 of this unit-context model is below 0.05.

The average across the 5,000 simulations of the estimation errors for each area represent the empirical biases of the considered area estimators. The Stata script to replicate these simulations can be found in the appendix (section 5.5.1.1).

One could argue that, in this scenario, the R^2 of unit-context models is much lower than that one in the applications of Masaki et al. (2020) and of Lange, Pape, and Pütz (2018). For this reason, the simulation experiment is repeated modifying slightly the data generating process to increase the R^2. Specifically, in this experiment, the covariate x_7 is now generated from a random Poisson variable with mean $\lambda = 3\left(\frac{c}{20} - \frac{a}{100} + u\right)$, where u is a random uniform value between 0 and 1, and σ_e is increased from 0.5 to 0.6. This modification leads to an R^2 of the unit-context model between 0.15 and 0.20, while for unit-level models the R^2 exceeds 0.60. The Stata script to replicate these simulations can be found in the following section 5.5.1.2.

5.5.1.1 Unit-Context Models – Validation

The do-file below reproduces the simulation experiment described in section 5.4. Note that the model is fit on to the population and then estimates are obtained by simulating on to the same population.

```
set more off
clear all

global main     "C:\Users\\`c(username)´\OneDrive\SAE Guidelines 2021\"
global section  "$main\3_Unit_level\"
global mdata    "$section\1_data\"
global myfigs   "$section\3_figures\"
/*
Author: Paul Corral
Do file below is a test for a two fold nested error model. It follows the method
illustrated in the paper from Marhuenda et al. (2017) and others in the link
below.
```

```
We start off by creating a fake data set as illustrated in that same paper.
 https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/rssa.12306
*/
/*
Purpose of file is to test SAE model performance by imputing on to the
population instead of a sample. This should remove all other sources of bias.
*/

*================================================================================
// Parameters for simulated data set
*================================================================================
        version 15
        set seed 734137
        global numobs = 20000
        global outsample = 50
        global areasize  = 500
        global psusize   = 50

        //We have 2 location effects below
        global sigmaeta_psu   = 0.05
        global sigmaeta_area  = 0.1
        //We have household specific errors
        global sigmaeps   = 0.5
        //Poverty line fixed at 12
        global  pline  = 12
        global lnpline = ln(12)
        //locals
        local obsnum    = $numobs
        local areasize  = $areasize
        local psusize   = $psusize
        local total_sim = 5000

*================================================================================
//1.Create simulated data
*================================================================================
//Start off with # of observations
set obs `=`obsnum´/`areasize´´
        gen area = _n
                lab var area "Area identifier"
        //expand to create 10 psu per area
        expand `=`areasize´/`psusize´´
        sort area
        //PSUs labelled from 1 to 10 within each area
        gen psu = _n - (area-1)*10
                lab var psu "PSU identifier"
        //expand to create 50 observations by psu
        expand `psusize´
        sort area psu
        //Household id
        gen hhid = _n
                lab var hhid "Household identifier"

        //Covariates, some are corrlated to the area and psu´s label
        gen x1=runiform()<=(0.3+.5*area/(`obsnum´/`areasize´) + ///
        0.2*psu/(`areasize´/`psusize´))
        gen x2=runiform()<=(0.2)
        gen x3= runiform()<=(0.1 + .2*area/int(`obsnum´/`areasize´))
        gen x4= runiform()<=(0.5+0.3*area/int(`obsnum´/`areasize´) + ///
        0.1*psu/int(`areasize´/`psusize´))
        gen x5= round(max(1,rpoisson(3)*(1-.1*area/int(`obsnum´/`areasize´))),1)
        gen x6= runiform()<=0.4
        gen x7= runiform()>=(0.2+0.4*area/int(`obsnum´/`areasize´) + ///
```

```
        0.1*psu/int(`areasize´/`psusize´))

        //note that this matches the model from eq. 3 of Corral et al. (2021)
        gen XB = 3+ .09* x1-.04* x2 - 0.09*x3 + 0.4*x4 - 0.25*x5 + 0.1*x6 + 0.33*x7
                lab var XB "Linear fit"

        //Create psu level means...
        groupfunction, mean(x*) merge by(area psu)

        //Indicate first area observation
        bysort area: gen area_1st = 1 if _n==1
        //Indicate first psu observation
        bysort area psu: gen psu_1st = 1 if _n==1
        sort hhid
        //We need weights for SAE command
        gen hhsize = 1
                lab var hhsize "HH size for command"

//Save population´s Xs  and linear fit
save "$mdata\popX.dta", replace

*================================================================================
//2. Import data for SAE
*================================================================================
sae data import, datain("$mdata\popX.dta") varlist( mean_x1 mean_x2 mean_x3 ///
mean_x4 mean_x5 mean_x6 mean_x7 x1 x2 x3 x4 x5 x6 x7 hhsize) ///
area(area) uniqid(hhid) dataout("$mdata\census")

*================================================================================
//3. Run the simulations
*================================================================================

/*
Now, we will run 5,000 simulations where we follow the model´s assumpitons.
under each simulation we will add to XB the psu and area effect, as well
as the household specific error.
Then, under each population we will obtain CensusEB estimates under
unit-level CensusEB, unit-level ELL, and unit-context models. For each
population and the EB estimates obtained we will calculate the difference
between the true poverty rate and the estimate, and the squared difference.
After 5000 simulations these are our empirical bias and MSE.
*/

//Add random location effects and household errors
forval z=1/`total_sim´{
        use "$mdata\popX.dta", clear
                //random area effects
                gen double eta_a = rnormal(0,$sigmaeta_area) if area_1st==1
                        replace eta_a = eta_a[_n-1] if missing(eta_a)
                gen double eta_p = rnormal(0,$sigmaeta_psu)  if psu_1st ==1
                        replace eta_p = eta_p[_n-1] if missing(eta_p)
                //household errors
                gen eps = rnormal(0,$sigmaeps)
                //Y, normally distributed
                egen double Y  = rsum(XB eta_a eta_p eps)

        tempfile myPop
        save `myPop´

        //Seed stage for simulations, changes after every iteration!
        local seedstage `c(rngstate)´

        gen double e_y = exp(Y)
        //Create true values
```

```
    forval a = 0/2{
            gen fgt`a´ = (e_y<$pline)*(1-e_y/$pline)^`a´
    }
    preserve
            //true values by area
            groupfunction [aw=hhsize], mean(fgt* e_y Y) by(area)
            rename e_y mean
            tempfile true
            save `true´
    restore

    //Bring in the 20K pop and use it as a survey
    use `myPop´, clear

    //Obtain UC SAE
    preserve
            sae sim h3 Y mean_x1 mean_x2 mean_x3 mean_x4 mean_x5 mean_x6 ///
mean_x7,  area(area)   mcrep(50) bsrep(0) matin("$mdata\census") ///
lny seed(`seedstage´) pwcensus(hhsize) indicators(FGT0 FGT1 FGT2) ///
          aggids(0) uniq(hhid) plines($pline)
                  rename avg_fgt* uc_fgt*
                  rename Unit area
                  rename Mean uc_mean
            tempfile h3area
            save `h3area´
    restore

    //Obtain UC SAE, without transforming
    preserve
            sae sim h3 Y mean_x1 mean_x2 mean_x3 mean_x4 mean_x5 mean_x6 ///
mean_x7,  area(area) mcrep(50) bsrep(0) matin("$mdata\census") ///
            seed(`seedstage´) pwcensus(hhsize) indicators(FGT0 FGT1 FGT2) ///
            aggids(0) uniq(hhid) plines($lnpline)
                  rename avg_fgt* ucn_fgt*
                  rename Unit area
                  rename Mean ucn_Y
            tempfile h3arean
            save `h3arean´
    restore

    //Obtain CensusEB SAE
    preserve
            sae sim h3 Y x1 x2 x3 x4 x5 x6 x7,  area(area) mcrep(50) ///
            bsrep(0) matin("$mdata\census") lny seed(`seedstage´) ///
            pwcensus(hhsize) indicators(FGT0 FGT1 FGT2) aggids(0) ///
            uniq(hhid) plines($pline)
                  rename avg_fgt* ceb_fgt*
                  rename Unit area
                  rename Mean ceb_mean
            tempfile h3eb
            save `h3eb´
    restore

    //Without transforming...CensusEB
    preserve
            sae sim h3 Y x1 x2 x3 x4 x5 x6 x7,  area(area)  ///
            mcrep(50) bsrep(0) matin("$mdata\census") seed(`seedstage´) ///
            pwcensus(hhsize) indicators(FGT0 FGT1 FGT2) aggids(0) ///
            uniq(hhid) plines($lnpline)
                  rename avg_fgt* cebn_fgt*
                  rename Unit area
                  rename Mean cebn_Y
```

```
                tempfile h3ebn
                save `h3ebn´
        restore


        //Open true point estimates
        use `true´, clear

        //Merge in the model based estimates
        merge 1:1 area using `h3area´, keepusing(uc_*)
                drop _m
        merge 1:1 area using `h3eb´  , keepusing(ceb_*)
                drop _m
        merge 1:1 area using `h3arean´, keepusing(ucn_*)
                drop _m
        merge 1:1 area using `h3ebn´  , keepusing(cebn_*)
                drop _m

        //Calculate bias and MSE
        foreach j in fgt0 fgt1 fgt2 mean{
                foreach i in ceb uc cebn ucn{
                        if ("`j´"=="mean" & ("`i´"=="cebn"|"`i´"=="ucn")) local j Y
                        gen double `i´_bias_`j´ = (`i´_`j´-`j´)/`total_sim´
                        gen double `i´_mse_`j´  = ((`i´_`j´-`j´)^2)/`total_sim´
                }
        }
        keep area *_bias_* *_mse_*

        //For first sim we rename the vector to *T
        if (`z´==1){
                rename *_bias_* *_bias_*T
                rename *_mse_*  *_mse_*T

                tempfile Stats
                save `Stats´
        }
        else{ //After the first sim, we add the bias and MSE to *T
                merge 1:1 area using `Stats´
                        drop _m

                foreach j in fgt0 fgt1 fgt2 mean{
                        foreach i in ceb uc cebn ucn{
                                if ("`j´"=="mean" & ("`i´"=="cebn"|"`i´"=="ucn"))  local j Y
                                replace `i´_bias_`j´T = `i´_bias_`j´T + `i´_bias_`j´
                                replace `i´_mse_`j´T  = `i´_mse_`j´T + `i´_mse_`j´

                                drop `i´_bias_`j´ `i´_mse_`j´
                        }
                }
                tempfile Stats
                save `Stats´
        }

}
save "$mdata\bias_in_mymodel.dta", replace
```

