Variance Estimation of Indicators on Social Exclusion and Poverty using the R Package laeken

Matthias Templ[1], Andreas Alfons[2]

Abstract This vignette illustrates the application of variance estimation procedures to indicators on social exclusion and poverty using the R package **laeken**. To be more precise, it describes a general framework for estimating variance and confidence intervals of indicators under complex sampling designs. Currently, the package is focused on bootstrap approaches. While the naive bootstrap does not modify the weights of the bootstrap samples, a calibrated version allows to calibrate each bootstrap sample on auxiliary information before deriving the bootstrap replicate estimate.

1 Introduction

When point estimates of indicators are computed from samples, it is important to also obtain variance estimates and confidence intervals in order to account for variability due to sampling. Other sources of variability such as data editing or imputation may need to be considered as well, but this is not further discussed in this paper. While this vignette targets the topic of variance and confidence interval estimation for the indicators on social exclusion and poverty according to Eurostat (2004, 2009), the aim is not to describe and evaluate the different approaches that have been proposed to date. Instead, the aim is to present the functionality for the statistical environment R (R Development Core Team 2011) implemented in the add-on package **laeken** (Alfons et al. 2012).

It should be noted that the basic design of the package, as well as standard point estimation of the indicators on social exclusion and poverty, is discussed in detail in vignette `laeken-standard` (Templ and Alfons 2011). In addition, vignette `laeken-pareto` (Alfons et al. 2011b) presents more sophisticated methods for point estimation of the indicators, which are less influenced by outliers. Those documents can be viewed from within R with the following commands:

```
R> vignette("laeken-standard")
R> vignette("laeken-pareto")
```

The data basis for the estimation of the indicators on social exclusion and poverty is the *European Union Statistics on Income and Living Conditions* (EU-SILC), which is an annual panel survey conducted in EU member states and other European countries. Package **laeken** provides the synthetic example data `eusilc` consisting of 14 827 observations from 6 000 households. Furthermore, the data were generated from Austrian EU-SILC survey data from 2006 using the data simulation methodology proposed by Alfons et al. (2011a) and implemented in the R package **simPopulation** (Alfons and Kraft 2010). The data set `eusilc` is used in the code examples throughout the paper.

[1] Department of Statistics and Probability Theory, Vienna University of Technology
Methods Unit, Statistics Austria
E-mail: templ@tuwien.ac.at

[2] ORSTAT Research Center, Faculty of Business and Economics, KU Leuven
E-mail: andreas.alfons@kuleuven.be

```
R> library("laeken")
R> data("eusilc")
```

The rest of the paper is organized as follows. Section 2 presents the general wrapper function for estimating variance and confidence intervals of indicators in package **laeken**. The naive and calibrated bootstrap approaches are discussed in Sections 3 and 4, respectively. Section 5 concludes.

2 General wrapper function for variance estimation

The function `variance()` provides a flexible framework for estimating the variance and confidence intervals of indicators such as the *at-risk-of-poverty rate*, the *Gini coefficient*, the *quintile share ratio* and the *relative median at-risk-of-poverty gap*. For a mathematical description and details on the implementation of these indicators in the R package **laeken**, the reader is referred to vignette `laeken-standard` (Templ and Alfons 2011). In any case, `variance()` acts as a general wrapper function for computing variance and confidence interval estimates of indicators on social exclusion and poverty with package **laeken**. The arguments of function `variance()` are shown in the following:

```
R> args(variance)

function (inc, weights = NULL, years = NULL, breakdown = NULL,
    design = NULL, data = NULL, indicator, alpha = 0.05, na.rm = FALSE,
    type = "bootstrap", gender = NULL, method = "mean", ...)
NULL
```

All these arguments are fully described in the R help page of function `variance()`. The most important arguments are:

inc: the income vector.

weights: an optional vector of sample weights.

breakdown: an optional vector giving different domains in which variances and confidence intervals should be computed.

design: an optional vector or factor giving different strata for stratified sampling designs.

data: an optional `data.frame`. If supplied, each of the above arguments should be specified as a character string or an integer or logical vector specifying the corresponding column.

indicator: an object inheriting from the class `"indicator"` that contains the point estimates of the indicator, such as `"arpr"` for the at-risk-of-poverty rate, `"qsr"` for the quintile share ratio, `"rmpg"` for the relative median at-risk-of-poverty gap, or `"gini"` for the Gini coefficient.

type: a character string specifying the type of variance estimation to be used. Currently, only `"bootstrap"` is implemented for variance estimation based on bootstrap resampling.

