STATBEL
A division of the Ministry of Economic Affairs

No 3 May 2001

Statistics Belgium Working Paper

Generalised Calibration at Statistics Belgium
SPSS® Module g-CALIB-S and Current Practices

Camille Vanderhoeft
Actuary, Section for Methodology and Co-ordination

Generalised Calibration at Statistics Belgium

SPSS® Module g-CALIB-S and Current Practices

Statistics Belgium

Camille Vanderhoeft
Actuary, Section for Methodology and Co-ordination

May 2001

Dear reader,

You are about to read a publication of the new methodological series of Statistics Belgium, the Belgian National Institute for Statistics. This series is called “Statistics Belgium Working Papers”. Our intention is to provide our users and everyone who in business and academic life engages in statistics, with studies, reports and preparatory documents. We also wish to give our staff, statisticians, methodologists and others a chance to disseminate their ideas and the results of their work.
These *working papers* aim to contribute to the development of statistical knowledge and the exchange of ideas.
As *working papers*, they will not reflect the official view of either Statistics Belgium (INS-NIS) or the Belgian Government. The authors only should be held responsible for their content.
Statistics Belgium launches this initiative hoping to raise the level of discussions on statistical needs and methods, and to ensure a better dissemination of ideas and conclusions.
As a matter of convenience, *Statistics Belgium Working Papers* will be issued in the language of the original paper.
The staff and all the officials of Statistics Belgium, and especially the authors, are anxious to receive any comment or question on this paper.

Brussels,
Claude CHERUY
Director-General,
Statistics Belgium

Contents

Introduction

During the last decade, the literature on *calibration* (or (up-)weighting, grossing(-up)), has vastly been growing. At the start of the nineties, some influential works were published. *Model Assisted Survey Sampling* from C.-E. Särndal, B. Swensson en J. Wretman, which appeared in 1992, is undoubtedly one of the most important reference works for the survey statistician; the book is often called "The bible of survey statistics", or "The yellow book". In this book the authors create a uniform framework for the theoretical treatment of survey statistics. It encloses a review, within a single framework, of survey methodology as it has been developed by themselves and by many other eminent researchers before them. Moreover, they offered an efficient tool for future developments of that methodology. Within the theory of *generalised regression* (GREG) *estimation*, they also treat calibration techniques. In the same year, 1992, an article entitled *Calibration Estimators in Survey Sampling*, from J.-C. Deville and C.-E. Särndal, was published in JASA. This leading article suddenly opened an even broader class of calibration techniques, of which GREG estimation is a sub-class. Many researchers, theoreticians and practitioners, all around the world in public statistical institutes and universities, have since then been inspired by the JASA-article, which resulted in a long series of studies on calibration. Apart from the statistical theory of the generalised calibration model, it is important for the present study to notice that Deville and Särndal based their treatment on the fact that, for a given sample, the calibrated weights can be found as the solution of a "convex mathematical programming problem". The latter idea is central in our study.

Indeed, theoretical as well as practical parts of this text are based on that convex mathematical program – which we call the *calibration problem* (CP) hereafter – that allows to compute the calibrated weights. The theory in this report is not a statistical theory of calibration, but rather a theoretical approach of various aspects of the numerical solution of the CP. Our knowledge about operations research and (linear) regression was very helpful in that respect. Searching for a rigorous mathematical formulation, we ended up with a matrix formulation of the CP, a "language" which proved to be very efficient for our purposes. As a result, we achieved a better theoretical understanding, which in turn resulted in some new developments, which were interesting and helpful both from the theoretical point of view and from the practical one. In fact, it would have been simply impossible for the author to treat, within the assigned time period, so many specific surveys, if he could not have based his reasoning on such a powerful formulation. We now try to explain this in more detail by giving an overview of the different chapters in this study.

Chapter I presents the statistical definition of calibration estimates of totals of random variables, i.e. as weighted sums of obrserved values of the variable; the weights in this sum are the *calibrated weights*, which have to satisfy certain linear constraints, the *calibration constraints*. Next we discuss some popular estimation techniques and show that they fit into the calibration framework. These traditional techniques are: (1°) the *post-stratification* technique, which assures that, within each post-stratum, the estimated population size equals a pre-specified value, (2°) the *raking (ratio)* technique, in which weights are such that marginal constraints in a cross-tabulation are satisfied, and (3°) the *ratio* estimation technique, which assures that, for a given *auxiliary* variable (a *proxy* for the study variable), the estimated total equals a given value. At Statistics Belgium, (almost) only the post-stratification estimation technique has been applied until now.

In the second chapter, we present "a" theory of generalised calibration. We discussed this already here before. Chapter II is the most mathematical part of the text, and may therefore be less appealing to some readers. For the author, however, it covers and reflects an important part of the work. The mathematical approach, building on matrix theory, on mathematical programming, and on linear

regression, finally resulted into a compact formulation of the CP. In section II.C, this led to a proposal for a solution to the supplementary problem of determining an extreme lower bound *L* and an extreme upper bound *U* for the correction factors (i.e. the ratios between the calibrated weights and the initial weights) for certain sub-classes of the CP. Chapter II further deals with the existence and uniqueness of the solution to the CP, and finally with the general algorithm that is the core of our software g-CALIB-S. This algorithm, as well as its mathematical justification, takes largely advantage of the properties of *generalised inverses* of matrices. If we could not have used that mathematical device, then our own implementation, in SPSS® 9.0, for calculation of calibrated weights would not have been very original, and would moreover have been useless, as well as virtually impossible to be realised.

Chapter III brings the reader back to statistics. We there return to more traditional methods, discussed already in chapter I. On the other hand, without being occupied by specific practical extrapolation problems, we demonstrate various ingenious applications of generalised calibration. Of course, this brings us back to other researchers' findings for complex situations. One of the strengths of this report, in our opinion, is the way the problem is treated formally, which makes it possible to make new derivations, in an abstract but controlled way, and to implement these later in a practical problem. Of course, we do not want to state that theory precedes practice! Indeed, statistical methodology starts with real-life statistical problems. It can however not be argued that one of them is more important than the other. But it is definitely true that a good understanding of a concrete problem is only possible within a suitable framework, or model. An excellent illustration is our discussion in part III.E about simultaneous calibration on external information available at two levels. An important summarising table in that context is table 3.8 in section III.E. This table can be used as an outline or *aide-mémoire* when setting up an application of calibration.

The central topic of this report in fact is our software g-CALIB-S, developed with SPSS® syntax language. This software resulted from the need to have access to a tool for improved calibration, using more external information when estimating from surveys. Statistics Belgium only a few years ago purchased SPSS®, so a little experimentation with calibration in SPSS® could not be postponed for a long time. With CALMAR (section IV.A.3) as an example, our colleague *Etienne Waeytens* started the implementation of calibration as developed by Deville and Särndal. When the author of this report saw the syntax, he immediately proposed an improvement by replacing the matrix fuction *inv*() with *ginv*(). More matrix functions were then also exploited to create macros for construction of the calibration design matrix. This way a thorough "theoretical" study on the one hand, and serious improvements to the software tools on the other, were onset. Implementation of the techniques in practice had to be delayed for a while: the software first had to be reliable, flexible and user-friendly enough, and the author had to study the different calibration problems Statistics Belgium is faced to.

Chapter IV is meant to be a manual for g-CALIB-S. The potential user can find in that chapter how the software has to be used, how s/he has to prepare the data that will be input to the software, and how to interpret and use the results that are produced by the software. The chapter closes with some comments on the software, pointing to some weaknesses of g-CALIB-S, but also to possible future developments, either to improve or to extend our syntax modules.

Chapter V is then devoted to implementation of generalised calibration, and consequently the software g-CALIB-S, in daily statistical practice at Statistics Belgium. The author invited the responsible statisticians in different statistical departments to propose practical applications of calibration, after instructing them about the contents of the project. Many practical problems were put forward. Unfortunately, they could not all be studied before the deadline for this report, so that only a few are discussed in chapter V. From the beginning it was stipulated that the main goal was to find out how current extrapolation practices fit into the generalised calibration framework, and to reproduce figures that were obtained earlier. The author benefited a lot from this task, since he was forced to understand the various problems in detail, which often resulted in an improvement of the theoretical framework or model. And the task has been successfully "finished". However, we decided not to reproduce simply figures obtained earlier, but to concentrate more on making suggestions for future work on calibration, since the latter seemed much more interesting.

The most interesting applications being dealt with are the *Time Use Survey* (TUS) in section V.D, "generalised raking" of data on *labour volume and labour compensation* (LVC) in section V.F, and the *Structural Business Survey* of enterprises (SBS) in section V.G. The first application (TUS) is an illustration of using auxiliary information at two levels (here: households and individuals). It discusses sophisticated and powerful calibration techniques, which will be useful for many other surveys too. The second application (LVC) is a special one, due to the fact that the data were presented in cross-tabulation format. A study of this application resulted in the discussion in section III.C (and in some sense also in section III.D). The LVC problem therefore broadened our understanding of the problem of calibration, which could not have been predicted when we first saw the data. The SBS case in another way contributed to better insight into the problem: we had to tackle here the problem of *over-coverage* of the sampling frame and how it is up to now being dealt with in the context of calibration.

The *Travel Survey* (TS) takes a special place in this report. Since we finally had access to a well-designed basic data file for this survey, we were able to show that preparatory work on the data, to make them ready for calibration, does not have to be that hard. Given a well-structured and well-documented database, it is possible to set up a systematic and extensive preparation of data, which may be reflected in SPSS® syntax files. Such programs, which are merely the implementation of procedures and algorithms, can be used in and be adapted to other circumstances (at other time points, or for other surveys) if necessary. So we can build on experience from the past.

The reader will not find many tables with figures in this report. That is because this study is first of all a methodological one. This means that we have to point out what can be done and how to do it. Partly, these instructions can be found in our SPSS® syntax programs, which the reader will find in appendix (chapter VII). As a mathematical formula or equation, such programs often tell more about strategies than a long verbal description. So the reader is invited to have a look at the programs when s/he wants to start an application of generalised calibration. Obviously, these programs (apart from those in section VII.A!) are not immediately ready-for-use in other circumstances, but they can serve very well as a starting point. It can be noticed that they also need to be improved for the survey for which they were intended.

We want to emphasize that the present study aims to initiate a new treatment of surveys at Statistics Belgium. A lot of work still has to be done before implementation of the techniques is a fact. Therefore we will organise some courses at our institute, both on advanced use of SPSS® and on generalised calibration methodology and practice. This explains one of the main purposes of this report: to provide our statisticians with some sort of manual or guidance when they start applying the techniques in practice. Hence the pragmatic approach of this study.

Finally, we whish to state that this work should be a start of an in-depth study of calibration for each survey were the techniques have to be used. We expect that survey specific studies on calibration will be published in the future. This should be accompanied with a methodological evaluation of the sampling-design, resulting moreover in the systematic and correct calculation of variance estimates, which are an important means of quality evaluation for our surveys.

Chapter I

Classical estimation techniques

I.A GENERALITIES AND NOTATION – THE CALIBRATION PROBLEM

Consider a *population U* of size *N*, an *initial sample s* from *U* of size *n*, and a sample *r* of *respondents* of size *m*. For simplicity in this chapter we assume that the sample *s* is drawn by *simple random sampling* (SRS), although this is not a crucial assumption here; the response mechanism need not be specified. We have: $r \subset s \subset U$ and $0 < m \leq n \leq N$. The overall *sampling fraction* is $f = n/N$ and the overall *response rate* or *response probability* is $p = m/n$. A subscript *h* (or *j*, etc) is included in these notations wherever sub-populations and corresponding sub-samples are to be considered.