5.5.1.2 Unit-Context Models – Validation with Better Model Fit

The do-file below reproduces the simulation experiment described in section 5.4, but considering unit-context models with a better R^2 and producing estimates for 2 different poverty thresholds. Note that

the model is fit to the whole set of population data and then estimates are also obtained by simulating on to the population.

```
set more off
clear all
global main     "C:\Users\\`c(username)´\OneDrive\SAE Guidelines 2021\"
global section  "$main\3_Unit_level\"
global mdata    "$section\1_data\"
global myfigs   "$section\3_figures\"
/*
Author: Paul Corral
Version @2 differs from previous one in that we create a model where
UC models have a better fit (R2 ~ 0.18), also welfare is somewhat more skewed

We start off by creating a fake data set illustrated in Marhuenda et al. (2017).
 https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/rssa.12306
*/
/*
Purpose of file is to test SAE model performance by imputing on to the
population instead of a sample. This should remove all other sources of bias.
*/
*================================================================================
// Parameters for simulated data set
*================================================================================
        version 15
        set seed 734137
        global numobs = 20000
        global areasize  = 500
        global psusize   = 50

        //We have 2 location effects below
        global sigmaeta_psu   = 0.05
        global sigmaeta_area  = 0.1
        //We have household specific errors
        global sigmaeps   = 0.6
        //Poverty line fixed at 27.8
        global pline    = 13
        global lnpline = ln($pline)
        global pline1   = 28
        global lnpline1 = ln($pline1)
        local lines $pline $pline1
        //locals
        local obsnum    = $numobs
        local areasize  = $areasize
        local psusize   = $psusize
        local total_sim = 1000

*================================================================================
//1.Create simulated data
*================================================================================
//Start off with # of observations
set obs `=`obsnum´/`areasize´´
        gen area = _n
                lab var area "Area identifier"
        //expand to create 10 psu per area
        expand `=`areasize´/`psusize´´
        sort area
        //PSUs labelled from 1 to 10 within each area
        gen psu = _n - (area-1)*`=`areasize´/`psusize´´
```

```
        lab var psu "PSU identifier"
    //expand to create 50 observations by psu
    expand `psusize´
    sort area psu
    //Household id
    gen hhid = _n
        lab var hhid "Household identifier"

    //Covariates, some are corrlated to the area and psu´s label
    gen x1=runiform()<=(0.3+.5*area/(`obsnum´/`areasize´) + ///
    0.2*psu/(`areasize´/`psusize´))
    gen x2=runiform()<=(0.2)
    gen x3= runiform()<=(0.1 + .2*area/int(`obsnum´/`areasize´))
    gen x4= runiform()<=(0.5+0.3*area/int(`obsnum´/`areasize´) + ///
    0.1*psu/int(`areasize´/`psusize´))
    gen x5= round(max(1,rpoisson(3)*(1-.1*area/int(`obsnum´/`areasize´))),1)
    gen x6= runiform()<=0.4
    gen x7=rpoisson(3)*(1*psu/int(`areasize´/`psusize´)- 1*area/int(`obsnum´/`areasize´) ///
    + 1*uniform())

    //note that this matches the model from eq. 3 of Corral et al. (2021)
    gen XB = 3+ .09* x1-.04* x2 - 0.09*x3 + 0.4*x4 - 0.25*x5 + 0.1*x6 + 0.33*x7
        lab var XB "Linear fit"

    //Create psu level means...
    preserve
    collapse (mean) x*, by(area psu)
    rename x* meanpsu_x*
    tempfile psumeans
    qui save `psumeans´
    restore

    preserve
    collapse (mean) x*, by(area)
    rename x* meanarea_x*
    tempfile areameans
    qui save `areameans´
    restore

    merge n:1 area psu using `psumeans´, assert(3) nogen
    merge n:1 area using `areameans´, assert(3) nogen

    //Indicate first area observation
    bysort area: gen area_1st = 1 if _n==1
    //Indicate first psu observation
    bysort area psu: gen psu_1st = 1 if _n==1
    sort hhid
    //We need weights for SAE command
    gen hhsize = 1
        lab var hhsize "HH size for command"

    //Create hierarchical identifier
    gen uno = 100+area
    gen dos = 100+psu
    gen double  HID = real(string(uno)+string(dos))

//Save population´s Xs  and linear fit
save "$mdata\popXT.dta", replace

*================================================================================
//2. Import data for SAE
*================================================================================
unab themeans : mean*
```

```
sae data import, datain("$mdata\popXT.dta") ///
varlist(`themeans´ x1 x2 x3 x4 x5 x6 x7 hhsize) ///
area(area) uniqid(hhid) dataout("$mdata\census")

*================================================================================
//3. Run the simulations
*================================================================================

/*
Now, we will run 5,000 simulations where we follow the model´s assumpitons.
under each simulation we will add to XB the psu and area effect, as well
as the household specific error.
Then, under each population we will obtain CensusEB estimates under
unit-level CensusEB, and unit-context models. For each
population and the EB predictions obtained we will calculate the difference
between the true poverty rate and the predicted one, and the squared difference.
After 5000 simulations these are our empirical bias and MSE.
*/

// For each simulation we need to add random location effects and
// household errors
forval z=1/`total_sim´{
qui{
        use "$mdata\popXT.dta", clear
                //random area effects
                gen double eta_a = rnormal(0,$sigmaeta_area) if area_1st==1
                        replace eta_a = eta_a[_n-1] if missing(eta_a)
                gen double eta_p = rnormal(0,$sigmaeta_psu)  if psu_1st ==1
                        replace eta_p = eta_p[_n-1] if missing(eta_p)
                //household errors
                gen eps = rnormal(0,$sigmaeps)
                //Generate Y adding the XB and the drawn errors
                egen double Y  = rsum(XB eta_a eta_p eps)
                gen double e_y = exp(Y)
        tempfile myPop
        save `myPop´

        if (`z´==1){
                reg Y x*
                predict res, res
                reg Y  meanpsu_x1 meanpsu_x2 meanpsu_x3 meanpsu_x4 meanpsu_x5 ///
                meanpsu_x6 meanpsu_x7
                predict resA, res

                twoway (kdensity res) (kdensity resA)
        }

        //Seed stage for simulations, changes after every iteration!
        local seedstage `c(rngstate)´


        //Create true values
        gen fgt0_$pline = (e_y<$pline)*(1-e_y/$pline)^0
        gen fgt0_$pline1 = (e_y<$pline1)*(1-e_y/$pline1)^0

        preserve
                //true values by area
                groupfunction [aw=hhsize], mean(fgt* e_y Y) by(area)
                rename e_y mean
                tempfile true
                save `true´
        restore

        //Bring in the 20K pop and use it as a survey
```

```
use `myPop´, clear

//Do model selection for Area
if (`z´==1){
        lnskew0 y1 = e_y
        lassoregress y1 mean*,  numfolds(5)
        local vv = e(varlist_nonzero)
        global area_lnskew `vv´
        drop y1

        bcskew0 y1 = e_y
        lassoregress y1 mean*,  numfolds(5)
        local vv = e(varlist_nonzero)
        global area_bc `vv´
        drop y1

        lassoregress Y mean*,  numfolds(5)
        local vv = e(varlist_nonzero)
        global area_vars1 `vv´

}


//Obtain UC SAE, without transforming
preserve
        sae sim h3 e_y $area_lnskew,  area(area)  ///
        mcrep(50) bsrep(0) matin("$mdata\census") seed(`seedstage´) lnskew ///
        pwcensus(hhsize) indicators(FGT0) aggids(0) uniq(hhid) plines($pline $pline1)
                rename avg_fgt* uc_fgt*
                rename Unit area
                rename Mean uc_mean
        tempfile h3area
        save `h3area´
restore

preserve
        sae sim h3 Y $area_vars1,  area(area)  ///
        mcrep(50) bsrep(0) matin("$mdata\census") seed(`seedstage´) lny ///
        pwcensus(hhsize) indicators(FGT0) aggids(0) uniq(hhid) plines($pline $pline1)
                rename avg_fgt* ucn_fgt*
                rename Unit area
                rename Mean ucn_mean
        tempfile h3arealn
        save `h3arealn´
restore

preserve
        sae sim h3 e_y $area_bc,  area(area)  ///
        mcrep(50) bsrep(0) matin("$mdata\census") seed(`seedstage´) bcox ///
        pwcensus(hhsize) indicators(FGT0) aggids(0) uniq(hhid) plines($pline $pline1)
                rename avg_fgt* ucb_fgt*
                rename Unit area
                rename Mean ucb_mean
        tempfile h3areabc
        save `h3areabc´
restore


//CensusEB
preserve
        sae sim h3 Y x1 x2 x3 x4 x5 x6 x7,  area(area) lny ///
```

```
        mcrep(50) bsrep(0) matin("$mdata\census") seed(`seedstage´) ///
        pwcensus(hhsize) indicators(FGT0) aggids(0) uniq(hhid) plines($pline $pline1)
                rename avg_fgt* ceb_fgt*
                rename Unit area
                rename Mean ceb_mean
        tempfile h3eb
        save `h3eb´
restore



//Open true point estimates
use `true´, clear

//Merge in the model based estimates
merge 1:1 area using `h3area´, keepusing(uc_*)
        drop _m
merge 1:1 area using `h3eb´  , keepusing(ceb_*)
        drop _m
merge 1:1 area using `h3arealn´  , keepusing(ucn_*)
        drop _m
merge 1:1 area using `h3areabc´  , keepusing(ucb_*)
        drop _m



//Calculate bias and MSE
local j mean
foreach i in ceb ucn uc ucb{
        gen double `i´_bias_`j´ = (`i´_`j´-`j´)/`total_sim´
        gen double `i´_mse_`j´  = ((`i´_`j´-`j´)^2)/`total_sim´
}

foreach line of local lines{
        foreach i in ceb ucn uc ucb{
                gen double `i´_bias_fgt0_`line´ = ///
                (`i´_fgt0_`line´-fgt0_`line´)/`total_sim´
                gen double `i´_mse_fgt0_`line´  = ///
                ((`i´_fgt0_`line´-fgt0_`line´)^2)/`total_sim´
        }
}
keep area *_bias_* *_mse_*

//For first sim we rename the vector to *T
if (`z´==1){
        rename *_bias_* *_bias_*T
        rename *_mse_*  *_mse_*T

        tempfile Stats
        save `Stats´
}
else{ //After the first sim, we add the bias and MSE to *T
        merge 1:1 area using `Stats´
                drop _m
        local j mean
        foreach i in ceb ucn uc ucb{
                replace `i´_bias_`j´T = `i´_bias_`j´T + `i´_bias_`j´
                replace `i´_mse_`j´T  = `i´_mse_`j´T + `i´_mse_`j´
        }
        foreach line of local lines{
                foreach i in ceb ucn uc ucb{
                         replace `i´_bias_fgt0_`line´T = ///
```

```
            `i´_bias_fgt0_`line´T + `i´_bias_fgt0_`line´
                                  replace `i´_mse_fgt0_`line´T  = ///
                                 `i´_mse_fgt0_`line´T + `i´_mse_fgt0_`line´

                                  drop `i´_bias_fgt0_`line´ `i´_mse_fgt0_`line´
                          }
                    }
                    tempfile Stats
                    save `Stats´
            }
    }
    dis as error "Sim num `z´"
    }

    save "$mdata\bias_in_mymodel_twolines.dta", replace
```