In the following sections, two bootstrap methods for estimating the variance and confidence intervals of point estimates for complex survey data are described. Furthermore, their application using the function `variance()` from package **laeken** is demonstrated.

3 Naive bootstrap

Let $\boldsymbol{X} := (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$ denote a survey sample with n observations and p variables. Then the *naive bootstrap algorithm* for estimating the variance and confidence interval of an indicator can be summarized as follows:

1. Draw R independent bootstrap samples $\boldsymbol{X}_1^*, \ldots, \boldsymbol{X}_R^*$ from $\boldsymbol{X}$.

2. Compute the bootstrap replicate estimates $\hat{\theta}_r^* := \hat{\theta}(\boldsymbol{X}_r^*)$ for each bootstrap sample $\boldsymbol{X}_r^*$, $r = 1, \dots, R$, where $\hat{\theta}$ denotes an estimator for a certain indicator of interest. Of course the sample weights always need to be considered for the computation of the bootstrap replicate estimates.

3. Estimate the variance $V(\hat{\theta})$ by the variance of the R bootstrap replicate estimates:

$$\hat{V}(\hat{\theta}) := \frac{1}{R-1} \sum_{r=1}^{R} \left(\hat{\theta}_r^* - \frac{1}{R} \sum_{s=1}^{R} \hat{\theta}_s^* \right)^2 . \tag{1}$$

4. Estimate the confidence interval at confidence level $1 - \alpha$ by one of the following methods (for details, see Davison and Hinkley 1997):

   Percentile method: $\left[\hat{\theta}^*_{((R+1)\frac{\alpha}{2})}, \hat{\theta}^*_{((R+1)(1-\frac{\alpha}{2}))}\right]$, as suggested by Efron and Tibshirani (1993).

   Normal approximation: $\hat{\theta} \pm z_{1-\frac{\alpha}{2}} \cdot \hat{V}(\hat{\theta})^{1/2}$ with $z_{1-\frac{\alpha}{2}} = \Phi^{-1}(1 - \frac{\alpha}{2})$.

   Basic bootstrap method: $\left[2\hat{\theta} - \hat{\theta}^*_{((R+1)(1-\frac{\alpha}{2}))}, 2\hat{\theta} - \hat{\theta}^*_{((R+1)\frac{\alpha}{2})}\right]$.

   For the percentile and the basic bootstrap method, $\hat{\theta}^*_{(1)} \leq \dots \leq \hat{\theta}^*_{(R)}$ denote the order statistics of the bootstrap replicate estimates.

In the following example, the variance and confidence interval of the at-risk-of-poverty rate are estimated with the naive bootstrap procedure. The output of function `variance()` is an object of the same class as the point estimate supplied as the `indicator` argument, but with additional components for the variance and confidence interval. In addition to the point estimate, the income and the sample weights need to be supplied. Furthermore, a stratified sampling design can be considered by specifying the `design` argument, in which case observations are resampled separately within the strata. To ensure reproducibility of the results, the seed of the random number generator is set.

```
R> a <- arpr("eqIncome", weights = "rb050", data = eusilc)
R> variance("eqIncome", weights = "rb050", design = "db040",
+     data = eusilc, indicator = a, bootType = "naive", seed = 123)

Value:
[1] 14.44422

Variance:
[1] 0.0920564

Confidence interval:
   lower    upper
13.87663 15.19417

Threshold:
[1] 10859.24
```

One of the most convenient features of package **laeken** is that indicators can be evaluated for different subdomains using a single command. This also holds for variance estimation. Using the `breakdown` argument, the example below produces variance and confidence interval estimates for each NUTS2 region in addition to the overall values.

```
R> b <- arpr("eqIncome", weights = "rb050", breakdown = "db040", data = eusilc)
R> variance("eqIncome", weights = "rb050", breakdown = "db040", design = "db040",
+     data = eusilc, indicator = b, bootType = "naive", seed = 123)
```

```
Value:
[1] 14.44422

Variance:
[1] 0.0920564

Confidence interval:
   lower    upper
13.87663 15.19417

Value by stratum:
        stratum    value
1    Burgenland 19.53984
2     Carinthia 13.08627
3 Lower Austria 13.84362
4      Salzburg 13.78734
5        Styria 14.37464
6         Tyrol 15.30819
7 Upper Austria 10.88977
8        Vienna 17.23468
9    Vorarlberg 16.53731

Variance by stratum:
        stratum       var
1    Burgenland 3.5105237
2     Carinthia 1.4133369
3 Lower Austria 0.4456053
4      Salzburg 1.2937926
5        Styria 0.4615967
6         Tyrol 1.0299617
7 Upper Austria 0.3785766
8        Vienna 0.6384621
9    Vorarlberg 1.7601223

Confidence interval by stratum:
        stratum     lower    upper
1    Burgenland 16.072806 23.63099
2     Carinthia 10.640776 15.23716
3 Lower Austria 12.196265 15.17182
4      Salzburg 11.913708 16.13315
5        Styria 13.020339 15.89730
6         Tyrol 13.084487 17.51124
7 Upper Austria  9.960467 12.54200
8        Vienna 15.712609 18.96003
9    Vorarlberg 13.604720 19.23431

Threshold:
[1] 10859.24
```