Let *y* be a *study variable* or *variable of interest*, with value y_k for the *k*-th population element. Our main goal is to estimate the population total $t_y = \sum_{k \in U} y_k$ of the variable *y*. A *linear estimator* for this total takes the form

$$\hat{t}_y = \sum_{k \in r} w_k y_k \,, \tag{I.1}$$

i.e. a weighted sum of available values for the study variable: the sum is over the respondents only. The central idea of *calibration* is to calculate the weights w_k for respondents $k \in r$ such that one or more *calibration constraints* are satisfied. A calibration constraint takes the general form

$$\sum_{k \in r} w_k x_k = t_x \,, \tag{I.2}$$

where *x* is considered to be a variable, with known value x_k for respondent *k*, and t_x is a known *calibration benchmark* for that variable. A benchmark often is the total of the variable *x* for the population *U*, whence the notation t_x, consistent with t_y. Calibration benchmarks will often be called *calibration totals*; a benchmark can sometimes be an estimate for a population total.

A *x*-variable in this chapter is always an *indicator variable*, corresponding to a sub-population of *U*, with value 1 for elements *k* inside the sub-population and value 0 for elements *k* outside that sub-population. An indicator variable is thus a *membership variable* for the sub-population considered. Notice that in regression theory, these variables are often called *dummies*; they may correspond to some category of a *qualitative explanatory variable*, or to cells induced by a cross-classification by several qualitative explanatory variables. A subscript *h* (or *j*, etc) will be used in the next sub-sections to distinguish indicator variables from each other.

The symbol *z* is used in this chapter to denote a quantitative variable. The value of this variable for respondent *k* is denoted z_k, and a benchmark corresponding to *z* is denoted t_z.

The purpose of this introductory chapter I is to recall that some classical techniques of estimation of totals of survey variables *y* often can be studied as *calibration techniques*. I.e. we show that commonly used estimators for t_y can be written in the form (I.1), with weights that satisfy (I.2). The classical techniques considered in this chapter are the *post-stratified estimator*, the *raking estimator* and the *ratio estimator*. Statistical properties of these estimators of totals, such as (un-) biasedness, consistency and efficiency are not considered here. By the way, the theory of generalised calibration presented in the next chapter too is not a statistical theory!

I.B THE POST-STRATIFIED ESTIMATOR

The *post-stratified estimator*, also called *post-hoc* stratification estimator by Barnett (1991), is discussed at length in most textbooks; see e.g. Cochran (1977) and Särndal *et al* (1992). This estimator is defined as follows, given post-strata indexed by the subscript $h = 1, \ldots H$:

$$\hat{t}_{y,ps} = \sum_h N_h \bar{y}_h \,, \tag{I.3}$$

where $\bar{y}_h = \frac{1}{m_h} \sum_{k \in r_h} y_k$ is the observed average of y in the h-th respondent sub-sample. This estimator can be rewritten as $\hat{t}_{y,ps} = \sum_h \frac{N_h}{m_h} \sum_{k \in r_h} y_k = \sum_{k \in r} \frac{N_h}{m_h} y_k$, which has the form (I.1) if $w_k = \frac{N_h}{m_h}$ for $k \in r_h$ and $h = 1, \ldots, H$.

Now, to each post-stratum h corresponds an indicator variable x_h, as pointed out in the preceding sub-section. The post-stratified estimator of the population total of this indicator variable, which is N_h, is then equal to $\hat{t}_{x_h,ps} = \sum_{h'} \frac{N_h}{m_{h'}} \sum_{k \in r_{h'}} x_{hk} = \frac{N_h}{m_h} m_h + \sum_{h' \neq h} \frac{N_h}{m_{h'}} 0 = N_h$. In other words, the weights $w_k = \frac{N_h}{m_h}$ satisfy the H calibration constraints, which are of the form (I.2):

$$\sum_{k \in r} w_k x_{hk} = N_h = t_{x_h} \qquad (h = 1, \ldots, H). \tag{I.4}$$

This proves that the post-stratification estimator is indeed a calibration estimator. The weights $w_k = \frac{N_h}{m_h}$ are called *up-weighting factors*, or *extrapolation coefficients*, in statistical practice; from now on we may call them *calibrated weights*.

Notice that $w_k = \frac{N_h}{n_h} \frac{n_h}{m_h} = f_h^{-1} p_h^{-1}$, which suggests that extrapolation can be seen as a two-step correction procedure, with correction for non-response in the first step (up-weighting from r to s), and adjustment for sampling error in the second step (up-weighting from s to U). Moreover, this suggests that post-strata ideally should coincide with sub-populations that are homogeneous with respect to response behaviour. In other words, the post-strata membership variables x_h should be suitable determinants for non-response.

Post-stratification is not only considered in the context of SRS. Often in practice the sampling situation is more complex, and care then has to be taken. This will be discussed in more detail in section III.B.1.

I.C THE RAKING ESTIMATOR

Post-strata may correspond to the categories of a qualitative variable, or to the cells in a cross-classification by two or more qualitative variables. Consider, for simplicity, only two such classification variables, and let subscript *i* be used to index the categories of one of the variables, and subscript *j* for the other. The post-stratification technique can be used to estimate the total of a study variable *y* if (1°) the sub-samples r_{ij} of respondents are all non-empty and if (2°) the sub-population sizes N_{ij} are all known.

Suppose that at least one of these conditions is not satisfied. Then a possible solution is to use only the *marginal* population counts N_{i+} and N_{+j}, and to apply the iterative method of *raking* (or *iterative proportional fitting*) to find weights w_{ij} that, at convergence of the iterative procedure, satisfy the constraints $w_{i+} = \frac{N_{i+}}{N}$ for all *i* and $w_{+j} = \frac{N_{+j}}{N}$ for all *j*, and consequently also $w_{++} = 1$. The summations denoted by the subscript "+" have to be interpreted as follows. Notice first that w_{ij} is a common weight for all elements $k \in r_{ij}$. Let $r_{i\bullet} = \bigcup_j r_{ij}$, then the subscript "+" denotes summation over all elements in $r_{i\bullet}$, i.e. $w_{i+} = \sum_{k \in r_{i\bullet}} w_{ij} = \sum_j \sum_{k \in r_{ij}} w_{ij} = \sum_j m_{ij} w_{ij}$; similar for summation over the index *i*. The raking-ratio estimator for the total of any study variable *y* is then equal to $\hat{t}_{y,rak} = N \sum_{i,j} w_{ij} \sum_{k \in r_{ij}} y_k = \sum_{k \in r} w_k y_k$, with $w_k = N w_{ij}$ for all $k \in r_{ij}$. Hence the raking estimator has the form (I.1). The iterative procedure is not fully described here; we refer for instance to Deville *et al* (1993). Notice that there is no explicit algebraic formula for the weights.

To show that the raking-ratio estimator is a calibration estimator, we have to show that the weights w_{ij} satisfy some calibration constraints. Therefore we only have to show that the above-mentioned constraints can be rewritten as calibration constraints. To that end, we define two sets of indicator variables: variables $x_i^{(1)}$, say, corresponding to the categories of the first classification variable, and variables $x_j^{(2)}$, say, corresponding to the categories of the second classification variable. It is then easy to verify that $t_{x_i^{(1)}} = \sum_{k \in U} x_{ik}^{(1)} = N_{i+}$ for any *i* and $t_{x_j^{(2)}} = \sum_{k \in U} x_{jk}^{(2)} = N_{+j}$ for any *j*. Finally, we derive the following calibration constraints for the weights w_k:

$$\begin{aligned} &\sum_{k \in r} w_k x_{ik}^{(1)} = t_{x_i^{(1)}} \quad \text{any } i \\ &\sum_{k \in r} w_k x_{jk}^{(2)} = t_{x_j^{(2)}} \quad \text{any } j \end{aligned} \tag{I.5}$$

This means that the raking estimator is a calibration estimator. More about raking, and its relation to post-stratification, will be discussed in sections II.B.2 and III.C. It may be noticed that the raking weights are not just *any* solution of the system (I.5), but a very special one. This will be discussed later in the context of generalised calibration.

I.D THE RATIO ESTIMATOR

Consider now the situation where a quantitative variable *z* is known for all respondents *k*, and the population total t_z is known as well. The *ratio estimator* of the total t_y of a study variable *y* is defined (still under SRS) as

$$\hat{t}_{y,rat} = \frac{t_x}{\sum_{k \in r} x_k} \sum_{k \in r} y_k = \sum_{k \in r} w_k y_k \ , \tag{I.6}$$

with constant weights $w_k = \frac{t_x}{\sum_{k' \in r} x_{k'}}$ ($k \in r$). To show that this ratio estimator is also a calibration estimator, we simply have to notice that the weights are satisfying trivially the single calibration constraint

$$\hat{t}_{x,rat} = \sum_{k \in r} w_k x_k = t_x \ . \tag{I.7}$$

The ratio estimator works well if the linear (population) regression model $E(y_k) = \beta x_k$ (through the origin) fits well. Notice that (I.7) does have many solutions; the one leading to the above constant weights (under SRS) is a particular *regression estimator* if the variance structure of the population regression model is determined as $V(y_k) = \sigma^2 x_k$. We refer to Särndal *et al* (1992) for a detailed discussion of ratio and regression estimators.

The ratio estimator is thus a particular calibration estimator. However, it is not covered by the theory of generalised calibration as presented in this text. That is because we have worked throughout with a slightly simplified version of the generalised calibration model introduced by Deville and Särndal (1992) and Deville *et al* (1993). More about this can be found in section II.A; see also IV.C.2.viii.

Chapter II

A theory of generalised calibration

II.A THE GENERALISED CALIBRATION PROBLEM AS A MATHEMATICAL OPTIMISATION PROBLEM

Consider a probability sample *s* of size *n* from a population *U* of size *N*. Let the *n* sample elements be selected according to a sampling design with positive *first order inclusion probabilities* $\pi_1,\ldots,\pi_N$ for all elements in *U*. This means that only *random* or *probability sampling* is being dealt with here. Where needed we shall also assume that the *second order inclusion probabilities* $\pi_{kl}\ (k,l=1,\ldots,N)$ are known. With these two conditions, the sampling design is said to be *measurable* (Särndal *et al*, 1992).

Suppose that measurements on *m auxiliary* or *calibration variables* are available for all sample elements; let x_{kj} be the value of the *j*-th calibration variable for the *k*-th sample element $(j=1,\ldots,m;k=1,\ldots,n)$. It is assumed that qualitative variables are already transformed into sets of indicator variables, etc. Finally, the population totals $t_j\ (j=1,\ldots,m)$ for the calibration variables must be available. The calibration problem consists of adjusting some *initial weights* d_k, resulting in *adjusted* or *calibrated weights* $w_k = g_k d_k$, where g_k are the adjustment factors or *g-weights*. The initial weights often are the *sampling weights* $1/\pi_k$, but these could already have been corrected for non-response before calibration takes place. Notice that *s* might be a respondent sample, instead of an initial sample.

The *generalised calibration problem*, i.e. the problem of calculating the calibrated weights or the g-weights <u>for a given sample s</u>, can be formulated as a *non-linear optimisation problem* as follows.

(C1) Minimise the *distance* $\sum_{k=1}^{n} d_k G\left(\frac{w_k}{d_k}\right)$,

(C2) subject to *m calibration constraints* $\sum_{k=1}^{n} w_k x_{kj} = t_j\ (j=1,\ldots,m)$,

(C3) and, occasionally, subject to *boundary constraints* $L \le \frac{w_k}{d_k} \le U\ (k=1,\ldots,n)$, with $0 \le L \le 1 \le U$.

The so-called *distance function G* is measuring the difference between the g-weights $g_k = w_k/d_k$ and 1. This function must satisfy the following regularity conditions: $G(.)$ is strictly convex and twice continuously differentiable (on the interior of its domain); $G(1)=0$, $G(.)\ge 0$; $G'(1)=0$ and $G''(1)=1$. The inverse of the function G' is called the *calibration function* $F(.)=G'^{-1}(.)$, whence $F(0)=1$.