5.5.2 Simulation Experiment 2 for Unit-Context Models

To further explore the bias of unit-context models and compare its performance to other methods, a final simulation is conducted following the data generation noted in section 5.5.1.2, but with some modifications. Firstly, the population size is $N = 500,000$, and the observations are allocated among $A = 100$ areas ($a = 1, \ldots, A$). Within each area a, observations are uniformly allocated over $C_a = 20$ clusters ($c = 1, \ldots, C_a$). Each cluster c consists of $N_{ac} = 250$ observations. In this simulation experiment, we take a simple random sample of $n_{ac} = 10$ households per cluster, and this sample is kept fixed across simulations. Using a sample, we can also compare with estimators based on FH model (discussed in Chapter 3). The model that generates the population data contains both cluster and area effects. Cluster effects are simulated as $\eta_{ac} \overset{iid}{\sim} N(0, 0.1)$, area effects as $\eta_a \overset{iid}{\sim} N\left(0, 0.15^2\right)$ and household specific residuals as $e_{ach} \overset{iid}{\sim} N\left(0, 0.5^2\right)$, where $h = 1, \ldots, N_{ac}$, $c = 1, \ldots, C_a$, $a = 1, \ldots, A$. Finally, x_7 is generated from a random Poisson variable with mean $\lambda = 3\left(\frac{c}{20} - \frac{a}{100} + u\right)$, where u is a random uniform value between 0 and 1. In this experiment, we take a grid of 99 poverty thresholds, corresponding to the 99 percentiles of the very first population generated. In total, 1,000 populations are generated. In each of the 1,000 populations, the following quantities are computed for each of the 99 poverty lines in each area:

1. True poverty indicators τ_a, using the "census".
2. CensusEB estimators $\hat{\tau}_a^{CEB_a}$ presented in Corral, Molina, and Nguyen (2021), based on a nested error model with only **area** random effects and including the unit-level values of the covariates, and obtained using a Monte Carlo approximation with $M = 50$ replicates. The R^2 for this model is roughly 0.60.
3. Unit-context CensusEB estimators $\hat{\tau}_a^{UC-CEB_a}$ based on a nested error model with random effects at the **area-level** obtained using a Monte Carlo approximation with $M = 50$ replicates. This estimator follows the approach of Masaki et al. (2020) and uses a model selected using lasso, as described in section 6.2. The R^2 of the resulting model hovers around 0.17.
4. Area-level FH estimators $\hat{\tau}_a^{FH_a}$ based on the model described in section 3.4. In this case, a separate model is needed for each of the 99 different poverty lines. Hence, the R^2 depends on the poverty line, but it ranges from 0.15 to 0.70.

The average across the 1,000 simulations represent the empirical bias for each area. The Stata script to replicate these simulations can be found in the appendix (section 5.5.2.1).[12]

[12] Depending on the computing power, this may take longer than 2 days to run.

5.5.2.1 Unit-Context Models – Validation Across All Poverty Lines

The Stata code below produces the simulations described in section 5.4. Here, a sample is drawn from the population and then, estimates are obtained for 99 different poverty lines. Each poverty line corresponds to a percentile of the very first generated population.

```
set more off
clear all

global main     "C:\Users\\`c(username)´\OneDrive\SAE Guidelines 2021\"
global section  "$main\3_Unit_level\"
global mdata    "$section\1_data\"
global myfigs   "$section\3_figures\"
/*
Author: Paul Corral
Version @2 differs from previous one in that we create a model where
UC models have a better fit (R2 ~ 0.18), also welfare is somewhat more skewed

We start off by creating a fake data set illustrated in Marhuenda et al. (2017).
 https://rss.onlinelibrary.wiley.com/doi/pdf/10.1111/rssa.12306
*/
/*
Purpose of file is to test SAE model performance by obtaining
estimates at different points of the tru distribution.
*/

*===============================================================================
// Parameters for simulated data set
*===============================================================================
        version 15
        set seed 734137
        global numobs = 500000
        global areasize  = 5000
        global psusize   = 250

        //We have 2 location effects below
        global sigmaeta_psu   = 0.1
        global sigmaeta_area  = 0.15
        //We have household specific errors
        global sigmaeps   = 0.5
        //Poverty line fixed at 26
        global  pline  = 26
        global lnpline = ln($pline)
        //locals
        local obsnum    = $numobs
        local areasize  = $areasize
        local psusize   = $psusize
        local total_sim = 1000
        local myrep     = 50
        local from0     = 1

*===============================================================================
//1.Create simulated data
*===============================================================================
//Start off with # of observations
set obs `=`obsnum´/`areasize´´
        gen area = _n
                lab var area "Area identifier"
        //expand to create 10 psu per area
        expand `=`areasize´/`psusize´´
```

```
sort area
//PSUs labelled from 1 to 10 within each area
gen psu = _n - (area-1)*`=`areasize´/`psusize´´
        lab var psu "PSU identifier"
//expand to create 50 observations by psu
expand `psusize´
sort area psu
//Household id
gen hhid = _n
        lab var hhid "Household identifier"

//Covariates, some are corrlated to the area and psu´s label
gen x1=runiform()<=(0.3+.5*area/(`obsnum´/`areasize´) + ///
0.2*psu/(`areasize´/`psusize´))
gen x2=runiform()<=(0.2)
gen x3= runiform()<=(0.1 + .2*area/int(`obsnum´/`areasize´))
gen x4= runiform()<=(0.5+0.3*area/int(`obsnum´/`areasize´) + ///
0.1*psu/int(`areasize´/`psusize´))
gen x5= round(max(1,rpoisson(3)*(1-.1*area/int(`obsnum´/`areasize´))),1)
gen x6= runiform()<=0.4
gen x7=rpoisson(3)*(1*psu/int(`areasize´/`psusize´)- 1*area/int(`obsnum´/`areasize´)+ ///
1*uniform())

//note that this matches the model from eq. 3 of Corral et al. (2021)
gen XB = 3+ .09* x1-.04* x2 - 0.09*x3 + 0.4*x4 - 0.25*x5 + 0.1*x6 + 0.33*x7
        lab var XB "Linear fit"

//Create psu level means...
preserve
collapse (mean) x*, by(area psu)
rename x* meanpsu_x*
tempfile psumeans
qui save `psumeans´
restore

preserve
collapse (mean) x*, by(area)
rename x* meanarea_x*
tempfile areameans
qui save `areameans´
restore

merge n:1 area psu using `psumeans´, assert(3) nogen
merge n:1 area using `areameans´, assert(3) nogen

//Indicate first area observation
bysort area: gen area_1st = 1 if _n==1
//Indicate first psu observation
bysort area psu: gen psu_1st = 1 if _n==1
sort hhid
//We need weights for SAE command
gen hhsize = 1
        lab var hhsize "HH size for command"

//Create hierarchical identifier
gen uno = 100+area
gen dos = 100+psu
gen double  HID = real(string(uno)+string(dos))

preserve
sample 10, by(HID) count
tempfile msvy
```

```
        save `msvy´
        restore

        merge 1:1 hhid using `msvy´
        gen svy = _m==3
        drop _m
//Save population´s Xs  and linear fit
sort hhid
save "$mdata\popX.dta", replace

*================================================================================
//2. Import data for SAE
*================================================================================
unab themeans : mean*
sae data import, datain("$mdata\popX.dta") varlist(`themeans´ x1 x2 x3 x4 x5 x6 x7 hhsize) ///
area(area) uniqid(hhid) dataout("$mdata\thecensus")

*================================================================================
//3. Simulation
*================================================================================
local seedstage `c(rngstate)´

if (`from0´==1) local start 1
else{
        use "$mdata\bias_in_mymodel_allplines.dta", clear
        local seedstage = `_dta[note1]´
        local start = subinstr(`_dta[note2]´,"Sim num ","",.)
        local start = `start´+1
        global plines `_dta[note3]´
        dis "`seedstage´"
        dis "`start´"
}

forval z=1/`total_sim´{
        set seed `seedstage´
        use "$mdata\popX.dta", clear
                gen double eta_a = rnormal(0,$sigmaeta_area) if area_1st==1
                        replace eta_a = eta_a[_n-1] if missing(eta_a)
                gen double eta_p = rnormal(0,$sigmaeta_psu)  if psu_1st ==1
                        replace eta_p = eta_p[_n-1] if missing(eta_p)
                //household errors
                gen eps = rnormal(0,$sigmaeps)
                //Generate Y adding the XB and the drawn errors
                egen double Y  = rsum(XB eta_a eta_p eps)


        if (`z´==1){
                pctile double y_p = Y, nq(100)
                replace y_p = round(exp(y_p),0.01)
                levelsof y_p, local(mp)
                local thelines
                local aa = 1
                foreach y of local mp{
                        if (`y´<10) local nm = substr("`y´",1,4)
                        if (`y´>=10 & `y´<100) local nm = substr("`y´",1,5)
                        if (`y´>100)  local nm = substr("`y´",1,6)
                        local thelines `thelines´  `nm´

                        local aa= `aa´+1
                }
                global plines `thelines´
        }
```

```
gen double e_y = exp(Y)
        //Create true values
foreach j of global plines{
        local linea = subinstr("`j´",".","_",.)
        forval a = 0/0{
                gen fgt`a´_`linea´ = (e_y<`j´)*(1-e_y/`j´)^`a´
        }
}


preserve
timer on 1
        //true values by area
        groupfunction [aw=hhsize], mean(fgt* e_y Y) by(area)
        rename e_y mean

        tempfile true
        save `true´
timer off 1
timer list
restore


keep if svy==1

if (`z´==1){
        lassoregress Y mean*,  numfolds(5)
        local vv = e(varlist_nonzero)
        global area_vars `vv´

        reg Y $area_vars
        gen touse = e(sample)
        mata: ds = _f_stepvif("$area_vars","hhsize",3,"touse")
        global area_vars `vifvar´
        reg Y $area_vars
}

tempfile msvy
save `msvy´

preserve
        //R2 of roughly 0.16
        sae sim h3 Y $area_vars,  area(area)  ///
        mcrep(`myrep´) lny bsrep(0) matin("$mdata\thecensus") seed(`seedstage´) ///
        pwcensus(hhsize) indicators(FGT0) aggids(0) uniq(hhid) plines($plines)
                rename avg_fgt* ucln_fgt*
                rename Unit area
                rename Mean ucln_mean
        tempfile h3arealn
        save `h3arealn´
restore

preserve
        sae sim h3 Y x1 x2 x3 x4 x5 x6 x7,  area(area)  ///
        mcrep(`myrep´) lny bsrep(0) matin("$mdata\thecensus") seed(`seedstage´) ///
        pwcensus(hhsize) indicators(FGT0) aggids(0) uniq(hhid) plines($plines)
                rename avg_fgt* cebln_fgt*
                rename Unit area
                rename Mean cebln_mean
        tempfile h3ebln
        save `h3ebln´
restore
```

```
	foreach pp of global plines{
		preserve
			local nm = subinstr("`pp´",".","_",.)
			qui: proportion fgt0_`nm´, over(area)
			mata: fgt0     = st_matrix("e(b)")
			mata: fgt0     = fgt0[`=1+`obsnum´/`areasize´´..`=2*`obsnum´/`areasize´´]´
			mata: fgt0_var = st_matrix("e(V)")
			mata: fgt0_var = ///
diagonal(fgt0_var)[`=1+`obsnum´/`areasize´´..`=2*`obsnum´/`areasize´´]

			gen num_obs = 1
			groupfunction, rawsum(num_obs) mean(fgt0_`nm´) first(meanarea*) by(area)
			sort area //ordered to match proportion output

			//Pull proportion´s results
			getmata dir_fgt0 = fgt0 dir_fgt0_var = fgt0_var
			gen double mifgt0 = dir_fgt0
			replace dir_fgt0_var = . if dir_fgt0_var==0
			replace dir_fgt0     = . if missing(dir_fgt0_var)

			qui: gen meanarea_x7sq = meanarea_x7^2

			if (`z´==1){
				fhsae dir_fgt0 meanarea_x7 meanarea_x7sq, ///
   re(dir_fgt0_var) method(chandra)
				test meanarea_x7sq
				local sq = r(p)
				test meanarea_x7
				local regular = r(p)
				if (`regular´<0.1 & `sq´<0.1) global fh_`nm´ meanarea_x7 ///
				meanarea_x7sq
				else global fh_`nm´ meanarea_x7
			}

			fhsae dir_fgt0 ${fh_`nm´}, re(dir_fgt0_var) method(reml) fh(fh_fgt0_`nm´)
			//for cases where the direct estimate is 1 or 0, the fh
			// estimate will be missing. In these cases we make it
			// equal to the direct estimate
			qui:replace fh_fgt0_`nm´ = mifgt0 if missing(fh_fgt0_`nm´)


			keep fh_fgt0 area
			tempfile fh_`nm´
			save `fh_`nm´´
		restore
	}


	//Open true point estimates
	use `true´, clear

	//Merge in the model based estimates
	merge 1:1 area using `h3arealn´, keepusing(ucln_*)
		drop _merge
	merge 1:1 area using `h3ebln´, keepusing(cebln_*)
		drop _merge

	foreach pp of global plines{
		local nm = subinstr("`pp´",".","_",.)
		merge 1:1 area using `fh_`nm´´
			drop _merge
	}
```

```
        foreach k of global plines{
                local linea = subinstr("`k´",".","_",.)
                foreach g in cebln ucln fh{
                        qui{
                        gen double `g´_bias_fgt0_`linea´ =   ///
             (`g´_fgt0_`linea´-fgt0_`linea´)/`total_sim´
                        gen double `g´_absbias_fgt0_`linea´ = ///
                        abs(`g´_fgt0_`linea´-fgt0_`linea´)/`total_sim´
                        gen double `g´_mse_fgt0_`linea´  =  ///
                        ((`g´_fgt0_`linea´-fgt0_`linea´)^2)/`total_sim´
                        }
                }
        }

        keep area *_bias_* *_mse_*  *_absbias_*

        //For first sim we rename the vector to *T
        if (`z´==1){
                rename *_bias_*    *_bias_*T
                rename *_absbias_* *_absbias_*T
                rename *_mse_*     *_mse_*T

                tempfile Stats
                save `Stats´
        }
        else{ //After the first sim, we add the bias and MSE to *T
                merge 1:1 area using `Stats´
                        drop _merge

                local todrop
                foreach g in cebln ucln fh{
                        foreach k of global plines{
                                local linea = subinstr("`k´",".","_",.)
                                qui{
                                replace `g´_bias_fgt0_`linea´T    = ///
                                `g´_bias_fgt0_`linea´T +    `g´_bias_fgt0_`linea´
                                replace `g´_absbias_fgt0_`linea´T = ///
                                `g´_absbias_fgt0_`linea´T + `g´_absbias_fgt0_`linea´
                                replace `g´_mse_fgt0_`linea´T     = ///
                                `g´_mse_fgt0_`linea´T +     `g´_mse_fgt0_`linea´

                                local todrop `todrop´ `g´_bias_fgt0_`linea´ ///
                                `g´_absbias_fgt0_`linea´ `g´_mse_fgt0_`linea´
                                }
                        }
                }
                local todrop: list uniq todrop
                drop `todrop´
                tempfile Stats
                save `Stats´
        }
        local seedstage `c(rngstate)´
        if (mod(`z´,50)==0){
                //seed to begin next sim if it stops before hand...
                note: "`seedstage´"
                note: "Sim num `z´"
                note: "$plines"
                save "$mdata\bias_in_mymodel_allplines.dta", replace
        }

        dis as error "SIMULATION NUMBER : `z´"
}
```

```
local a=1
foreach x of global plines{
        local nm = subinstr("`x´",".","_",.)
        rename *_bias_fgt0_`nm´T     *_bias_fgt0_ptile`a´
        rename *_absbias_fgt0_`nm´T  *_absbias_fgt0_ptile`a´
        rename *_mse_fgt0_`nm´T      *_mse_fgt0_ptile`a´
        local a =`a´+1
}
save "$mdata\bias_in_mymodel_allplines.dta", replace
```

