DACSEIS

IST-2000-26057

Workpackage 1

Variance Estimation in Complex Surveys

Deliverable 1.2

2004

List of contributors:

Kersten Magg, Ralf Münnich, University of Tübingen
Harm Jan Boonstra, Statistics Netherlands
Doris Eckmair, Andreas Quatember, Helga Wagner, University of Linz
Jean-Pierre Renfer, Ueli Oetliker, SFSO, and Sylvain Sardy, EPFL

Main responsibility:

Ralf Münnich, University of Tübingen

IST–2000–26057–DACSEIS

The DACSEIS research project is financially supported within the IST programme of the European Commission. Research activities take place in close collaboration with Eurostat.

`http://europa.eu.int/comm/eurostat/research/`

`http://www.cordis.lu/ist/`

`http://www.dacseis.de/`

Preface

Deliverable D1.2 aims to present parts of the simulation study going beyond the scope of the general workpackages. First, an overview to the simulation programmes of the comparative study is given. Chapter 2 contains specialised simulations on the Dutch LFS, the Austrian Microcensus, and the Swiss HBS. The Appendix contains selected programme codes.

Most simulation results from the studies in this deliverable are added to the recommended practice manual in deliverable D12.2. The recommended practice manual also contains an overview to all variables from the surveys used in the simulation study as well as a description of the simulation tasks.

The different parts of this deliverable are written by:

1	Kersten Magg, Ralf Münnich, University of Tübingen
2.1	Harm Jan Boonstra, Statistics Netherlands
2.2	Doris Eckmair, Andreas Quatember, Helga Wagner, University of Linz
2.3	Jean-Pierre Renfer, Ueli Oetliker, SFSO, and Sylvain Sardy, EPFL

Ralf Münnich University of Tübingen

Contents

List of Figures

List of Tables

Chapter 1

The DACSEIS Monte-Carlo Simulation Study

1.1 Overview to the Simulation Study

During the DACSEIS research project, several simulation studies had been implemented, performed, and evaluated. Those simulations which were applied in order to support the workpackage related research were included directly in the workpackages, e.g. for the workpackages 5, 6, 7, 8, 9, and 10. However, additional specialized simulations were performed going beyond the scope of several workpackages. Their results will be presented in the next chapter of this report.

The aim of the main simulation study is to compare the recommended methodology coming from several workpackages, i. e. the variance estimation methodology under non-response in a practical environment. The methods comprise

- Raking and calibration estimators (workpackage 8),
- Calibration for non-response (cf. Lundstrøm and Särndal, 2002),
- Resampling techniques (workpackage 5), and
- Non-response and imputation (workpackage 11).

The aim was to elaborate *best practice recommendations* for the usage of adequate variance estimation methods in accordance with an close-to-reality environment. The very heterogenous structure of the different surveys made it necessary to develop a very flexible simulation platform while respecting for the widespread methodology. This should enable the researcher to easily fit the tasks of interest into *control files* in order to quickly receive simulation results. The following section gives an overview to the main simulation programmes and the interface between the different items.

1.2 Structure of the Simulation Programmes

The simulation programmes are fully implemented in R allowing for easy code transformation into other languages. However, language specific characteristics can never be avoided and hardly be translated in either direction. Some specialised code that doesn't use standard definitions should be explained in comments.

The general hierarchy of the file structure is as follows

```
0:   ~/
1:   ~/FinalSim/
2:   ~/NSample/Survey/
3:   ~/SimData/
4:   ~/SimData/SurveySpec/
5:   ~/SimData/DataSets/
6:   ~/SimData/Universe/
7:   ~/SimData/Estimators/
8:   ~/Output/
```

The $\sim$ / refers to the main simulation directory which could be placed anywhere. The files for the simulation study are stored in the directories in the way as follows according to the enumeration above:

0: The main simulation file, e.g. `DACSEIS.simulation.main.R` (see programme A.1), will be stored in the main simulation directory;

1: all programmes and estimators will be stored here except the main simulation file;

2: the samples of the corresponding universe are stored in this directory as 1.stpr ... 10000.stpr in the subdirectory according to the survey (DLFS, FLFS, GMC, AMC, SHBS, and EVS). In general, 10,000 samples will be used for the simulation;

3: this directory includes all information concerning specifications as well as all auxiliary variables (cf. items 4 to 7);

4: this directory concludes the survey specification as described in the metadata in deliverable D1.1. All specifications including estimation variables, auxiliary variables, non-response, etc. (cf. programme A.3). Further, the `.master` files can be found here allowing the automatisation of the generation of the simulation data sets;

5: this directory contains the simulation control files that specify the tasks to be simulated;

6: the matrices of true values are stored in this directory. However, this highly depends on the stratification used. The names should be according to the sample-directory under `~/NSample/` plus stratification abbreviation. If small area estimates are to be considered, the corresponding true values for the small areas should be saved in the same way;

7: this directory contains the simulation control files that specify the considered estimators (including further information such as calibration, imputation, etc.);

8: in this directory, the simulation output will be stored where each concrete simulation should be stored in a separate directory. Special attention has to be paid to the name convention between simulation tasks and output directories to guarantee a uniform structure.

The main path for the simulation root directory and the corresponding subdirectory is stored in a separate path file which will be read first in the simulation main programme. For different computers and operating systems, different path statements should be written (cf. programme A.7).

The general structure of the simulation programmes including their interdependencies is shown in Figure 1.1. The chosen structure should enable the user to clearly distinguish the structure of the different tasks and interfaces from each other while allowing for easy automatisation of the simulation process. The core of the simulation is defined in `MasterSim.R`, which calls all estimators via generalised interfaces. This programme is controlled by the main simulation programme. All other files contain standardised information on the simulation tasks, e.g. by specifying the variables of interest and the corresponding methodology.

This structure will allow for an easy update of the simulation programme with further estimators, other survey specifications and new simulation tasks while respecting the general flow of the simulation programmes. Further descriptions can be drawn from the following chart:

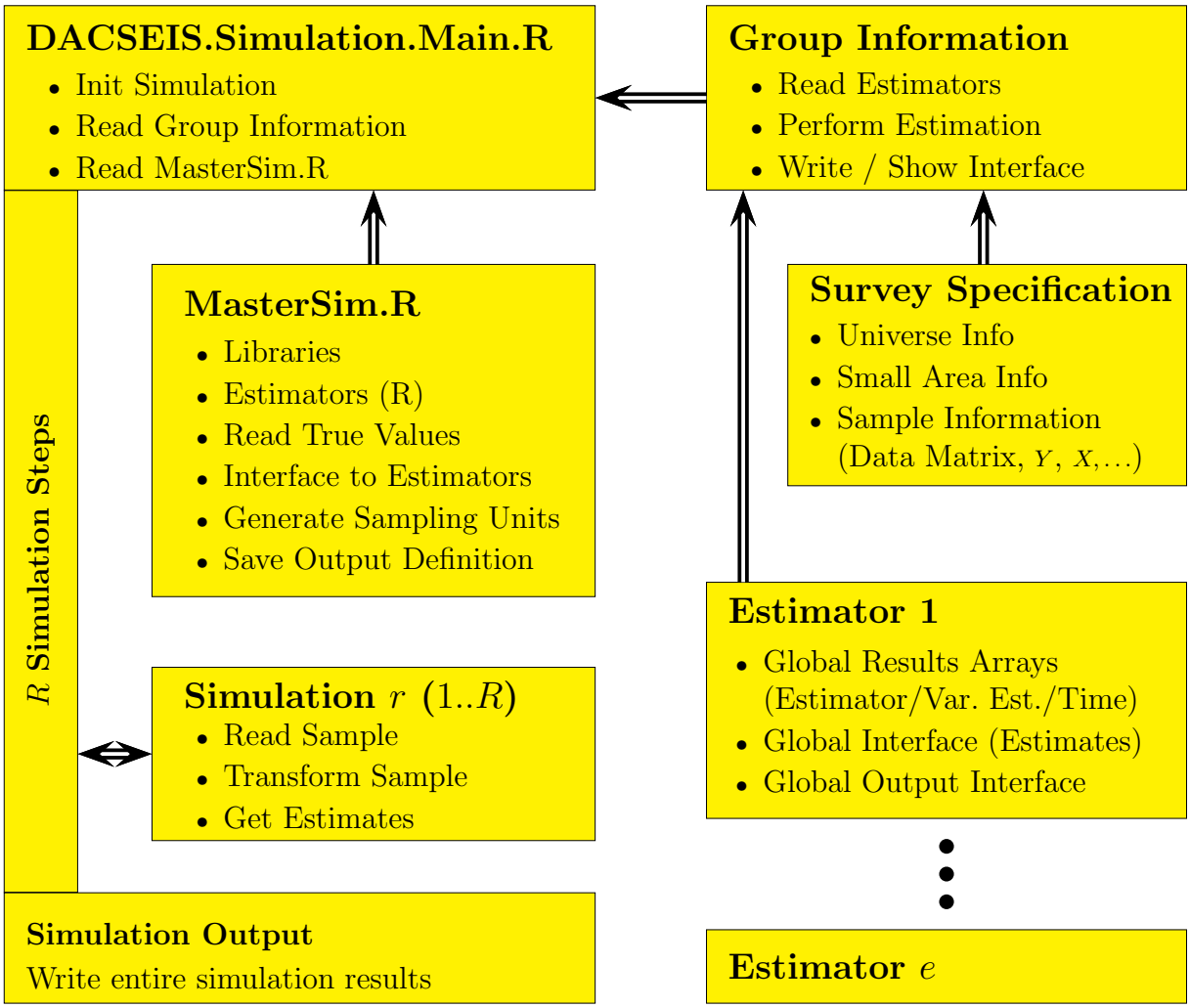

Figure 1.1: Structure of simulation programmes

DACSEIS.Simulation.Main.R The main simulation control programme (cf. listing A.1) is reduced to minimal requirements. First, the path's information are read containing all necessary paths for the simulation platform (cf. listing A.7). Next, the simulation task has to be read which is defined in the **Group Information** file as well as the **MasterSim.R** (descriptions see below). The simulation itself is a primitive loop used to subsequently read all samples. In a second step estimates are gained from the samples. Finally, the results are written. All three steps use a general interface which is defined in one of the general files.

MasterSim.R This file contains all relevant information and interfaces for the simulation study. First, necessary libraries (e.g. the library **boot** for bootstrap methods) and the source files for estimators and other methods used during the simulation, such as imputation and calibration methods, are read. Next, the routines for reading universe or small area information like *true values* or values for auxiliary variables are provided. A major part of this file is the definition of the interfaces to the estimators. The interface is designed such that within the simulation the general call of an estimator is

```
value = Estimator(data, parms)
```

with the matrix `data` and the parameter vector `parms`. Both will be generated during the simulation when transforming the full sample in this specialised format according to the definition in **Survey Specification**. Since the sample generally consists of individual data and sometimes data need to be aggregated, e.g. in households or in clusters for regression estimation, specialised routines for building these sampling units are added. The grouping must be defined in **Survey Specification**. Finally, the output routines to save the results are defined in **MasterSim.R** as well as some further functions, e.g. for special evaluations on the sample data sets. An example file can be drawn from listing A.2.

Group Information The group information file is designed to allow for simultaneous running estimates on the corresponding data. Within this file, the specialised estimators on concrete simulation tasks are loaded. Further, general interfaces to the estimators for a given survey specification are given. The name of the general simulation task, which is defined in the main simulation file must be consistent with the survey specification, the concrete estimators, and the grouping file (cf. listing A.4 and others).

Survey Specification The survey specification file is used to define all characteristics of the data in the simulation. This includes the interfaces to the universe data which may be available for the simulation as auxiliary data, e.g. population totals for regression estimation or calibration, and the *true values* for evaluation purposes. The same interface will be used for small area estimation. The main purpose, however, is the generation of the simulation matrix from the sample input file. To achieve this, the variables of interest for the simulation will be stored columnwise in a globally available matrix. Parameter information will indicate in which columns the information is stored. The main parameter indicators are `parms$Y` for the estimation variable, `parms$X` for the auxiliary variable, `parms$Ind` for the stratum indicator, and `parms$NR` for the non-response indicator variable. The global arrays will ensure efficient use of the data matrix. However, in use for estimators, one should generally try to avoid changing values within this matrix since in runtime the entire data matrix will be copied in functions under R (R uses references to global variables if data remain unchanged). The global variables for a specific simulation task are also defined in this file. An example file can be drawn from listing A.3.

Estimator The estimator information file enables the user to apply a general interface to the specific estimators that will be used in **MasterSim.R**. It contains on the one hand an interface to the estimator including time measurement, on the other hand a routine to print the results. The estimated values are stored in variables globally available respecting the number of simulation runs R that are defined in the main simulation file. The listings A.5 and A.6 give two examples.

There is one major drawback. One has to carefully define simulation tasks with adequate grouping structures in order to ensure consistent naming. To make this easier, a general *PERL* routine was written which automatically generates all information files on the right hand side of Figure 1.1 from master files and creates all directories for output files needed. The names of the output files must be consistent with the grouping files that all simulation tasks can be easily identified.

1.3 Contents of the Simulation Study

The DACSEIS simulation study was carried out by five different participating institutions. Essentially the project dealt with point and variance estimation, classifying and evaluating different variance estimation methods based on close-to-reality data-sets. For this purpose several types of target and auxiliary variables were applied. In most cases the target variable *unemployed* was implemented in the simulation study. Additionally variables like *employed*, *income*, *status of the education*, *number of peoples in special age classes*, etc. were used as target or auxiliary variables. With respect to the data set the simulation study was based on six different kinds of universes (cf. table 1.1). Alternative non-response mechanisms and rates were applied.

Kind of Survey	Country	Universe
Labour Force Surveys	Netherlands	Dutch Labour Force Survey
	Finland	Finnish Labour Force Survey
Microcensus Surveys	Germany	German Microcensus
	Austria	Austrian Microcensus
Household Budget Surveys	Germany	German Sample Survey of Income and Expenditure
	Switzerland	Swiss Household Budget Surveys

Table 1.1: Overview to the six pseudo universes used in the simulation study

Detailed description of the simulations performed are included in the deliverable reports from workpackage WP5 – WP10, and for the imputation coding in WP11. Further specialised simulations are presented in the next chapter. A general overview to the simulation study and its outcomes which are presented in the DACSEIS recommended practice manual can be drawn from deliverable D12.3.

Chapter 2

Specialised Simulations

2.1 Repeated Weighting in the Dutch LFS

In workpackage 7, a strategy for estimation of a set of multi-dimensional tables from several data sources is described. The estimation strategy, called repeated weighting, is such that all resulting estimated tables in the set are mutually consistent concerning common marginal cells. Deliverable D7.2 contains a description of this strategy and derives the repeated weighting (RW) estimator as well as estimators of its sampling variance. Deliverable D7.3 reports about a simulation study carried out to investigate the repeated sampling behaviour of the RW estimators and variance estimators. Here we summarize the results of this simulation study and report about additional simulations.

The simulations described in D7.3 are based on simple random samples drawn from part of the province Noord-Brabant data from the Dutch LFS pseudo-universe. The additional simulations are carried out on samples drawn from the Amsterdam subpopulation of the Dutch LFS pseudo-universe.

2.1.1 The Simulation Setup

The population is based on the Dutch pseudo-universe consisting of the simulated target population for the Dutch LFS. The following person characteristics are used in the simulation studies:

1. MUN: municipality
2. SEX: sex in 2 categories
3. AGE: age in 6 categories
4. MST: marital status in 3 categories
5. $ETHN$: ethnicity in 3 categories
6. $EMPL$: employment in 3 categories

Two subpopulations are considered: the province of Noord-Brabant and Amsterdam. From these subpopulations, samples are generated and the repeated weighting estimation procedure is applied to two different target tables for each sample. Also, simple expansion estimators and general regression estimators as well as corresponding variance estimators are computed for comparison.

The two target tables used for the simulations are $SEX \times AGE \times ETHN$ and $SEX \times MST \times EMPL$. For the first table, $ETHN$ is considered the target variable, i.e. the variable only known from the sample, and for the second table $EMPL$ is the target variable. The other variables are supposed to be in the register and are used as auxiliary information in the estimation process.

2.1.2 Simulations Using the Amsterdam Subpopulation

The simulations on Amsterdam are carried out using the complete pseudo-universe for Amsterdam. The samples are roughly 1% samples drawn according to a sampling design that approximates the design used for the Dutch LFS, see the combined deliverables D3.1 and D3.2, Section 3.1.3. Point estimators are computed for 10,000 simulation runs, and variance estimates are computed for 1,000 of these 10,000 runs.

1. *Target table* $SEX \times AGE \times ETHN$

The first target table simulated is the table of counts $SEX \times AGE \times ETHN$ (abbreviated as SAE), in which $ETHN$ is considered as the variable observed only in the sample, and the other variables are registered for the whole population. The overall weighting scheme used here is $SEX \times MST + AGE$.

The simple expansion estimator of the vector of population counts t_{SAE} is given by

$$\widehat{t}^{d}_{SAE} = \sum_{i \in S} d_i y_i \ = N \overline{y}_S \,, \tag{2.1}$$

where y is a vector of dummy variables representing the target table's cells, d_i are design weights for simple random sampling equal to N/n, and $\overline{y}_S$ is the sample mean of y.

The expansion estimator serves as starting point for the general regression estimator which uses the overall weighting scheme to improve on it. The general regression estimator can be written as a weighted sum

$$\widehat{t}^{w}_{SAE} = \sum_{i \in S} w_i y_i \tag{2.2}$$

with regression weights w_i calibrated to the known population totals of the overall weighting scheme.

The general regression estimator on its turn is the starting point for the repeated weighting (RW) estimator given by

$$\widehat{t}^{RW}_{SAE} = \widehat{t}^{w}_{SAE} + (\widehat{B}^{w}_{SAE;SA+SE+AE})^t \begin{pmatrix} t_{SA} - \widehat{t}^{w}_{SA} \\ 0 \\ 0 \end{pmatrix} , \tag{2.3}$$

where $\widehat{t}^w_{SA}$ is the $SEX \times AGE$ margin of $\widehat{t}^w_{SAE}$, and t_{SA} represents the register totals of $SEX \times AGE$. The zeroes between the brackets in (2.3) are due to the fact that the corresponding margins for $SEX \times ETHN$ and $AGE \times ETHN$ estimated using the overall weighting scheme weights are already consistent with the register. For details we refer to deliverables D7.2 and D7.3.

An alternative repeated weighting estimator using only a minimal weighting scheme, i.e. the minimal re-weighting scheme such that consistency with the register is obtained, is

$$\widehat{t}^{MW}_{SAE} = \widehat{t}^w_{SAE} + (\widehat{B}^w_{SAE;SA})^t (t_{SA} - \widehat{t}^w_{SA})\,. \tag{2.4}$$

This estimator only calibrates $\widehat{t}^w_{SAE}$ to $SEX \times AGE$ register totals and unlike (2.3) does not preserve the $SEX \times ETHN$ and $AGE \times ETHN$ margins of $\widehat{t}^w_{SAE}$.

Approximate sampling variances can be computed from the 10,000 point estimates. They are used to evaluate the performance of the following variance estimators. Variance estimates for the expansion estimator $\widehat{t}^d_{SAE}$ are given by the diagonal elements of the estimated covariance matrix

$$v^{(d)} = \frac{N(N-n)}{n}\, s^2_y\,, \tag{2.5}$$

where $s^2_y \equiv \frac{1}{n-1}\sum_{i\in S}(y_i - \overline{y})(y_i - \overline{y})^t$ denotes the sample covariance matrix of y.

For the regression estimator, the usual linearisation variance estimates with and without g-weights are computed. They are given by the diagonal elements of

$$v^{(w)} = \frac{N(N-n)}{n}\, s^2_e\,, \tag{2.6}$$

and

$$v^{(w)}_g = \frac{N(N-n)}{n}\, s^2_{ge}\,, \tag{2.7}$$

where s^2_e is the sample covariance matrix of the residuals e_k of the target vector-variable y with respect to the overall weighting scheme and s^2_{ge} is the sample covariance matrix of the products $g_k e_k$, where g_k are the g-weights corresponding to the overall weighting scheme.