Deville and Särndal (1992) have defined a slightly more general model, by incorporating an additional factor $/\ q_k\ (q_k>0)$ for element $k\ (\in s)$ in the objective function in (C1). I.e. to obtain their model we should replace (C1) with the following

(C1') Minimise the *distance* $\sum_{k=1}^{n} \frac{d_k}{q_k} G\left(\frac{w_k}{d_k}\right)$.

The ratio estimator in section I.D is a special case of an estimator for which the calibrated weights are based on the calibration problem (C1')–(C3), but not of (C1)–(C3); see Deville and Särndal (1992). As mentioned in section I.D we work throughout this text with the simplified model (C1)–(C3). It would however be straightforward to incorporate the factors q_k throughout; see section II.B for more details. Our software too can easily be extended: see IV.C.2.viii.

(C3) seems to indicate that the g-weights are scattered around 1 (a g-weight g_k equal to 1 means that no correction to the initial weight is needed, for element k). In order to make the above optimisation problem feasible, particularly by appropriate specification of the bounds L and U in (C3), a global a priori adjustment to the initial weights may be necessary, making the implicit assumption of g-weights that are scattered around 1 more plausible. We ignore this secondary problem in our theoretical exposition, but have incorporated such an overall correction factor in our software. This factor is called the *scale* (parameter); it is discussed further in section III.A.2.

The distance function G, or, equivalently, the calibration function F, can be chosen conveniently, considering practical properties of the resulting g-weights. Deville *et al* (1993) introduce four different "methods" corresponding to four different distance functions: (1) the *linear method* with *quadratic* distance function and linear calibration function, (2) the *raking ratio* or *multiplicative method* with exponential calibration function, (3) the *truncated linear method* with *quadratic* distance function and linear calibration function, and (4) the *logit method* with *logistic* calibration function. An overview is presented in table 2.1, together with some properties and the corresponding calibration functions. The following notation is used: $\Re = (-\infty,+\infty)$; $\Re_0^+ = (0,+\infty)$. Deville and Särndal (1992) have considered a few more distance functions.

Table 2.1 *Distance function G, additional constraints (C3) and calibration function F for four calibration methods*: **(1)** *linear method;* **(2)** *multiplicative method;***(3)** *truncated linear method;* **(4)** *logit method*

	Distance function $G(x)$	Additional constraint (C3)	Calibration function $F(u)$
(1)	$\frac{(x-1)^2}{2}$ for $x \in \Re$	*None*	$1+u$ for $u \in \Re$
(2)	$x\ln(x) - x + 1$ for $x \in \Re_0^+$ $-x+1$ for $x = 0$	*None*	e^u for $u \in \Re$
(3)	$\frac{(x-1)^2}{2}$ for $x \in \Re$	$x \in [L,U]$ with $0 \le L \le 1 \le U$	$1+u$ for $u \in [L-1, U-1]$ L for $u = L-1$ U for $u = U-1$
(4)	$\left[(x-L)\ln\frac{x-L}{1-L} + (U-x)\ln\frac{U-x}{U-1}\right]A^{-1}$ for $x \in (L,U)$ $\left[(U-L)\ln\frac{U-L}{U-1}\right]A^{-1}$ for $x \le L$ $\left[(U-L)\ln\frac{U-L}{1-L}\right]A^{-1}$ for $x \ge U$ with $0 \le L < 1 < U$	*None*	$\frac{L(U-1)+U(1-L)e^{Au}}{(U-1)+(1-L)e^{Au}}$ for $u \in \Re$ where $A = \frac{U-L}{(U-1)(1-L)}$

Often a quadratic function is used, i.e. $G(x) = \frac{1}{2}(x-1)^2$; the calibration method is then said to be *linear*. Estimators based on this method are *generalised regression* (GREG) estimators. A disadvantage of the linear method is that the calibrated weights can be negative. Other choices of *G* can force the calibrated weights being positive. Under the linear method, the additional constraints (C3) can be used to restrict the g-weights: this is the truncated linear method (3). Notice that Calmar (Sautory, 1993) and Bascula (Nieuwenbroek, 1997) are using different algorithms to implement (C3); our implementation is as in Calmar.

The calibration methods (1) to (4) are compared in figures 2.1 and 2.2. Figure 2.1 shows the distance functions *G*. Figure 2.2 shows the calibration functions *F*. We have set L = .15 and U = 1.4. Notice that the domain of the logit distance function is $[L,U]$. We will show later that the *g*-weights g_k are equal to $F(u_k)$, where u_k depends on the auxiliary information and the initial weight for sample element *k*. Thus the shape of the calibration function determines the adjustments that are made by the calibration technique. The figure shows that methods (1), (3) and (4) are very close to each other within some interval for *u*. Outside this interval methods (3) and (4) are truncating the adjustments factors. Method (4) is doing this more smoothly than method (3). Method (2) tends to shift the *g*-weights upward in a systematic way, compared with all other methods; the *u*-range where the value of its calibration function is close to that for the other methods is rather small. It may be expected that the results from methods (1), (3) and (4) are close, as far as the *g*-weights are not extreme (i.e. not to much different from the central value 1). The results from method (2) may be substantially different.

Fig. 2.1 *Comparing four types of distance functions G*

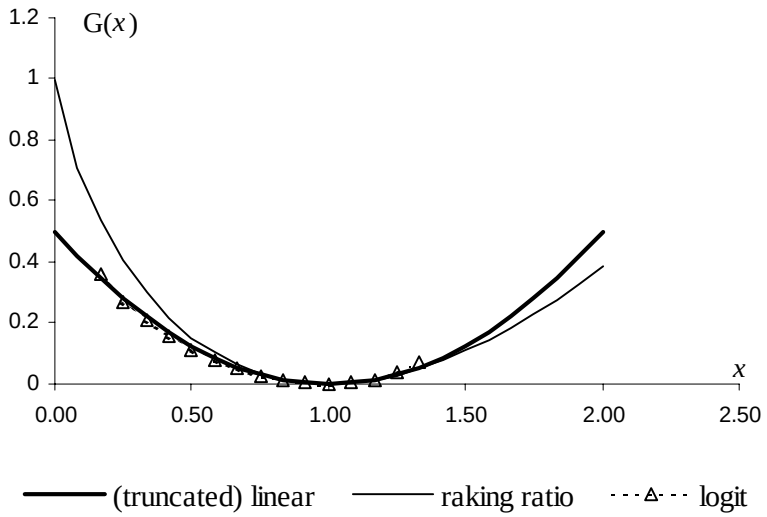
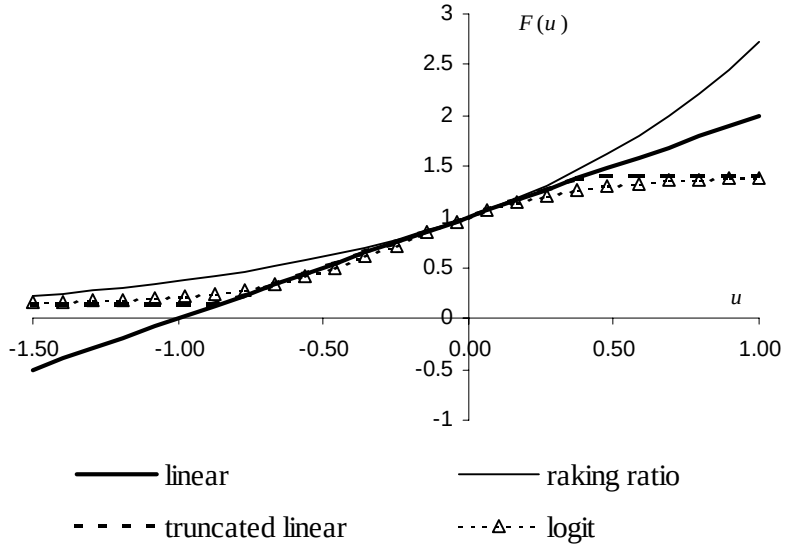


Fig. 2.2 *Comparing four types of calibration functions F*



II.B TERMINOLOGY AND NOTATION

We now introduce some matrix notation. This is used extensively in the sequel, since it makes the mathematical treatment more compact. Notice that a lot of mathematical formulations and derivations are similar to the mathematics for (linear) regression methodology. So the reader may not completely be unaware of the following notations. Anyway, we would like to encourage the reader to learn to read the mathematics in the rest of this report, as we believe that it may help him/her to understand more thoroughly the procedures of generalised calibration.

We define:

- the *auxiliary m*-vector for the *k*-th sample element $\mathbf{x}_k = (x_{k1},...,x_{km})^T$;
- the *n*-vector of *initial* (*sampling*) *weights* $\mathbf{d} = (d_1,...,d_n)^T$;
- the *n*-vector of *calibrated weights* $\mathbf{w} = (w_1,...,w_n)^T$;
- the *n*-vector of *g-weights* $\mathbf{g} = (g_1,...,g_n)^T$;
- the $n \times m$ (*calibration*) *design matrix* $\mathbf{X} = (\mathbf{X}_1,...,\mathbf{X}_m)$, where $\mathbf{X}_j = (x_{1j},...,x_{nj})^T$ is the *j*-th column (i.e. the vector of *n* measurements on the *j*-th calibration variable) and $\mathbf{x}_k^T$ is the *k*-th row;
- the *m*-vector of *population totals* $\mathbf{t} = (t_1,...,t_m)^T$;
- the *n*-vector of 1's: $\mathbf{1}_n = (1,...,1)^T$;
- the $n \times n$ *initial weights matrix* $\mathbf{D} = diag(\mathbf{d})$ with *k*-th diagonal element d_k;
- the $n \times n$ *calibrated weights matrix* $\mathbf{W} = diag(\mathbf{w})$ with *k*-th diagonal element w_k;
- the $n \times n$ *g-weights matrix* $\mathbf{G} = diag(\mathbf{g})$ with *k*-th diagonal element g_k;
- a vector of zeros $\mathbf{0} = (0,...,0)^T$, whose length follows from the context;
- the *identity matrix* $\mathbf{I}$, whose dimension follows from the context.

Notice that, for example, $diag(\mathbf{d})\mathbf{1}_n = \mathbf{D}\mathbf{1}_n = \mathbf{d}$. Then we have $\mathbf{W} = \mathbf{DG} = \mathbf{GD}$ and $\mathbf{w} = \mathbf{W}\mathbf{1}_n = \mathbf{Dg} = \mathbf{Gd}$.

The calibration constraints (C2) can then be written as $\mathbf{X}^T\mathbf{w} = \mathbf{t}$, or $\mathbf{X}^T\mathbf{W}\mathbf{1}_n = \mathbf{t}$, or $\mathbf{X}^T\mathbf{Dg} = \mathbf{t}$, or finally $\breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}$, where $\breve{\mathbf{X}} = \mathbf{DX}$ is the *expanded design matrix,* i.e. the matrix with elements $\breve{x}_{ij} = d_i x_{ij} = \frac{x_{ij}}{\pi_i}$ (provided the initial weights are the sampling weights). These are all linear systems with *m* equations in *n* "variables" (either the calibrated weights or the g-weights). If the calibration system is *consistent*, then it has at least one solution $\mathbf{w}^*$ (or $\mathbf{g}^*$). This means that the auxiliary information is not over-identifying the weights, or that there is no contradiction in the calibration constraints. On the other hand, as we will see later, some of this information may be redundant. Formally, this means that the design matrix $\mathbf{X}$ need not be of full rank, although it will be assumed that there are more rows than columns in this matrix: $n \geq m$.