References

Arora, Vipin, Partha Lahiri, and Kanchan Mukherjee (1997). "Empirical Bayes Estimation of Finite Population Means from Complex Surveys". In: *Journal of the American Statistical Association* 92.440, pp. 1555–1562.

Banerjee, Abhijit V, Angus Deaton, Nora Lustig, Kenneth Rogoff, and Edward Hsu (2006). "An Evaluation of World Bank Research, 1998-2005". In: *Available at SSRN 2950327*.

Blumenstock, Joshua, Jonathan Lain, Isabella Smythe, and Tara Vishwanath (2021). *Using Big Data and Machine Learning to Locate the Poor in Nigeria*. URL: https://blogs.worldbank.org/opendata/using-big-data-and-machine-learning-locate-poor-nigeria.

Chen, Tianqi and Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pp. 785–794.

Chi, Guanghua, Han Fang, Sourav Chatterjee, and Joshua E Blumenstock (2021). "Micro-estimates of Wealth for All Low-and Middle-income Countries". In: *arXiv preprint arXiv:2104.07761*.

Corral, Paul, Kristen Himelein, Kevin McGee, and Isabel Molina (2021). "A Map of the Poor or a Poor Map?" In: *Mathematics* 9.21. ISSN: 2227-7390. DOI: 10.3390/math9212780. URL: https://www.mdpi.com/2227-7390/9/21/2780.

Corral, Paul, Isabel Molina, and Minh Cong Nguyen (2021). "Pull Your Small Area Estimates up by the Bootstraps". In: *Journal of Statistical Computation and Simulation* 91.16, pp. 3304–3357. DOI: 10.1080/00949655.2021.1926460. URL: https://www.tandfonline.com/doi/abs/10.1080/00949655.2021.1926460.

Efron, Bradley and Carl Morris (1975). "Data Analysis Using Stein's Estimator and Its Generalizations". In: *Journal of the American Statistical Association* 70.350, pp. 311–319.

Fay, Robert E and Roger A Herriot (1979). "Estimates of Income for Small Places: An Application of James-Stein Procedures to Census Data". In: *Journal of the American Statistical Association* 74.366a, pp. 269–277.

González-Manteiga, Wenceslao, Maria J Lombardía, Isabel Molina, Domingo Morales, and Laureano Santamaría (2008). "Bootstrap Mean Squared Error of a Small-area EBLUP". In: *Journal of Statistical Computation and Simulation* 78.5, pp. 443–462.

Jean, Neal, Marshall Burke, Michael Xie, W Matthew Davis, David B Lobell, and Stefano Ermon (2016). "Combining Satellite Imagery and Machine Learning to Predict Poverty". In: *Science* 353.6301, pp. 790–794. URL: https://www.science.org/doi/abs/10.1126/science.aaf7894.

Lange, Simon, Utz Johann Pape, and Peter Pütz (2018). "Small Area Estimation of Poverty under Structural Change". In: *World Bank Policy Research Working Paper* 9383.

Marhuenda, Yolanda, Isabel Molina, Domingo Morales, and JNK Rao (2017). "Poverty Mapping in Small Areas under a Twofold Nested Error Regression Model". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 180.4, pp. 1111–1136. DOI: 10.1111/rssa.12306.

Masaki, Takaaki, David Newhouse, Ani Rudra Silwal, Adane Bedada, and Ryan Engstrom (2020). "Small Area Estimation of Non-monetary Poverty with Geospatial Data". In: *World Bank Policy Research Working Paper* 9383.

Molina, Isabel (2019). *Desagregación De Datos En Encuestas De Hogares: Metodologías De Estimación En áreas Pequeñas*. CEPAL. URL: https://repositorio.cepal.org/handle/11362/44214.

Molina, Isabel and JNK Rao (2010). "Small Area Estimation of Poverty Indicators". In: *Canadian Journal of Statistics* 38.3, pp. 369–385.

Natekin, Alexey and Alois Knoll (2013). "Gradient Boosting Machines, a Tutorial". In: *Frontiers in neurorobotics* 7, p. 21.

Nguyen, Viet Cuong (2012). "A Method to Update Poverty Maps". In: *The Journal of Development Studies* 48.12, pp. 1844–1863. DOI: 10.1080/00220388.2012.682983. eprint: https://doi.org/10.1080/00220388.2012.682983. URL: https://doi.org/10.1080/00220388.2012.682983.

Torabi, Mahmoud and JNK Rao (2014). "On Small Area Estimation under a Sub-area Level Model". In: *Journal of Multivariate Analysis* 127, pp. 36–55.

Chapter 6

Model Diagnostics

Small area estimation models used in Chapters 3 and 4 are particular cases of linear mixed models.[1] Under model-based SAE, using either unit-level (chapter 4) or area-level (chapter 3) models, a series of useful model diagnostics can help verify model assumptions and assess model fit. These checks also include residual analysis to detect deviations from the assumed model and detection of influential observations (Rao and Molina 2015). Performing thorough and rigorous model diagnostics as part of the SAE exercise is crucial for ensuring the validity of small area estimates.

This chapter describes the underlying model considered in Section 6.1. Then it provides brief recommendations for model and variable selection in Section 6.2 and concludes with regression diagnosis in Section 6.3.

6.1 The nested-error model

The model used for small area estimation of poverty and welfare, such as the ones proposed by Elbers, Lanjouw, and Lanjouw (2003) and Molina and Rao (2010), assume that the transformed welfare y_{ch}, for each household h within each location c in the population is linearly related to a $1 \times K$ vector of characteristics (or correlates) x_{ch} for that household, according to the nested error model:

$$y_{ch} = x_{ch}\beta + \eta_c + e_{ch},\ h = 1, \ldots, N_c,\ c = 1, \ldots, C, \tag{6.1}$$

where η_c and e_{ch} are respectively location and household-specific idiosyncratic errors, assumed to be independent from each other, following:

$$\eta_c \overset{iid}{\sim} N\left(0, \sigma_\eta^2\right),\ e_{ch} \overset{iid}{\sim} N\left(0, \sigma_e^2\right),$$

where the variances σ_η^2 and σ_e^2 are unknown. Here, C is the number of locations in which the population is divided and N_c is the number of households in location c, for $c = 1, \ldots, C$. Finally, β is the $K \times 1$ vector of coefficients.