For the repeated weighting estimator $\widehat{t}^{RW}_{SAE}$, the linearisation variance estimates are the diagonal elements of

$$v^{RW} = \frac{N(N-n)}{n}\, s^2_{e^{RW}}\,, \tag{2.8}$$

where

$$e^{RW}_k = e_k + (\widehat{B}^w_{SAE;SA+SE+AE})^t \begin{pmatrix} -e_{SA;k} \\ 0 \\ 0 \end{pmatrix}\,, \tag{2.9}$$

with $e_{SA;k}$ the $SEX \times AGE$ margins of e_k. For $\widehat{t}^{MW}_{SAE}$ no specific variance estimates are computed. The variance estimates are computed in 1,000 of the 10,000 simulation runs.

Performance of the point estimators

For ten cells of the table we list the results (in terms of relative root mean squared errors (RRMSE)) in Table 2.1 below. For more complete simulation results we refer to the

Table 2.1: RRMSE (in %) over 10,000 simulation runs for selected cells of table $SEX \times AGE \times ETHN$. The second column contains the population totals.

Cell name	t_{SAE}	$\widehat{t}^{d}_{SAE}$	$\widehat{t}^{w}_{SAE}$	$\widehat{t}^{RW}_{SAE}$	$\widehat{t}^{MW}_{SAE}$
SEXAGE1ETHN1	4,412	14.1	13.8	13.6	13.6
SEXAGE2ETHN1	1,546	23.9	23.6	23.5	23.5
SEXAGE3ETHN1	955	30.2	30.0	29.9	29.9
SEXAGE4ETHN1	4,127	14.6	14.2	14.1	14.1
SEXAGE5ETHN1	4,382	14.0	13.7	13.7	13.7
SEXAGE1ETHN2	14,196	7.7	7.1	6.7	6.7
SEXAGE2ETHN2	12,707	8.1	7.4	7.0	7.0
SEXAGE3ETHN2	12,762	8.1	7.4	7.0	7.0
SEXAGE4ETHN2	10,002	9.2	8.7	8.5	8.5
SEXAGE5ETHN2	8,018	10.5	10.1	10.1	10.1

electronic RPM. More information about the naming convention for cells of the frequency table and about the definition of the evaluation criteria used can be found in deliverable D7.3.

The main finding for the point estimators is that the differences in accuracy between the various point estimators are small. In particular, the differences between the two repeated weighting estimators $\widehat{t}^{RW}_{SAE}$ and $\widehat{t}^{MW}_{SAE}$ are negligible. The repeated weighting estimators perform slightly better in terms of MSE than the regression estimator which on its turn performs slightly better than the simple expansion estimator. The most important difference between the estimators concerns their calibration to, and therefore consistency with, marginal totals from the register, which cannot be seen from Table 2.1. Whereas the expansion estimator is not calibrated to register totals, the regression estimator is calibrated to SEX and AGE totals due to the presence of these variables in the overall weighting scheme, and the repeated weighting estimators are calibrated in addition to the interaction in $SEX \times AGE$ due to the re-weighting.

Performance of the variance estimators

Table 2.2 shows relative root mean squared errors of the variance estimators described above for the same selection of cells of the target table.

The RRMSE of the repeated weighting variance estimator seems somewhat smaller than the regression variance estimators. This is especially clear for most cells involving $ETHN2$, which have larger population and sample size than the $ETHN1$ cells. Note that the orders of magnitude of the RRMSE of the variance estimators are the same as those of the corresponding point estimators.

Coverage rates have also been computed but are not displayed. Coverage rates of the estimator/variance estimator combinations considered are all close to the nominal 95% level with only slightly reduced coverage for the smaller cells. There are no noticeable differences in coverage between the estimator/variance estimator combinations.

Table 2.2: RRMSE (in %) of variance estimates over 1,000 simulation runs.

Cell name	$v^{(d)}(\widehat{t}^{d}_{SAE})$	$v^{(w)}(\widehat{t}^{w}_{SAE})$	$v^{(w)}_{g}(\widehat{t}^{w}_{SAE})$	$v^{RW}(\widehat{t}^{RW}_{SAE})$
SEXAGE1ETHN1	13.8	13.2	13.3	12.9
SEXAGE2ETHN1	23.6	23.3	23.2	22.8
SEXAGE3ETHN1	30.9	30.5	30.9	30.1
SEXAGE4ETHN1	14.1	13.4	13.4	13.1
SEXAGE5ETHN1	14.1	14.0	13.8	13.9
SEXAGE1ETHN2	8.1	7.7	7.8	6.0
SEXAGE2ETHN2	8.8	9.1	9.2	5.9
SEXAGE3ETHN2	8.5	7.9	8.0	6.0
SEXAGE4ETHN2	9.0	8.5	8.9	7.4
SEXAGE5ETHN2	10.4	9.8	9.9	9.6

2. *Target table $SEX \times MST \times EMPL$*

The second simulation is carried out on target table $SEX \times MST \times EMPL$ (abbreviated by SME), where $EMPL$ is considered as the variable observed only in the sample, and the other variables are register variables. The overall weighting scheme is now taken as $SEX \times AGE + ETHN$.

The repeated weighting estimator of the totals t_{SME} is

$$\widehat{t}^{RW}_{SME} = \widehat{t}^{w}_{SME} + (\widehat{B}^{w}_{SME;SM+SE+ME})^{t} \begin{pmatrix} t_{SM} - \widehat{t}^{w}_{SM} \\ 0 \\ \widehat{t}^{RW}_{ME} - \widehat{t}^{w}_{ME} \end{pmatrix}, \tag{2.10}$$

where $\widehat{t}^{w}_{SME}$ is the regression estimator of t_{SME} based on the overall weighting scheme, t_{SM} are the register totals of $SEX \times MST$, and

$$\widehat{t}^{RW}_{ME} = \widehat{t}^{w}_{ME} + (\widehat{B}^{w}_{ME;M+E})^{t} \begin{pmatrix} t_{M} - \widehat{t}^{w}_{M} \\ 0 \end{pmatrix}. \tag{2.11}$$

As can be seen from (2.10) and (2.11), this example involves a recursive use of the repeated weighting estimator, i.e. one of the margins is calibrated on totals estimated themselves by a repeated weighting estimator.

In Section 6 of D7.2 a non-recursive approximation to the RW estimator was proposed. In general, this approximation does not necessarily satisfy all consistency requirements. However, its main function is to provide a simpler variance estimator for the RW estimator, as we shall see below. Here the approximation of the RW estimator takes the form

$$\widehat{t}^{SRW}_{SME} = \widehat{t}^{w}_{SME} + (\widehat{B}^{w}_{SME;SM+SE})^{t} \begin{pmatrix} t_{SM} - \widehat{t}^{w}_{SM} \\ 0 \end{pmatrix}, \tag{2.12}$$

the re-weighting being only with respect to the margins $SEX \times MST$ and $SEX \times EMPL$, i.e. those margins that do not need re-weighting themselves. We shall refer to this estimator as the simplified RW (SRW) estimator.

Finally, the minimal weighting scheme for calibrating $\widehat{t}^{w}_{SME}$ to the register is $SEX \times MST$. The corresponding re-weighted estimates are given by

$$\widehat{t}^{MW}_{SME} = \widehat{t}^{w}_{SME} + (\widehat{B}^{w}_{SME;SM})^t (t_{SM} - \widehat{t}^{w}_{SM}) \,. \tag{2.13}$$

The three repeated weighting estimators described above as well as the HT estimator and regression estimator based on the overall weighting scheme are again computed for 10,000 samples drawn according to the same sampling scheme.

Five different variance estimators have been computed for a subset of 1,000 simulation runs, one for the HT estimator, two for the regression estimator $\widehat{t}^{w}_{SME}$ and two for the repeated weighting estimator $\widehat{t}^{RW}_{SME}$. For the HT and regression estimators the variance estimates computed are of the same form as given in (2.5), (2.6) and (2.7).

For the repeated weighting estimator $\widehat{t}^{RW}_{SME}$, the linearisation variance estimates are the diagonal elements of

$$v^{RW} = \frac{N(N-n)}{n} \, s^2_{e^{RW}} \,, \tag{2.14}$$

where

$$e^{RW}_k = e_k + (\widehat{B}^{w}_{SME;SM+SE+ME})^t \begin{pmatrix} -e_{SM;k} \\ 0 \\ e^{RW}_{ME;k} - e_{ME;k} \end{pmatrix} \,, \tag{2.15}$$

$e_{SM;k}$ are the $SEX \times MST$ margins of the overall weighting scheme regression residuals e_k, and

$$e^{RW}_{ME;k} = e_{ME;k} + (\widehat{B}^{w}_{ME;M+E})^t \begin{pmatrix} -e_{M;k} \\ 0 \end{pmatrix} \,. \tag{2.16}$$

A simplified variance estimator can be based on the estimator $\widehat{t}^{SRW}_{SME}$. The linearisation variance estimates for the latter are the diagonal elements of

$$v^{SRW} = \frac{N(N-n)}{n} \, s^2_{e^{SRW}} \,, \tag{2.17}$$

where

$$e^{SRW}_k = e_k + (\widehat{B}^{w}_{SME;SM+SE})^t \begin{pmatrix} -e_{SM;k} \\ 0 \end{pmatrix} \,. \tag{2.18}$$

We consider v^{SRW} as a variance estimator for the original repeated weighting estimator $\widehat{t}^{RW}_{SME}$. Variance estimation for $\widehat{t}^{SRW}_{SME}$ and $\widehat{t}^{MW}_{SME}$ is not simulated.

Performance of the point estimators

For a selection of four cells of the target table we list the results (in terms of relative root mean squared errors) in Table 2.3.

From Table 2.3 it is clear that the mean squared errors of all three repeated weighting estimators $\widehat{t}^{RW}_{SME}$, $\widehat{t}^{SRW}_{SME}$ and $\widehat{t}^{MW}_{SME}$ are approximately the same. Cell $SEXMST1EMPL2$ is clearly improved by repeated weighting, whereas $SEXMST2EMPL2$ is slightly improved. Smaller cells, involving $EMPL1$, do not benefit noticeably from the repeated weighting. The use of auxiliary information is more effective for the larger cells.

Table 2.3: RRMSE (in %) over 10,000 runs for table $SEX \times MST \times EMPL$. The second column contains the population totals.

Cell name	t_{SME}	$\widehat{t}^{d}_{SME}$	$\widehat{t}^{w}_{SME}$	$\widehat{t}^{RW}_{SME}$	$\widehat{t}^{SRW}_{SME}$	$\widehat{t}^{MW}_{SME}$
SEXMST1EMPL1	1,458	24.4	24.2	24.1	24.0	24.0
SEXMST2EMPL1	5,705	12.4	12.2	12.2	12.2	12.2
SEXMST1EMPL2	83,990	3.1	2.4	1.1	1.1	1.1
SEXMST2EMPL2	50,841	4.0	3.6	3.0	3.0	3.0

Performance of the variance estimators

Table 2.4 shows relative root mean squared errors of the variance estimators described above for the same selection of four cells of the target table.

For cell $SEXMST2EMPL2$ the repeated weighting variance estimators have clearly smaller MSE than do the other variance estimators. Even for the smallest cell $SEXMST1EMPL1$ the repeated weighting variance estimators perform similarly to or slightly better (in MSE sense) than the regression variance estimators. It might then come as a surprise that for the much larger cell $SEXMST1EMPL2$, repeated weighting variance estimators have much larger *relative* RMSE than the regression variance estimators. However, the surprise will not be so great after noticing from Table 2.3 that the repeated weighting point estimators for this cell perform much better than the regression estimators. Therefore, the decrease in accuracy is only relative; the absolute mean squared errors of regression and repeated weighting variance estimators are apparently not very different. The differences between the full linearisation variance estimator v^{RW} and its simplified version v^{SRW} are small, except that v^{SRW} seems to perform somewhat worse for cell $SEXMST1EMPL2$. Regarding also the cells not listed here and the results of deliverable D7.3 we can cautiously conclude that v^{SRW} is a reasonable simple alternative for v^{RW}.

Table 2.4: RRMSE (in %) of variance estimates over 1,000 simulation runs.

Cell name	$v^{(d)}(\widehat{t}^{d}_{SME})$	$v^{(w)}(\widehat{t}^{w}_{SME})$	$v^{(w)}_{g}(\widehat{t}^{w}_{SME})$	$v^{RW}(\widehat{t}^{RW}_{SME})$	$v^{SRW}(\widehat{t}^{RW}_{SME})$
SEXMST1EMPL1	24.6	24.1	24.0	23.5	23.6
SEXMST2EMPL1	12.4	12.1	12.1	12.0	12.0
SEXMST1EMPL2	6.2	3.0	2.7	6.3	9.5
SEXMST2EMPL2	3.7	4.0	3.9	2.7	3.3

Coverage rates of the estimator/variance estimator combinations considered are found to be close to the nominal 95% level with only slightly reduced coverage for the smaller cells. There are no noticeable differences in coverage between the estimator/variance estimator combinations.

Computational issues

The simulations have been carried out in S-Plus 2000 under Windows 2000 on a PC with 1.5GHz Pentium 4 processor and 256MB internal memory. To give a rough indication of

computation times, the total time taken for all simulations on target table $SEX \times MST \times EMPL$ was about 11.5 hours, of which 6 hours were spent reading sample data. So the computations of all 5×10,000 point estimates and all 5×1,000 variance estimates took about 5.5 hours. Since the computations of different estimators and variance estimators involved many common subcalculations we have not attempted to measure the differences in computation times between the various estimators. However, it is clear that none of the studied estimators/variance estimators take unreasonable amounts of time to compute, and differences in ease of implementation seem more relevant.

2.1.3 Non-response

The same simulations as described above for the Amsterdam subpopulation were performed another time with the addition of a non-response mechanism according to an artificial response model. The model was chosen such that the total non-response fraction was 40% with different response rates for different classes of the variable $EMPL$. In particular, for the response rate for $EMPL1$, the unemployed, a very high non-response rate of 80% was chosen. This corresponds to a NMAR or non-ignorable non-response mechanism. Of course, such a worst-case model has great impact on the estimates of $EMPL$ with large negative biases for the number of unemployed and consequently much reduced coverage rates. The effects on target table $SEX \times AGE \times ETHN$ are rather small with mainly a loss of accuracy corresponding to the reduced sample size, but no large biases in the point estimators. The point of interest here is, however, whether non-response affects the various estimators differently. This is not the case in the present simulation study. Repeated weighting estimators suffer as much from non-response as regression and expansion estimators. In general, of course, this depends on the appropriateness of the auxiliary information used in the regression and repeated weighting estimators, i.e., on the correlation structure between the auxiliaries, the target variable and the non-response.

2.1.4 Simulations on Data from the Province Noord-Brabant

For deliverable D7.3 simulations have been carried out on a subset of the pseudo-universe subpopulation corresponding to the province Noord-Brabant. For the same two target tables, two simulation studies have been performed, with 15,000 simulation runs each, the first with sample size $n = 500$ and the second with sample size $n = 5{,}000$. Approximate sampling variances are computed from the 15,000 point estimates. Variance estimates are computed in 3,000 of the 15,000 simulation runs.

Details about this simulation study can be found in deliverable D7.3. The main conclusions, which are very similar to those described above for the Amsterdam simulations, are as follows. As a general observation, differences between several variants of repeated weighting estimators (splitting-up, minimal, simplified) are small. In almost all cases the repeated weighting estimators perform as well as or (slightly) better than the regression estimator based on the overall weighting scheme, even for as small a sample size as $n = 500$. The main advantage of repeated weighting estimators is, however, their numerical consistency, in this case numerical consistency with all register counts.

We found no signs for any stability problems of repeated weighting point and variance estimators, which one might have expected on the grounds that repeated weighting estimators involve more estimated regression coefficients as well as estimated calibration totals. Finally, the simplified variance estimator computed in the simulation of table $SEX \times MST \times EMPL$ seems a reasonable and simple alternative for the full linearisation variance estimator of the repeated weighting estimator.

2.2 Special Features of the Austrian Microcensus

2.2.1 Simulation Studies

The sampling procedure and the estimation were programmed in C++.

For the Austrian Microcensus (=AMC)

- the estimation for the total of some interesting variables
- the comparison of various direct variance estimates
- the estimation of a ratio of two totals and of its direct variance estimate

as well as the comparison of the use of three different universes was done in C++. Later for a better comparability the complex variance estimator was programmed in R.

The results of this additional work are presented here:

The simulation was conducted by $nsim = 10,000$ repeated drawings according to the fixed sampling plan of the AMC. For the simulation study the essential notations of the pseudo-universe are

l	part 1 or 2 of the dwelling stock (for details see: Report of WP2)
b	federal state
h	stratum
i	Primary Sampling Unit (=PSU) in part 2
W_{lbh}	number of dwellings of the pseudo-universe in lbh
i_W, i_M, i_N	dwelling number, household number, personal number
i_{MW}	household number within a dwelling
x_{1j} x_{2j}	personal characteristics age and gender
x_{3j} x_{4j} x_{5j}	personal characteristics education, labor force concept and nationality.

2.2.2 Weighting Procedure

The weighting in the AMC is carried out per federal state, part, stratum and PSU (see: Report of WP1).

Weighting Procedure for Part 1 of the Universe

For each federal state b and stratum h the weight is given by

$$\frac{W_{1bh}}{w_{1bh}}$$

for empty dwellings and by

$$\frac{W_{1bh}}{w_{1bh}} \cdot \frac{w^{(1)}_{1bh} + w^{(2)}_{1bh}}{w^{(1)}_{1bh}}$$

for dwellings and its occupants, where

$1bh$	Part 1, federal state b and stratum h; short: stratum $1bh$
W_{1bh}	number of dwellings in stratum $1bh$ of the universe
w_{1bh}	number of dwellings in stratum $1bh$ of the sample
$w^{(1)}_{1bh}$	number of interviewed dwellings in stratum $1bh$ of the sample
$w^{(2)}_{1bh}$	number of empty dwellings in stratum $1bh$ of the sample, with the categories 'no one at home' or 'refusal'
$w^{(3)}_{1bh}$	number of empty dwellings in stratum $1bh$ of the sample, with the categories 'dwelling demolished', 'no longer used as dwelling', 'seasonal housing unit' or 'vacant'

with

$$w_{1bh} = w^{(1)}_{1bh} + w^{(2)}_{1bh} + w^{(3)}_{1bh}. \tag{2.19}$$

Weighting Procedure for Part 2 of the Universe

In Part 2 for each federal state b, stratum h and PSU i the weighting is given by

$$\frac{\sum_{i=1}^{C_{2bh}} W_{2bhi}}{\sum_{i=1}^{c_{2bh}} W_{2bhi}} \cdot \frac{W_{2bhi}}{w_{2bhi}}$$

for empty dwellings and by

$$\frac{\sum_{i=1}^{C_{2bh}} W_{2bhi}}{\sum_{i=1}^{c_{2bh}} W_{2bhi}} \cdot \frac{W_{2bhi}}{w_{2bhi}} \cdot \frac{w^{(1)}_{2bhi} + w^{(2)}_{2bhi}}{w^{(1)}_{2bhi}}$$

for dwellings and its occupants with analog notation, where $2bhi$ means Part 2, federal state b, stratum h and PSU i and

C_{2bh}	number of PSUs in stratum $2bh$ of the universe
c_{2bh}	number of PUSs in stratum $2bh$ of the sample

with

$$w_{2bhi} = w^{(1)}_{2bhi} + w^{(2)}_{2bhi} + w^{(3)}_{2bhi}. \tag{2.20}$$

2.2.3 The Estimation of a Total

The Horvitz-Thompson-Estimator of a total τ (e.g. the number of employees) of the universe U

$$\tau = \sum_U y_k$$

is given by

$$\widehat{\tau}_\pi = \sum_S \frac{y_k}{\pi_k},$$

where S means the drawn sample, y_k the kth element of the interesting variable y and π_k its inclusion probability.