The distance or *objective function* in (C1) can be written shortly as $D(\mathbf{d},\mathbf{w})$, to emphasise that some distance, *D*, between the initial weights **d** and the calibrated weights **w**, is considered. Notice that $D(\mathbf{d},\mathbf{w})$ is a weighted sum of distance measures $G(w_k/d_k) = G(g_k)$; the *k*-th weight in this sum is the initial weight d_k. Thus: $D(\mathbf{d},\mathbf{w}) = \sum_{k\in s} d_k G(w_k/d_k) = \sum_{k\in s} d_k G(g_k)$. The quantity $G(g_k)$ measures the distance between the initial weight d_k and the calibrated weight w_k, or, as mentioned before,

between the *g*-weight g_k and 1. If we now write $G(\mathbf{g}) = \left(G(g_1),...,G(g_n)\right)^T$, then the objective function becomes $D(\mathbf{d},\mathbf{w}) = \mathbf{d}^T G(\mathbf{g})$.

Finally, the additional boundary constraints (C3) can be written shortly as $\mathbf{g} \in \Omega_B$, where Ω_B is the bounded subset $[L,U]^n$ of the Euclidean space $\Re^n$. If, more generally, Ω_B is allowed to be any, bounded or unbounded, subset of $\Re^n$, making it explicit in which area the *g*-weights are looked after, either implicitly through *D* or explicitly, then any calibration problem (C1-C2) or (C1-C3) can be written briefly as

$$\left\{\min D(\mathbf{d},\mathbf{w}); \mathbf{X}^T\mathbf{w} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}.$$

It follows from the above that the calibration problem alternatively can be formulated in terms of the g-weights:

$$\left\{\min \mathbf{d}^T G(\mathbf{g}); \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}.$$

More details about the calibration problem, and further specification of the set Ω_B is found in the next section II.C.

The extended model (C1')–(C3) (see section II.A) can also be formulated in matrix notation. We therefore introduce the matrix $\mathbf{Q} = diag(q_k)$, i.e. the diagonal matrix with *k*-th diagonal element q_k. Then $\mathbf{Q}^{-1} = diag(1/q_k)$, and the objective function in (C1') can then be written as $\mathbf{d}^T\mathbf{Q}^{-1}G(\mathbf{g})$. So the model (C1')–(C3) can be formulated in matrix notation as

$$\left\{\min \mathbf{d}^T\mathbf{Q}^{-1}G(\mathbf{g}); \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}.$$

II.C EXISTENCE AND UNIQUENESS OF A SOLUTION TO THE CALIBRATION PROBLEM – THE BOUNDARY PROBLEM

In the previous section, we have formulated the generalised calibration problem as a *mathematical programming problem*, in terms of the g-weights or correction factors **g**, as follows:

$$\left\{\min \mathbf{d}^T G(\mathbf{g}); \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\} \tag{II.1}$$

We also define $\Omega_C = \left\{\mathbf{g} \in \Re^n; \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}\right\}$; this does not depend on the calibration method. The set $\Omega_C \cap \Omega_B$ is the *feasible region* for the mathematical programming problem (II.1). For each of the calibration methods, we can easily specify the feasible region $\Omega_C \cap \Omega_B$ first, and, given $\Omega_C = \left\{\mathbf{g} \in \Re^n; \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}\right\}$, we then derive the set Ω_B. Table 2.2 lists the sets $\Omega_C \cap \Omega_B$ and Ω_B for the four calibration methods (1) to (4).

Table 2.2 *The feasible region $\Omega_C \cap \Omega_B$ and the set Ω_B for four calibration methods*: **(1)** *linear method;* **(2)** *multiplicative method;* **(3)** *truncated linear method;* **(4)** *logit method*

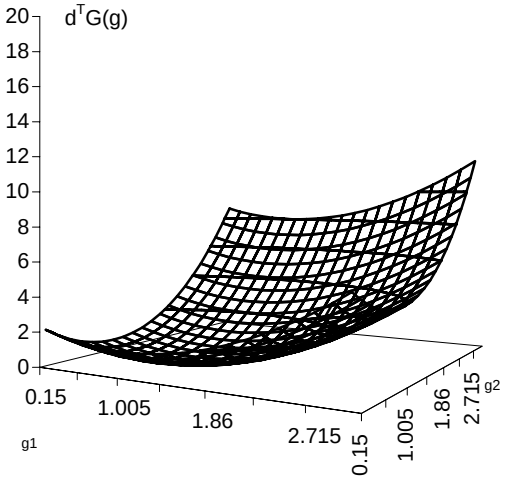
	$\Omega_C \cap \Omega_B$	Ω_B
(1)	$\left\{\mathbf{g} \in \Re^n; \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}\right\}$	$\Re^n$
(2)	$\left\{\mathbf{g} \in \Re^n; \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}, \mathbf{g} \geq \mathbf{0}\right\}$	$\left\{\mathbf{g} \in \Re^n; \mathbf{g} \geq \mathbf{0}\right\} = [0,+\infty)^n = \Re^{+n}$
(3)	$\left\{\mathbf{g} \in \Re^n; \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}, L\mathbf{1}_n \leq \mathbf{g} \leq U\mathbf{1}_n\right\}$	$\left\{\mathbf{g} \in \Re^n; L\mathbf{1}_n \leq \mathbf{g} \leq U\mathbf{1}_n\right\} = [L,U]^n$
(4)	$\left\{\mathbf{g} \in \Re^n; \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}, L\mathbf{1}_n \leq \mathbf{g} \leq U\mathbf{1}_n\right\}$	$\left\{\mathbf{g} \in \Re^n; L\mathbf{1}_n \leq \mathbf{g} \leq U\mathbf{1}_n\right\} = [L,U]^n$

The feasible region $\Omega_C \cap \Omega_B$ is defined by a (finite) set of *linear equality* and/or *inequality constraints*. Hence, this set is *convex*. The feasible region is also *closed*.

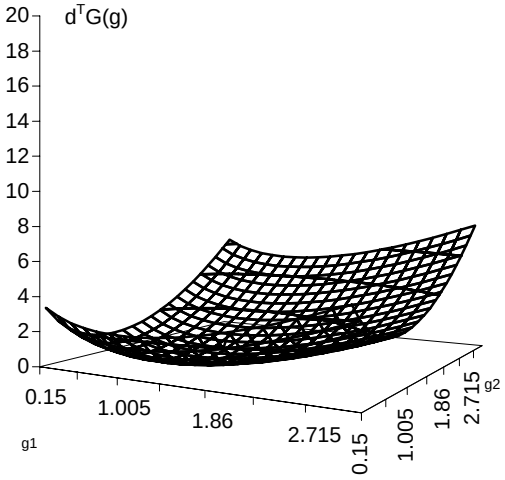
Since each linear equality constraint – a calibration constraint! – can equivalently be replaced with two inequality constraints, the feasible region $\Omega_C \cap \Omega_B$ is the intersection of a finite number of *half-spaces*, and therefore called a *polyhedron* (Cameron, 1985, p.31).

It follows from the definition of the (scalar-valued) distance functions *G* (see table 2.1), that, for any calibration method, the objective function $\mathbf{d}^T G(\mathbf{g})$ in (II.1) is *strictly convex* on the corresponding set Ω_B, and therefore also on the corresponding feasible region $\Omega_C \cap \Omega_B$. Moreover, it is easy to see that the reduced problem $\left\{\min \mathbf{d}^T G(\mathbf{g}); \mathbf{g} \in \Omega_B\right\}$ is a trivial one: the solution always exists, is unique, and is equal to $\mathbf{g} = \mathbf{1}_n$. Notice that we assume throughout that $0 \leq L \leq 1 \leq U$ for method (3) and $0 \leq L < 1 < U$ for method (4). It follows then that Ω_B never is empty. For *n* = 2 we have displayed, for each calibration method, in figure 2.3, the surface $\mathbf{d}^T G(\mathbf{g})$ on a set $[0.15, 3]^2$, which contains $\mathbf{1}_2$.

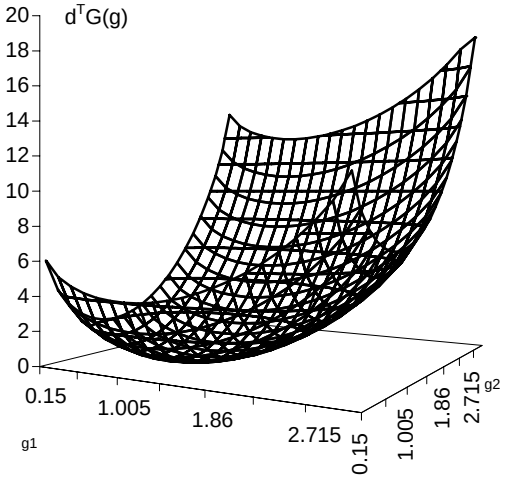
Fig. 2.3 *The surface* $\mathbf{d}^T G(\mathbf{g})$, *for* $n = 2$, *on the set* $[0.15, 3]^2$



(1) *linear method and* (3) *truncated linear method*



(2) *multiplicative method*



(4) *logit method*

The calibration system $\breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}$ can, in principle, be *inconsistent*. In other words, $\Omega_C = \left\{\mathbf{g} \in \Re^n; \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}\right\}$ can be empty. Then, the feasible region is empty and the optimisation problem has no solution at all; the calibration problem is then said to be *infeasible*. However, in practice, the calibration constraints will be carefully set up, resulting into a *consistent linear system* $\breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}$. So, from now on, Ω_C is assumed to be non-empty, or, equivalently, $\breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}$ is assumed to be consistent.

Although both the sets Ω_C and Ω_B are (assumed to be) non-empty, the feasible region $\Omega_C \cap \Omega_B$ can occasionally be empty. Then the optimisation problem (II.1) has no solution.

So let us now assume that $\Omega_C \cap \Omega_B$ is a non-empty feasible region, i.e. that the calibration problem is a *feasible* optimisation problem. Since $\mathbf{d}^T G(\mathbf{g})$ is strictly convex on the feasible region $\Omega_C \cap \Omega_B$, and since $\mathbf{g} = \mathbf{1}_n$ is always a solution of the reduced problem $\left\{\min \mathbf{d}^T G(\mathbf{g}); \mathbf{g} \in \Omega_B\right\}$, it easily follows that the convex programming problem (II.1) is *bounded*, which means that it has a finite optimal solution. Moreover, it follows immediately that this solution is unique. Notice that the calibration constraints do not necessarily have a unique solution. The solution of (II.1), if it exists, is the one that satisfies the calibration constraints and, in some sense, provides a minimal adjustment for the initial weight of each sample element.