As illustrated in Chapter 4, the assumption of normality plays a considerable for EB methods. Deviations from this assumption may lead to biased and noisier estimates, as shown in Corral et al. (2021). Isolated deviations from the model (outliers) and influential observations or even outlying locations may also

[1] Except for the gradient boosting application shown.

exist. Thus, the following sections provide some insights towards selecting a suitable model, as well as checks that may be done to ensure that the chosen model for SAE is adequate.

6.2 Model and variable selection

The objective of model selection is to determine the relevant covariates out of a pool of candidate model variables such that the resulting SAE model generates the most precise estimates possible, noting that it must be possible to measure this precision accurately. Classic approaches to model selection include lasso or stepwise regression. Other literature on the subject includes the fence method, described in Pfeffermann (2013), which involves selecting a model out of a subgroup of correct models that fulfill specific criteria. Rao and Molina (2015) also elaborate on other methods such as Akaike Information Criteria (AIC) type methods, which rely on the marginal restricted log likelihood based on normality of the random effects and the errors. For variable selection under area-level models, particularly Fay-Herriot models, Lahiri and Suntornchost (2015) propose a method where the approximation error converges to zero in probability for large sample sizes.[2]

Since the aim is to arrive at the "true" model, removing all non-significant covariates from the model is recommended as they may introduce noise. It is important not to confuse the estimated noise and the true noise of a given small area estimate. The most common uncertainty measure for an area-specific prediction is the MSE (Tzavidis et al. 2018). In applications of SAE, such as those illustrated for unit-level models, an estimate of the MSE for the small area estimator is obtained through a parametric bootstrap procedure (see section 4.4.2.2). This method differs considerably from the method used in the ELL method, where a single computational algorithm is used to assess the uncertainty. Corral, Molina, and Nguyen (2021), through model-based simulations, present evidence of the fact that the single computational algorithm to estimate noise used in ELL could underestimate the actual MSE of the method.[3]

The script example provided in section 4.5.1 of Chapter 4 uses a lasso approach for model selection that initially includes all suitable covariates and uses 10 fold cross-validation with a shrinkage parameter λ that is within one standard error of the one that minimizes the cross-validated average prediction squared error. The lasso approach employed here does not consider the nested error structure of the model; that is, it is done with the corresponding linear model without the random area effects. Practitioners who are comfortable with R may rely on the `glmmlasso` R package, which may be used for model selection in the model with the assumed nested error structure (see Groll and Tutz (2014) and Groll (2017)).

The lasso selection process yields a selected set of covariates, although some of the included covariates may be non-significant. Consequently, the next step is to remove all the non-significant covariates sequentially. The process starts by removing the most non-significant covariates, one by one. When a covariate is removed, the significance of other covariates may change; thus, after removing each covariate, the model is fit again to determine which covariate to remove next until the remaining ones are all significant. Note that the process used here ignores the magnitude of the coefficients and thus could be further improved.

Finally, it is recommended to remove highly collinear covariates. Once these are removed, the following steps are to identify outliers and influential observations, which may lead to considerably different estimated model parameters, see the next section. For the Fay-Herriot model of Chapter 3 a different

[2]The authors define the approximation error as the difference between standard variable selection criterion and the ideal variable selection criterion, where it is assumed that the direct estimates are not measured with noise.

[3]See Corral et al. (2021) for a detailed discussion on the previous ELL bootstrap. Also see Elbers, Lanjouw, and Lanjouw (2002) for the sources of noise.

approach than the one detailed here was taken. In the taken approach for area-level models, the model started with all possible covariates and began removing covariates, starting with the least significant ones. The removal was done considering the random effects (see section 3.2).

6.3 Regression diagnostics

After the fitting process, checking whether the assumptions from the underlying model are satisfied is recommended. Regression diagnostics for Linear Mixed Models (LMM)[4] are more difficult to interpret than those of the standard linear models, since these models include random effects, which lead to a different covariance structure. Moreover, tools for diagnostics of linear mixed models are less common in software packages, including Stata. Thus, practitioners either must code their own diagnostics or rely on diagnostics often used for linear regression. The model assumptions to keep in mind are: [5]

1. Linearity: the dependent variable y_{ch} is a linear function of the selected vector of covariates x_{ch}.
2. Normality: random effects η_c and errors e_{ch} are normally distributed.
3. Homoskedasticity: errors' variance σ_e^2 is constant, although this assumption may be relaxed by modeling the heteroskedasticity using a model such as the alpha model specification provided by ELL (2003). This can be done when choosing Henderson's Method III fitting in Stata's `sae` package.
4. Independence: errors e_{ch} are independently distributed. Under the nested error model, the assumption is extended so that e_{ch} in location c is unrelated to the corresponding location effect η_c, as well as to all other location effects η_l, $l \neq c$.

Other issues to consider are the detection of influential observations and outliers and, as discussed above, multicollinearity. Although these are not part of the assumptions, their presence may affect the precision of estimates of model parameters and, in turn, model predictions.

The following subsections include some formal and informal ways to check if the assumptions of the underlying model are satisfied. The process starts by eliminating covariates with high multicollinearity, followed by influence analysis and, finally, residual analysis which encompasses most assumptions.[6] Model diagnostics based on residuals and influence measures for special cases of the general linear mixed model can be found in Rao and Molina (2015).

6.3.1 Multicollinearity

Collinearity reduces the accuracy of estimates of regression coefficients, leading to larger standard errors of the coefficients. Under the multiple imputation (MI) inspired bootstrap methods, such as the often used one in the ELL method (see Ch. 4), larger standard errors in the coefficients typically led to larger estimates of noise for the estimators of the indicators of interest. Thus, care was taken to avoid collinearity and multicollinearity. A simple way to detect collinearity is via a correlation matrix, where large absolute correlation coefficients may suggest collinearity problems (James et al. 2013). However,

[4] LMM are an extension of general linear models which include both fixed and random effects.

[5] These assumptions are similar to the ones of a classic linear regression, but adding those for the random effects inclusion.

[6] Many of the commands provided in the following subsections can be easily reviewed by typing `help regress postestimation` in Stata's command window.

collinearity may occur between more than a pair of variables leading to multicollinearity, which cannot be detected under the pairwise correlation matrix (*ibid*).

Under the presence of multicollinearity, the inspection of the correlation matrix is no longer sufficient, and one must instead compute the variance inflation factors (VIFs) (James et al. 2013). The smallest possible value for a VIF is 1. There are multiple rules of thumb as to what is an acceptable VIF. According to James et al. (2013), values exceeding 5 or 10 may require action. After the model is fit, the command `estat vif` may be used to check the variance inflation factor for the covariates included in the model.

Example 1:

Variance inflation factor values above 10 might require special attention since the variable in question could be a linear combination of other independent variables. In the example below, a special Mata function,[7] `_f_stepvif()`, is used to remove covariates with a VIF above a specified threshold; in this case, the chosen value is 3.[8] The function expects the covariate list as its first argument, followed by the sample weights, and finally, the threshold. After the removal of high VIF covariates, the resulting covariate list is returned in a Stata local macro called `vifvar`.

```
*================================================================================
// Collinearity
*================================================================================

reg y $postsign [aw=Whh],r

    //Check for multicollinearity, and remove highly collinear (VIF>3)
        cap drop touse              //remove vector if it is present to avoid error in next step
        gen touse = e(sample)           //Indicates the observations used
        estat vif                       //Variance inflation factor
        local hhvars $postsign

    //Remove covariates with VIF greater than 3
        mata: ds = _f_stepvif("`hhvars´","Whh",3,"touse")
        global postvif `vifvar´

        //VIF check
        reg y $postvif [aw=Whh], r
        vif

    // For ilustration
    // Henderson III GLS - model post removal of non-significant
    sae model h3 y $postsign [aw=Whh], area(HID_mun)

    // Henderson III GLS - model post removal of non-significant
        sae model h3 y $postvif [aw=Whh], area(HID_mun)
```

6.3.2 Influence analysis

Influence analysis is used to detect observations that may have considerable impact on the estimates of the parameters and, consequently, model predictions. These observations include outliers (observations

[7] The Mata function is included in Stata's `sae` package.
[8] The specific threshold value is up to the practitioner, although it is not recommended to use thresholds above 10.

with a large residual, i.e., an observation poorly predicted by the model) and influential observations (the omission of which considerably changes the point estimates of β). These observations can be identified by measuring how far the observation's value for a predictor variable is from the mean or by the size of their studentized residual. Influence analysis is recommended prior to calculating any small area estimate, as these observations may impact the bias and noise of the final small area estimates.

Cook's distance (Cook 1977), also known as Cook's D, measures the effect in the estimated coefficients when an observation is left out or deleted (Rao and Molina 2015). Under ordinary least squares regression, Cook's distance can be obtained easily in Stata with the `cooksd` option after the `predict` command. Nevertheless, under regular OLS, Cook's distance focuses solely on isolated observations, whereas, under the nested error model used for SAE shown in equation 6.1, the analysis of the influence of particular locations may be more relevant. The Stata package `mlt` by Möhring and Schmidt-Catran (2013) may be used to assess the influence on the estimated parameters of particular locations.[9] The `mlt` command estimates Cook's D empirically, making it computationally intensive. The rule of thumb for classifying influential locations is absolute Cook's D values greater than $4/C$, where C is the number of locations into which the population is divided. The `mlt` command will also calculate DFBETAs, which measure the influence of a single location on the coefficient of each covariate. It represents the standardized difference between the coefficient with and without the given location (Möhring and Schmidt-Catran 2013). The rule of thumb for classifying locations as influential is a DFBETA absolute value above $2/\sqrt{C}$, although this should be applied with caution.

Leverage measures the influence on the fitted values of a given observation. Unfortunately, data packages that can obtain leverage under the assumed model are not available. Cameron and Trivedi (2005) present an alternative toward handling influential observations under OLS by using the post estimation command `dfits`, which shows the difference in fits (predictions) with and without the unusual observation. The command combines outliers and leverage into a single statistic. A rule of thumb to identify these observations is if $|dfits| > 2\sqrt{k/n}$, where k is the number of covariates and n is the number of observations. Nevertheless, the removal of observations always entails loss of information (which may be fair) and thus much care should be taken before deciding on removal, and should be done only when, after inspecting the offender, it is determined to be mistaken.

Example 2:

After fitting the model and calculating residuals, problematic observations are identified using several rules of thumb: $Cooks'd > 4/n$, $leverage > (2k+2)/n$ and $abs(rstu) > 2$.