Using the *sample membership indicator* (Särndal *et al.*, 1992, p. 30) of the k-th element with

$$I_k = \begin{cases} 1 & \dots \text{if} \quad k \in S \\ 0 & \dots \text{if} \quad \text{not} \end{cases}$$

the estimator can be expressed as

$$\widehat{\tau}_\pi = \sum_U I_k \frac{y_k}{\pi_k} \qquad .$$

For the reciprocal h_k of the inclusion probability π_k the estimator gives

$$\widehat{\tau}_\pi = \sum_S y_k h_k \tag{2.21}$$

Because sampling in the strata and PSUs is independent the Horvitz-Thompson-Estimator for the two parts can be calculated as

$$\widehat{\tau}_\pi \quad = \quad \sum_s y_k h_k = \sum_{b=1}^{B_1} \sum_{h=1}^{H_{1b}} y_{1bh} h_{1bh} + \sum_{b=1}^{B_2} \sum_{h=1}^{H_{2b}} \sum_{i=1}^{C_{2bh}} y_{2bhi} h_{2bhi}. \tag{2.22}$$

where B_1 respectively B_2 are the numbers of federal states in parts 1 and 2 of the dwellings stock and H_{1b} respectively H_{2b} are the numbers of strata in federal state b of parts 1 and 2.

Part 1 of the Universe

Now the weights h_k can be inserted in (2.22). Hence, the Horvitz-Thompson-Estimator for part 1 takes the form

$$\widehat{\tau}_{\pi 1} \quad = \quad \sum_{b=1}^{B_1} \sum_{h=1}^{H_{1b}} \Big[\frac{W_{1bh}}{w_{1bh}} \cdot \sum_{i=1}^{w^{(3)}_{1bh}} y_{1bhi} + \frac{W_{1bh}}{w_{1bh}} \cdot \frac{w^{(1)}_{1bh} + w^{(2)}_{1bh}}{w^{(1)}_{1bh}} \cdot \sum_{j=1}^{w^{(1)}_{1bh}} y_{1bhj} \Big], \tag{2.23}$$

where

$\sum_{i=1}^{w^{(3)}_{1bh}} y_{1bhi}$	the sum of variable y within the empty dwellings of 1bh
$\sum_{j=1}^{w^{(1)}_{1bh}} y_{1bhj}$	the sum of variable y within the responding dwellings of 1bh.

Part 2 of the Universe

The total of the considered variables is zero in the case of empty dwellings, therefore inserting the weights h_k also for part 2, expression (2.22) gives

$$\begin{aligned}\widehat{\tau}_{\pi^*} &= \sum_{b=1}^{B_1}\sum_{h=1}^{H_{1b}}\left[\frac{W_{1bh}}{w_{1bh}}\cdot\frac{w^{(1)}_{1bh}+w^{(2)}_{1bh}}{w^{(1)}_{1bh}}\cdot\sum_{i=1}^{w^{(1)}_{1bh}} y_{1bhi}\right] + \\ &+ \sum_{b=1}^{B_2}\sum_{h=1}^{H_{2b}}\left[\frac{W_{2bh}}{\sum_{i=1}^{c_{2bh}} W_{2bhi}}\cdot\sum_{i=1}^{c_{b2h}}\left(\frac{W_{2bhi}}{w_{2bhi}}\cdot\frac{w^{(1)}_{2bhi}+w^{(2)}_{2bhi}}{w^{(1)}_{2bhi}}\cdot\sum_{j=1}^{w^{(1)}_{2bhi}} y_{2bhij}\right)\right], \qquad (2.24)\end{aligned}$$

where $W_{2bh}=\sum_{i=1}^{C_{2bh}} W_{2bhi}$ and $\sum_{j=1}^{w^{(1)}_{2bhi}} y_{2bhij}$ is the sum of variable y within all responding dwellings of part 2 of federal state b, stratum h and PSU i.

As the weight in part 2 is multiplied in each PSU with the variable ratio

$$\frac{W_{2bh}}{\sum_{i=1}^{c_{2bh}} W_{2bhi}} \quad ,$$

the estimator is not exactly a Horvitz-Thompson Estimator. To avoid confusion we will use the notation π^* if it is not exactly a Horvitz-Thompson-Estimator given by (2.21).

Simulation Results of the Total Estimator for Three Different Universes

For the simulation we used three different pseudo-populations (POP). The first population consists of only occupied dwellings, the second of occupied and empty (=uninhabited) dwellings and the third population also includes unit non-response. The non-response was generated according to the mechanism, which is described in detail in Section 5.1 of deliverable D1.1 of workpackage 1 of the DACSEIS project.

Table 2.5: Different pseudo-populations

POP	description
OOU	*Only Occupied Universe*
OUU	*Occupied and Unoccupied Universe*
INU	*Include Non-Response Universe*

The results below show the total estimators of the variables

- LFC: number of employees
- TOT: populations size
- UNI: number of university graduates

Table 2.6: True values of LFC, TOT and UNI per federal state and for Austria as whole

FST	LFC	TOT	UNI
BGL	106,273.00	265,903.00	6,951.00
KTN	197,272.00	514,700.00	14,291.00
NOE	600,704.00	1,437,630.00	46,401.00
OOE	538,002.00	1,265,354.00	36,789.00
SBG	202,824.00	465,514.00	19,264.00
STM	443,356.00	1,101,788.00	28,069.00
TIR	261,315.00	624,345.00	20,632.00
VBG	136,301.00	321,451.00	8,702.00
WIE	652,619.00	1,466,117.00	123,482.00
OE	3,138,666.00	7,462,802.00	304,581.00

for the three pseudo-universes.

The true values of the pseudo-population per federal state are shown in Table 2.6.

Tables 2.7 - 2.9 show for $r = 10,000$ runs the estimated values and their biases. The bias is given by the difference of the estimated poulation mean and the true value of the pseudo-population.

Table 2.7: Total estimator for the OOU per federal state and for Austria as whole

FST	LFC	Bias	TOT	Bias	UNI	Bias
BGL	106,280.11	7.11	265,809.31	-93.69	6,904.83	-46.17
KTN	197,306.27	34.27	514,746.16	46.16	14,307.73	16.73
NOE	600,832.54	128.54	1,437,435.77	-194.23	46,416.75	15.75
OOE	538,000.42	-1.58	1,265,424.61	70.61	36,769.94	-19.06
SBG	202,870.02	46.02	465,431.15	-82.85	19,291.11	27.11
STM	443,224.84	-131.16	1,101,792.87	4.87	28,023.48	-45.52
TIR	261,494.63	179.63	624,512.89	167.89	20,596.71	-35.29
VBG	136,261.34	-39.66	321,426.63	-24.37	8,699.80	-2.20
WIE	652,730.83	111.83	1,466,207.87	90.87	123,475.62	-6.38
OE	3,139,001.00	335.00	7,462,787.26	-14.74	304,485.96	-95.04

Table 2.8: Total estimator for the OUU per federal state and for Austria as whole

FST	LFC	Bias	TOT	Bias	UNI	Bias
BGL	106,375.77	102.77	265,862.68	-40.32	6,971.13	20.13
KTN	197,329.26	57.26	514,769.48	69.48	14,298.47	7.47
NOE	600,853.28	149.28	1,437,281.30	-348.70	46,406.13	5.13
OOE	537,806.84	-195.16	1,265,228.49	-125.51	36,788.82	-0.18
SBG	202,708.30	-115.70	465,278.72	-235.28	19,261.76	-2.24
STM	443,188.42	-167.58	1,101,620.86	-167.14	27,999.04	-69.96
TIR	261,674.69	359.69	624,687.83	342.83	20,610.26	-21.74
VBG	136,385.54	84.54	321,692.17	241.17	8,712.18	10.18
WIE	652,737.50	118.50	1,466,315.97	198.97	123,541.59	59.59
OE	3,139,059.62	393.62	7,462,737.49	-64.51	304,589.37	8.37

Table 2.9: Total estimator for the INU per federal state and for Austria as whole

FST	LFC	Bias	TOT	Bias	UNI	Bias
BGL	106,357.44	84.44	265,706.42	-196.58	6,960.39	9.39
KTN	197,203.07	-68.93	514,398.93	-301.07	14,228.83	-62.17
NOE	601,289.61	585.61	1,437,318.11	-311.89	46,451.40	50.40
OOE	538,006.47	4.47	1,264,838.64	-515.36	36,908.17	119.17
SBG	202,857.05	33.05	465,338.76	-175.24	19,274.64	10.64
STM	443,487.36	131.36	1,101,521.65	-266.35	27,790.89	-278.11
TIR	261,444.15	129.15	624,836.41	491.41	20,633.53	1.53
VBG	136,425.73	124.73	321,726.24	275.24	8,707.15	5.15
WIE	652,599.47	-19.53	1,465,851.95	-265.05	123,471.21	-10.79
OE	3,139,670.35	1,004.35	7,461,537.11	-1,264.89	304,426.21	-154.79

2.2.4 Estimation of a Ratio

Is τ_y the total of an interesting variable y and τ_z the total of an interesting variable z, the ratio

$$R = \frac{\tau_y}{\tau_z} = \frac{\sum_U y_k}{\sum_U z_k} \tag{2.25}$$

can be estimated by the nonlinear estimator

$$\widehat{R} = \frac{\widehat{\tau}_{y\pi}}{\widehat{\tau}_{z\pi}} \quad . \tag{2.26}$$

For an interesting ratio 'labor-force-participation-rate', short 'LFR', the numerator is given by the total estimator 'LFC' and the denominator is given by the total estimator 'TOT'. The estimation of both variables follows exactly the same procedure given by (2.24). Thus, the results of the estimated ratio can be shown immediately.

Simulation Results of the Ratio 'LFR' for the Three Different Universes

The true values of the pseudo-universe per federal state are shown in Table 2.10.

Table 2.10: True values of LFR per federal state and for Austria as whole

FST	LFR
BGL	0.399668
KTN	0.383276
NOE	0.417843
OOE	0.425179
SBG	0.435699
STM	0.402397
TIR	0.418543
VBG	0.424018
WIE	0.445134
OE	0.420575

Table 2.11 shows for $r = 10,000$ runs the estimated values and their biases for each pseudo-universe.

Table 2.11: LFR per federal state

	OOU		OUU		INU	
FST	LFR	Bias	LFR	Bias	LFR	Bias
BGL	0.399836	1.676257E-04	0.400115	4.471790E-04	0.400282	6.135050E-04
KTN	0.383308	3.221250E-05	0.383335	5.950911E-05	0.383366	9.031751E-05
NOE	0.417989	1.458819E-04	0.418048	2.052405E-04	0.418341	4.981033E-04
OOE	0.425154	-2.497654E-05	0.425067	-1.120693E-04	0.425356	1.767739E-04
SBG	0.435875	1.764370E-04	0.435671	-2.833331E-05	0.435934	2.351027E-04
STM	0.402276	-1.208253E-04	0.402306	-9.106682E-05	0.402613	2.165500E-04
TIR	0.418718	1.751194E-04	0.418889	3.460980E-04	0.418420	-1.224731E-04
VBG	0.423927	-9.123233E-05	0.423963	-5.508661E-05	0.424043	2.494196E-05
WIE	0.445183	4.868559E-05	0.445155	2.041509E-05	0.445201	6.716265E-05
OE	0.420620	4.572059E-05	0.420631	5.638093E-05	0.420781	2.059000E-04

2.2.5 Variance Estimation

The variance of the $nsim$ $(i = 1, \ldots, nsim)$ simulation results

$$\mathrm{V}(\widehat{\tau}) := \mathrm{V}^*(\widehat{\tau}) = \frac{1}{nsim - 1} \sum_{i=1}^{nsim} (\widehat{\tau}_i - \overline{\widehat{\tau}})^2 \tag{2.27}$$

is used as a reference value to compare different variance estimators $\widehat{\mathrm{V}}(\widehat{\tau})$.

In practice direct variance estimators have the advantage of being easy to compute. Here we compare four of them, partly in use for the AMC-results.

Simplified Variance Estimation of a Total in the AMC

Under simple random sampling without replacement, the variance estimator of the estimator $\widehat{\tau}$ is defined by

$$\widehat{\mathrm{V}}_1(\widehat{\tau}_{\pi^*}) = \frac{(N-n)(N-\widehat{\tau}_{\pi^*})\widehat{\tau}_{\pi^*}}{Nn} \tag{2.28}$$

where

N	the number of persons in the universe,
n	the number of persons in the sample
$\widehat{\tau}^*_\pi$	total estimator of an interesting variable

Under assumption that $\widehat{\tau}_b = \frac{N_b}{N} \cdot \widehat{\tau}$ the variance estimator is given by

$$\widehat{\mathrm{V}}_2(\widehat{\tau}_{\pi^*}) = \sum_{b=1}^{B} \frac{(N_b - n_b)(N-\widehat{\tau}_{\pi^*})N_b\widehat{\tau}_{\pi^*}}{N^2 n_b} \quad . \tag{2.29}$$

Accounting for different sampling fractions per federal state the estimator can be calculated as

$$\widehat{\mathrm{V}}_3(\widehat{\tau}_{\pi^*}) = \sum_{b=1}^{B} \frac{(N_b - n_b)(N_b-\widehat{\tau}_{b\pi^*})\widehat{\tau}_{b\pi^*}}{N_b n_b} \tag{2.30}$$

where $\sum_{b=1}^{B} N_b = N$ und $\sum_{b=1}^{B} n_b = n$.

Complex Variance Estimation of a Total in the AMC

As the sample of dwellings consists of occupied and unoccupied dwellings and dwellings with non-response the sample size has to be adopted. For the variance estimation the number of dwellings is smaller than the defined size w_{1bh} in (2.19) respectively w_{2bhi} in (2.20). That is because dwellings with the category *non-response* are not components of the sampling size used for variance estimation.

Hence, the new size for part 1 is defined by

$$w^*_{1bh} = w^{(1)}_{1bh} + w^{(3)}_{1bh} \tag{2.31}$$

respectively for part 2 by

$$w^*_{2bhi} = w^{(1)}_{2bhi} + w^{(3)}_{2bhi} \quad . \tag{2.32}$$

For the complex variance estimation that takes into account the complex design, but does not account the modification of the estimator in part 2 of the universe, so that it gives the variance estimator for a real Horvitz-Thompson-Estimator for a total, the parameters are

W_{1bh} number of dwellings in $1bh$ of the pseudo-universe
w^*_{1bh} number of dwellings in $1bh$ of the sample
W_{2bhi} number of dwellings in $2bhi$ of the pseudo-universe
w^*_{2bhi} number of dwellings in $2bhi$ of the sample
C_{2bh} number of PSUs in $2bh$ of the pseudo-universe
c_{2bh} number of PSUs in $2bh$ of the sample.

Part 1 of the Universe The complex variance estimator of part 1 is given by

$$\widehat{\mathrm{V}}_4(\widehat{\tau}_{\pi 1}) = \sum_{b=1}^{B_1}\sum_{h=1}^{H_{1b}} \frac{W_{1bh}^2}{w^*_{1bh}}\Big(1-\frac{w^*_{1bh}}{W_{1bh}}\Big)s^2_{1bh} \quad , \tag{2.33}$$

where the sample-variance in stratum $1bh$ is given by

$$s^2_{1bh} = \frac{1}{w^*_{1bh}-1}\sum_{i=1}^{w^*_{1bh}}(y_{1bhi}-\overline{y}_{1bh})^2 \tag{2.34}$$

and the stratum-mean is given by

$$\overline{y}_{1bh} = \frac{1}{w^*_{1bh}}\sum_{i=1}^{w^*_{1bh}} y_{1bhi} \quad . \tag{2.35}$$

Part 2 of the Universe The complex variance estimator of part 2 is given by

$$\begin{aligned}\widehat{\mathrm{V}}_4(\widehat{\tau}_{\pi 2}) &= \sum_{b=1}^{B_2}\sum_{h=1}^{H_{2b}}\Big[\frac{C^2_{2bh}}{c_{2bh}}\Big(1-\frac{c_{2bh}}{C_{2bh}}\Big)\cdot s^2_{\widehat{\tau}2bh} + \\ &+ \frac{C_{2bh}}{c_{2bh}}\sum_{i=1}^{c_{2bh}}\frac{W^2_{2bhi}}{w^*_{2bhi}}\cdot\Big(1-\frac{w^*_{2bhi}}{W_{2bhi}}\Big)s^2_{2bhi}\Big] \quad ,\end{aligned}$$

where the variance between the PSUs (Between-variance) is given by

$$s^2_{\widehat{\tau}2bh} = \frac{1}{c_{2bh}-1}\sum_{i=1}^{c_{2bh}}\Big(W_{2bhi}\cdot\overline{y}_{2bhi} - \frac{1}{c_{2bh}}\sum_{k=1}^{c_{2bh}} W_{2bhk}\cdot\overline{y}_{2bhk}\Big)^2 \tag{2.36}$$

the sample variance within one PSU i (Within-variance) is given by

$$s^2_{2bhi} = \frac{1}{w^*_{2bhi}-1}\sum_{j=1}^{w^*_{2bhi}}(y_{2bhij}-\overline{y}_{2bhi})^2 \tag{2.37}$$

and the PSU-mean is given by

$$\overline{y}_{2bhi} = \frac{1}{w^*_{2bhi}}\sum_{j=1}^{w^*_{2bhi}} y_{2bhij} \quad . \tag{2.38}$$

Complex Variance Estimator For both parts the variance estimator can now be calculated as

$$\begin{aligned}\widehat{\mathrm{V}}_4(\widehat{\tau}_\pi) &= \sum_{b=1}^{B_1}\sum_{h=1}^{H_{1b}} \frac{W_{1bh}^2}{w_{1bh}^*}\Big(1-\frac{w_{1bh}^*}{W_{1bh}}\Big)s_{1bh}^2 + \\ &+ \sum_{b=1}^{B_2}\sum_{h=1}^{H_{2b}}\Big[\frac{C_{2bh}^2}{c_{2bh}}\Big(1-\frac{c_{2bh}}{C_{2bh}}\Big)s_{\widehat{\tau}2bh}^2 + \\ &+ \frac{C_{2bh}}{c_{2bh}}\sum_{i=1}^{c_{2bh}}\frac{W_{2bhi}^2}{w_{2bhi}^*}\cdot\Big(1-\frac{w_{2bhi}^*}{W_{2bhi}}\Big)s_{2bhi}^2\Big] \quad . \end{aligned} \tag{2.39}$$

Simulation Results of the Variance Estimation in Three Different Universes

The simulation results for $nsim = 10,000$ samples are shown for the three POPs.

Variance Estimation of the LFC Table 2.12 shows the relevant statistics of the total estimator LFC.

Table 2.12: Statistics of the LFC

LFC							
POP	$E(\widehat{\tau})$	$\mathrm{V}(\widehat{\tau})$	$SE(\widehat{\tau})$	$B(\widehat{\tau})$	$MSE(\widehat{\tau})$	$CV(\widehat{\tau})$	$BR(\widehat{\tau})$
OOU	3,139,001.00	4.482615E+08	21,172.19	335.00	4.483737E+08	6.7449E-03	1.5823E-02
OUU	3,139,059.62	5.405967E+08	23,250.74	393.62	5.407517E+08	7.4069E-03	1.6930E-02
INU	3,139,670.35	5.950878E+08	24,394.42	1,004.35	5.960965E+08	7.7697E-03	4.1171E-02

Table 2.13 shows the results of the variance estimation of the LFC.