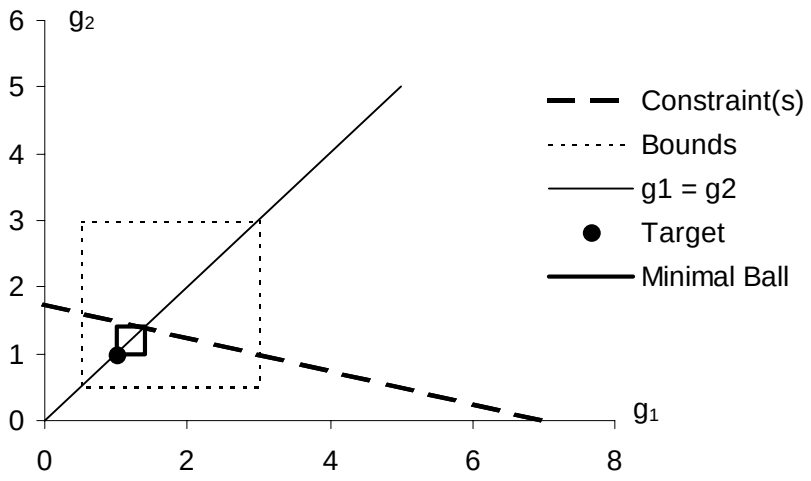
One practical problem remains to be discussed. We have assumed that Ω_C is non-empty. For the linear method this set is also the feasible region. Hence, if the calibration system is consistent, then a solution to (II.1) always exists if the linear method is used. For the multiplicative method, the assumption of non-empty feasible region implies that the calibration system not only must have a solution, but that at least one non-negative solution ($\mathbf{g} \geq \mathbf{0}$) exists. This is not necessarily true, although it is a desirable property of the g-weights. The existence of such a solution cannot be guaranteed; if it doesn't exist, the feasible region is empty. The problem of whether the feasible region is empty is even more difficult for the truncated linear and the logit method. Then, although it may be assumed that the calibration system is consistent, one has to choose the bounds L and U such that the feasible region is non-empty. In the statistical literature on generalised calibration, one argues that L and U should be chosen arbitrarily. If a solution exists for the chosen values, one might consider a possible "improvement" of L and U. "Improvement" should be understood as an increase in L and/or a decrease in U, in order to restrict the g-weights or adjustment factors further. This process of tightening $\left[L,U\right]^n$ is, according to the literature (Deville *et al*, 1993; Sautory, 1993), a trial and error procedure, and consequently can cost a lot of computer time, since for each choice of the bounds, the full calibration model (II.1) has to be solved, which is itself an iterative procedure. We here propose an alternative, more economic, approach.

For illustrative purposes, consider an extremely small (and unrealistic) calibration problem:

$$\begin{aligned}
&\left\{\min \mathbf{d}^T G(\mathbf{g}); \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\} \\
&= \left\{\min \begin{pmatrix}2 & 4\end{pmatrix}\begin{pmatrix}G(g_1)\\ G(g_2)\end{pmatrix}; \begin{pmatrix}1 & 4\end{pmatrix}\begin{pmatrix}g_1\\ g_2\end{pmatrix} = 7, L \leq g_1, g_2 \leq U\right\}. \qquad \text{(II.2)} \\
&= \left\{\min 2G(g_1) + 4G(g_2); g_1 + 4g_2 = 7, L \leq g_1, g_2 \leq U\right\}
\end{aligned}$$

Thus we have $n = 2$ (2 sample elements) and $m = 1$ (1 calibration constraint). The distance function G need not be specified. If we choose $L = 0.5$ and $U = 3$, then a graphical representation of the single calibration constraint, the set Ω_B, and the feasible region for the problem (II.2) is as in figure 2.4.

Fig. 2.4 *Calibration constraints and feasible region for problem* (II.2)



The calibration system is consistent, since there is only one calibration constraint; Ω_C is the straight line corresponding to this constraint. The big square bordered by a dashed line (labelled "Bounds" in the legend) is the set $\Omega_B = [L,U]^2 = [0.5,3]^2$. For this choice of L and U the feasible region is non-empty: it is the line segment on the straight line inside the square. By increasing L and decreasing U, many squares can be found such that the feasible region is non-empty. The small square in the graph is a minimal square, in the following sense: it is the smallest square that contains the so-called "*target point*" $\mathbf{1}_n = (1 \quad 1)^T$ and a minimum number of points (here only 1) on the calibration constraint Ω_C. The target point is introduced because of the assumption $L \le 1 \le U$, whence the square $\Omega_B = [L,U]^2$ must contain the target. The opposite corner point, on the calibration constraint Ω_C, is in some sense a point on Ω_C that is closest to the target. Notice that this point need not be on the line $g_1 = g_2$, in general. The distance between the target and the opposite corner point is measured by the *maximum-norm* (or L_∞ *-norm*). For this norm, the points at the same fixed distance from a fixed point are on a square, with the fixed point in the middle. In other words, a L_∞ *-ball* is a square. Once the "closest" point (or points) on the calibration constraint(s) is (or are) found, the largest L and the smallest U can easily be calculated as, respectively, the minimum of the co-ordinates of the target point and the closest point(s) on the calibration constraints, and the maximum of those co-ordinates. These values, L^* and U^* say, determine the *minimal ball* $[L^*,U^*]^2$.

We now generalise the technique introduced in the example. Using the maximum-norm, to find a point on the calibration constraints, as close as possible to the target point $\mathbf{1}_n$, we solve the following mathematical programming problem:

$$\left\{ \min \max_{1 \le k \le n} |g_k - 1| ; \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t} \right\}. \tag{II.3}$$

Suppose $\mathbf{g}_0 = (g_{01},\ldots,g_{0n})^T$ is a solution to (II.3), then we find L^* and U^* from:

$$\begin{aligned} L^* &= \min\{1, g_{01}, \ldots, g_{0n}\} \\ U^* &= \max\{1, g_{01}, \ldots, g_{0n}\} \end{aligned}. \qquad \text{(II.4)}$$

Notice that if (II.3) has a solution, then the original problem (II.1) also has a solution. Hence problem (II.3) can be used to find out whether the feasible region of the original problem, for appropriate L and U in the truncated linear and the logit method, is empty or not. Therefore, it can be used to check whether the calibration constraints are consistent or not.

L^* in (II.4) can be negative. To avoid negative L^* (and negative final g-weights), (II.3) can be modified into:

$$\left\{\min \max_{1\le k\le n} |g_k - 1|; \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \mathbf{g} \ge 0\right\}, \qquad \text{(II.5)}$$

but then existence of a solution to (II.3) does not guarantee existence of a solution to (II.5).

Strictly speaking, L^* in (II.4) can also be equal to 1. Then, when using the logit method, one should set L just a little bit smaller than 1, since otherwise the logit distance G is not well-defined. A similar remark holds for U^* and U.

It is interesting to mention that (II.3) (and similarly (II.5)) can be reformulated as a *linear programming (LP) problem*:

$$\left\{\min z\,; \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t},\, g_k - 1 \le z, 1 - g_k \le z\right\}. \qquad \text{(II.6)}$$

The *simplex algorithm* might be used to solve this LP problem, after a final transformation into a standardised form. Implementation of this algorithm is not really difficult, although pivoting is not a technique that has to be programmed daily. If one considers the implementation of this algorithm to solve the LP problem, then it should be investigated how data storage can be reduced. The *revised simplex algorithm* could be a solution (Cameron, 1985); Brickman (1988) gives a nice and illuminating discussion of the simplex algorithm, based on what is defined as *condensed (simplex) tableaux* (Brickman, 1988, p.9).

We have used the target point $\mathbf{1}_n$ in the above discussion. This does not restrict the applicability of our ideas, or of our software. Justification of this special target point is related to the *scale* parameter, introduced in section II.A. So any point in the n-dimensional Euclidian space could serve as target point.

We feel it would be very useful to solve (II.3) (or (II.6)) or (II.5) immediately after defining the calibration constraints. It would be great if that were possible in almost real-time. One would then have a clear indication of possible values for the lower bound L and the upper bound U, if these were to be set, which is the case if the logit or truncated linear method is finally chosen. Moreover, solving this problem gives a lot of information on the central problem (II.1) itself. We strongly believe that this information will help to set up the final calibration problem, from which the final g-weights will be obtained, and to make the statistician more confident about the solution of the calibration problem.

We have not yet programmed an algorithm to solve (II.3) or (II.5). However, for small problems one can use Microsoft Excel, which has a very powerful optimisation tool, called the Solver. We have used this to obtain the results for the above example. Figure 2.4 also has been created with Microsoft Excel.

II.D NUMERICAL SOLUTION TO THE CALIBRATION PROBLEM: THE BASIC ALGORITHM

Recall that the constraints $\mathbf{g} \in \Omega_B$ in (II.1) are rather implicit for methods (1), (2) and (4). To develop the basic algorithm for solving the calibration problem we will therefore ignore these constraints in the present section. So we consider now the *simplified calibration problem*

$$\left\{\min D(\mathbf{d},\mathbf{w});\ \mathbf{X}^T\mathbf{w} = \mathbf{t}\right\} \quad \text{or} \quad \left\{\min \mathbf{d}^T G(\mathbf{g});\ \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}\right\}, \tag{II.7}$$

which is a convex mathematical programming problem, with strictly convex objective function and linear equality constraints. A solution is assumed to exist (see section II.C).

The method of solving (II.7) is straightforward and well known, using the technique of the Lagrange multipliers. Let $\boldsymbol{\lambda} = (\lambda_1,...,\lambda_m)^T$ be the vector of Lagrange multipliers. Defining the Lagrangian function $L(\mathbf{w},\boldsymbol{\lambda}) = \sum_{k=1}^{n} d_k G\left(\frac{w_k}{d_k}\right) + \boldsymbol{\lambda}^T\left(\mathbf{X}^T\mathbf{w} - \mathbf{t}\right)$, it can be shown that the $(n+m) \times (n+m)$ system $\frac{\partial L}{\partial \mathbf{w}} = 0, \frac{\partial L}{\partial \boldsymbol{\lambda}} = 0$ transforms into the following $m \times m$ system in $\boldsymbol{\lambda}$:

$$\boldsymbol{\Phi}(\boldsymbol{\lambda}) = \sum_{k=1}^{n} d_k F\left(\mathbf{x}_k^T\boldsymbol{\lambda}\right)\mathbf{x}_k - \mathbf{t} = \mathbf{X}^T\mathbf{w}(\boldsymbol{\lambda}) - \mathbf{t} = \mathbf{0}, \tag{II.8}$$

where $\mathbf{w}(\boldsymbol{\lambda})$ is the n-vector of calibrated weights, with k-th component defined as:

$$w_k = w_k(\boldsymbol{\lambda}) = d_k F\left(\mathbf{x}_k^T\boldsymbol{\lambda}\right) = d_k g_k(\boldsymbol{\lambda}) = d_k g_k. \tag{II.9}$$

In matrix notation: $\mathbf{w}(\boldsymbol{\lambda}) = \mathbf{D}F(\mathbf{X}\boldsymbol{\lambda})$. The Lagrange multipliers $\boldsymbol{\lambda} = (\lambda_1,...,\lambda_m)^T$ can be obtained by solving iteratively the non-linear system $\boldsymbol{\Phi}(\boldsymbol{\lambda}) = \mathbf{0}$ of m transformed calibration constraints. The formula (II.9) allows calculating the final calibrated weights, once the Lagrange multipliers are found. Notice that $g_k = F\left(\mathbf{x}_k^T\boldsymbol{\lambda}\right) = w_k / d_k$, or $\mathbf{g}(\boldsymbol{\lambda}) = F(\mathbf{X}\boldsymbol{\lambda})$. To solve iteratively the system of non-linear equations (II.8), the Newton-Raphson method is used. This is based on a first order Taylor series expansion of the left hand sides $\boldsymbol{\Phi}(\boldsymbol{\lambda})$, resulting into the following set of *updating equations*:

$$\boldsymbol{\lambda}^{(l)} = \boldsymbol{\lambda}^{(l-1)} - \left(\boldsymbol{\Phi}'\left(\boldsymbol{\lambda}^{(l-1)}\right)\right)^{-}\boldsymbol{\Phi}\left(\boldsymbol{\lambda}^{(l-1)}\right), \tag{II.10}$$

which allows calculating successive updates $\boldsymbol{\lambda}^{(l)}$ (l =1, 2, …) for the Lagrange multipliers $\boldsymbol{\lambda}$, starting from initial values $\boldsymbol{\lambda}^{(0)}$. It is convenient to start from $\boldsymbol{\lambda}^{(0)} = \mathbf{0}$, as it will be seen later. At each iteration l = 1,2,…, $\mathbf{w}\left(\boldsymbol{\lambda}^{(l-1)}\right) = \mathbf{D}F\left(\mathbf{X}\boldsymbol{\lambda}^{(l-1)}\right)$ is first evaluated, from which then the m-vector $\boldsymbol{\Phi}\left(\boldsymbol{\lambda}^{(l-1)}\right) = \mathbf{X}^T\mathbf{w}\left(\boldsymbol{\lambda}^{(l-1)}\right) - \mathbf{t}$ and the matrix $\boldsymbol{\Phi}'\left(\boldsymbol{\lambda}^{(l-1)}\right) = \mathbf{X}^T\mathbf{W}\left(\boldsymbol{\lambda}^{(l-1)}\right)\mathbf{X}$, where $\mathbf{W}\left(\boldsymbol{\lambda}^{(l-1)}\right) = diag\left(\mathbf{w}\left(\boldsymbol{\lambda}^{(l-1)}\right)\right)$ (see Proposition II.4 in section II.E.4), are calculated. Evaluation of equation (II.10) involves computation of a *g-inverse* $\left(\boldsymbol{\Phi}'\left(\boldsymbol{\lambda}^{(l-1)}\right)\right)^{-}$ of the $m \times m$ matrix $\boldsymbol{\Phi}'\left(\boldsymbol{\lambda}^{(l-1)}\right)$. The usage of g-inverses turns out to be a very efficient mathematical device in practice. We comment on this in the next paragraphs.