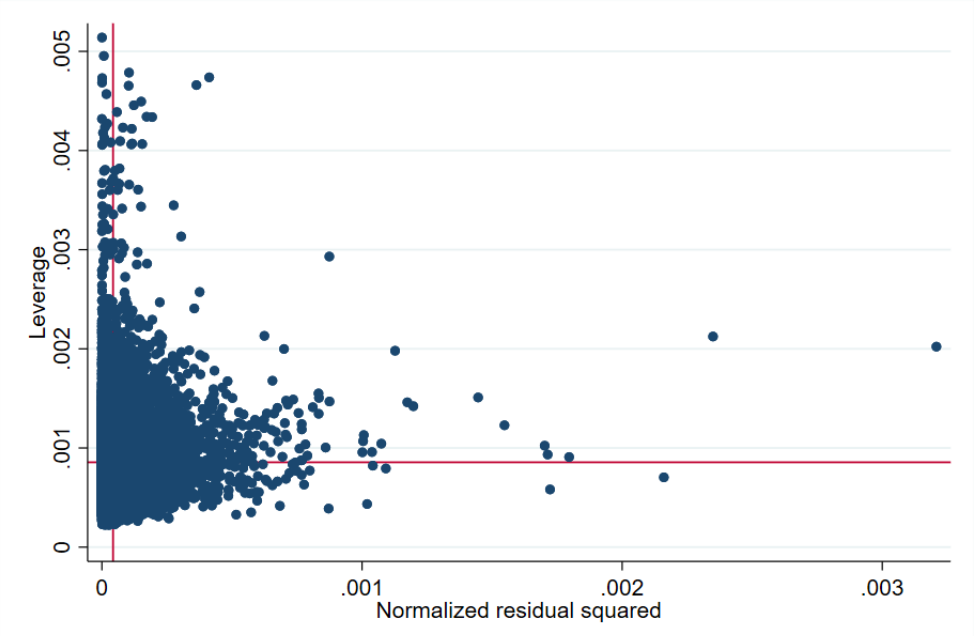
```
// Step 1
    reg y $postvif

    // After regression without weights...

    // Calculate measures to identify influential observations
        predict cdist, cooksd      // calculates the Cook´s D influence statistic
        predict rstud, rstudent    // calculates the Studentized (jackknifed) residuals
// Step 2
    reg y $postvif [aw=Whh]
```

[9] https://www.stata.com/meeting/germany12/abstracts/desug12_moehring.pdf

```
// Predict leverage and residuals
    predict lev, leverage  // calculates the diagonal elements of the
                        // projection ("hat") matrix
    predict r, resid       // calculates the residuals

// Save useful locals
local myN=e(N)              // # observations
    local myK=e(rank)           // rank or k
local KK =e(df_m)           // degrees of freedom (k-1)

sum cdist, d
* return list
    local max = r(max)          // max value
    local p99 = r(p99)                  // percentile 99

// Step 3

// For ilustration...
// We have influential data points...
reg lny $postvif if cdist<4/`myN´ [aw=Whh]
reg lny $postvif if cdist<`p99´   [aw=Whh]
    reg lny $postvif if cdist<`max´   [aw=Whh]

// Identified influential / outliers observations
gen nogo = abs(rstud)>2 & cdist>4/`myN´ & lev>(2*`myK´+2)/`myN´

count if nogo==1        // these are the obs that we want to eliminate
```

A graphical representation of the squared normalized residual versus leverage (`lvr2plot`), before and after the elimination of influential observations, is an easy way to identify potentially influential observations and outliers (Fig. 6.1). The two reference lines are the means of the leverage and the squared normalized residual. Many points are outside these two reference points and should be scrutinized before deciding to remove them.

Figure 6.1: Squared normalized residual vs leverage



Source: own elaboration from code in Appendix 6.4.2. The red lines are references to the mean leverage on the Y axis and the mean squared normalized residual on the X axis. Observations that are far away from these reference points should be inspected more closely.

6.3.3 Residual analysis

Most techniques for residual analysis rely on visual inspection of the graphed residuals. Residuals should be checked for linearity, normality, and constant variance in case homoskedasticity is assumed.

6.3.3.1 Linearity

The nested error model used in SAE (Eq. 6.1) assumes that the outcome variable y is a linear function of the covariates.[10] If a single covariate is used, a scatter plot of the residual versus the covariate is enough to see if a linear relationship exists. When several covariates are used, checking for linearity is somewhat more complex.

To assess linearity in a simple regression, use `scatter` to produce a plot of y versus x, `lfit` to fit a regression line, and `lowess` to show a smoothed fit.

```
. reg depvar indepvar
. twoway (scatter depvar indepvar)(lfit depvar indepvar)(lowess depvar indepvar)
```

For multiple regression:

```
. reg depvar indepvars
. predict r, residual
. scatter r indepvar1
. scatter r indepvar2
```

Other checks for non-linearity include `acprlot` and `cprplot`, which produce graphs of an augmented component-plus-residual plot and a residual plot, respectively.

```
. reg depvar indepvars
. acprplot indepvar1, lowess lsopts(bwidth(1))
. acprplot indepvar2, lowess lsopts(bwidth(1))
```

When a clear non-linear pattern is observed, transformations of the independent variable might help.

```
. graph matrix depvar indepvars, half
. kdensity indepvar, normal
. gen logvar=log(indepvar)
. kdensity logvar, normal
```

```
// Linearity
    reg y $postsign, r

    // Augmented component-plus-residual plot; works better than the
    // component-plus-residual plot for identifying nonlinearities in the data.
    acprplot age_hh, lowess lsopts(bwidth(1)) graphregion(color(white)) msize(small)

    graph export "$figs\acprplot_age_hh.png", as(png) replace
```

[10] Much of the information in this section is borrowed from UCLA: Statistical Consulting Group (2022); https://stats.idre.ucla.edu/stata/webbooks/reg/chapter2/stata-webbooksregressionwith-statachapter-2-regression-diagnostics/

```
// Kernel density plot  for log_age_hh with a normal density overlaid
kdensity age_hh, normal graphregion(color(white)) msize(small)

graph export "$figs\kdensity_age_hh.png", as(png) replace

// log transformation
gen log_age_hh =log(age_hh)

// Kernel density plot  for log_age_hh with a normal density overlaid
kdensity log_age_hh, normal graphregion(color(white)) msize(small)

graph export "$figs\kdensity_log_age_hh.png", as(png) replace
```

6.3.3.2 Tests for normality of residuals and random effects

Model errors and random effects are assumed to be normally distributed under the nested error model used for SAE. Deviations from normality may lead to considerable bias in the final small area estimates. Scatter plots of residuals against fitted values and normal Q-Q plots of residuals provide a natural way to identify outliers or influential observations that might affect the precision of the estimates. West, Welch, and Galecki (2014) mentioned that the random effects vector is assumed to follow a multivariate normal distribution. Thus, the information from the observations sharing the same random effect is used to predict (instead of estimating) the values of that random effect in the model.

The usual predictors of the random effects under linear mixed models are known as Empirical Best Linear Unbiased Predictors (EBLUPs), since they are the most precise linear unbiased predictors. They use the weighted least squares estimates of β, and the variance parameters are replaced by suitable estimates (Robinson 1991). Rao and Molina (2015) provide a comprehensive formal derivation of the EBLUPs, and applications beyond small area estimation can be found in West, Welch, and Galecki (2014).

The `xtmixed` or `mixed` command may be used to check the validity of a linear mixed model in Stata. Deviations from normality can be observed in a normal Q-Q plot of residuals (Figure 6.2), which displays sample quantiles of unit-level residuals against the theoretical quantiles of a normal distribution plot:

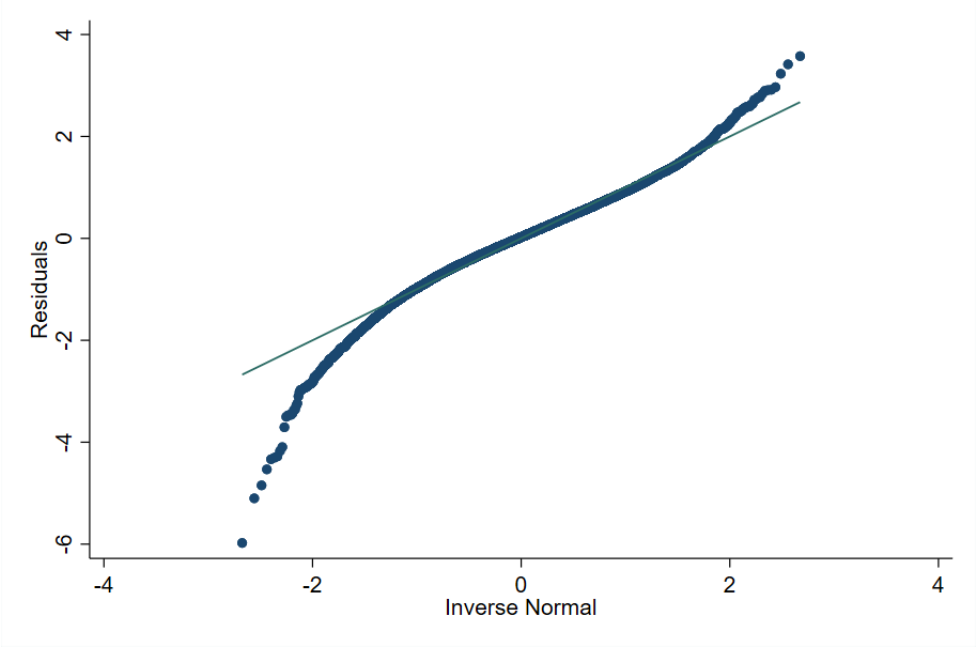
```
mixed depvar indepvars || area:, reml

predict res, residual

qnorm res
```

It is also important to check the assumptions regarding the random effect's distribution. This may be done after fitting the linear mixed model and obtaining the predicted random effects (Figure 6.3):

```
predict eta, reffects

bysort area:  gen first = _n

qnorm eta if first==1
```

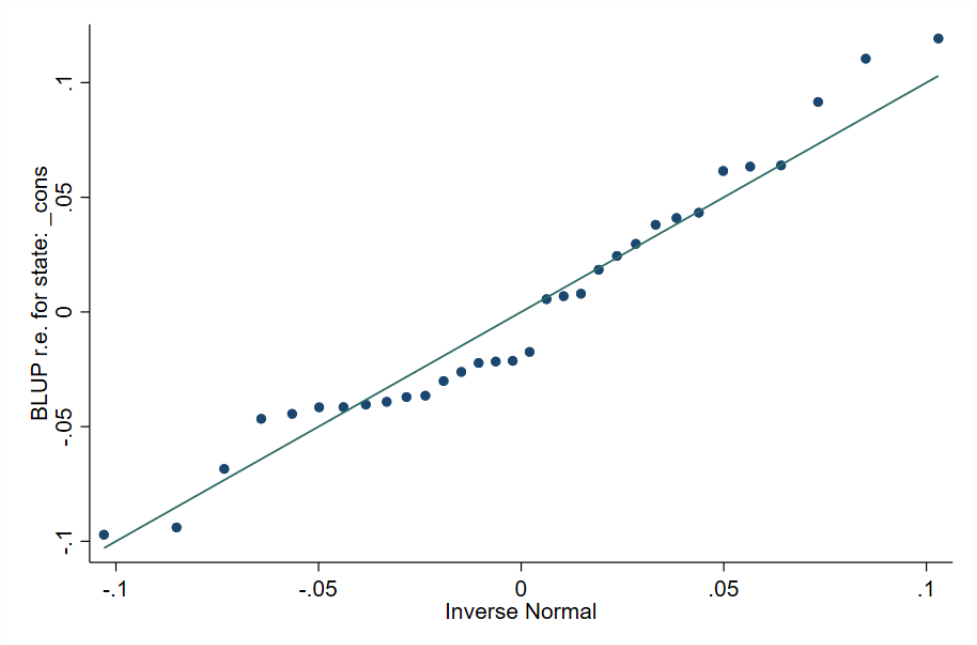
The plots of empirical quantiles compared to theoretical quantiles of the normal distribution (normal Q-Q plots) are helpful to detect deviations from normality. Transformations of the dependent variable are often taken to ensure that empirical quantiles are better aligned to the theoretical ones. As Marhuenda et al. (2017) note, in real life applications, the exact fit to a distribution is hardly met.