Table 2.13: Variance estimation of the LFC

LFC						
POP	OOU		OUU		INU	
	$\widehat{\mathrm{V}}_\cdot(\widehat{\tau})$	$[\widehat{\mathrm{V}}_\cdot(\widehat{\tau})]^{\frac{1}{2}}$	$\widehat{\mathrm{V}}_\cdot(\widehat{\tau})$	$[\widehat{\mathrm{V}}_\cdot(\widehat{\tau})]^{\frac{1}{2}}$	$\widehat{\mathrm{V}}_\cdot(\widehat{\tau})$	$[\widehat{\mathrm{V}}_\cdot(\widehat{\tau})]^{\frac{1}{2}}$
$\widehat{\mathrm{V}}_1(\widehat{\tau})$	1.837303E+08	13,554.71	1.837092E+08	13,553.94	2.062698E+08	14,362.10
$\widehat{\mathrm{V}}_2(\widehat{\tau})$	2.195938E+08	14,818.70	2.196041E+08	14,819.05	2.498763E+08	15,807.47
$\widehat{\mathrm{V}}_3(\widehat{\tau})$	2.198119E+08	14,826.06	2.198083E+08	14,825.93	2.501162E+08	15,815.06
$\widehat{\mathrm{V}}_4(\widehat{\tau})$	5.357135E+08	23,145.48	5.969341E+08	24,432.24	6.427462E+08	25,352.44

Figure 2.1 shows the distribution of the four variance estimators of the LFC using the INU.

It is shown that the estimators $\widehat{\mathrm{V}}_1(\widehat{\tau})$-$\widehat{\mathrm{V}}_3(\widehat{\tau})$ underestimate the real variance, whereas $\widehat{\mathrm{V}}_4(\widehat{\tau})$ estimates the true variance best.

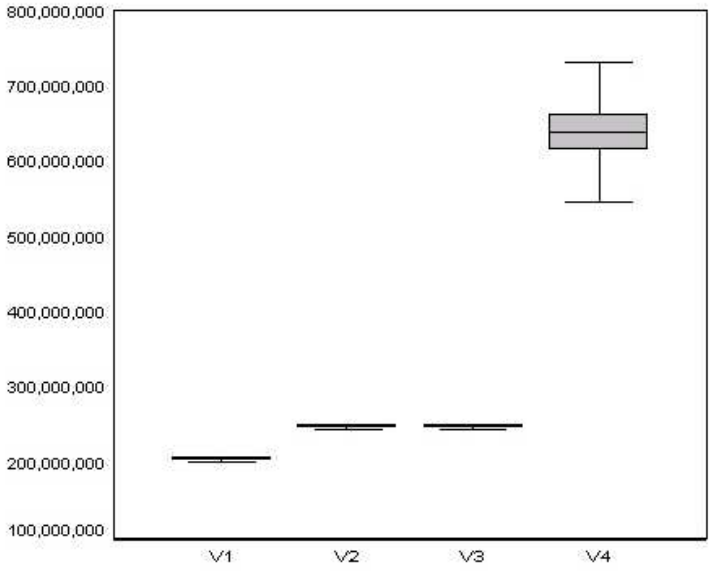

Figure 2.1: Boxplot of the LFC

Variance Estimation of the TOT Table 2.14 shows the relevant statistics of the total estimator TOT.

Table 2.14: Statistics of the TOT

TOT							
POP	$E(\widehat{\tau})$	$\mathrm{V}(\widehat{\tau})$	$SE(\widehat{\tau})$	$B(\widehat{\tau})$	$MSE(\widehat{\tau})$	$CV(\widehat{\tau})$	$BR(\widehat{\tau})$
OOU	7,462,787.26	8.775408E+08	29,623.32	-14.74	8.775410E+08	3.9695E-03	-4.9766E-04
OUU	7,462,737.49	1.382518E+09	37,182.23	-64.51	1.382522E+09	4.9824E-03	-1.7350E-03
INU	7,461,537.11	1.485310E+09	38,539.71	-1,264.89	1.486910E+09	5.1651E-03	-3.2820E-02

Table 2.15 shows the results of the variance estimation of the TOT. As the results of the estimator of the population total $\widehat{\tau}$ is around the population size N the simplified variance estimators can cause negative results. Therefore they are useless and the results are only given for the complex variance estimator.

Table 2.15: Variance estimation of TOT

TOT						
POP	OOU		OUU		INU	
	$\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})$	$[\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})]^{\frac{1}{2}}$	$\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})$	$[\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})]^{\frac{1}{2}}$	$\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})$	$[\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})]^{\frac{1}{2}}$
$\widehat{\mathrm{V}}_4(\widehat{\tau})$	1.367896E+09	36,985.08	1.671517E+09	40,884.19	1.814300E+09	42,594.60

Figure 2.2 shows the distribution of the variance estimator of the TOT using the INU.

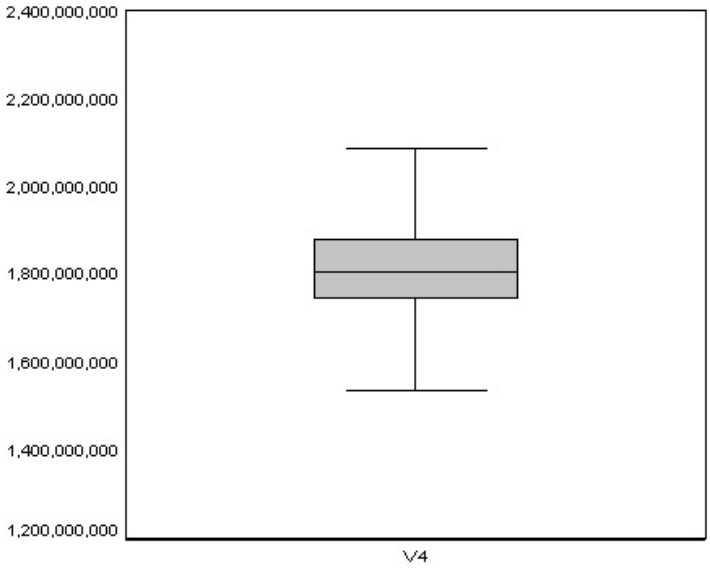

Figure 2.2: Boxplot of the TOT

Variance Estimation of the UNI Table 2.16 shows the relevant statistics of the total estimator UNI.

Table 2.16: Statistics of the UNI

UNI							
POP	$E(\widehat{\tau})$	$\mathrm{V}(\widehat{\tau})$	$SE(\widehat{\tau})$	$B(\widehat{\tau})$	$MSE(\widehat{\tau})$	$CV(\widehat{\tau})$	$BR(\widehat{\tau})$
OOU	304,485.96	4.595807E+07	6,779.24	-95.04	4.596710E+07	2.2265E-02	-1.4019E-02
OUU	304,589.37	4.741918E+07	6,886.16	8.37	4.741925E+07	2.2608E-02	1.2159E-03
INU	304,426.21	5.391412E+07	7,342.62	-154.79	5.393808E+07	2.4120E-02	-2.1081E-02

Table 2.17 shows the results of the variance estimation of the UNI.

Table 2.17: Variance estimation of the UNI

UNI						
POP	OOU		OUU		INU	
	$\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})$	$[\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})]^{\frac{1}{2}}$	$\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})$	$[\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})]^{\frac{1}{2}}$	$\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})$	$[\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})]^{\frac{1}{2}}$
$\widehat{\mathrm{V}}_1(\widehat{\tau})$	2.950565E+07	5,431.91	2.951178E+07	5,432.47	3.311723E+07	5,754.76
$\widehat{\mathrm{V}}_2(\widehat{\tau})$	3.526495E+07	5,938.43	3.527782E+07	5,939.51	4.011804E+07	6,333.88
$\widehat{\mathrm{V}}_3(\widehat{\tau})$	3.957852E+07	6,291.15	3.958724E+07	6,291.84	4.542699E+07	6,739.95
$\widehat{\mathrm{V}}_4(\widehat{\tau})$	4.591535E+07	6,776.09	4.749491E+07	6,891.65	5.179009E+07	7,196.53

Figure 2.3 shows the distribution of the variance estimators of the UNI using the INU-universe. All four estimators underestimate the true variance with the fourth once again being the best.

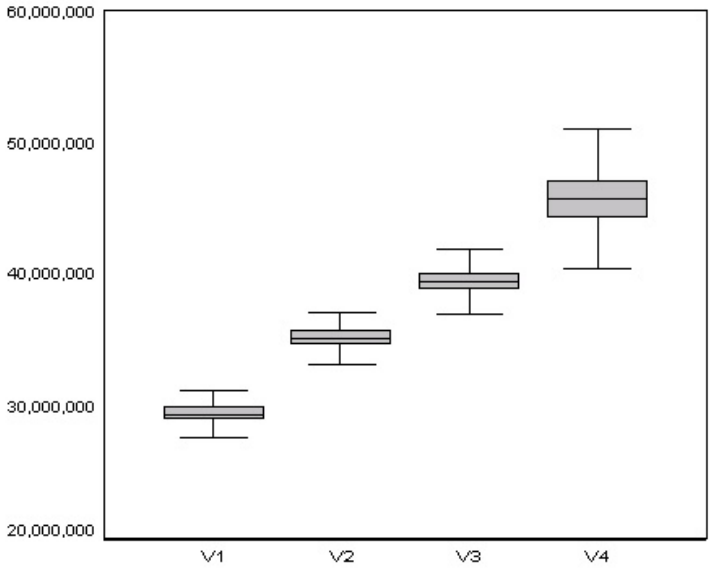

Figure 2.3: Boxplot of the UNI

Coverage Rate A confidence interval with confidence level $1 - \alpha = 0.95$ for the estimated parameter $\widehat{\tau}$ for each sample i $(i = 1, \ldots, nsim)$ is given by

$$\left[\widehat{\tau} - z_{1-\alpha/2} \cdot [\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})]^{1/2} \leq \tau \leq \widehat{\tau} + z_{1-\alpha/2} \cdot [\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})]^{1/2}\right] \quad . \tag{2.40}$$

Table 2.18 shows the coverage rates $CR_{\cdot}(\tau)$ for all variance estimators $\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})$.

Table 2.18: Coverage rates

	LFC				TOT	UNI			
POP	$CR_1(\tau)$	$CR_2(\tau)$	$CR_3(\tau)$	$CR_4(\tau)$	$CR_4(\tau)$	$CR_1(\tau)$	$CR_2(\tau)$	$CR_3(\tau)$	$CR_4(\tau)$
OOU	0.7934	0.8306	0.8307	0.9675	0.9856	0.8829	0.9148	0.9309	0.9513
OUU	0.7484	0.7880	0.7882	0.9607	0.9688	0.8777	0.9077	0.9265	0.9514
INU	0.7539	0.7947	0.7950	0.9584	0.9692	0.8741	0.9074	0.9262	0.9441

Variance Estimation of a Ratio

The estimation of the variance of a ratio is more complex because the covariance of the two estimated totals has to be accounted. For simplified variance estimation we will ignore the covariance and define $\widehat{R} := \widehat{p}$, with $\widehat{p}$, the proportion of two total estimators of interest.

Simplified Variance Estimation of a Ratio Under assumption that there exists no covariance of the two variables y and z the variance can be estimated by (2.28). As

the population total of the pseudo-universe is now a random variable it also has to be estimated by $\widehat{N} = \widehat{\tau}_{z\pi}$. Thus, the variance estimator of the estimated ratio $\widehat{p}$ can be written as

$$\widehat{\mathrm{V}}_1(\widehat{p}) = \frac{\widehat{p}(1-\widehat{p})}{n} \cdot \frac{\widehat{N}-n}{\widehat{N}} \quad . \tag{2.41}$$

Under assumption that $\widehat{p}_b = \widehat{p}$ the variance estimator is given by

$$\widehat{\mathrm{V}}_2(\widehat{p}) = \sum_{b=1}^{B} \frac{\widehat{p}(1-\widehat{p})}{n_b} \cdot \frac{(\widehat{N}_b - n_b)\widehat{N}_b}{\widehat{N}^2} \tag{2.42}$$

and in consideration of the sampling fraction per federal state with $\widehat{p} = \sum_{b=1}^{B} \widehat{p}_b \cdot \frac{\widehat{N}_b}{\widehat{N}}$ the variance estimator can be calculated as

$$\widehat{\mathrm{V}}_3(\widehat{p}) = \sum_{b=1}^{B} \left(\frac{\widehat{N}_b}{\widehat{N}}\right)^2 \cdot \widehat{\mathrm{V}}_3(\widehat{p}_b) = \sum_{b=1}^{B} \cdot \left(\frac{\widehat{N}_b}{\widehat{N}}\right)^2 \cdot \frac{\widehat{p}_b(1-\widehat{p}_b)}{n_b} \cdot \frac{\widehat{N}_b - n_b}{\widehat{N}_b}. \tag{2.43}$$

Complex Variance Estimation of a Ratio For complex variance estimation the Taylor linearisation technique can be used to find an approximate variance of the ratio $\widehat{R}$ (SÄRNDAL *et al.*, 1992, p. 177).

The variance estimator of $\widehat{R}$ can be calculated as

$$\widehat{\mathrm{V}}(\widehat{R}) = \frac{1}{\widehat{\tau}_{z\pi}^2} \cdot \left[\widehat{\mathrm{V}}(\widehat{\tau}_{y\pi}) + \widehat{R}^2\widehat{\mathrm{V}}(\widehat{\tau}_{z\pi}) - 2\widehat{R}\widehat{C}(\widehat{\tau}_{y\pi}, \widehat{\tau}_{z\pi})\right], \tag{2.44}$$

where the covariance of the estimators $\widehat{\tau}_{y\pi}$ and $\widehat{\tau}_{z\pi}$ is given by $\widehat{C}(\widehat{\tau}_{y\pi}, \widehat{\tau}_{z\pi})$. (SÄRNDAL *et al.*, 1992, p. 178).

Both estimators of total $\widehat{\tau}_{y\pi}$ and $\widehat{\tau}_{z\pi}$ can be calculated by (2.24) and both variance estimators $\widehat{\mathrm{V}}(\widehat{\tau}_{y\pi})$ and $\widehat{\mathrm{V}}(\widehat{\tau}_{z\pi})$ by (2.39). The only unknown size is given by the covariance. The following derivation of the covariance can be found in (QUATEMBER, 2003).

Covariance in Part 1 The covariance in part 1 is given by

$$\widehat{C}_1(\widehat{\tau}_{y_1\pi}, \widehat{\tau}_{z_1\pi}) = \sum_{b=1}^{B_1} \sum_{h=1}^{H_{1b}} \frac{(1-f_{1bh})W_{1bh}^2}{w_{1bh}^*} \cdot S_{yz1bh}, \tag{2.45}$$

where the the sampling fraction within each stratum $1bh$ is given by

$$f_{1bh} = \frac{w_{1bh}^*}{W_{1bh}}$$

and the sample covariance of y and z within $1bh$ is given by

$$\begin{aligned} S_{yz1bh} &= \frac{1}{w^*_{1bh}-1}\Big(\sum_{1bh} y_k \cdot z_k - \frac{1}{w^*_{1bh}}\sum_{1bh} y_k \cdot \sum_{1bh} z_k\Big) = \\ &= \frac{1}{w^*_{1bh}-1}\sum_{1bh}(y_k - \overline{y}_{1bh}) \cdot (z_k - \overline{z}_{1bh}) \quad . \end{aligned}$$

Covariance in Part 2 The covariance of part 2 in is given by

$$\begin{aligned} \widehat{C}_2(\widehat{\tau}_{y_2\pi}, \widehat{\tau}_{z_2\pi}) &= \sum_{b=1}^{B_2}\sum_{h=1}^{H_{2b}}\Big\{\sum_{i=1}^{c_{2bh}} \frac{1}{f^*_{2bh}\cdot f_{2bhi}\cdot(w^*_{2bhi}-1)} \cdot \Big[(W_{2bhi} - w^*_{2bhi}) \\ &\cdot \sum_{j=1}^{w^*_{2bhi}} y_{2bhij}\cdot z_{2bhij} - \Big(W_{2bhi} - 1 - \frac{w^*_{2bhi}-1}{f^*_{2bh}\cdot f_{2bhi}}\Big)\cdot \sum_{j=1}^{w^*_{2bhi}} y_{2bhij}\cdot \sum_{j=1}^{w^*_{2bhi}} z_{2bhij}\Big] \\ &+ \Big(1 - \frac{f^*_{2bh}\cdot(C_{2bh}-1)}{c_{2bh}-1}\Big)\cdot \sum_{i\neq i'} \frac{1}{f^{*2}_{2bh}\cdot f_{2bhi}\cdot f_{2bhi'}} \cdot \sum_{j=1}^{w^*_{2bhi}} y_{2bhij}\cdot \sum_{j=1}^{w^*_{2bhi'}} z_{2bhi'j}\Big\} \end{aligned} \tag{2.46}$$

where the sampling fraction within each stratum $2bhi$ is given by

$$f_{2bhi} = \frac{w^*_{2bhi}}{W_{2bhi}},$$

the sampling fraction of selected PSUs in stratum $2bh$ is given by

$$f^*_{2bh} = \frac{c_{2bh}}{C_{2bh}}$$

and i denotes a PSU i and i' a PSU $i' \neq i$.

Covariance The formula (2.45) and (2.46) can now be used to estimate the variance of the estimated ratio

$$\widehat{\mathrm{V}}(\widehat{R}) = \frac{1}{\widehat{\tau}^2_{z\pi}}\cdot\Big[\widehat{\mathrm{V}}(\widehat{\tau}_{y\pi}) + \widehat{R}^2\cdot\widehat{\mathrm{V}}(\widehat{\tau}_{z\pi}) - 2\cdot\widehat{R}\cdot[\widehat{C}_1(\widehat{\tau}_{y_1\pi},\widehat{\tau}_{z_1\pi}) + \widehat{C}_2(\widehat{\tau}_{y_2\pi},\widehat{\tau}_{z_2\pi})]\Big]. \tag{2.47}$$

Simulation Results of the Variance Estimation of a Ratio

Table 2.19 shows the relevant statistics of the ratio estimator LFR.

Table 2.20 shows the results of the variance estimation. The variance estimator $\widehat{\mathrm{V}}_4(\widehat{\tau})$ gives the results of the used Taylor linearisation.

Figure 2.4 shows the variance estimators of the LFR.

Table 2.19: Statistics of the LFR

LFR							
POP	$E(\widehat{\tau})$	$\mathrm{V}(\widehat{\tau})$	$SE(\widehat{\tau})$	$B(\widehat{\tau})$	$MSE(\widehat{\tau})$	$CV(\widehat{\tau})$	$BR(\widehat{\tau})$
OOU	0.420620	4.889280E-06	2.2112E-03	4.572059E-05	4.891370E-06	5.2569E-03	2.0677E-02
OUU	0.420631	4.887657E-06	2.2108E-03	5.638093E-05	4.890836E-06	5.2559E-03	2.5502E-02
INU	0.420781	5.506905E-06	2.3467E-03	2.059000E-04	5.549300E-06	5.5770E-03	8.7741E-02

Table 2.20: Variance estimation of the LFR

LFR						
POP	OOU		OUU		INU	
	$\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})$	$[\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})]^{\frac{1}{2}}$	$\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})$	$[\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})]^{\frac{1}{2}}$	$\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})$	$[\widehat{\mathrm{V}}_{\cdot}(\widehat{\tau})]^{\frac{1}{2}}$
$\widehat{\mathrm{V}}_1(\widehat{\tau})$	3.299015E-06	0.001816	3.298670E-06	0.001816	3.70393E-06	0.001925
$\widehat{\mathrm{V}}_2(\widehat{\tau})$	3.942661E-06	0.001986	3.942516E-06	0.001986	4.48597E-06	0.002118
$\widehat{\mathrm{V}}_3(\widehat{\tau})$	3.946977E-06	0.001987	3.946822E-06	0.001987	4.49106E-06	0.002119
$\widehat{\mathrm{V}}_4(\widehat{\tau})$	4.865754E-06	0.002206	4.912969E-06	0.002217	5.24153E-06	0.002289

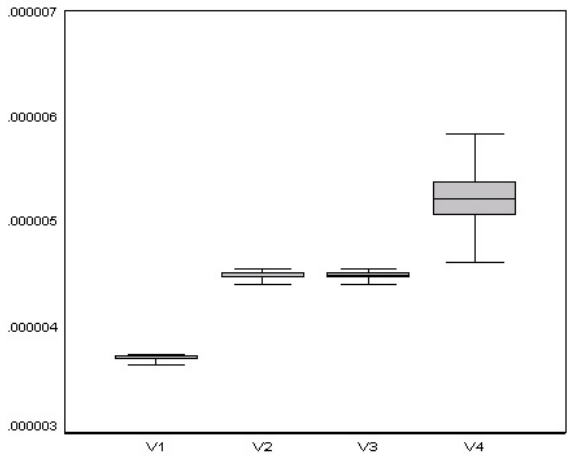

Figure 2.4: Boxplot of the variance estimator of the LFR

2.3 Simulation of Unit Non-response in the Swiss HBS Universe

Unit non-response is a challenging problem in the Swiss HBS as illustrated in Section 8 of deliverable D2.1. Since very little auxiliary information is available from the sampling frame, non-response cannot be addressed in a straightforward manner by calibration or post-stratification. A two-level weighting scheme was therefore developed that uses information from partially participating households. Level 1 corresponds to households with a minimum set of basic information (geographical region, household size and age, socio-economic group and nationality of its reference person) while level 2 contains house-

holds with full participation. The response probabilities between level 1 and level 2 were estimated using a logistic regression (for each one of the 12 waves) with the following variables:

- geographical region (binary variable: 1 if in region Ticino or Région lémanique, 0 otherwise);
- socio-economic group of the reference person (binary variable: 1 if employee, 0 otherwise);
- nationality of the reference person (binary variable: 1 if Swiss, 0 otherwise);
- size of the household (numerical variable).