The following general algorithm is implemented in our SPSS module g-CALIB-S:

BASIC ALGORITHM

Step i Initialise the Lagrange multipliers: $\boldsymbol{\lambda}^{(0)} = \mathbf{0}$. The initial value of the calibrated weights is $\mathbf{w}^{(0)} = \mathbf{D}F\left(\mathbf{X}\boldsymbol{\lambda}^{(0)}\right) = \mathbf{D}\mathbf{1}_n = \mathbf{d}$.

Step ii Calculate the first update of the Lagrange multipliers from formula (II.10): $\boldsymbol{\lambda}^{(1)} = -\left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-}\left(\mathbf{X}^T\mathbf{d} - \mathbf{t}\right)$, and calculate the first updated value of the calibrated weight vector $\mathbf{w}^{(1)} = \mathbf{D}F\left(\mathbf{X}\boldsymbol{\lambda}^{(1)}\right)$. Let $l = 1$.

Step iii If convergence is attained, then go to Step vi, otherwise continue with Step iv.

Step iv Let $\mathbf{W}^{(l)} = diag\left(\mathbf{w}^{(l)}\right)$. Calculate $\boldsymbol{\Phi}^{(l)} = \mathbf{X}^T\mathbf{w}^{(l)} - \mathbf{t}$ and $\boldsymbol{\Phi}'^{(l)} = \mathbf{X}^T\mathbf{W}^{(l)}\mathbf{X}$.

Step v Set l to $l + 1$. Calculate $\boldsymbol{\lambda}^{(l)} = \boldsymbol{\lambda}^{(l-1)} - \left(\boldsymbol{\Phi}'^{(l-1)}\right)^{-}\boldsymbol{\Phi}^{(l-1)}$, or $\boldsymbol{\lambda}^{(l)} = \boldsymbol{\lambda}^{(l-1)} - \left(\mathbf{X}^T\mathbf{W}^{(l-1)}\mathbf{X}\right)^{-}\left(\mathbf{X}^T\mathbf{w}^{(l-1)} - \mathbf{t}\right)$. The new update for the calibrated weights is $\mathbf{w}^{(l)} = \mathbf{D}F\left(\mathbf{X}\boldsymbol{\lambda}^{(l)}\right)$. Return to Step iii.

Step vi The final solution is $\mathbf{w}^{*} = \mathbf{w}^{(l)} = \mathbf{D}F\left(\mathbf{X}\boldsymbol{\lambda}^{(l)}\right) = \mathbf{D}\mathbf{g}\left(\boldsymbol{\lambda}^{(l)}\right)$.

Converge is attained if absolute change in successive updates of the g-weights is smaller than a pre-specified *tolerance level* $\varepsilon > 0$, i.e. if $\max_{1\le k\le n}\left|g_k^{(l)} - g_k^{(l-1)}\right| \le \varepsilon$. This maximum-norm criterion is implemented in g-CALIB-S. Alternative convergence criteria could be implemented; e.g. $\max_{1\le k\le n}\left|\frac{w_k^{(l)} - w_k^{(l-1)}}{w_k^{(l-1)}}\right| \le \varepsilon$, or $\left\|\left(\mathbf{W}^{(l)} - \mathbf{W}^{(l-1)}\right)\left(\mathbf{W}^{(l-1)}\right)^{-}\mathbf{1}_n\right\| \le \varepsilon$, where $\|.\|$ is the Euclidean norm. We have not compared the performance of the algorithm for different convergence criteria. The maximum-norm criterion (for the g-weights) is also implemented in Calmar (Sautory, 1993).

Before we proceed with the inclusion of the additional constraints $\mathbf{g} \in \Omega_B$, resulting into an extended algorithm, we notice that two matrices involved in the calculation are generally not uniquely determined. First, there is the design matrix **X**. The reader will know from linear regression that a design matrix for a given regression problem can take different forms; we here have a similar problem. Second, as we discuss in section II.E, a g-inverse of a given matrix is generally not unique. The next section II.E is completely devoted to a very technical in-depth treatment of these identification problems. More specifically, we will demonstrate that the algorithm is invariant for both the choice of the design matrix, and for the way of computing g-inverses.

II.E MATHEMATICAL JUSTIFICATION

II.E.1 Preliminaries, and results from linear regression theory

The theory of generalised calibration is in many respects similar to the theory of linear regression, which in turn uses a lot of results from matrix theory and the theory of linear transformations. In the next paragraphs we will justify the algorithm to solve the generalised calibration problem. As it will be seen, knowledge of matrix-based theory of linear regression will be very helpful. So we repeat here below basic concepts from matrix theory and linear regression. We consider real matrices only.

Linear regression theory is also useful in advanced theoretical research on the properties (weaknesses and strengths) of calibration estimators. See for instance Chambers and Skinner (1999) , where it is argued that "*the choice of calibration constraints is synonymous with an implicit linear model specification for the regression of the survey variable on the auxiliary variables defining these constraints*". Deville (2000) chooses for a treatment based on *generalised linear modelling* to justify calibration as a technique to correct for non-response. Generalised linear modelling (Francis *et al*, 1993; Lindsey, 1997) has a lot in common with traditional linear regression modelling.

NOTATION AND BASIC MATRIX RESULTS

For any $n \times m$ matrix $\mathbf{A}$, $M(\mathbf{A})$ denotes the subspace of $\Re^n$ generated by the columns of $\mathbf{A}$; it is called the *range space* or *column space* of $\mathbf{A}$. The *dimension* of this subspace is the *rank* $r(\mathbf{A})$ of $\mathbf{A}$; this is at most m if $n \geq m$.

A *generalised inverse* or *g-inverse* of a matrix $\mathbf{A}$ is any matrix $\mathbf{B}$ that satisfies $\mathbf{ABA} = \mathbf{A}$; a g-inverse of a matrix $\mathbf{A}$ is usually denoted as $\mathbf{A}^-$. A *reflexive g-inverse* of $\mathbf{A}$ is a g-inverse that also satisfies $\mathbf{A}^-\mathbf{A}\mathbf{A}^- = \mathbf{A}^-$. The *Moore-Penrose inverse* of $\mathbf{A}$ is the reflexive g-inverse, denoted $\mathbf{A}^+$, for which both $\mathbf{A}\mathbf{A}^+$ and $\mathbf{A}^+\mathbf{A}$ are symmetric. The Moore-Penrose inverse always exists and is unique. Other g-inverses are not necessarily unique. The Moore-Penrose inverse, as well as any other g-inverse, is the *ordinary inverse*, denoted $\mathbf{A}^{-1}$, if $\mathbf{A}$ is *square* ($n = m$) and has full rank: $r(\mathbf{A}) = n = m$. The Moore-Penrose inverse can often be obtained directly (through a simple matrix function) in software packages (SPSS, 1999b; SAS, 1990).

An important general result for g-inverses is the following (Rao, 1972; (vi)(c) in 1b.5):

Invariance property of g-inverses
For non-null matrices $\mathbf{B}$ and $\mathbf{C}$, $\mathbf{B}\mathbf{A}^-\mathbf{C}$ is invariant for any choice of the g-inverse $\mathbf{A}^-$ iff $M(\mathbf{B}^T) \subset M(\mathbf{A}^T)$ and $M(\mathbf{C}) \subset M(\mathbf{A})$.

In the discussion here below we recall a lot of results from linear regression theory. Following Rao (1972), we use the short notation $(\mathbf{y}, \mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$ for the fundamental linear regression problem in which $\mathbf{y}$ has the *expectation* $E(\mathbf{y}) = \mathbf{X}\boldsymbol{\beta}$ and *dispersion matrix* $D(\mathbf{y}) = \sigma^2\mathbf{I}$ ($\sigma > 0$). Similarly, the generalised linear regression problem is denoted as $(\mathbf{y}, \mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a *positive definite* (pd) dispersion matrix. The dispersion matrix thus has an ordinary inverse $\boldsymbol{\Sigma}^{-1}$; the latter then has a unique pd *square root* $\boldsymbol{\Sigma}^{-\frac{1}{2}}$, which is the unique matrix that satisfies $\boldsymbol{\Sigma}^{-\frac{1}{2}}\boldsymbol{\Sigma}^{-\frac{1}{2}} = \boldsymbol{\Sigma}^{-1}$. Of course $\boldsymbol{\Sigma}$ itself has a unique pd square root $\boldsymbol{\Sigma}^{\frac{1}{2}}$. Notice that $\boldsymbol{\Sigma}^{-\frac{1}{2}} = \left(\boldsymbol{\Sigma}^{\frac{1}{2}}\right)^{-1}$.

ORDINARY LEAST SQUARES (OLS) ESTIMATION

Consider the linear regression problem $(\mathbf{y}, \mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$. The following results are well known; for more details we refer to Rao (1972; 1b.5-6, 1c.4, 4a.6, 4i.4).

(i) The system of linear equations $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$ is not necessarily *consistent*. So an exact solution is not being searched for, but instead we are interested in the OLS solution, which is the solution $\boldsymbol{\beta}$ that minimises the *quadratic form* $(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})^T(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})$.

(ii) The system of *normal equations* $\mathbf{X}^T\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\mathbf{y}$ is always consistent. The rank of $\mathbf{X}^T\mathbf{X}$ satisfies $r(\mathbf{X}^T\mathbf{X}) = r(\mathbf{X}^T)$; also $M(\mathbf{X}^T\mathbf{X}) = M(\mathbf{X}^T) \subset \Re^m$.

(iii) A solution to the set of normal equations is $\hat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^- \mathbf{X}^T\mathbf{y}$, which is an OLS solution to the regression problem. This solution is unique only if the design matrix $\mathbf{X}$ has full rank $r(\mathbf{X}) = m$; then the only g-inverse $(\mathbf{X}^T\mathbf{X})^-$ is the ordinary inverse $(\mathbf{X}^T\mathbf{X})^{-1}$.

(iv) The dispersion of the estimator $\hat{\boldsymbol{\beta}}$ is $\sigma^2(\mathbf{X}^T\mathbf{X})^-$. This shows that the g-inverse $(\mathbf{X}^T\mathbf{X})^-$ has statistical significance.

(v) The matrix $\mathbf{X}(\mathbf{X}^T\mathbf{X})^-\mathbf{X}^T = \mathbf{P}$ is invariant for any choice of the g-inverse $(\mathbf{X}^T\mathbf{X})^-$. Hence the OLS estimator $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}} = \mathbf{P}\mathbf{y}$ for $\mathbf{y}$ is unique.