Figure 6.2: Sample quantiles of residuals against theoretical quantiles of a Normal distribution



Source: own elaboration from code in Appendix 6.4.2. Deviations from normality can be observed in the figure particularly at the bottom. Marhuenda et al. (2017) notes that, in applications using real data, the exact fit to a distribution is hardly met; it is recommended that practitioners apply several transformations to the dependent variable and select the one that provides the best approximation to the nested error model's assumptions.

Figure 6.3: Sample quantiles of predicted random effects against theoretical quantiles of a Normal distribution



Source: own elaboration.

6.4 Appendix

6.4.1 Useful definitions and concepts

Definitions are based on Molina and Marhuenda (2015), Ghosh and Rao (1994), Rao and Molina (2015), and Cochran (2007).

- **Small area:** a domain (area) is regarded as small if the domain-specific sample size is not large enough to support direct estimates of adequate precision. That is to say, a small area is any domain of interest for which direct estimates of adequate precision cannot be produced.
- **Large area:** an area or domain where a direct estimate of the parameter of interest for the area has a sufficiently large sample to yield estimates with the desired precision.
- **Domain:** domain may be defined by geographic delimitations/territories (state, county, school district, health service area), socio-demographic groups, or both (specific age-sex-race group within a large geographic area) or even other types of sub-populations (set of firms belonging to a census division by industry group).
- **Direct (domain) estimators:** estimators based only on the domain-specific sample area. These estimators are typically design-unbiased but tend to have low precision in small areas.
- **Indirect (domain) estimators:** estimator that uses information from other areas, under the assumption that there exists some homogeneity relationship between them.
- **Target parameter:** indicator to be estimated. Some examples are population mean, proportion, and rate.
- **Efficiency/precision:** 1/variance when an estimator is unbiased; 1/MSE otherwise.
- **Sampling error:** error from using a sample from the population rather than the whole population.

For a better understanding of statistical inference, the following concepts/definitions are necessary. Definitions and observations are from Molina and García-Portugues (2021) and Rao and Molina (2015). Note that, when dealing with SAE, the bias for each area $c = 1, \ldots, C$ is of interest, not the bias on average over all the areas.

- **Unbiased estimator:** if an estimator's bias is equal to zero for all the parameter values. If the expected value of the estimator is equal to the parameter. The estimator $\hat{\vartheta}_c$ of parameter ϑ_c is unbiased if and only if $E(\hat{\vartheta}_c - \vartheta_c) = 0$
- **Estimation error:** the estimation error $\hat{\vartheta}_c - \vartheta_c$ is typically different from zero even if the estimator is unbiased. The bias is the mean estimation error: $Bias(\hat{\vartheta}_c) = E(\hat{\vartheta}_c - \vartheta_c)$.
- **Mean squared (estimation) error (MSE):** is also called mean squared prediction error (MSPE) or prediction mean squared error (PMSE). $MSE(\hat{\vartheta}_c) = E[(\hat{\vartheta}_c - \vartheta_c)^2]$.
- **Standard error (SE) of $\hat{\vartheta}_c$:** is the standard deviation of the sampling distribution of $\hat{\vartheta}_c$.
- **Coefficient of variation (CV) of ϑ:** $cv(\hat{\vartheta}) = s(\hat{\vartheta})/\vartheta$ is the associated standard error of the estimate over the true value of ϑ. It is also known as the relative standard deviation (RSD). The estimated CV is then $\hat{cv}(\hat{\vartheta}) = s(\hat{\vartheta})/\hat{\vartheta}$.
- **Consistency:** when increasing the sample size n, the probability that the estimator differs from the true value ϑ_c by more than ε vanishes for every $\varepsilon > 0$.

6.4.2 Regression diagnostics do-file

The following do-file provides an example of how to do regression diagnostics. Note that the checks are not exhaustive and only serve as a guide for the practitioner.

```
clear all
set more off
/*==============================================================================
Do-file prepared for SAE Guidelines
- Regression Diagnostics
- authors Paul Corral, Minh Nguyen & Sandra Segovia
*==============================================================================*/

global main       "C:\Users\\`c(username)´\OneDrive\SAE Guidelines 2021"
global section    "$main\4_Model_selection"

global data      "$section\1_data"
global dofile    "$section\2_dofiles"
global figs      "$section\3_figures"

local survey       "$data\survey_public.dta"
*local census      "$main\3_Unit_level\1_data\census_public.dta"

//global with candidate variables.
global myvar rural  lnhhsize age_hh male_hh  piped_water no_piped_water ///
no_sewage sewage_pub sewage_priv electricity telephone cellphone internet ///
computer washmachine fridge television share_under15 share_elderly ///
share_adult max_tertiary max_secondary HID_* mun_* state_*

version 15
set seed 648743

local graphs graphregion(color(white)) xsize(9) ysize(6) msize(small)
*==============================================================================
// End of preamble
*==============================================================================

// First part is just as the model selection dofile

// Load in survey data
use "`survey´", clear

        //Remove small incomes affecting model
        drop if e_y<1

    // Kernel density plot  for e_y with a normal density overlaid
    kdensity e_y, normal `graphs´

    graph export "$figs\kdensity_e_y.png", as(png) replace

        //Log shift transformation to approximate normality
        lnskew0 double bcy = exp(lny)

    // Kernel density plot  for lnywith a normal density overlaid
    kdensity lny, normal `graphs´

    graph export "$figs\kdensity_lny.png", as(png) replace

    // Kernel density plot  for bcy with a normal density overlaid
    kdensity bcy, normal `graphs´

    graph export "$figs\kdensity_bcy.png", as(png) replace

    // removes skeweness from distribution
```

```
sum e_y, d
    sum lny, d
    sum bcy, d

    // Data has already been cleaned and prepared. Data preparation and the creation
    // of eligible covariates is of extreme importance.
    // In this instance, we skip these comparison steps because the sample is
    // literally a subsample of the census.
    codebook HID //10 digits, every single one
    codebook HID_mun //7 digits every single one

    //We rename HID_mun
    rename HID_mun MUN
    //Drop automobile, it is missing
    drop *automobile* //all these are missing

    //Check to see if lassoregress is installed, if not install
    cap which lassoregress
    if (_rc) ssc install elasticregress

    //Model selection - with Lasso
    gen lnhhsize = ln(hhsize)
    lassoregress bcy  $myvar [aw=Whh], lambda1se epsilon(1e-10) numfolds(10)
    local hhvars = e(varlist_nonzero)
    global postlasso  `hhvars´

    //Try Henderson III GLS
    sae model h3 bcy $postlasso [aw=Whh], area(MUN)

    //Rename HID_mun
    rename MUN HID_mun

    //Loop designed to remove non-significant covariates sequentially
    forval z= 0.5(-0.05)0.05{
            qui:sae model h3 bcy `hhvars´ [aw=Whh], area(HID_mun)
            mata: bb=st_matrix("e(b_gls)")
            mata: se=sqrt(diagonal(st_matrix("e(V_gls)")))
            mata: zvals = bb´:/se
            mata: st_matrix("min",min(abs(zvals)))
            local zv = (-min[1,1])
            if (2*normal(`zv´)<`z´) exit

            foreach x of varlist `hhvars´{
                    local hhvars1
                    qui: sae model h3 bcy `hhvars´ [aw=Whh], area(HID_mun)
                    qui: test `x´
                    if (r(p)>`z´){
                            local hhvars1
                            foreach yy of local hhvars{
                                    if ("`yy´"=="`x´") dis ""
                                    else local hhvars1 `hhvars1´ `yy´
                            }
                    }
                    else local hhvars1 `hhvars´
                    local hhvars `hhvars1´
            }
    }

    global postsign `hhvars´

*================================================================================
```

```
// Regression diagnostics
*================================================================================

/*This is not a complete diagnostic; it is just a preview, steps & repetitions
depend on the underlying model. Check all vars, check different transformations,
do not forget a model for heteroskedasticity (alpha model) if needed */

rename bcy y

*================================================================================
// Collinearity
*================================================================================

reg y $postsign [aw=Whh],r

    //Check for multicollinearity, and remove highly collinear (VIF>3)
        cap drop touse              //remove vector if it is present to avoid error in next step
        gen touse = e(sample)           //Indicates the observations used
        estat vif                                   //Variance inflation factor
        local hhvars $postsign

    //Remove covariates with VIF greater than 3
        mata: ds = _f_stepvif("`hhvars´","Whh",3,"touse")
        global postvif `vifvar´

        //VIF check
        reg y $postvif [aw=Whh], r
        vif

    // For ilustration
    // Henderson III GLS - model post removal of non-significant
    sae model h3 y $postsign [aw=Whh], area(HID_mun)

    // Henderson III GLS - model post removal of non-significant
        sae model h3 y $postvif [aw=Whh], area(HID_mun)

*================================================================================
// Residual Analysis
*================================================================================

// Linearity

    reg y $postsign, r

    // Augmented component-plus-residual plot; works better than the
    // component-plus-residual plot for identifying nonlinearities in the data.
    acprplot age_hh, lowess lsopts(bwidth(1)) `graphs´

    graph export "$figs\acprplot_age_hh.png", as(png) replace

    // Kernel density plot  for log_age_hh with a normal density overlaid
    kdensity age_hh, normal `graphs´

    graph export "$figs\kdensity_age_hh.png", as(png) replace

    // log transformation
    gen log_age_hh =log(age_hh)

    // Kernel density plot  for log_age_hh with a normal density overlaid
    kdensity log_age_hh, normal `graphs´

    graph export "$figs\kdensity_log_age_hh.png", as(png) replace
```

```
// Normality

    reg y $postvif [aw=Whh],r
    predict resid, resid

    // Kernel density plot  for residuals with a normal density overlaid
    kdensity resid, normal `graphs´

    graph export "$figs\kdensity_resid.png", as(png) replace

    // Standardized normal probability
    pnorm resid , `graphs´

    graph export "$figs\pnorm.png", as(png) replace

    // Quantiles of a variable against the quantiles of a normal distribution
    qnorm resid , `graphs´

    graph export "$figs\qnorm.png", as(png) replace

    // Numerical Test:  Shapiro-Wilk W test for normal data
    swilk resid


// Heteroscedasticity

    reg y $postvif

    // Residuals vs fitted values with a reference line at y=0
    rvfplot , yline(0)  `graphs´

    graph export "$figs\rvfplot_1.png", as(png) replace

    // Cameron & Trivedi´s decomposition of IM-test / White test
    estat imtest

    // Breusch-Pagan / Cook-Weisberg test for heteroskedasticity
    estat hettest


*================================================================================
// Influence Analysis
*================================================================================

// Graphic method < before >

    reg y $postvif [aw=Whh]

    // residuals vs fitted vals
    rvfplot , yline(0)  `graphs´

    graph export "$figs\rvfplot_2.png", as(png) replace


    // normalized residual squared vs leverage
    lvr2plot ,  `graphs´

    graph export "$figs\lvr2plot.png", as(png) replace

// Numerical method

// Step 1

    reg y $postvif

    // After regression without weights...
```

```
    // Calculate measures to identify influential observations
        predict cdist, cooksd      // calculates the Cook´s D influence statistic
        predict rstud, rstudent    // calculates the Studentized (jackknifed) residuals
// Step 2
    reg y $postvif [aw=Whh]

    // Predict leverage and residuals
        predict lev, leverage  // calculates the diagonal elements of the
                           // projection ("hat") matrix
        predict r, resid       // calculates the residuals
    // Save useful locals
    local myN=e(N)              // # observations
        local myK=e(rank)           // rank or k
    local KK =e(df_m)           // degrees of freedom (k-1)

    sum cdist, d
    * return list
        local max = r(max)          // max value
        local p99 = r(p99)                  // percentile 99
 // Step 3

    // For ilustration...
    // We have influential data points...
    reg lny $postvif if cdist<4/`myN´ [aw=Whh]
    reg lny $postvif if cdist<`p99´   [aw=Whh]
        reg lny $postvif if cdist<`max´   [aw=Whh]
    // Identified influential / outliers observations
    gen nogo = abs(rstud)>2 & cdist>4/`myN´ & lev>(2*`myK´+2)/`myN´

    count if nogo==1        // these are the obs that we want to eliminate


// Graphic method < after >
    reg y $postvif [aw=Whh] if nogo==0

    // residuals vs fitted vals
    rvfplot , yline(0)  `graphs´

    graph export "$figs\rvfplot_2_after.png", as(png) replace


    // normalized residual squared vs leverage
    lvr2plot ,        `graphs´

    graph export "$figs\lvr2plot_after.png", as(png) replace


*================================================================================
// Model Specification tests
*================================================================================

   reg y $postvif

// Wald test for ommited vars < will compare with previous regression>
   boxcox y $postvif, nolog // Box - Cox model


// Functional form of the conditional mean
```

```
    reg y $postvif

    estat ovtest // performs regression specification error test (RESET) for omitted variables

    linktest //performs a link test for model specification

// Omnibus tests + Heteroscedasticity tests
    reg y $postvif

    estat imtest   // Cameron & Trivedi´s decomposition of IM-test / White test

    estat hettest // Breusch-Pagan / Cook-Weisberg test for Heteroscedasticity

*================================================================================
// Diagnostics for random effects
*================================================================================

// Multilevel mixed-effects linear regression
    mixed y $postvif || state:, reml

    predict res, residual

    predict eta, reffects

    bysort state: gen first=_n

    // For state == 1
    qnorm eta if first==1 , `graphs´

    graph export "$figs\qnorm_mixed_1.png", as(png) replace

// Quantiles of a variable against the quantiles of a normal distribution
    qnorm res , `graphs´

    graph export "$figs\qnorm_mixed.png", as(png) replace
```