The pseudo-universe for the simulation (with around 3.2 million households) was created based on the 9,295 level-2-households corrected by the weights obtained from the weighting scheme described in Section 8.6 of deliverable D2.1. In the universe, the households have therefore relative frequencies corresponding to the 15,434 level-1-households of the survey. Because of file size restrictions, the number of variables had to be drastically reduced compared to the survey. Only the following five variables were retained:

- geographical region (GREGION; 7 regions);
- socio-economic group of the reference person (STASOCIO; 6 groups);
- household type (TYPE; 8 classes);
- total of expenditures (EXPENDITURES; continuous variable);
- total of income (INCOME; continuous variable).

In order to simulate unit non-response, three mechanisms were proposed, inspired by the ideas that led to the weighting scheme. Non-response was simulated during the drawing process by randomly attributing each household drawn (around 15,000 per sample) to "non-responding" (around 6,000 per sample) or "responding" (around 9,000 per sample) with a probability that depended on the following auxiliary variables:

- Mechanism A (simple): a three-parameter logistic regression model with variables GREGION and STASOCIO. The response probabilities are calculated using the equation
$$\rho_k = \frac{\exp(\beta_0 + \beta_1 \mathrm{SA}_k + \beta_2 \mathrm{G13}_k)}{1 + \exp(\beta_0 + \beta_1 \mathrm{SA}_k + \beta_2 \mathrm{G13}_k)}.$$
The estimated coefficients obtained by logistic regression are listed in Table 2.21. The four estimated response probabilities are compared with the observed ones in Table 2.22.
- Mechanism C (intermediate): a 42-parameter logistic regression model with the observed response rates in each cell of all possible combinations of levels of GREGION and STASOCIO. Table 2.23 compares the observed response rates under Mechanism C to the estimated and observed response rates under Mechanism A.

Table 2.21: Parameter values used for non-response Mechanism A.

β_0 (constant)	0.313
β_1 (SA: GREGION $\neq$ 2 or 7)	0.562
β_2 (G13: STASOCIO = 1)	-0.338

Table 2.22: Calculated and observed response probabilities ρ (%) under non-response Mechanism A.

SA (GREGION $\neq$ 2 or 7)	0	1	0	1
G13 (STASOCIO = 1)	0	0	1	1
Observed response probability ρ_{obs}	58.02	49.30	70.43	63.14
Calculated response probability ρ_k	57.75	49.37	70.56	63.11

- Mechanism D (complex): a 48-parameter logistic regression model with intercept β_{0i}, GREGION β_{1i} (SA: GREGION $\neq$ 2 or 7), STASOCIO β_{2i} (G13: STASOCIO = 1) and TYPE β_{3i} (TY: TYPE = 0) for 12 waves ($i = 1, \ldots, 12$). Table 2.24 lists the corresponding parameter values. The response probabilities are calculated using the equation

$$\rho_k = \frac{\exp(\beta_{0i} + \beta_{1i}\text{SA}_k + \beta_{2i}\text{G13}_k + \beta_{3i}\text{TY}_k)}{1 + \exp(\beta_{0i} + \beta_{1i}\text{SA}_k + \beta_{2i}\text{G13}_k + \beta_{3i}\text{TY}_k)}.$$

  The estimated coefficients obtained by logistic regression are given in Table 2.24.

Table 2.24: Parameter values used for imputation mechanism D.

	Wave1	Wave2	Wave3	Wave4	Wave5	Wave6
β_{0i}	0.482	0.651	0.353	0.619	0.514	0.281
β_{1i}	-0.174	-0.463	-0.058	-0.700	-0.126	-0.159
β_{2i}	0.423	0.525	0.543	0.393	0.373	0.563
β_{3i}	-0.216	-0.276	-0.110	-0.332	-0.246	-0.227
	Wave7	Wave8	Wave9	Wave10	Wave11	Wave12
β_{0i}	0.125	0.196	0.322	0.456	0.137	0.540
β_{1i}	-0.218	-0.158	-0.092	-0.717	-0.258	-0.772
β_{2i}	0.705	0.559	0.348	0.659	0.799	0.556
β_{3i}	-0.080	-0.232	-0.217	-0.090	-0.126	-0.218

In the simulations, we compared the following variance estimators (see deliverable D5.1) capable of handling calibration and unit non-response: bootstrap, jackknife, the Lundstrøm–Särndal formula, multiple imputation and jackknife linearisation. The effect of using a weighting scheme based on two levels of response was studied. The implemented function `SFSOCalcWeights` follows the unit non-response model A (see the `R` code listing below).

Table 2.23: Comparison of response probabilities ρ (%) of Mechanisms A and C.

GREGION	SA	STASOCIO	G13	Mechanism A ρ_{obs}	Mechanism A ρ_k	Mechanism C ρ_{obs}
1	1	-1	0	49.30	49.37	63.95
1	1	1	1	63.14	63.11	64.74
1	1	2	0	49.30	49.37	48.03
1	1	3	0	49.30	49.37	58.62
1	1	4	0	49.30	49.37	60.00
1	1	5	0	49.30	49.37	47.10
2	0	-1	0	58.02	57.75	64.29
2	0	1	1	70.43	70.56	70.66
2	0	2	0	58.02	57.75	48.88
2	0	3	0	58.02	57.75	50.00
2	0	4	0	58.02	57.75	75.00
2	0	5	0	58.02	57.75	61.62
3	1	-1	0	49.30	49.37	58.75
3	1	1	1	63.14	63.11	61.08
3	1	2	0	49.30	49.37	45.42
3	1	3	0	49.30	49.37	75.00
3	1	4	0	49.30	49.37	40.00
3	1	5	0	49.30	49.37	46.51
4	1	-1	0	49.30	49.37	53.85
4	1	1	1	63.14	63.11	62.96
4	1	2	0	49.30	49.37	52.44
4	1	3	0	49.30	49.37	66.67
4	1	4	0	49.30	49.37	57.89
4	1	5	0	49.30	49.37	45.25
5	1	-1	0	49.30	49.37	48.65
5	1	1	1	63.14	63.11	62.49
5	1	2	0	49.30	49.37	47.17
5	1	3	0	49.30	49.37	56.90
5	1	4	0	49.30	49.37	43.48
5	1	5	0	49.30	49.37	51.78
6	1	-1	0	49.30	49.37	62.50
6	1	1	1	63.14	63.11	64.63
6	1	2	0	49.30	49.37	53.85
6	1	3	0	49.30	49.37	48.39
6	1	4	0	49.30	49.37	38.46
6	1	5	0	49.30	49.37	49.15
7	0	-1	0	58.02	57.75	53.33
7	0	1	1	70.43	70.56	69.90
7	0	2	0	58.02	57.75	52.78
7	0	3	0	58.02	57.75	100.00
7	0	4	0	58.02	57.75	76.92
7	0	5	0	58.02	57.75	53.78

It was used within the bootstrap variance estimation and lead to the results presented as a boxplot in Figure 2.5 under “Bootstrap SFSO”.

Figure 2.5 presents the results for the first universe and regression imputation. The results with the other universes are similar and lead to the same conclusions. The reference value (dotted line) is larger with the SFSO weights since it takes into account the variability due to fitting a logistic regression model to calculate the weights. The bootstrap method is able to some extent to reflect this increase of variance. Not taking it into account would

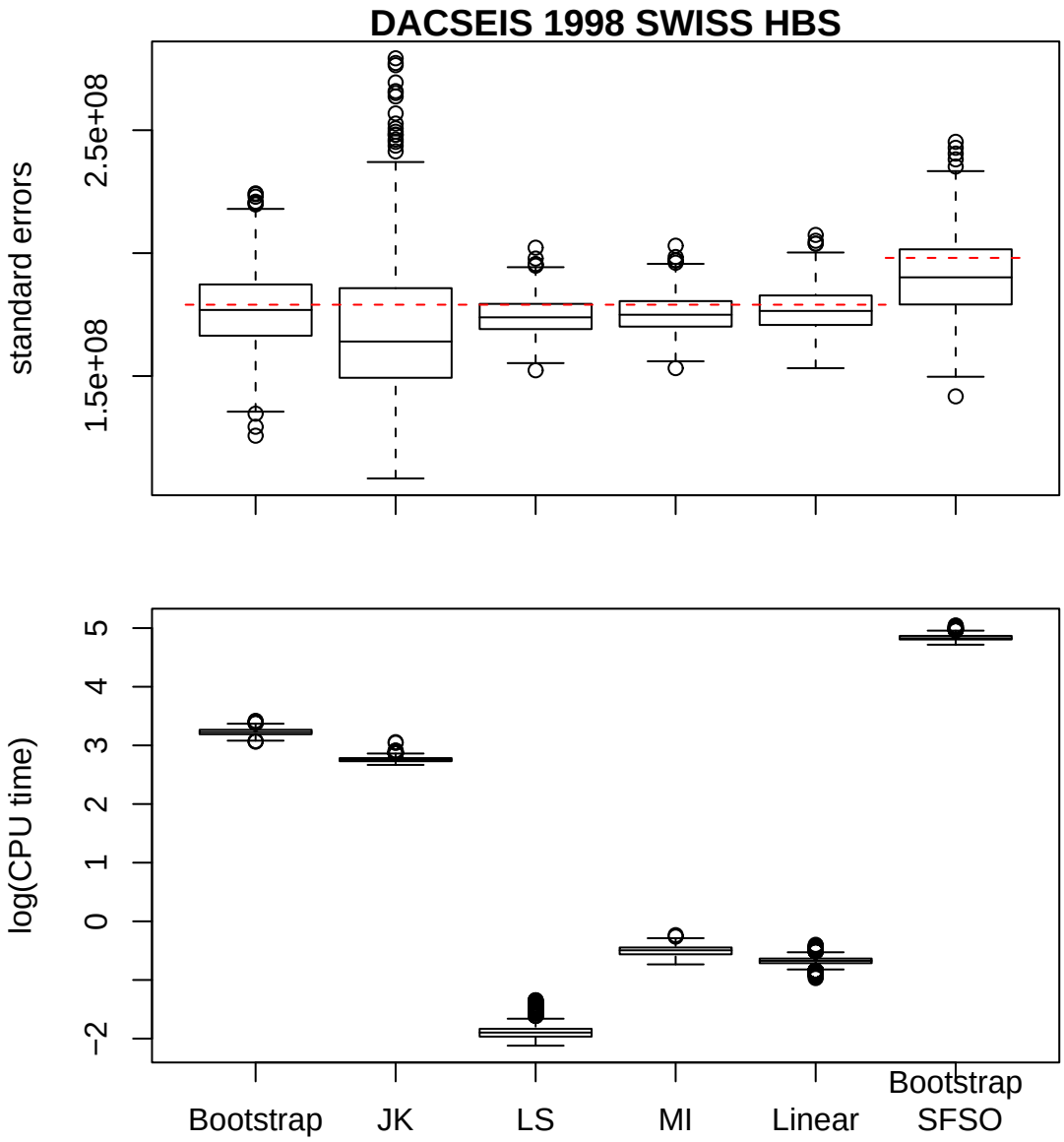

Figure 2.5: Boxplots of the 1000 estimated standard errors and reference value at the dotted line (top graph), and corresponding CPU time on a logarithmic scale. From left to right: bootstrap, jackknife (JK), Lundstrøm–Särndal formula (LS), multiple imputation (MI), jackknife linearisation (Linear), and bootstrap with SFSO weights.

result in an underestimation of the variance of the calibrated estimator. The bootstrap and the jackknife compared to the other methods seem to be more unstable, especially the jackknife. As far as computational efficiency is concerned, the bootstrap and the jackknife are many order of magnitudes slower than the other three methods. The bootstrap SFSO is even slower due to the computational time when fitting the logistic regression model.

These results are encouraging to develop a jackknife linearisation variance estimator with the SFSO weights. The technology of jackknife linearisation presented in Section 3.2 of deliverable D5.1 should be capable of use in order to efficiently estimate the variance of the Horvitz-Thompson-Estimator using the SFSO logistic regression model weights.

Appendix A

Codes of the Simulation Programmes

A.1 Main Simulation Files

Listing A.1: Main simulation file

```
##############################################################################
##                                                                          #
##       General simulation routine for the                                 #
##           DACSEIS simulation study                                       #
##                                                                          #
##############################################################################
## Include general path environment (machine dependent)

source("e:/Sim_Tabelle_MI/FinalSim/GetPathsNS2.R")

## Initialisation
## ------------------------------------------------------------------------
## R     : number of runs (generally 1000)
## BR    : bootstrap replications (Shao/Sitter)
## MIR   : multiple imputation replications (Rubin) / MIR=30,
##         MIR0 = 5, MIR1 = 10
## JKm, JK
## rr    : rr rounding for ShowSim(r)

Rest    <- 10000
R       <- 1000
BR      <- 100
BR0     <- 50
MIR     <- 30
MIR0    <- 5
MIR1    <- 15
JKm     <- 100
JKd     <- 10
rr      <- 2
```

```
eps <- 1e-005
ets <- 1e-005
maxIta <- 100

IgnoreErr <- T
TrySilent <- F

options(error = expression(CurrentWarn <<- Warn(10,F,CurrentWarn)))

## Simulation identifier (group simulation file)
## ------------------------------------------------------------------------
## ELO0001       : estimation of unemployed

SIdent  <- "GMC.1.T.ELOEB.SAL.49.3.sub0" ; source(paste
            (DATPATH,"DataSets/",SIdent,".dat",sep=""))

## Additional programmes and estimators needed
## ------------------------------------------------------------------------
## MasterSim     : include master simulation set-up

source(paste(PATH,"MasterSim.R",sep=""))

## Get universe files
## ------------------------------------------------------------------------
## SA    : small area model (group simulation file)

Univ <- GetUniverse()

Sim.Init(Univ = Univ, Univ.SA)

## Main programme
## ------------------------------------------------------------------------
## start of master simulation routine

print(paste("Simulation",SIdent,"started!",sep="␣"))
StartTime <- proc.time()[3]

#Sim.Read()
for (r in 1:Rest) {

    S <- GetSample(r)

    Sim.Est(S,r)

    print(paste("Simulation␣no:␣",r,sep=""))
    print(paste("TrueVal:␣␣␣␣␣␣␣␣␣␣␣␣",sum(Univ$ELOA1F),sep="␣"))
    Sim.Show(r)

    if (r %% 10 == 0) { # print estimated completion time
        TimeGone <- proc.time()[3] - StartTime
        if (r == R) StartTime <- proc.time()[3]
        if (r<R) {
            TotalTimeEst <- (R/r * TimeGone - TimeGone)
            print(paste("Estimated␣time␣for␣completion␣of␣Phase␣I:
␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣",round(TotalTimeEst) %/% 60, "␣minutes␣",
                round(TotalTimeEst) %% 60, "␣seconds",sep=""))
        }
```

```
        if (r>R) {
            TotalTimeEst <- ((Rest-R)/(r-R) * TimeGone - TimeGone)
            print(paste("Estimated time for completion of Phase II:
                ",round(TotalTimeEst) %/% 60, " minutes ",
                round(TotalTimeEst) %% 60, " seconds",sep=""))
        }
        Sim.Write(r)
    }

}

## Writing output
## ------------------------------------------------------------------------
## Writing simulation metadata to output

Sim.Write(Rest)

print("Simulation finalised with success!")
```