It should be noted that the workhorse function `bootVar()` is called internally by `variance()` for bootstrap variance and confidence interval estimation. The function `bootVar()` could also be called directly by the user in exactly the same manner. Moreover, variance and confidence interval estimation for any other indicator implemented in package **laeken** is straightforward—the application using function `variance()` or `bootVar()` remains the same.

4 Calibrated bootstrap

Rao and Wu (1988) showed that the naive bootstrap is biased when used in the complex survey context. They propose to increase the variance estimate in the h-th stratum by a factor of $\frac{n_h-1}{n_h}$ (if the bootstrap sample is of the same size). In addition, they describe extensions to sampling without replacement, unequal probability sampling, and two-stage cluster sampling with equal probabilities and without replacement.

Deville and Särndal (1992) and Deville et al. (1993) provide a general description on how to calibrate sample weights to account for known population totals. The naive bootstrap does not include the recalibration of bootstrap samples in order to fit known population totals and therefore is, strictly formulated, not suitable for many practical applications. However, even though a bias might be introduced, the naive bootstrap works well in many situations and is faster to compute than the calibrated version. Hence it is a popular method often used in practice.

In real-world data, the inclusion probabilities for observations in the population are in general not all equal, resulting in different *design weights* for the observations in the sample. Furthermore, the initial design weights are in practice often adjusted by calibration, e.g., to account for non-response or so that certain known population totals can be precisely estimated from the survey sample. To give a simplified example, if the population sizes in different regions are known, the sample weights may be calibrated so that the Horvitz-Thompson estimates (Horvitz and Thompson 1952) of the population sizes equal the known true values. However, when bootstrap samples are drawn from survey data, resampling observations has the effect that such known population totals can no longer be precisely estimated. As a remedy, the sample weights of each bootstrap sample should be calibrated.

The calibrated version of the bootstrap thus results in more precise variance and confidence interval estimation, but comes with higher computational costs than the naive approach. In any case, the *calibrated bootstrap algorithm* is obtained by adding the following step between Steps 1 and 2 of the naive bootstrap algorithm from Section 3:

1b. Calibrate the sample weights for each bootstrap sample $\boldsymbol{X}_r^*$, $r = 1, \dots, R$. Generalized raking procedures are thereby used for calibration: either a multiplicative method known as *raking*, an additive method or a logit method (see Deville and Särndal 1992, Deville et al. 1993).

The function call to `variance()` for the calibrated bootstrap is very similar to its counterpart for the naive bootstrap. A matrix of auxiliary calibration variables needs to be supplied via the argument `aux`. In addition, the argument `totals` can be used to supply the corresponding population totals. If the `totals` argument is omitted, as in the following example, the population totals are computed from the sample weights of the original sample. This follows the assumption that those weights are already calibrated on the supplied auxiliary variables.

```
R> variance("eqIncome", weights = "rb050", design = "db040",
+      data = eusilc, indicator = a, X = calibVars(eusilc$db040),
+      seed = 123)

Value:
[1] 14.44422

Variance:
[1] 0.09165169

Confidence interval:
   lower    upper
13.87817 15.19303

Threshold:
[1] 10859.24
```

Note that the function `calibVars()` transforms a factor into a matrix of binary variables, as required by the calibration function `calibWeights()`, which is called internally. While the default is to use raking for calibration, other methods can be specified via the `method` argument.

5 Conclusions

Both bootstrap procedures for variance and confidence interval estimation of indicators on social exclusion and poverty currently implemented in the R package **laeken** have their strengths. While the naive bootstrap is faster to compute, the calibrated bootstrap in general leads to more precise results. The implementation of other procedures such as linearization techniques (Kovačević and Binder 1997, Deville 1999, Hulliger and Münnich 2006, Osier 2009) or the delete-a-group jackknife (Kott 2001) is future work.