References

Cameron, A Colin and Pravin K Trivedi (2005). *Microeconometrics: Methods and Applications*. Cambridge university press. ISBN: 978-0-521-84805-3.

Cochran, William G (2007). *Sampling Techniques*. John Wiley & Sons.

Cook, R Dennis (1977). "Detection of Influential Observation in Linear Regression". In: *Technometrics* 19.1, pp. 15–18.

Corral, Paul, Kristen Himelein, Kevin McGee, and Isabel Molina (2021). "A Map of the Poor or a Poor Map?" In: *Mathematics* 9.21. ISSN: 2227-7390. DOI: 10.3390/math9212780. URL: https://www.mdpi.com/2227-7390/9/21/2780.

Corral, Paul, Isabel Molina, and Minh Cong Nguyen (2021). "Pull Your Small Area Estimates up by the Bootstraps". In: *Journal of Statistical Computation and Simulation* 91.16, pp. 3304–3357. DOI: 10.1080/00949655.2021.1926460. URL: https://www.tandfonline.com/doi/abs/10.1080/00949655.2021.1926460.

Elbers, Chris, Jean O Lanjouw, and Peter Lanjouw (2003). "Micro–level Estimation of Poverty and Inequality". In: *Econometrica* 71.1, pp. 355–364.

Elbers, Chris, Jean Olson Lanjouw, and Peter Lanjouw (2002). "Micro-level Estimation of Welfare". In: *World Bank Policy Research Working Paper* 2911.

Ghosh, Malay and JNK Rao (1994). "Small Area Estimation: An Appraisal". In: *Statistical science* 9.1, pp. 55–76.

Groll, Andreas (2017). "glmmLasso: Variable Selection for Generalized Linear Mixed Models by L1-penalized Estimation". In: *R package version* 1.1, p. 25.

Groll, Andreas and Gerhard Tutz (2014). "Variable Selection for Generalized Linear Mixed Models by L 1-penalized Estimation". In: *Statistics and Computing* 24.2, pp. 137–154.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani (2013). *An Introduction to Statistical Learning*. Vol. 112. Springer. ISBN: 978-1-0716-1418-1. URL: https://link.springer.com/book/10.1007/978-1-0716-1418-1?noAccess=true.

Lahiri, Partha and Jiraphan Suntornchost (2015). "Variable Selection for Linear Mixed Models with Applications in Small Area Estimation". In: *Sankhya B* 77.2, pp. 312–320.

Marhuenda, Yolanda, Isabel Molina, Domingo Morales, and JNK Rao (2017). "Poverty Mapping in Small Areas under a Twofold Nested Error Regression Model". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 180.4, pp. 1111–1136. DOI: 10.1111/rssa.12306.

Möhring, Katja and Alexander Schmidt-Catran (2013). *MLT: Stata Module to Provide Multilevel Tools*. URL: https://econpapers.repec.org/software/bocbocode/s457577.htm.

Molina, Isabel and Eduardo García-Portugues (2021). *A First Course on Statistical Inference*. Accessed: 2010-06-22. Bookdown.org. URL: https://bookdown.org/egarpor/inference/.

Molina, Isabel and Yolanda Marhuenda (2015). "Sae: An R Package for Small Area Estimation". In: *The R Journal* 7.1, pp. 81–98.

Molina, Isabel and JNK Rao (2010). "Small Area Estimation of Poverty Indicators". In: *Canadian Journal of Statistics* 38.3, pp. 369–385.

Pfeffermann, Danny (2013). "New Important Developments in Small Area Estimation". In: *Statistical Science* 28.1, pp. 40–68.

Rao, JNK and Isabel Molina (2015). *Small Area Estimation*. 2nd. John Wiley & Sons.

Robinson, George K (1991). "That BLUP Is a Good Thing: The Estimation of Random Effects". In: *Statistical science* 6.1, pp. 15–32.

Tzavidis, Nikos, Li-Chun Zhang, Angela Luna, Timo Schmid, and Natalia Rojas-Perilla (2018). "From Start to Finish: A Framework for the Production of Small Area Official Statistics". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 181.4, pp. 927–979.

UCLA: Statistical Consulting Group (2022). *Regression with Stata Chapter 2 – Regression Diagnostics*. URL: https://stats.oarc.ucla.edu/stata/dae/robust-regression/.

West, Brady T, Kathleen B Welch, and Andrzej T Galecki (2014). *Linear Mixed Models: A Practical Guide Using Statistical Software*. 2nd. Chapman and Hall/CRC. DOI: 10.1201/b17198. URL: https://www.taylorfrancis.com/books/mono/10.1201/b17198/linear-mixed-models-brady-west-kathleen-welch-andrzej-galecki.

Chapter 7

Concluding Remarks

The objective of these Guidelines is to illustrate the application of small area estimation methods to obtain estimates of welfare indicators at a geographical level below that achieved with regular sample surveys. They concentrate on the production of small area estimates of poverty and related indicators through model-based small area estimation – both area- and unit-level models. The Guidelines are expected to serve as a practical guide to individuals working on the production of small area estimates of poverty so that informed choices can be made.

The main contribution of the guidelines is to provide a clear and easy-to-follow set of recommendations. Toward that goal, it organizes some of the existing evidence on the best practices for poverty mapping under the considered methodologies to help practitioners make considered choices given their existing data limitations. The Guidelines also provide easy-to-follow and replicable scripts for much of the analysis presented, such that interested readers may deepen their understanding of the methods. Finally, the advantages and disadvantages of each method are presented to facilitate the user's decision-making.

In summary, the Guidelines present evidence on what approach may be best suited in different data environments. Evidence suggests area-level models such as Fay-Herriot are quite useful when access to census microdata is not possible, although the gains in precision may be limited. When access to census microdata is possible, and data between survey and census is comparable, unit-level methods such as CensusEB will likely yield estimates of the highest quality. The importance of data transformation to approximate the model's assumptions has also been highlighted. Additionally, the importance of estimating location effects at the same level as the one where estimates will be reported is noted. Evidence is also provided showing that when the level of random location effects and the level of reporting are not aligned, it will yield noisier estimates.

The Guidelines also note that the small area estimation of poverty is an active research field, and methods are constantly being improved. Some of the more recent modeling approaches discussed in the Guidelines require further methodological work. Interesting approaches have been suggested for instances where the census and survey data are not aligned. However, there is still much work to be done to refine these methodologies and address the concerns highlighted in these guidelines and elsewhere. In particular, hybrid alternatives such as unit-context models, because these are unable to replicate the full welfare distribution, are likely to yield biased estimates. Moreover, since the noise estimates for unit-context models are likely incorrect, their precision cannot be adequately evaluated. Similarly, the Guidelines also suggest that the lack of adequate noise estimates for machine learning approaches such as gradient-boosting limits their usefulness for the purposes of small area estimation of poverty, despite the method showing promising results in a validation using the Mexican Intracensal survey. Future research

should focus on resolving these methodological concerns and constraints, as this would facilitate the mainstreaming of these methods and further enhance the practitioners' ability to produce reliable small area estimates of poverty when access to census microdata is not available.