Listing A.2: General simulation control file (MasterSim.R)

```
##############################################################################
##                                                                          #
##  General simulation functions                                            #
##      for the DACSEIS simulation study                                    #
##                                                                          #
##############################################################################
## libraries to load
## ------------------------------------------------------------------------
## boot      : Davison/Hinkley/Canty bootstrap routines
## mvtnorm   : multivariate normal distribution

library(boot)
library(mvtnorm)

## Estimators to be used
## ------------------------------------------------------------------------
## HT.strat.total           : Horvitz-Thompson, stratified
## GREG.strat.total         : GREG-estimator, stratified
## Bootstrap                : Bootstrap Shao/Sitter
## SingleImpute             : Single imputation routines
## MultipleImpute           : Multiple imputation routines

source(paste(PATH,"HT.strat.total.R",sep=""))
source(paste(PATH,"HT.strat.total.cal.R",sep=""))
source(paste(PATH,"HT.raking.total.R",sep=""))
source(paste(PATH,"GREG.strat.total.R",sep=""))
source(paste(PATH,"gGREG.strat.total.R",sep=""))

source(paste(PATH,"Bootstrap.R",sep=""))
source(paste(PATH,"ddJK.orig.R",sep=""))
source(paste(PATH,"SingleImpute.R",sep=""))
source(paste(PATH,"MultipleImpute.R",sep=""))
```

```
## General inits
## ------------------------------------------------------------------------
## S         : sample matrix in simulation
## SAset     : set of small area options (+ in array!)
##                   0: no SA
##                   1: RS
##                   2: GGK
##                   3: RS x GGK(1,2,3,4+5)
##                   4: RS | 8
## DoVarEst : flag for variance estimation (more estimates
##                           may be needed within simulation)

S        <- numeric(0)
SAset    <- c("no SA", "RS", "GGK", "RSxGGK", "NUTS5")
DoVarEst <- T

## Get sample as R data.frame
## ------------------------------------------------------------------------
## S     : sample read routine (enlarge for more fed states!!!)

GetSample <- function(r) {
   S <- read.table(paste(SPATH,BL,"/",r,".stpr",sep=""),header=T)
   return(S)
}

## Get universe info as R data.frame
## ------------------------------------------------------------------------
## Univ     : sample read routine (enlarge for more fed states!!!)
## Univ.SA : sample read routine (enlarge for more fed states!!!)

GetUniverse <- function() {
    dat <- switch(UNIV,
        GMC = read.table(paste(DATPATH,"Univ/",UNIV,"/",BL,
            ".RSxGGK.total.dat",sep=""),header=T),
        DLFS = read.table(paste(DATPATH,"Univ/",UNIV,"/",BL,
            ".total.dat",sep=""),header=T)
        )
    dat <- as.list(lapply(dat,function(U) tapply(U,StratInd,sum)))
    if (sum(ifelse(StratInd>0,0,1))>0) {
        print("Stratum reduction!")
        print(StratInd)
        snum <- length(unique(StratInd))
        dat <- as.list(lapply(dat,function(U) U[2:snum]))
    }
    return(dat)
}

GetSmallArea <- function(SA) {
    if (SA == 0) return(NULL)
    U <- switch(SA,
        read.table(paste(DATPATH,BL,".RSxGGK.total.dat",sep="")),
        read.table(paste(DATPATH,BL,".RSxGGK.total.dat",sep="")),
        read.table(paste(DATPATH,BL,".SA-RSxGGK.total.dat",sep="")),
        read.table(paste(DATPATH,BL,".SA-NUTS5.total.dat",sep=""))
    )
    if (SA < 3) {
```

```
        rows <- dim(U)[1]
        cols <- dim(U)[2]
        U <- array(as.matrix(U),dim=c(5,rows/5,32))
        if (SA < 2) {
            U <- apply(U,c(2,3),sum)
        } else {
            U <- apply(U,c(1,3),sum)
        }
    }
    return(U)
}

## Functions for estimation
## ------------------------------------------------------------------------
## HT       : Horvitz-Thompson estimator (expansion, stratified)
## GREG     : GREG estimator (combined)

GetHT.strat.total <- function(dat, parms = parms, TrueVal=TrueVal,
    Yimp = NULL, V=T) {
    if (length(Yimp)>0) {
        EstObj <- HT.strat.total(Yimp,dat[,parms$Ind],,TrueVal$Nh,V)
    } else {
        EstObj <- HT.strat.total(dat[,parms$Y],dat[,parms$Ind],,
            TrueVal$Nh,V)
    }
    return(c(EstObj$HT.strat.total,EstObj$HT.strat.total.var))
}

GetHT.strat.total.SI <- function(dat, parms = parms, TrueVal=TrueVal,
        SIMeth = SIMeth, V = T) {
    Yimp <- SIMeth(dat,parms)
    EstObj <- HT.strat.total(Yimp,dat[,parms$Ind],,TrueVal$Nh,V)
    return(c(EstObj$HT.strat.total,EstObj$HT.strat.total.var))
}

GetHTcal.strat.total <- function(dat, parms = parms, TrueVal=TrueVal) {
    marginals <- apply(TrueVal$X,2,sum)
    z <- (!is.na(dat[,parms$NR]))
    Y <- GetBySampUnits(as.matrix(dat[z,parms$Y]),dat[z,parms$SU])
    X <- GetBySampUnits(as.matrix(dat[z,parms$X]),dat[z,parms$SU])
    Ind <- GetIndBySampUnits(dat[z,],parms)
    # N.h for German Microcensus
    N.h <- 100 * tapply(Y,Ind,length)

    EstObj <- HT.cal.total(Y,X,rep(100,length(Y)),Ind,marginals,TRUE)
    return(c(EstObj$theta.hat,EstObj$v.est))
}

GetHT.raking.total <- function(dat, parms = parms, TrueVal=TrueVal,
        V = T, WOR=T) {
    EstObj <- HT.raking.total(dat[,parms$NR],as.matrix(dat[,parms$X]),
        dat[,parms$Ind],TrueVal$X,dat[,parms$IIP],V,WOR)
    return(c(EstObj$PEst.total,EstObj$VEst.total))
}

GetHT.raking.JK.total <- function(dat, parms = parms, TrueVal=TrueVal,
        V = T, WOR=T) {
```

```
    EstObj <- HT.raking.JK.total(dat[,parms$NR],as.matrix(dat[,parms$X]),
        dat[,parms$Ind],TrueVal$X,dat[,parms$IIP],V,WOR)
    return(c(EstObj$PEst.total,EstObj$VEst.total))
}

GetHT.GREG.total <- function(dat, parms = parms, TrueVal=TrueVal,
        V = T, WOR=T) {
    EstObj <- HT.GREG.total(dat[,parms$NR],as.matrix(dat[,parms$X]),
        dat[,parms$Ind],TrueVal$X,dat[,parms$IIP],V,WOR)
    return(c(EstObj$PEst.total,EstObj$VEst.total))
}

GetHT.GREG.JK.total <- function(dat, parms = parms, TrueVal=TrueVal,
        V = T, WOR=T) {
    EstObj <- HT.GREG.JK.total(dat[,parms$NR],as.matrix(dat[,parms$X]),
        dat[,parms$Ind],TrueVal$X,dat[,parms$IIP],V,WOR)
    return(c(EstObj$PEst.total,EstObj$VEst.total))
}

GetHT.MLraking.total <- function(dat, parms = parms, TrueVal=TrueVal,
        V = T, WOR=T) {
    EstObj <- HT.MLraking.total(dat[,parms$NR],as.matrix(dat[,parms$X]),
        dat[,parms$Ind],TrueVal$X,dat[,parms$IIP],V,WOR)
    return(c(EstObj$PEst.total,EstObj$VEst.total))
}

GetHT.MLraking.JK.total <- function(dat, parms = parms, TrueVal=TrueVal,
        V = T, WOR=T) {
    EstObj <- HT.MLraking.JK.total(dat[,parms$NR],as.matrix(
        dat[,parms$X]),dat[,parms$Ind],TrueVal$X,dat[,parms$IIP],V,WOR)
    return(c(EstObj$PEst.total,EstObj$VEst.total))
}

GetHT.strat.total.cal <- function(dat = dat, parms = parms,
        TrueVal=TrueVal, V) {
    EstObj <- HT.strat.total.cal(dat[,parms$NR],cbind(1,dat[,parms$X]),
        dat[,parms$IIP],dat[,parms$Ind],w=1,cbind(TrueVal$N,TrueVal$X),
        TrueVal$N, V)
    return(c(EstObj[1],EstObj[2]))
}

GetGREG.strat.total <- function(dat, parms = parms, TrueVal = TrueVal,
        Yimp = NULL, V=T) {
    if (length(Yimp)>0) {
        Y <- Yimp
    } else {
        Y <- dat[,parms$Y]
    }
    EstObj <- GREG.strat.total(Y,as.matrix(dat[,parms$X]),
        dat[,parms$Ind],,,TrueVal$X,TrueVal$N,V)
    return(c(EstObj$GREG.strat.total,EstObj$GREG.strat.total.var))
}

GetgGREG.strat.total <- function(dat, parms = parms, TrueVal = TrueVal,
        Yimp = NULL, V=T) {
    if (length(Yimp)>0) {
        Y <- Yimp
```

```
    } else {
        Y <- dat[,parms$Y]
    }
    EstObj <- gGREG.strat.total(y=Y,X=cbind(1,dat[,parms$X]),
        IIP=dat[,parms$IIP],stratind=dat[,parms$Ind],Xtot=TrueVal$X,
        N.h=TrueVal$N,V=V)
    return(c(EstObj$GREG.strat.total,EstObj$GREG.strat.total.var))
}

GetGREG.strat.total.SI <- function(dat, parms = parms, TrueVal = TrueVal,
        SIMeth = SIMeth, V = T) {
    Yimp <- SIMeth(dat,parms)
    EstObj <- GREG.strat.total(Yimp,as.matrix(dat[,parms$X]),
        dat[,parms$Ind],,,TrueVal$X,TrueVal$N,V)
    return(c(EstObj$GREG.strat.total,EstObj$GREG.strat.total.var))
}

GetgGREG.strat.total.SI <- function(dat, parms = parms, TrueVal = TrueVal,
        SIMeth = SIMeth, V = T) {
    Yimp <- SIMeth(dat,parms)
    EstObj <- gGREG.strat.total(y=Yimp,X=cbind(1,dat[,parms$X]),
        IIP=dat[,parms$IIP],stratind=dat[,parms$Ind],Xtot=TrueVal$X,
        N.h=TrueVal$N,V=V)
    return(c(EstObj$GREG.strat.total,EstObj$GREG.strat.total.var))
}

GetGREG.MZ.total <- function(dat, parms = parms, TrueVal = TrueVal,
        Yimp = NULL, V=T) {
    if (length(Yimp)>0) {
        Y <- GetBySampUnits(as.matrix(Yimp),dat[,parms$SU])
    } else {
        Y <- GetBySampUnits(as.matrix(dat[,parms$Y]),dat[,parms$SU])
    }
    X <- GetBySampUnits(as.matrix(dat[,parms$X]),dat[,parms$SU])
    Ind <- GetIndBySampUnits(dat,parms)
    # N.h for German Microcensus
    N.h <- 100 * tapply(Y,Ind,length)
    EstObj <- GREG.strat.total(Y,X,Ind,,,TrueVal$X,N.h,V)
    return(c(EstObj$GREG.strat.total,EstObj$GREG.strat.total.var))
}

GetGREG.MZ.total.SI <- function(dat, parms = parms, TrueVal = TrueVal,
        SIMeth = SIMeth, V = T) {
    Yimp <- SIMeth(dat,parms)
    Y <- GetBySampUnits(as.matrix(Yimp),dat[,parms$SU])
    X <- GetBySampUnits(as.matrix(dat[,parms$X]),dat[,parms$SU])
    Ind <- GetIndBySampUnits(dat,parms)
    # N.h for German Microcensus
    N.h <- 100 * tapply(Y,Ind,length)
    EstObj <- GREG.strat.total(Y,X,Ind,,,TrueVal$X,N.h,V)
    return(c(EstObj$GREG.strat.total,EstObj$GREG.strat.total.var))
}

GetgGREG.MZ.total.SI <- function(dat, parms = parms, TrueVal = TrueVal,
        SIMeth = SIMeth, V = T) {
    Yimp <- SIMeth(dat,parms)
    Y <- GetBySampUnits(as.matrix(Yimp),dat[,parms$SU])
```

```
    X <- GetBySampUnits(as.matrix(dat[,parms$X]),dat[,parms$SU])
    Ind <- GetIndBySampUnits(dat,parms)
    # N.h for German Microcensus
    N.h <- 100 * tapply(Y,Ind,length)
    EstObj <- gGREG.strat.total(y=Y,X=cbind(1,X),
        IIP=as.vector(rep(100,length(Y))),stratind=Ind,Xtot=TrueVal$X,
        N.h=N.h,V=V)
    return(c(EstObj$GREG.strat.total,EstObj$GREG.strat.total.var))
}


## Sampling untis
## ------------------------------------------------------------------
## definition of sampling units for estimation
## in general individual
## if needed, then AWB

SampUnits <- function(S) {
    Strata <- 5*(S$RS-min(S$RS))+S$GGK
    Zone <- ((S$AWB-1) %/% 100)
    Ind <- tapply(S$AWB,Strata,length)
    AddVal <- c(0,cumsum(tapply(((S$AWB-1) %/% 100),(S$RS-min(S$RS))*
        5+S$GGK,function(s) length(unique(s))))[1:length(unique(Strata))]
    Zone <- Zone + rep(AddVal,Ind)
    return(Zone)
}

GetBySampUnits <- function(X,ZI) {
    return(apply(X,2,function(x) tapply(x,ZI,sum)))
}

GetIndBySampUnits <- function(dat,parms) {
    return(tapply(dat[,parms$Ind],dat[,parms$SU],function(s) s[1]))
}

## General output
## ------------------------------------------------------------------
## DACSEIS  : for DACSEIS output
## EURAREA  : for EURAREA output
## group writing should be handled in extra file

Save.output <- function(Est, VarEst, Time, Warn, TrueVal,MasterFile,r) {
   write.table(Est[1:r],file=paste(OutPATH,MasterFile,"/pest.dat",sep=""))
   write.table(VarEst[1:min(r,R)],file=paste(OutPATH,MasterFile,
        "/vest.dat",sep=""))
   write.table(Time[1:min(r,R)],file=paste(OutPATH,MasterFile,"/time.dat",
        sep=""))
   write.table(Warn[1:r],file=paste(OutPATH,MasterFile,"/warn.dat",
        sep=""))
   tv <- c(sum(TrueVal$Y),sum(TrueVal$N))
   names(tv) <- c("Y","N")
   write.table(t(tv) ,file=paste(OutPATH,MasterFile,"/trueval.dat",
        sep=""),col.names=T)
}


## Specialised funtioncs
```

```
## ------------------------------------------------------------------------
## self declaring definitions

MS <- max(StratInd)
utapply <- function(x) {
    return(tapply(c(x,1:MS),c(x,1:MS),length) - 1)
}

## Variables of interest
## ------------------------------------------------------------------------
## self declaring definitions

NRclass <- function(S) {
    help <- ifelse(S$NAT>0,6,0) +
        ifelse(S$HS>1,ifelse(S$HS>2,6,5),
            ifelse(S$SEX==0,ifelse(S$AGE<60,1,2),ifelse(S$AGE<60,3,4)))
    return(help)
}

AGEclass <- function(S) {
    help <- ifelse(S$AGE<14,0,
                ifelse(S$AGE<25,1,
                ifelse(S$AGE<35,2,
                ifelse(S$AGE<45,3,
                ifelse(S$AGE<55,4,
                ifelse(S$AGE<65,5,6))))))
    return(help)
}

AGEclassA <- function(S) {
    h <- ifelse(S$AGE<=20,0,
                ifelse(S$AGE<=30,1,
                ifelse(S$AGE<=40,2,
                ifelse(S$AGE<=50,3,
                ifelse(S$AGE<=65,4,0)))))
    return(h)
}

STRATA <- function(sind) {
    return(StratInd[sind])
}

Unemp <- function(S) {                  # unemployed
    return(ifelse(S$ELO==1,1,0))
}

UnempU25 <- function(S) {               # unemployed under 25
    return(ifelse(S$ELO==1 & S$AGE<25,1,0))
}

UnempU25ng <- function(S) {             # unemployed under 25, non-German
    return(ifelse(S$ELO==1 & S$AGE<25 & S$NAT>0,1,0))
}

CurrentWarn <- 0
Warn <- function(error,stat,oldWarn) {
    if (!stat) {
```

```
        a <- (oldWarn %% 2^(error-1)) + 2^(error-1) + 2^error *
            (oldWarn %/% 2^error)
    } else {
        a <- oldWarn
    }
    return(a)
}
```

Listing A.3: Survey specification file

```
##############################################################################
##                                                                          #
##  General simulation dataset:                                             #
##  GMC.1.T.ELOEB.SAL.49.3.sub0                                             #
##                                                                          #
##############################################################################
## Variables in use:
##
## Y:        EstVar            :: ELOEB
## X:        AuxVar            :: ALO, DM, DF, AM
## Ind       Strata            :: RS x GGK
## NR        Nonresponse       :: 25% non-response
## Xnr       NR AuxVar         :: ALOEB
## SU        Sampling Units    :: AWB (by RS and GGK)
## SA        Small Areas       :: no
##
##############################################################################
##                                                                          #
## Additional constants:                                                    #
##                                                                          #
##############################################################################
## BL        : Federal state for simulation
## SA        : Small area problem (cf. MasterSim)

UNIV    <- "GMC"
BL      <- "SAL"
SA      <- 0

StratInd <- 1:15


##############################################################################
##                                                                          #
## Metadata for GMC.1.T.ELOEB.SAL.49.3.sub0:                                #
##                                                                          #
##############################################################################
## .dat      : data matrix (if group data used, NULL)
## .parms    : parameter vector


GMC.1.T.ELOEB.SAL.49.3.sub0.univ        <- NULL
GMC.1.T.ELOEB.SAL.49.3.sub0.smallarea   <- NULL
GMC.1.T.ELOEB.SAL.49.3.sub0.dat         <- NULL
```

```
GMC.1.T.ELOEB.SAL.49.3.sub0.parms        <- NULL
GMC.1.T.ELOEB.SAL.49.3.sub0.names        <- NULL

GMC.1.T.ELOEB.SAL.49.3.sub0.Univ <- function(U) {
    Y <- U$ELOA1F + U$ELOA2F + U$ELOA3F + U$ELOA4F + U$ELOA1M +
         U$ELOA2M + U$ELOA3M + U$ELOA4M
    X <- cbind(U$ALO,U$DM,U$DF,U$AM)
    Nh <- U$N
    GMC.1.T.ELOEB.SAL.49.3.sub0.univ <<- list(Y = Y, X = X, Nh = Nh)
}

GMC.1.T.ELOEB.SAL.49.3.sub0.Sample <- function(S) {
    Strata <- STRATA(5*(S$RS-min(S$RS))+S$GGK)
    S <- S[Strata>0,]
    Strata <- Strata[Strata>0]

#    Zone <- SampUnits(S)
    dat <- cbind(
        ifelse(S$ELO==1 & S$AGE>13,1,0),
        S$ALO,
        ifelse(S$NAT==0 & S$SEX==0,1,0),
        ifelse(S$NAT==0 & S$SEX==1,1,0),
        ifelse(S$NAT>0 & S$SEX==0,1,0),
        ifelse(S$ALO==1 & S$AGE>13,1,0),
        S$NR25,
        Strata,
        100)

    GMC.1.T.ELOEB.SAL.49.3.sub0.names <<- c("UnempEB", "ALO",
        "Na0Se0", "Na0Se1", "Na1Se0", "ALOEB", "NR25", "Strata",
        "IIP")

    Y       <- c(1)
    X       <- c(2:5)
    Ind     <- c(8)
    NR      <- c(7)
    Xnr     <- c(6)
    Xnr1    <- c(6)
    SU      <- c(8)
    SA      <- NULL
    IIP      <- c(9)

    # set NR vector to either NR or y-value
    dat[,NR] <- ifelse(dat[,NR]==0,NA,dat[,Y])

    GMC.1.T.ELOEB.SAL.49.3.sub0.dat <<- dat
    GMC.1.T.ELOEB.SAL.49.3.sub0.parms <<- list(Y = Y, X = X,
        Ind = Ind, NR = NR, Xnr = Xnr, Xnr1 = Xnr1, SU = SU,
        SA = SA, IIP = IIP)
}
```

Listing A.4: Group information file

```
##############################################################################
##                                                                          #
## Estimation Metadata  :                                                   #
##                                                                          #
## Survey specifiation  :   GMC.1.T.ELOEB.SAL.49.3.sub0                     #
##                                                                          #
## Estimator / Variance estimator :                                         #
##       :   GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct              #
##       :   GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct            #
##                                                                          #
## NR correction        :   no                                              #
## weighting            :   no                                              #
##                                                                          #
##############################################################################
## SurveySpec
source(paste(DATPATH,"SurveySpec/GMC.1.T.ELOEB.SAL.49.3.sub0.dat",sep=""))

## Estimators of interest

source(paste(DATPATH,"DataSets/GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.
    total.direct.dat",sep=""))
source(paste(DATPATH,"DataSets/GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.
    total.direct.dat",sep=""))

## General routines

Sim.Init <- function(Univ = Univ, Univ.SA = Univ.SA) {
    GMC.1.T.ELOEB.SAL.49.3.sub0.Univ(Univ)
}

Sim.Est <- function(S,r) {

    GMC.1.T.ELOEB.SAL.49.3.sub0.Sample(S)

    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.Est(r)
      print("HT.strat.total.direct done")
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.Est(r)
      print("GREG.strat.total.direct done")
}


Sim.Write <- function(r) {

    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.Write(r)
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.Write(r)
}

Sim.Read <- function() {

    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.Read()
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.Read()

}
```

```
Sim.Show <- function(r) {

    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.Show(r)
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.Show(r)

}
```