Furthermore, Alfons et al. (2009) demonstrated how the variance of indicators computed from data with imputed values may be underestimated in bootstrap procedures, depending on the indicator itself and the imputation procedure used. They proposed to use the method described in Little and Rubin (2002), which consists of drawing bootstrap samples from the original data with missing values, and to impute the missing data for each bootstrap sample before computing the corresponding bootstrap replicate estimate. Of course, this results in an additional increase of the computation time. The implementation of this procedure in package **laeken** is future work. It should also be noted that multiple imputation is a further possibility to consider the additional uncertainty from imputation when estimating the variance of an indicator (see Little and Rubin 2002).

Acknowledgments

This work was partly funded by the European Union (represented by the European Commission) within the 7th framework programme for research (Theme 8, Socio-Economic Sciences and Humanities, Project AMELI (Advanced Methodology for European Laeken Indicators), Grant Agreement No. 217322). Visit `http://ameli.surveystatistics.net` for more information on the project.

References

A. Alfons and S. Kraft. ***simPopulation**: Simulation of synthetic populations for surveys based on sample data*, 2010. URL `http://CRAN.R-project.org/package=simPopulation`. R package version 0.2.1.

A. Alfons, M. Templ, and P. Filzmoser. On the influence of imputation methods on Laeken indicators: simulations and recommendations. UNECE Worksession on Statistical Data Editing, Neuchâtel, Switzerland, 2009.

A. Alfons, S. Kraft, M. Templ, and P. Filzmoser. Simulation of close-to-reality population data for household surveys with application to EU-SILC. *Statistical Methods & Applications*, 20(3): 383–407, 2011a.

A. Alfons, M. Templ, P. Filzmoser, and J. Holzer. Robust pareto tail modeling for the estimation of social inclusion indicators using the R package **laeken**. Research Report CS-2011-2, Department of Statistics and Probability Theory, Vienna University of Technology, 2011b. URL `http://www.statistik.tuwien.ac.at/forschung/CS/CS-2011-2complete.pdf`.

A. Alfons, J. Holzer, and M. Templ. ***laeken**: Laeken indicators for measuring social cohesion*, 2012. URL `http://CRAN.R-project.org/package=laeken`. R package version 0.3.3.

A.C. Davison and D.V. Hinkley. *Bootstrap Methods and their Applications*. Cambridge University Press, 1997. ISBN 0 521 57471 4.

J.-C. Deville. Variance estimation for complex statistics and estimators: Linearization and residual techniques. *Survey Methodology*, 25(2):193–203, 1999.

J.-C. Deville and C.-E. Särndal. Calibration estimators in survey sampling. *Journal of the American Statistical Association*, 87(418):376–382, 1992.

J.-C. Deville, C.-E. Särndal, and O. Sautory. Generalized raking procedures in survey sampling. *Journal of the American Statistical Association*, 88(423):1013–1020, 1993.

B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, New York, 1993. ISBN 0-412-04231-2.

Eurostat. Common cross-sectional EU indicators based on EU-SILC; the gender pay gap. EU-SILC 131-rev/04, Unit D-2: Living conditions and social protection, Directorate D: Single Market, Employment and Social statistics, Eurostat, Luxembourg, 2004.

Eurostat. Algorithms to compute social inclusion indicators based on EU-SILC and adopted under the Open Method of Coordination (OMC). Doc. LC-ILC/39/09/EN-rev.1, Unit F-3: Living conditions and social protection, Directorate F: Social and information society statistics, Eurostat, Luxembourg, 2009.

D.G. Horvitz and D.J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, 47(260):663–685, 1952.

B. Hulliger and R. Münnich. Variance estimation for complex surveys in the presence of outliers. In *Proceedings of the Section on Survey Research Methods*, pages 3153–3156. American Statistical Association, 2006.

P.S. Kott. The delete-a-group jackknife. *Journal of Official Statistics*, 17(4):521–526, 2001.

M.S. Kovačević and D.A. Binder. Variance estimation for measures for income inequality and polarization – the estimating equations approach. *Journal of Official Statistics*, 13(1):41–58, 1997.

R.J.A. Little and D.B. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, New York, 2nd edition, 2002. ISBN 0-471-18386-5.

G. Osier. Variance estimation for complex indicators of poverty and inequality using linearization techniques. *Survey Research Methods*, 3(3):167–195, 2009.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL http://www.R-project.org. ISBN 3-900051-07-0.

J.N.K. Rao and C.F.J. Wu. Resampling inference with complex survey data. *Journal of the American Statistical Association*, 83(401):231–241, 1988.

M. Templ and A. Alfons. Standard methods for point estimation of social inclusion indicators using the R package **laeken**. Research Report CS-2011-1, Department of Statistics and Probability Theory, Vienna University of Technology, 2011. URL http://www.statistik.tuwien.ac.at/forschung/CS/CS-2011-1complete.pdf.