(vi) The matrix $\mathbf{P}$ is *idempotent* ($\mathbf{P}^2 = \mathbf{P}$) and *symmetric* ($\mathbf{P}^T = \mathbf{P}$), so $\mathbf{P}$ is an *orthogonal projection* matrix. The estimator $\hat{\mathbf{y}} = \mathbf{P}\mathbf{y}$ is the projection of $\mathbf{y}$ on the subspace $M(\mathbf{X}) = M(\mathbf{P}) \subset \Re^n$ along the *orthogonal subspace* $M(\mathbf{I} - \mathbf{P}) \subset \Re^n$. The latter space contains the residual vector $\mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{P})\mathbf{y}$. This establishes a geometric interpretation of the regression problem.

GENERALISED LEAST SQUARES (GLS) ESTIMATION

Now, consider the generalised linear regression problem $(\mathbf{y}, \mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma})$. The dispersion matrix $\boldsymbol{\Sigma}$ is symmetric, but not necessarily diagonal, which means that the observations $y_1, ..., y_n$ can be correlated. For notational convenience we will assume that $\boldsymbol{\Sigma}$ is pd (non-singular). This problem can be transformed into the fundamental linear regression problem $(\mathbf{y}', \mathbf{X}'\boldsymbol{\beta}, \mathbf{I})$ (the fundamental regression problem with $\sigma^2 = 1$) by the linear transformation with matrix $\boldsymbol{\Sigma}^{-\frac{1}{2}}$, i.e. $\mathbf{y}' = \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{y}$ and $\mathbf{X}' = \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{X}$. The previous results (i-vi) are generalised as follows:

(vii) The system of linear equations $\mathbf{X}\boldsymbol{\beta} = \mathbf{y}$ is not necessarily consistent. So an exact solution is again not being searched for, but instead we are interested in the GLS solution, which is the solution $\boldsymbol{\beta}$ that minimises the quadratic form $(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})^T\boldsymbol{\Sigma}^{-1}(\mathbf{X}\boldsymbol{\beta} - \mathbf{y})$.

(viii) The system of normal equations $\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{y}$ is always consistent. The rank of $\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X}$ satisfies $r(\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X}) = r(\mathbf{X}^T)$; also $M(\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X}) = M(\mathbf{X}^T\boldsymbol{\Sigma}^{-\frac{1}{2}}) \subset \Re^m$.

(ix) A solution to the set of normal equations is $\hat{\boldsymbol{\beta}}_{GLS} = \left(\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X}\right)^{-}\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{y} = \left(\mathbf{X}'^T\,\mathbf{X}'\right)^{-}\mathbf{X}'^T\,\mathbf{y}'$, which is a GLS solution to the regression problem $(\mathbf{y}, \mathbf{X}\boldsymbol{\beta}, \boldsymbol{\Sigma})$. As in (iii) this solution is unique only if the design matrix $\mathbf{X}$ has full rank $r(\mathbf{X}) = m$.

(x) The dispersion of the estimator $\hat{\boldsymbol{\beta}}_{GLS}$ is $\left(\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X}\right)^{-}$.

(xi) The matrix $\mathbf{X}'\left(\mathbf{X}'^T\,\mathbf{X}'\right)^{-}\mathbf{X}'^T = \boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{X}\left(\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X}\right)^{-}\mathbf{X}^T\boldsymbol{\Sigma}^{-\frac{1}{2}} = \mathbf{P}'$ is invariant for any choice of the g-inverse $\left(\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X}\right)^{-}$. Obviously then $\boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{P}'\boldsymbol{\Sigma}^{-\frac{1}{2}} = \mathbf{X}\left(\mathbf{X}^T\boldsymbol{\Sigma}^{-1}\mathbf{X}\right)^{-}\mathbf{X}^T\boldsymbol{\Sigma}^{-1} = \mathbf{P}^*$ is invariant too. Hence the GLS estimator $\hat{\mathbf{y}}_{GLS} = \mathbf{X}\hat{\boldsymbol{\beta}}_{GLS} = \mathbf{P}^*\mathbf{y}$ for $\mathbf{y}$ is unique.

(xii) The matrix $\mathbf{P}'$ is idempotent and symmetric, so $\mathbf{P}'$ is an orthogonal projection onto the subspace $M(\mathbf{X}') = M\left(\boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{X}\right) = M(\mathbf{P}') \subset \Re^n$. The matrix $\mathbf{P}^*$ is idempotent (but generally not symmetric), so $\mathbf{P}^*$ is a projection onto the subspace $M(\mathbf{X})$. The estimator $\hat{\mathbf{y}}_{GLS} = \mathbf{P}^*\mathbf{y}$ is the projection of $\mathbf{y}$ on the subspace $M(\mathbf{X}) = M\left(\mathbf{P}^*\right) \subset \Re^n$ along the subspace $M\left(\mathbf{I} - \mathbf{P}^*\right) \subset \Re^n$. The latter space contains the residual vector $\mathbf{y} - \hat{\mathbf{y}}_{GLS} = \left(\mathbf{I} - \mathbf{P}^*\right)\mathbf{y}$. The subspaces $M\left(\mathbf{P}^*\right)$ and $M\left(\mathbf{I} - \mathbf{P}^*\right)$ are generally not orthogonal, since $\mathbf{P}^*$ is generally not symmetric!

(xiii) For any $\mathbf{y}$, we have $\mathbf{P}^*\mathbf{y} = \boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{P}'\boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{y} = \boldsymbol{\Sigma}^{\frac{1}{2}}\mathbf{P}'\mathbf{y}'$. This means that the orthogonal projection of $\mathbf{y}$ by $\mathbf{P}^*$ onto $M(\mathbf{X})$ can be obtained by successive application of the projection by $\mathbf{P}'$ of the linear transformation $\mathbf{y}'$ onto $M\left(\boldsymbol{\Sigma}^{-\frac{1}{2}}\mathbf{X}\right)$ and the linear (back) transformation by $\boldsymbol{\Sigma}^{\frac{1}{2}}$ of the latter projection $\mathbf{P}'\mathbf{y}'$.

II.E.2 Equivalent design matrices

A design matrix (in a linear regression problem as well as in a calibration problem) is not unique. However, the result of the regression of $\mathbf{y}$ on a given set of explanatory variables should not depend on the choice of the design matrix derived from those explanatory variables. This suggests the following definition.

Definition 1
Matrices $\mathbf{X}$ and $\mathbf{Z}$ are *equivalent design matrices* if and only if $M(\mathbf{X}) = M(\mathbf{Z})$. The symbol "$\approx$" denotes equivalence of design matrices. Hence $\mathbf{X} \approx \mathbf{Z}$ iff $M(\mathbf{X}) = M(\mathbf{Z})$.

In linear regression, the choice of the design matrix will have an effect on the interpretation of the regression coefficients $\boldsymbol{\beta}$.

Here is an example of three equivalent design matrices:

$$\begin{bmatrix} 1 & 1 & 0 & 2 \\ 1 & 1 & 0 & 7 \\ 1 & 0 & 1 & 4 \\ 1 & 0 & 1 & 5 \end{bmatrix} \approx \begin{bmatrix} 1 & 1 & 2 \\ 1 & 1 & 7 \\ 1 & 0 & 4 \\ 1 & 0 & 5 \end{bmatrix} \approx \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 3.5 \\ 0 & 1 & 2 \\ 0 & 1 & 2.5 \end{bmatrix}$$

Equivalent design matrices (for a given linear regression problem) may have different numbers of columns, but always have the same number of rows (which is the number of cases or observations in statistical problems). However, if $\mathbf{X} \approx \mathbf{Z}$, then $r(\mathbf{X}) = r(\mathbf{Z})$, which is the dimension of the range space $M(\mathbf{X}) = M(\mathbf{Z})$. Now, we will write $\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-}\mathbf{X}^T = \mathbf{P}_{\mathbf{X}}$ (instead of $\mathbf{P}$) for the projection on $M(\mathbf{X})$, given $\mathbf{X}$. Similarly we write $\mathbf{P}_{\mathbf{Z}}$; $\mathbf{P}_{\mathbf{X}}^{'} = \mathbf{P}_{\mathbf{X}'}$ (instead of $\mathbf{P}'$; see (xi) in the previous section) and $\mathbf{P}_{\mathbf{Z}}^{'} = \mathbf{P}_{\mathbf{Z}'}$; $\mathbf{P}_{\mathbf{X}}^{*}$ (instead of $\mathbf{P}^{*}$; see (xi) in the previous section) and $\mathbf{P}_{\mathbf{Z}}^{*}$. The following result then follows easily from the results (v) and (xi) on projection matrices in section II.E.1.

Proposition II.1

If $\mathbf{X} \approx \mathbf{Z}$ then $\mathbf{P}_{\mathbf{X}} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-}\mathbf{X}^T = \mathbf{Z}\left(\mathbf{Z}^T\mathbf{Z}\right)^{-}\mathbf{Z}^T = \mathbf{P}_{\mathbf{Z}}$, for any choice of the g-inverses involved.

Now let $\mathbf{\Sigma}$ be as in section II.E.1, and write $\mathbf{\Sigma}^{-\frac{1}{2}}M(\mathbf{X})$ for the image of the subspace $M(\mathbf{X})$ under the linear transformation $\mathbf{\Sigma}^{-\frac{1}{2}}$. Obviously $\mathbf{\Sigma}^{-\frac{1}{2}}M(\mathbf{X}) = M\left(\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{X}\right)$. Then it follows immediately from $M(\mathbf{X}) = M(\mathbf{Z})$ that $M\left(\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{X}\right) = M\left(\mathbf{\Sigma}^{-\frac{1}{2}}\mathbf{Z}\right)$. This leads to the following result.

Proposition II.2

If $\mathbf{X} \approx \mathbf{Z}$ and $\mathbf{\Sigma}$ is any square symmetric pd matrix, then $\mathbf{P}_{\mathbf{X}}^{'} = \mathbf{P}_{\mathbf{Z}}^{'}$ and $\mathbf{P}_{\mathbf{X}}^{*} = \mathbf{P}_{\mathbf{Z}}^{*}$, for any choice of the g-inverses involved in the computation of the projection matrices.

Propositions II.1 and II.2 are formulated in a most general way, i.e. they are not only valid in a regression context. We have recalled in the previous section more results from linear regression theory than we need here or in the remainder of this text. Such results should of course be of interest to any statistician, for whom this text has been written primarily. The reader with a statistical background should therefore feel more confident with the results presented here after going through the introductory section II.E.1.

The following result is important in our justification (see sections II.E.3 to II.E.5) of the algorithm to solve the generalised calibration problem. The result follows easily from proposition II.2.

Proposition II.3

Now we make a special choice for the matrix $\mathbf{\Sigma}$, i.e. $\mathbf{\Sigma} = \mathbf{W}^{-1}$, where $\mathbf{W}$ is any pd square diagonal weight matrix. Then for any pair of equivalent design matrices $\mathbf{X}$ and $\mathbf{Z}$, the following equality holds:

$$\mathbf{X}\left(\mathbf{X}^T\mathbf{W}\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{W} = \mathbf{P}_{\mathbf{X}}^{*} = \mathbf{P}_{\mathbf{Z}}^{*} = \mathbf{Z}\left(\mathbf{Z}^T\mathbf{W}\mathbf{Z}\right)^{-}\mathbf{Z}^T\mathbf{W}, \qquad \text{(II.11)}$$

for any choice of the g-inverses.

Proposition II.3 states a two-folded invariance property: invariance for the choice of the matrix $\mathbf{X}$ (the design matrix in the context of linear regression or calibration), and invariance for the way g-inverses are calculated. In the next sections II.E.3-5, we use this result to show that the algorithm to solve the generalised calibration problem has these invariance properties to, in each step of the iterative procedure. In section II.E.3, we treat the linear method, because it takes a special place in the general class of calibration methods: no iteration is required. Next, in section II.E.4 we return to calibration

problems of the simplified form (II.7) and the related Basic Algorithm. In section II.E.5, finally, we generalise the Basic Algorithm to the so-called Extended Algorithm, which is appropriate for solving the general calibration problem (II.1), and we discuss the invariance properties for the Extended Algorithm.