Listing A.5: Estimator information file

```
##############################################################################
##                                                                          #
## Estimation Metadata  :                                                   #
##      GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.dat               #
##                                                                          #
## Survey specifiation  :   GMC.1.T.ELOEB.SAL.49.3.sub0                     #
## Estimator            :   ESTS                                            #
## Variance Estimator   :   VESTS                                           #
## NR correction        :   NRCORR                                          #
## weighting            :   WEIGTHING                                       #
##                                                                          #
##############################################################################

GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.p.est
    <- array(NA,dim=c(Rest,1))
GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.v.est
    <- array(NA,dim=c(R,1))
GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.time
    <- array(NA,dim=c(R,1))
GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.warn
    <- array(0,dim=c(R,1))


GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.Est <- function(r) {

 stime <- proc.time()[1]
 CurrentWarn <<- 0
 if (r>R) {
    if (IgnoreErr) {
    try({
    EstRes <<- GetHT.MLraking.JK.total(
        GMC.1.T.ELOEB.SAL.49.3.sub0.dat,
        GMC.1.T.ELOEB.SAL.49.3.sub0.parms,
        GMC.1.T.ELOEB.SAL.49.3.sub0.univ, V = F, WOR=F)
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.p.est[r]
        <<- EstRes[1]
    }, silent=TrySilent)
    } else {
    EstRes <<- GetHT.MLraking.JK.total(
        GMC.1.T.ELOEB.SAL.49.3.sub0.dat,
        GMC.1.T.ELOEB.SAL.49.3.sub0.parms,
        GMC.1.T.ELOEB.SAL.49.3.sub0.univ, V = F, WOR=F)
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.p.est[r]
        <<- EstRes[1]
```

```
    }
 } else {
    if (IgnoreErr) {
    try({
    EstRes <<- GetHT.MLraking.JK.total(
        GMC.1.T.ELOEB.SAL.49.3.sub0.dat,
        GMC.1.T.ELOEB.SAL.49.3.sub0.parms,
        GMC.1.T.ELOEB.SAL.49.3.sub0.univ, V = T, WOR=F)
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.p.est[r]
        <<- EstRes[1]
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.v.est[r]
        <<- EstRes[2]
    }, silent=TrySilent)
    } else {
    EstRes <<- GetHT.MLraking.JK.total(
        GMC.1.T.ELOEB.SAL.49.3.sub0.dat,
        GMC.1.T.ELOEB.SAL.49.3.sub0.parms,
        GMC.1.T.ELOEB.SAL.49.3.sub0.univ, V = T, WOR=F)
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.p.est[r]
        <<- EstRes[1]
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.v.est[r]
        <<- EstRes[2]
    }
 }
 etime <- proc.time()[1]
GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.warn[r]
    <<- CurrentWarn
GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.time[r]
    <<- etime - stime
}


GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.Write <- function(r) {
    Save.output(
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.p.est,
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.v.est,
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.time,
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.warn,
    GMC.1.T.ELOEB.SAL.49.3.sub0.univ,
    "GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct",r)
}

GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.Read <- function() {
    MasterFile <- "GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct"
    h <- read.table(file=paste(OutPATH,MasterFile,"/pest.dat",sep=""))$x
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.p.est[1:length(h)]
        <<- h
    h <- read.table(file=paste(OutPATH,MasterFile,"/vest.dat",sep=""))$x
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.v.est[1:length(h)]
        <<- h
    h <- read.table(file=paste(OutPATH,MasterFile,"/time.dat",sep=""))$x
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.time[1:length(h)]
        <<- h
    h <- read.table(file=paste(OutPATH,MasterFile,"/warn.dat",sep=""))$x
    GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.warn[1:length(h)]
        <<- h
```

```
}


GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.Show <- function(r) {

    print(paste("HTMLrakJK direct:   "),
    round(mean(GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.
        p.est[1:r],na.rm=T),rr), round(var(GMC.1.T.ELOEB.SAL.49.3.sub0.HT.
        strat.total.direct.p.est[1:r],na.rm=T),rr),
        round(mean(GMC.1.T.ELOEB.SAL.49.3.sub0.HT.strat.total.direct.
        v.est[1:min(r,R)],na.rm=T),rr),"  "))

}
```

Listing A.6: Estimator information file

```
##############################################################################
##                                                                          #
## Estimation Metadata  :                                                   #
##      GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.dat             #
##                                                                          #
## Survey specifiation  :   GMC.1.T.ELOEB.SAL.49.3.sub0                     #
## Estimator            :   ESTS                                            #
## Variance Estimator   :   VESTS                                           #
## NR correction        :   NRCORR                                          #
## weighting            :   WEIGTHING                                       #
##                                                                          #
##############################################################################

GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.p.est <-
    array(NA,dim=c(Rest,1))
GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.v.est <-
    array(NA,dim=c(R,1))
GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.time  <-
    array(NA,dim=c(R,1))
GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.warn  <-
    array(0,dim=c(R,1))


GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.Est <- function(r) {

    stime <- proc.time()[1]
    CurrentWarn <<- 0
    if (r>R) {
        if (IgnoreErr) {
            try({
                EstRes <<- GetHT.MLraking.JK.total(
                    GMC.1.T.ELOEB.SAL.49.3.sub0.dat,
                    GMC.1.T.ELOEB.SAL.49.3.sub0.parms,
                    GMC.1.T.ELOEB.SAL.49.3.sub0.univ, V = F, WOR=F)
                GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.
                    p.est[r] <<- EstRes[1]
            }, silent=TrySilent)
```

```
        } else {
            EstRes <<- GetHT.MLraking.JK.total(
                GMC.1.T.ELOEB.SAL.49.3.sub0.dat,
                GMC.1.T.ELOEB.SAL.49.3.sub0.parms,
                GMC.1.T.ELOEB.SAL.49.3.sub0.univ, V = F, WOR=F)
            GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.
                p.est[r] <<- EstRes[1]
        }
    } else {
        if (IgnoreErr) {
            try({
                EstRes <<- GetHT.MLraking.JK.total(
                    GMC.1.T.ELOEB.SAL.49.3.sub0.dat,
                    GMC.1.T.ELOEB.SAL.49.3.sub0.parms,
                    GMC.1.T.ELOEB.SAL.49.3.sub0.univ, V = T, WOR=F)
                GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.
                    p.est[r] <<- EstRes[1]
                GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.
                    v.est[r] <<- EstRes[2]
        }, silent=TrySilent)
        } else {
            EstRes <<- GetHT.MLraking.JK.total(
                GMC.1.T.ELOEB.SAL.49.3.sub0.dat,
                GMC.1.T.ELOEB.SAL.49.3.sub0.parms,
                GMC.1.T.ELOEB.SAL.49.3.sub0.univ, V = T, WOR=F)
            GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.p.est[r]
                <<- EstRes[1]
            GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.v.est[r]
                <<- EstRes[2]
        }
    }
    etime <- proc.time()[1]
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.warn[r]
        <<- CurrentWarn
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.time[r]
        <<- etime - stime
}


GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.Write <- function(r) {
    Save.output(
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.p.est,
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.v.est,
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.time,
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.warn,
    GMC.1.T.ELOEB.SAL.49.3.sub0.univ,
    "GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct",r)
}

GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.Read <- function() {
    MasterFile <- "GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct"
    h <- read.table(file=paste(OutPATH, MasterFile,"/pest.dat",sep=""))$x
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.p.est[1:length(h)]
        <<- h
    h <- read.table(file=paste(OutPATH, MasterFile,"/vest.dat",sep=""))$x
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.v.est[1:length(h)]
        <<- h
```

```
    h <- read.table(file=paste(OutPATH,MasterFile,"/time.dat",sep=""))$x
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.time[1:length(h)]
        <<- h
    h <- read.table(file=paste(OutPATH,MasterFile,"/warn.dat",sep=""))$x
    GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.warn[1:length(h)]
        <<- h
}


GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.Show <- function(r) {

    print(paste("HTMLrakJK direct:   ",
    round(mean(GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.#
        p.est[1:r],na.rm=T),rr), round(var(GMC.1.T.ELOEB.SAL.49.3.sub0.
        GREG.strat.total.direct.p.est[1:r],na.rm=T),rr),
        round(mean(GMC.1.T.ELOEB.SAL.49.3.sub0.GREG.strat.total.direct.
        v.est[1:min(r,R)],na.rm=T),rr),"  "))

}
```

Listing A.7: Paths definition file

```
##############################################################################
##                                                                          #
##  General file of paths for the DACSEIS simulation study                  #
##                                                                          #
##############################################################################
## MainPATH      : general path
## PATH          : path to programmes
## DATPATH       : path to additional data files
## SPATH         : path to samples
## OutPATH       : path to save results in metadata format
## ZS            : sampling routine (only GMC)

MainPATH    <- "/home/muennich/FinalSim/"
ZS          <- "GMC/"

PATH        <- paste(MainPATH,"FinalSim/",sep="")
DATPATH     <- paste(MainPATH,"SimData/",sep="")
SPATH       <- paste("/simdat1/Samples/",ZS,sep="")
OutPATH     <- paste(MainPATH,"Output/",sep="")
```

A.2 Estimators

Listing A.8: HT estimator

```
##############################################################################
##                                                                          #
##  Weighted HT estimator                                                   #
##                                                                          #
##############################################################################
##  Variables:                                                              #
##                                                                          #
##  y        values of variable of interest                                 #
##  sind     stratum index vector                                           #
##  w        weights (eg for compensating NR)                               #
##  cw       additional weights (cf Särndal)                                #
##  pi       inclusion probabilites (1st order)                             #
##  N        number of units in universe                                    #
##  V        flag for including variance estimate (F -> NA)                 #
##                                                                          #
##  Variables in programme:                                                 #
##                                                                          #
##  n        number of units in sample                                      #
##  ll       number of regression coefficients (covar)                      #
##  g        Särndal g-weight with reg. coef. incl. prob                    #
##               pi and weights w and cw                                    #
##                                                                          #
##  Return values:                                                          #
##                                                                          #
##  HT.total       : HT estimate                                            #
##  HT.total.Var   : HT direct variance estimate                            #
##                                                                          #
##############################################################################


HT.strat.total <- function(y, sind, w = 0, N.h, V = FALSE) {

    StratNR <- unique(sind)
    H <- length(StratNR)
    n.h <- tapply(y,sind,length)

    if (sum(w) == 0) w <- rep(1,H)

    if (H < length(N.h)) {
        CurrentWarn <<- Warn(3,F,CurrentWarn)
        N.h <- N.h[StratNR]
    }

    HT <- sum(w*N.h/n.h * tapply(y,sind,sum))

    HTvar <- ifelse(V, sum(w^2*N.h^2 * tapply(y,sind,var)
                                / n.h * (1-n.h/N.h)), NA)

    list(HT.strat.total = HT, HT.strat.total.var = HTvar)

}
```

Listing A.9: GREG estimator

```
#############################################################################
##                                                                         #
##  GREG total estimator                                                   #
##                                                                         #
#############################################################################
##  Variables:                                                             #
##                                                                         #
##  y        values of variable of interest                                #
##  x        matrix of aux variables                                       #
##  w        weights (eg for compensating NR)                              #
##  pp       inclusion porbabilites (1st order)                            #
##  Xtot     matrix of true values                                         #
##  N        number of units in universe                                   #
##  W        flag for weighted regression                                  #
##  V        flag for including variance estimate (F -> NA)                #
##                                                                         #
##  Variables in programme:                                                #
##                                                                         #
##  n        number of units in sample                                     #
##  H        number of strata                                              #
##  ll       number of regression coefficients (covar)                     #
##  B        regression coeff. from Y on X with incl. prob                 #
##           pi and weights w                                              #
##                                                                         #
##  Return values:                                                         #
##                                                                         #
##  GREG.strat.total       : GREG estimate                                 #
##  GREG.strat.total.var   : linearised variance estimate from GREG        #
##                                                                         #
#############################################################################

GREG.strat.total <- function(y, x, sind, w=0, W=F, Xtot, N.h, V=F) {
    x <- as.matrix(x)
    StratNR <- sort(unique(sind))
    H <- length(StratNR)
    n.h <- tapply(y,sind,length)
    if (H < length(N.h)) {
        CurrentWarn <<- Warn(3,F,CurrentWarn)
        N.h <- N.h[StratNR]
    }
    n <- sum(n.h)
    ll <- dim(x)[2]
    Xtot <- array(Xtot,dim=c(H,ll))

    if (W) {
        w <- rep(1,n)
        pp <- rep(n.h/N.h,n.h)
        lreg <- lm(y ~ x, weights = w/pp)
    } else {
        lreg <- lm(y~x)
    }

        B <- matrix(coef(lreg)[2:(ll+1)], ncol=1)
        GREG <- sum(matrix(N.h/n.h * tapply(y,sind,sum),ncol=1)
```

```
                + (Xtot - matrix(rep(N.h/n.h,ll),ncol=ll)
                *apply(x,2,function(Z) tapply(Z,sind,sum))) %*% B)

    GREGvar <- ifelse(V, sum(N.h^2 * tapply(lreg$residuals,sind,var)
                                    / n.h * (1-n.h/N.h)), NA)

    list(GREG.strat.total = GREG, GREG.strat.total.var = GREGvar)

}
```

Listing A.10: Bootstrap methods

```
#############################################################################
##                                                                         #
##   Bootstrap estimates with Shao / Sitter                                #
##                                                                         #
#############################################################################

Boot.ShaoSitter.inner <- function(dat, boot.ind, EstMeth, SIMeth,
                                  TrueVal, parms, Nh) {
    Yimp <- SIMeth(dat[boot.ind,],parms)
    EstObj <- EstMeth(dat[boot.ind,], parms=parms, TrueVal=TrueVal,
                      Yimp=Yimp)
    return(EstObj[1])
}

Boot.ShaoSitter.modified.inner <- function(dat, boot.ind, EstMeth,
                        SIMeth, TrueVal, parms, Nh, ModParms) {
    Yimp <- SIMeth(dat[boot.ind,],parms)
    EstObj <- EstMeth(dat[boot.ind,], parms=parms, TrueVal=TrueVal,
                      Yimp=Yimp)
    return(EstObj[1])
}

Boot.ShaoSitter <- function(dat, BR, EstMeth, SIMeth, TrueVal,
                        orig = T, parms = parms, Nh = Nh) {
    if (orig) {
        outBoot <- boot(dat, Boot.ShaoSitter.inner, BR,
                   strata = dat[,parms$Ind], EstMeth = EstMeth,
                   SIMeth = SIMeth, TrueVal = TrueVal, parms=parms,
                   Nh = Nh)
    } else {
        ModParms <- SIMeth(dat[,c(parms$Xnr,parms$NR)])
        outBoot <- boot(dat, Boot.ShaoSitter.modified.inner, BR,
                   strata = dat[,parms$Ind], EstMeth = EstMeth,
                   SIMeth = SIMeth, TrueVal = TrueVal, parms=parms,
                   Nh = Nh, ModParms = ModParms)
    }
    return(c(var(outBoot$t),var(outBoot$t[1:BR0])))
}

Boot.varest.inner <- function(dat, boot.ind, EstMeth, TrueVal,
                              parms, Nh, Yimp = NULL) {
```

```
    EstObj <- EstMeth(dat[boot.ind,], parms=parms, TrueVal=TrueVal,
              Nh=Nh, Yimp=Yimp)
    return(EstObj[1])
}

Boot.varest <- function(dat, BR, EstMeth, TrueVal = TrueVal,
                        parms = parms, Nh = Nh, Yimp = NULL) {
    outBoot <- boot(dat, Boot.varest.inner, BR, strata =
               dat[,parms$Ind], EstMeth = EstMeth, TrueVal=TrueVal,
               parms = parms, Nh = Nh, Yimp = Yimp)
    return(c(outBoot$t0,var(outBoot$t)))
}
```

Listing A.11: Single Imputation methods

```
##############################################################################
##                                                                          #
##   Single Imputation routines                                             #
##                                                                          #
##############################################################################

#
# Bootstrap imputation routine
#

source(paste(PATH,"SILFS1.R",sep=""))
source(paste(PATH,"SILFS2.R",sep=""))
source(paste(PATH,"SILFS3.R",sep=""))
source(paste(PATH,"SILFS4.R",sep=""))
source(paste(PATH,"SILINREG.R",sep=""))
source(paste(PATH,"SIRATIO.R",sep=""))

SI.logit <- function(dat,parms)
{
    n <- dim(dat)[1]
    l <- length(parms$Xnr)
    logobs <- summary(testglm <- glm(dat[,parms$NR]~dat[,parms$Xnr],
              family = binomial(link = logit)), correlation=T,
              na.action="na.omit")

    if (!testglm$converged) {
        CurrentWarn <<- Warn(1,testglm$converged,CurrentWarn)
        logobs <- summary(testglm <- glm(dat[,parms$NR] ~
                  dat[,parms$Xnr], family = binomial(link = logit),
                  maxit=100), correlation=T, na.action="na.omit")
        CurrentWarn <<- Warn(2,testglm$converged,CurrentWarn)
    }

    betadachobs     <- logobs$coefficients[, 1]
    se.betadachobs  <- logobs$coefficients[, 2]
    cor.betadachobs <- logobs$correlation
    cov.betadachobs <- cor.betadachobs *
                       (se.betadachobs %*% t(se.betadachobs))
```

```
    beta <- as.vector(rmvnorm(1, as.vector(betadachobs),
                                    cov.betadachobs))
    prob <- 1/(1 + exp( - (beta[1] + matrix(dat[,parms$Xnr])
                                        %*% beta[2:1])))
    Ymis <- ifelse(runif(n) > prob, 0, 1)
    Yimp <- ifelse(is.na(dat[,parms$NR]), Ymis, dat[,parms$Xnr])
    return(Yimp)
}

EmptySI <- function(dat,parms) {
    return(dat[,parms$Y])
}

SI.lfs1 <- function(dat,parms) {
    out.impute<- SILFS1(dat[,parms$Xnr1],dat[,parms$NR])
    Yimp<-out.impute[,dim(out.impute)[2]]
    return(Yimp)
}

SI.lfs2 <- function(dat,parms) {
    out.impute<- SILFS2(as.matrix(dat[,parms$Xnr]),dat[,parms$NR])
    Yimp<-out.impute[,dim(out.impute)[2]]
    return(Yimp)
}

SI.lfs3 <- function(dat,parms) {
    out.impute<- SILFS3(as.matrix(dat[,parms$Xnr]),dat[,parms$NR])
    Yimp<-out.impute[,dim(out.impute)[2]]
    return(Yimp)
}

SI.lfs4 <- function(dat,parms) {
    out.impute<- SILFS4(as.matrix(dat[,parms$Xnr]),dat[,parms$NR])
    Yimp<-out.impute[,dim(out.impute)[2]]
    return(Yimp)
}

SI.linreg <- function(dat,parms) {
    out.impute<- SIlinreg(as.matrix(dat[,parms$Xnr]),dat[,parms$NR])
    Yimp<-out.impute[,dim(out.impute)[2]]
    return(Yimp)
}

SI.ratio <- function(dat,parms) {
    out.impute <- as.matrix(SIratio(dat[,parms$Ind],
                    as.matrix(dat[,parms$Xnr]),dat[,parms$NR]))
    Yimp<-out.impute[,dim(out.impute)[2]]
    return(Yimp)
}
```

Listing A.12: Multiple Imputation methods

```
##############################################################################
```

```
##                                                                          #
##   Multiple Imputation routines                                           #
##                                                                          #
############################################################################
##                                                                          #
##  MI Inference                                                            #
##  name: MIinference()                                                     #
##  Calculates the MI estimate, the between-imputation variance,            #
##  the within-imputaton variance, the total variance,                      #
##  the confidence interval, and, if possible, the coverage                 #
##                                                                          #
############################################################################


MIinference <- function(thetahat, varhat.thetahat, alpha = 0.05)
{
    m <- length(thetahat)
    lambda <- 1 - (alpha/2)
    MIestimate <- mean(thetahat)
    B <- var(thetahat)
    W <- mean(varhat.thetahat)
    total <- W + (1 + 1/m) * B
    DF <- (m - 1) * (1 + W/((1 + 1/m) * B))^2
    CI.low <- MIestimate - qt(lambda, DF) * sqrt(total)
    CI.up <- MIestimate + qt(lambda, DF) * sqrt(total)
    width <- CI.up - CI.low
    list(MI.Est = MIestimate, MI.Var = total, CI.low = CI.low,
                              CI.up = CI.up, BVar = B, WVar = W)
}


############################################################################
##                                                                          #
## name: MIlogit()                                                         #
## Performs proper multiple imputations for a binary variable with         #
##  missing data                                                            #
## according to Rubin (1987), pp. 169-170.                                 #
## Y is the univariate dichotomous variable which has missing values       #
##  coded with NA                                                           #
## Xorg can be a vector or a matrix containing the covariates              #
##                                                                          #
############################################################################

MI.logit.GetModel <- function(dat) {
    n <- dim(dat)[1]
    l <- dim(dat)[2]
    ind <- is.na(dat[,l])
    logobs <- summary(testglm <- glm(dat[,l] ~ dat[,1:(l-1)],
              family = binomial(link = logit), maxit=25),
              correlation=T, na.action="na.omit")
    if (!testglm$converged) {
    CurrentWarn <<- Warn(1,testglm$converged,CurrentWarn)
        logobs <- summary(testglm <- glm(dat[,l] ~ dat[,1:(l-1)],
                  family = binomial(link = logit), maxit=100),
                  correlation=T, na.action="na.omit")
    CurrentWarn <<- Warn(2,testglm$converged,CurrentWarn)
    print("long version ...")
```

```
    }
     betadachobs <- logobs$coefficients[, 1]
     se.betadachobs <- logobs$coefficients[, 2]
     cor.betadachobs <- logobs$correlation
     cov.betadachobs <- cor.betadachobs
                         *(se.betadachobs %*% t(se.betadachobs))
     list(MIbeta = betadachobs, MIbetaCov = cov.betadachobs)
}

MI.logit.GetData <- function(dat,MIparms) {
     n <- dim(dat)[1]
     l <- dim(dat)[2]
     beta <- as.vector(rmvnorm(1, as.vector(MIparms$MIbeta),
                                      MIparms$MIbetaCov))
     prob <- 1/(1 + exp( - (beta[1] + matrix(dat[,1:(l-1)])
                                        %*% beta[2:l])))
     Ymis <- ifelse(runif(n) > prob, 0, 1)
     Yimp <- ifelse(is.na(dat[,l]), Ymis, dat[,l])
     return(Yimp)
}

MI.logit <- function(dat, ModParms = NULL) {
     if (length(ModParms)>0) {
          return(MI.logit.GetData(dat, ModParms))
     } else {
          return(MI.logit.GetModel(dat))
     }
}

##########################################################################
#                                                                        #
# name: MIlinreg()                                                       #
# Performs proper multiple imputations for a continuous variable        #
#  with missing data                                                     #
# according to Rubin (1987), p. 167                                      #
# Y is the univariate variable which has missing values coded with      #
#  mis                                                                   #
# Xorg can be a vector or a matrix containing the covariates            #
# lower and upper bounds of the values to be imputed can be             #
#  considered                                                            #
# if necessary, the data can be transformed before imputing them        #
#                                                                        #
##########################################################################

MI.linreg.GetModel <- function(dat) {
     n <- dim(dat)[1]
     l <- dim(dat)[2]
     ind <- is.na(dat[,l])
     if(min(dat[,l], na.rm = T) < MI.linreg.lower) {
          print(c("Lower bound is too high, set to minimum observed
                                                            value"))
          lower <- min(dat[,l], na.rm = T)
     } else {
     lower <- MI.linreg.lower
     }
     if(max(dat[,l], na.rm = T) > MI.linreg.upper) {
          print(c("Upper bound is too low, set to maxmimum observed
```

```
                                                                    value"))
        upper <- max(dat[,l], na.rm = T)
    } else {
    upper <- MI.linreg.upper
    }
    if(MI.linreg.loglin) {
        dat[,l] <- log(dat[,l])
        if(lower != -Inf) {
            lower <- log(lower)
        }
        upper <- log(upper)
    }
    n1 <- length(na.omit(dat[,l]))

    ind <- is.na(dat[,l])

    Vobs <- solve(t(cbind(1,dat[!ind,1:(l-1)])) %*%
                      cbind(1,dat[!ind,1:(l-1)]))

    betadachobs <- Vobs %*% (t(cbind(1,dat[!ind,1:(l-1)])) %*%
                                                    dat[!ind,l])
    sigma2obssum <- sum((dat[!ind,l] - cbind(1,dat[!ind,1:(l-1)])
                                                %*% betadachobs)^2)

    list(MIbeta = betadachobs, MIsigsum = sigma2obssum, MIn1 = n1,
                        MIvobs = Vobs, lower = lower, upper = upper)
}

MI.linreg.GetData <- function(dat,MIparms) {

    n <- dim(dat)[1]
    l <- dim(dat)[2]
    sigma2 <- MIparms$MIsigsum / rchisq(1, (MIparms$MIn1 - l - 1))
    betatemp <- rmvnorm(1, as.vector(MIparms$MIbeta), sigma2
                                            * MIparms$MIvobs)
        redo <- TRUE
        while(redo) {
        Ymis <- rnorm(n, (cbind(1,dat[,1:(l-1)]) %*% t(betatemp)),
                                                    sqrt(sigma2))
            Yimp <- ifelse(is.na(dat[,l]), Ymis, dat[,l])
                Yimp <- ifelse(Yimp < MIparms$lower, NA, Yimp)
                Yimp <- ifelse(Yimp > MIparms$upper, NA, Yimp)

                if(length(na.omit(Yimp)) == n) {
                    redo <- FALSE
                }
        }
        if(MI.linreg.loglin) {
            return(exp(Yimp))
        }
        else {
            return(Yimp)
        }
}

MI.linreg <- function(dat, ModParms = NULL) {
    if (length(ModParms)>0) {
```

```
        return(MI.linreg.GetData(dat, ModParms))
    } else {
        return(MI.linreg.GetModel(dat))
    }
}

############################## MI master routine ############################

MI.varest <- function(dat, MIR, EstMeth = EstMeth, MIMeth = MIMeth ,
                      TrueVal = TrueVal, parms = parms, Nh = Nh, V=T) {
    Est <- array(0,dim=c(MIR,2))
    MIparms <- MIMeth(dat[,c(parms$Xnr,parms$NR)])
    for (i in 1:MIR) {
        Yimp <- MIMeth(dat[,c(parms$Xnr,parms$NR)],MIparms)
        Est[i,] <- EstMeth(dat, parms = parms, TrueVal=TrueVal,
                                                Yimp=Yimp, V)
    }
    if (V) {
        MIobj  <- MIinference(Est[,1],Est[,2])
        MIobj0 <- MIinference(Est[1:MIR0,1],Est[1:MIR0,2])
        MIobj1 <- MIinference(Est[1:MIR1,1],Est[1:MIR1,2])
        return(c(MIobj$MI.Est,MIobj$MI.Var,MIobj0$MI.Est,
               MIobj0$MI.Var,MIobj1$MI.Est,MIobj1$MI.Var))
    } else {
        return(c(mean(Est[,1]), 0, mean(Est[1:MIR0,1]), 0,
                                   mean(Est[1:MIR1,1]), 0))
    }
}

MI.varest.boot <- function(dat, MIR, BR, EstMeth = EstMeth,
                  MIMeth = MIMeth , TrueVal = TrueVal,
                  parms = parms, Nh = Nh) {
    Est <- array(0,dim=c(MIR,2))
    MIparms <- MIMeth(dat[,c(parms$Xnr,parms$NR)])
    for (i in 1:MIR) {
        Yimp <- MIMeth(dat[,c(parms$Xnr,parms$NR)],MIparms)
        Est[i,] <- Boot.varest(dat, BR, EstMeth, TrueVal = TrueVal,
                                        parms = parms, Yimp = Yimp)
        print(sum(Yimp))
    }
    MIobj <- MIinference(Est[,1],Est[,2])
    return(c(MIobj$MI.Est,MIobj$MI.Var))
}
```

A.3 Template Files

Listing A.13: Group information template

```
#########################################################################
##                                                                     #
```

```
## Estimation Metadata   :                                                    #
##                                                                           #
## Survey specifiation   :   SURVEYSPEC                                      #
##                                                                           #
## Estimator / Variance estimator : WHICHESTIMATORS                          #
## NR correction         :   no                                              #
## weighting             :   no                                              #
##                                                                           #
#############################################################################
## SurveySpec
source(paste(DATPATH,"SurveySpec/SURVEYSPEC.dat",sep=""))

## Estimators of interest

INPATHS

## General routines

Sim.Init <- function(Univ = Univ, Univ.SA = Univ.SA) {
    SURVEYSPEC.Univ(Univ)
}

Sim.Est <- function(S,r) {

    SURVEYSPEC.Sample(S)

DOESTIMATORS

}


Sim.Write <- function(r) {

WRITEESTIMATORS

}

Sim.Read <- function() {

READESTIMATORS

}

Sim.Show <- function(r) {

SHOWESTIMATORS

}
```

Listing A.14: Estimator information template

```
#############################################################################
##                                                                          #
```

```
##  General simulation dataset:                                              #
##      SURVEYSPEC                                                           #
##                                                                           #
#############################################################################
##
## Variables in use:
##
## Y:        EstVar           :: ELOEB
## X:        AuxVar           :: ALO, DM, DF, AM
## Ind       Strata           :: RS x GGK
## NR        Nonresponse      :: 25% non-response
## Xnr       NR AuxVar        :: ALOEB
## SU        Sampling Units   :: AWB (by RS and GGK)
## SA        Small Areas      :: no
##
#############################################################################
#############################################################################
##                                                                           #
## Additional constants:                                                    #
##                                                                           #
#############################################################################
## BL        : Federal state for simulation (all not yet implemented)
## SA        : Small area problem (cf. MasterSim)

UNIV     <- COUNTRY
BL       <- BUNDESLAND
SA       <- 0

StratInd <- 1:SURVEYNUMBL


#############################################################################
##                                                                           #
## Metadata for SURVEYSPEC:                                                 #
##                                                                           #
#############################################################################
## .dat      : data matrix (if group data used, NULL)
## .parms    : parameter vector


SURVEYSPEC.univ         <- NULL
SURVEYSPEC.smallarea    <- NULL
SURVEYSPEC.dat          <- NULL
SURVEYSPEC.parms        <- NULL
SURVEYSPEC.names        <- NULL

SURVEYSPEC.Univ <- function(U) {
    Y <- U$ELOA1F + U$ELOA2F + U$ELOA3F + U$ELOA4F + U$ELOA1M
                            + U$ELOA2M + U$ELOA3M + U$ELOA4M
    X <- cbind(U$ALO,U$DM,U$DF,U$AM)
    Nh <- U$N
    SURVEYSPEC.univ <<- list(Y = Y, X = X, Nh = Nh)
}

SURVEYSPEC.Sample <- function(S) {
    Strata <- STRATA(5*(S$RS-min(S$RS))+S$GGK)
    S <- S[Strata>0,]
```

```
    Strata <- Strata[Strata>0]

    dat <- cbind(
        ifelse(S$ELO==1 & S$AGE>13,1,0),
        S$ALO,
        ifelse(S$NAT==0 & S$SEX==0,1,0),
        ifelse(S$NAT==0 & S$SEX==1,1,0),
        ifelse(S$NAT>0 & S$SEX==0,1,0),
        ifelse(S$ALO==1 & S$AGE>13,1,0),
        S$NR25,
        Strata,
        100)

    SURVEYSPEC.names <<- c("UnempEB", "ALO", "Na0Se0", "Na0Se1",
                           "Na1Se0", "ALOEB", "NR25", "Strata", "IIP")

    Y       <- c(1)
    X       <- c(2:5)
    Ind     <- c(8)
    NR      <- c(7)
    Xnr     <- c(6)
    Xnr1    <- c(6)
    SU      <- c(8)
    SA      <- NULL
    IIP      <- c(9)

    # set NR vector to either NR or y-value
    dat[,NR] <- ifelse(dat[,NR]==0,NA,dat[,Y])

    SURVEYSPEC.dat <<- dat
    SURVEYSPEC.parms <<- list(Y = Y, X = X, Ind = Ind, NR = NR,
            Xnr = Xnr, Xnr1 = Xnr1, SU = SU, SA = SA, IIP = IIP)
}
```

A.4 PERL Routines

Listing A.15: PERL routine for generating simulation programmes

```
# /bin/perl -w
#
# Federal State

$BL = "SAL";
$surveynumbl = 15;
$country = "GMC";

$master = "GMC.1.T.ELOEB";
$FS = "49.3.sub0";

$pfad = "e:/Sim_Tabelle_MI";
```

```
$surveyspec = $master . "." . $BL . "." . $FS;
$GN = "G01";


# Start

if (-e "$pfad/SimData/SurveySpec/$master.$FS.master") {

    open(Mikro, "< $pfad/SimData/SurveySpec/$master.$FS.master")
                or die "Couldn't find Master-SurveySpec to read";
    open(OFILE, "> $pfad/SimData/SurveySpec/$surveyspec.dat")
                or die "Couldn't write to SurveySpec-File
                $pfad/SimData/SurveySpec/$surveyspec.dat\n";

    while ($line = <Mikro>) {
        $line =~ s/SURVEYSPEC/$surveyspec/g;
        $line =~ s/SURVEYNUMBL/$surveynumbl/g;
        $line =~ s/BUNDESLAND/"$BL"/g;
        $line =~ s/COUNTRY/"$country"/g;
        print OFILE $line;
    }
    close(OFILE);
    close(Mikro);

    $whichests = "";
    $outests = "";
    $doests = "";
    $writeests = "";
    $readests = "";
    $showests = "";
    for ($i = 0; $i<$anz; $i++) {
        if (-e "$pfad/SimData/DataSets/$surveyspec.$Ests[$i].dat") {
            print "$Ests[$i] does not exist\n";
        } else {
          $whichests = $whichests . "##\t\t:\t$surveyspec.$Ests[$i]\n";
          $outests = $outests . "source(paste(DATPATH,\"DataSets/
                                $surveyspec.$Ests[$i].dat\",sep=\"\"))\n";
          $doests = $doests . "\t$surveyspec.$Ests[$i].Est(r)\n\t
                                print(\"$Ests[$i] done\")\n";
          $writeests = $writeests . "\t$surveyspec.$Ests[$i].Write(r)\n";
          $readests = $readests . "\t$surveyspec.$Ests[$i].Read()\n";
          $showests = $showests . "\t\t $surveyspec.$Ests[$i].Show(r)\n";

          if (-e "$pfad/Output/$surveyspec.$Ests[$i]/") {
              print "!!! Caution $surveyspec.$Ests[$i]
                                already exists !!!\n";
          } else {
              mkdir("$pfad/Output/$surveyspec.$Ests[$i]",0777)
              or die "Directory $pfad/Output/$surveyspec.$Ests[$i]
                                couldn't be generated !!!\n\n";
              if($Ests[$i] =~ /.MI/) {
                mkdir("$pfad/Output/$surveyspec.$Ests[$i]0",0777)
                or die "Directory $pfad/Output/$surveyspec.$Ests[$i]0
                                couldn't be generated !!!\n\n";
                mkdir("$pfad/Output/$surveyspec.$Ests[$i]1",0777)
                or die "Directory $pfad/Output/$surveyspec.$Ests[$i]1
                                couldn't be generated !!!\n\n";
```

```
                };
                if($Ests[$i] =~ /.boot/) {
                    mkdir("$pfad/Output/$surveyspec.$Ests[$i]0",0777)
                    or die "Directory $pfad/Output/$surveyspec.$Ests[$i]0
                    couldn't be generated !!!\n\n";
                };
            }
            open(EFILE, "< $pfad/SimData/Estimators/$Ests[$i]")
                    or die "$Ests[$i] not found ...\n";
            open(OFILE, ">
                    $pfad/SimData/DataSets/$surveyspec.$Ests[$i].dat")
                    or die "Couldn't write to Estimator-File\n";
            while ($line = <EFILE>) {
              $line =~ s/SSPEC/$surveyspec/g;
              print OFILE $line;
            }
            close(OFILE);
            close(EFILE);
          }
    }

    open(IFILE, "< $pfad/SimData/SurveySpec/group.master")
        or die "Couldn't open input file ...\n";
    open(OFILE, "> $pfad/SimData/DataSets/$surveyspec.$GN.dat")
        or die "Couldn't write to Group-File
                $pfad/SimData/DataSets/$surveyspec.$GN.dat\n";
    while ($line = <IFILE>) {
        $line =~ s/SURVEYSPEC/$surveyspec/g;
        $line =~ s/INPATHS/$outests/g;
        $line =~ s/WHICHESTIMATORS/$whichests/g;
        $line =~ s/DOESTIMATORS/$doests/g;
        $line =~ s/WRITEESTIMATORS/$writeests/g;
        $line =~ s/READESTIMATORS/$readests/g;
        $line =~ s/SHOWESTIMATORS/$showests/g;
        print OFILE $line;
        }
        close(OFILE);
        close(IFILE);

} else {

    print("Master-Sourcefile
          ($pfad/SimData/SurveySpec/$master.$FS.master)
          not found!\n Programme stopped.
          Please generate SurveySpec or enter real name.\n\n");

}

exit(0);
```

Appendix B

Weighting in the Swiss HBS

Listing B.1: Weighting in the Swiss HBS

```
################### START FUNCTION SFSOCalcWeights ##########################
SFSOCalcWeights <- function(Sfull, weightsOfFullSample = T)
    {
    # This function calculates the weights per household by using
    # sampling weights and nonresponse weights according to model A.
    #
    # Parameters
    # - Sfull: Data frame with the sample data containing both
    #          non-responding and responding households.
    #          This data frame must contain 3 vectors of
    #          length n (= number of households at level 1):
    #          - REG (=GREGION): variable containing the stratum
    #            (and regional) information of each household
    #          - STA (=STASOCIO): variable containing the
    #            socio-economic group of each household
    #          - UNR0: binary response vector
    #            1: responding household
    #            0: non-responding household
    # - weightsOfFullSample: optional parameter to select the length
    #          of the returned weight vector:
    #          - full length (n: all households of the sample)
    #          - or only the responding households
    #
############## version 0.1: February 9th, 2004, UOe ##########################

    # Inclusion probabilities ph per stratum (sampling weights = 1/ph):
    #   The numbers of households of each stratum in the universe
    #   are stored in the global data frame "HBS.univ".
    p1  <- sum(Sfull$REG == 1) / HBS.univ$Nh[1]
    p2  <- sum(Sfull$REG == 2) / HBS.univ$Nh[2]
    p3  <- sum(Sfull$REG == 3) / HBS.univ$Nh[3]
    p4  <- sum(Sfull$REG == 4) / HBS.univ$Nh[4]
    p5  <- sum(Sfull$REG == 5) / HBS.univ$Nh[5]
```

```
    p6  <- sum(Sfull$REG == 6) / HBS.univ$Nh[6]
    p7  <- sum(Sfull$REG == 7) / HBS.univ$Nh[7]

    # Nonresponse model according to mechanism A using a
    #   logistic regression with 2 parameters and the intercept.
    sa       <- !( (Sfull$REG == 2) | (Sfull$REG == 7) )
    saNum    <- sa * 1
    g13      <- (Sfull$STA == 1)
    g13Num   <- g13 * 1
    rep      <- (Sfull$UNR0 == 1)
    repNum   <- rep * 1
    logisticFit <- glm(formula = repNum ~ saNum + g13Num,
        family = binomial(link = logit),
        data = Sfull,
        na.action = na.fail,
        control = list(epsilon = 0.0001, maxit = 50, trace = F)
        )
    exptemp <- exp(logisticFit$coefficient[1] +
        logisticFit$coefficient[2] * saNum +
        logisticFit$coefficient[3] * g13Num)
    repWeights  <- 1 / (exptemp / (1 + exptemp) )

    # Weights = response weights * sampling weights.
    weights <- repWeights * (
          (Sfull$REG == 1) * (1/p1)
        + (Sfull$REG == 2) * (1/p2)
        + (Sfull$REG == 3) * (1/p3)
        + (Sfull$REG == 4) * (1/p4)
        + (Sfull$REG == 5) * (1/p5)
        + (Sfull$REG == 6) * (1/p6)
        + (Sfull$REG == 7) * (1/p7))

    # Selection of size of returned vector.
    if(weightsOfFullSample) return(weights)
    else return(weights[Sfull$UNR0 == 1])
    }
################## END FUNCTION SFSOCalcWeights ############################
```

References

Lundstrøm, S. and Särndal, C. E. (2002): *Estimation in the Presence of Nonresponse and Frame Imperfections*. Statistics Sweden.

Quatember, A. (2003): The estimation of the variance of a ratio in the Austrian Microcensus. unpublished manuscript.

Särndal, C. E., Swensson, B. and Wretman, J. (1992): *Model Assisted Survey Sampling*. New York: Springer-Verlag.