II.E.3 The linear case

The problem is to solve $\left\{\min \mathbf{d}^T G(\mathbf{g}); \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}\right\}$, with quadratic distance function G. Then $G'(x) = x - 1$, the inverse of which is $G'^{-1}(u) = F(u) = 1 + u$. Substitution of $F\left(\mathbf{x}_k^T \boldsymbol{\lambda}\right) = 1 + \mathbf{x}_k^T \boldsymbol{\lambda}$ into (II.8) makes that system linear in $\boldsymbol{\lambda}$: $\sum_{k=1}^{n} d_k \left(1 + \mathbf{x}_k^T \boldsymbol{\lambda}\right) \mathbf{x}_k - \mathbf{t} = \mathbf{X}^T \mathbf{d} + \left(\mathbf{X}^T \mathbf{D} \mathbf{X}\right) \boldsymbol{\lambda} - \mathbf{t} = \mathbf{0}$, where $\mathbf{D} = diag(\mathbf{d})$. Equivalently:

$$-\left(\mathbf{X}^T \mathbf{D} \mathbf{X}\right) \boldsymbol{\lambda} = \mathbf{X}^T \mathbf{d} - \mathbf{t}. \tag{II.12}$$

That this system is consistent actually follows from the assumed consistency of the original calibration system $\mathbf{X}^T \mathbf{w} = \mathbf{t}$. We then know from linear algebra (Rao, 1972) that

$$\boldsymbol{\lambda}^* = -\left(\mathbf{X}^T \mathbf{D} \mathbf{X}\right)^- \left(\mathbf{X}^T \mathbf{d} - \mathbf{t}\right) \tag{II.13}$$

is a solution, for any g-inverse $\left(\mathbf{X}^T \mathbf{D} \mathbf{X}\right)^-$. No iteration is necessary. Interestingly, the solution (II.13) is exactly the first update $\boldsymbol{\lambda}^{(1)}$ that would be obtained from application of (II.10), with $\boldsymbol{\lambda}^{(0)} = \mathbf{0}$, $\boldsymbol{\Phi}\left(\boldsymbol{\lambda}^{(0)}\right) = \mathbf{X}^T \mathbf{d} - \mathbf{t}$ and $\boldsymbol{\Phi}'\left(\boldsymbol{\lambda}^{(0)}\right) = \mathbf{X}^T \mathbf{D} \mathbf{X}$. Thus, for the linear method the iterative procedure already converges after one step, which is a well-known property in generalised calibration methodology.

The solution $\boldsymbol{\lambda}^*$ is generally not unique, e.g. when $\mathbf{X}$ has not full rank, since $\mathbf{X}^T \mathbf{D} \mathbf{X}$ is then singular and has infinitely many g-inverses. However, it follows directly from the general invariance property in section II.E.1, with $\mathbf{A} = \mathbf{X}^T \mathbf{D} \mathbf{X}$, $\mathbf{B} = \mathbf{X}$ and $\mathbf{C} = \mathbf{X}^T \mathbf{d} - \mathbf{t}$, that $\mathbf{X} \boldsymbol{\lambda}^* = -\mathbf{X}\left(\mathbf{X}^T \mathbf{D} \mathbf{X}\right)^- \left(\mathbf{X}^T \mathbf{d} - \mathbf{t}\right)$ is invariant for the choice of the g-inverse. The k-th component of the m-vector $\mathbf{X} \boldsymbol{\lambda}^*$ is $\mathbf{x}_k^T \boldsymbol{\lambda}^*$, hence the calibrated weights $w_k = d_k \left(1 + \mathbf{x}_k^T \boldsymbol{\lambda}^*\right)$ are invariant for the choice of the g-inverse. In matrix notation: $\mathbf{w}^* = \mathbf{D}\left(\mathbf{1}_n + \mathbf{X} \boldsymbol{\lambda}^*\right)$. If $\mathbf{X}$ does have full rank m, the g-inverse becomes the ordinary inverse $\left(\mathbf{X}^T \mathbf{D} \mathbf{X}\right)^{-1}$ (provided all initial weights are strictly positive, as assumed). In that case, the solution $\boldsymbol{\lambda}^*$ is unique too.

Moreover it follows from proposition II.3 that the calibrated weights $\mathbf{w}^* = \mathbf{D}\left(\mathbf{1}_n + \mathbf{X} \boldsymbol{\lambda}^*\right)$ are independent for the choice of the design matrix, for given calibration problem $\left\{\min \mathbf{d}^T G(\mathbf{g}); \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}\right\}$. Notice that the g-weights in matrix notation are written as $\mathbf{g}^* = \mathbf{1}_n + \mathbf{X} \boldsymbol{\lambda}^*$.

The reader must be aware of the fact that under the linear method the solution $\boldsymbol{\lambda}^*$, and hence $\mathbf{w}^*$ and $\mathbf{g}^*$ can have zero and/or negative components. This is a major drawback of the linear method, while, on the other hand, the method is very simple to implement, since no iteration is needed to compute the solution. Moreover, the resulting calibrated estimators of totals of survey variables are the well-known and interesting GREG estimators.

II.E.4 The simplified calibration problem

We now consider the general simplified calibration problem(II.7), as discussed in section II.D. Notice that the linear calibration problem is a special case. In section II.D we have introduced the Basic Algorithm to solve the simplified calibration problem. We now discuss some interesting properties of this algorithm.

Propositions II.4 and II.5 follow by straightforward algebra.

Proposition II.4
The $m \times m$ matrix $\boldsymbol{\Phi}'(\boldsymbol{\lambda})$ can be written as $\boldsymbol{\Phi}'(\boldsymbol{\lambda}) = \mathbf{X}^T\mathbf{W}(\boldsymbol{\lambda})\mathbf{X} = \mathbf{X}^T\mathbf{D}\mathbf{G}(\boldsymbol{\lambda})\mathbf{X}$, where $\mathbf{W}(\boldsymbol{\lambda}) = diag(\mathbf{w}(\boldsymbol{\lambda}))$ and $\mathbf{G}(\boldsymbol{\lambda}) = diag(\mathbf{g}(\boldsymbol{\lambda}))$. It follows that $\boldsymbol{\Phi}'(\boldsymbol{\lambda})$ is a symmetric matrix.

Proposition II.5
The initialisation $\boldsymbol{\lambda}^{(0)} = \mathbf{0}$ implies:

(i) $g_k(\boldsymbol{\lambda}^{(0)}) = F(\mathbf{x}_k^T\boldsymbol{\lambda}^{(0)}) = 1$, or $\mathbf{g}(\boldsymbol{\lambda}^{(0)}) = F(\mathbf{X}\boldsymbol{\lambda}^{(0)}) = \mathbf{1}_m$;

(ii) $\mathbf{w}(\boldsymbol{\lambda}^{(0)}) = \mathbf{d}$, or $\mathbf{W}(\boldsymbol{\lambda}^{(0)}) = \mathbf{D}$;

(iii) $\boldsymbol{\Phi}(\boldsymbol{\lambda}^{(0)}) = \mathbf{X}^T\mathbf{d} - \mathbf{t} = \mathbf{t}_\pi - \mathbf{t}$;

(iv) $\boldsymbol{\Phi}'(\boldsymbol{\lambda}^{(0)}) = \mathbf{X}^T\mathbf{D}\mathbf{X}$, where $\mathbf{D} = diag(\mathbf{d})$.

The notation $\mathbf{t}_\pi = \mathbf{X}^T\mathbf{d}$ usually denotes the so-called π-estimator or Horvitz-Thompson estimator for the calibration totals $\mathbf{t}$. This assumes that $\mathbf{d}$ is the vector of sampling weights, i.e. $d_k = \pi_k^{-1}$. However, $\mathbf{d}$ may be any vector of initial weights, so $\mathbf{t}_\pi$ can be any vector of initial estimates of the calibration totals $\mathbf{t}$. It is always assumed that the initial weights are all positive, i.e. probability sampling is assumed throughout the text. In other words, we assume that **D** is pd.

Proposition II.6
If $\mathbf{w}(\boldsymbol{\lambda}) > \mathbf{0}$, i.e. $\mathbf{W}(\boldsymbol{\lambda})$ is pd, then $\boldsymbol{\Phi}'(\boldsymbol{\lambda})$ is positive semi-definite (psd) with rank $r(\boldsymbol{\Phi}'(\boldsymbol{\lambda})) = r(\mathbf{X}) \le m$. If both $\mathbf{w}(\boldsymbol{\lambda}) > \mathbf{0}$ and $\mathbf{X}$ has full rank, i.e. $r(\mathbf{X}) = m$, then $\boldsymbol{\Phi}'(\boldsymbol{\lambda})$ is pd.

If the rank of $\mathbf{X}$ is maximal, then no auxiliary information is redundant. The advantage of working with such a calibration design matrix is that $\boldsymbol{\Phi}'(\boldsymbol{\lambda})$ then has an ordinary inverse $(\boldsymbol{\Phi}'(\boldsymbol{\lambda}))^{-1}$. However, although it is always possible, it is usually more difficult to construct a full rank design matrix. Therefore we focus on the situation were $r(\mathbf{X}) < m$. On the other hand, $\mathbf{W}(\boldsymbol{\lambda})$ is always assumed to be pd, which means that none of its diagonal elements is zero or negative.

Theorem II.1

Suppose $\mathbf{D}$ *and* $\mathbf{W}(\boldsymbol{\lambda})$ *are pd, and let* $\boldsymbol{\lambda}^{(0)} = \mathbf{0}$. *Then, for given simplified calibration problem*(II.7), *the following expressions do not depend on the choice of the g-inverse involved or on the calibration design matrix* $\mathbf{X}$*:*

(i) $\mathbf{X}\left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{D}\mathbf{1}_n = \mathbf{X}\left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{d}$

$= \mathbf{X}\left(\mathbf{X}^T\mathbf{W}\left(\boldsymbol{\lambda}^{(0)}\right)\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{W}\left(\boldsymbol{\lambda}^{(0)}\right)\mathbf{1}_n = \mathbf{X}\left(\mathbf{X}^T\mathbf{W}\left(\boldsymbol{\lambda}^{(0)}\right)\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{w}\left(\boldsymbol{\lambda}^{(0)}\right)$;

(ii) $\mathbf{X}\left(\mathbf{X}^T\mathbf{W}(\boldsymbol{\lambda})\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{W}(\boldsymbol{\lambda})\mathbf{1}_n = \mathbf{X}\left(\mathbf{X}^T\mathbf{W}(\boldsymbol{\lambda})\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{w}(\boldsymbol{\lambda})$;

and, if the calibration constraints are consistent, this also holds for

(iii) $\mathbf{X}\left(\mathbf{X}^T\mathbf{W}(\boldsymbol{\lambda})\mathbf{X}\right)^{-}\mathbf{t}$ and $\mathbf{X}\left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-}\mathbf{t} = \mathbf{X}\left(\mathbf{X}^T\mathbf{W}\left(\boldsymbol{\lambda}^{(0)}\right)\mathbf{X}\right)^{-}\mathbf{t}$.

From proposition II.3 it immediately follows that invariance holds for the expressions (i) and (ii). To prove (iii), we assume that $\mathbf{w}^*$ is a solution, and rewrite the expression as

$$\begin{aligned}\mathbf{X}\left(\mathbf{X}^T\mathbf{W}(\boldsymbol{\lambda})\mathbf{X}\right)^{-}\mathbf{t} &= \mathbf{X}\left(\mathbf{X}^T\mathbf{W}(\boldsymbol{\lambda})\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{w}^* \\ &= \mathbf{X}\left(\mathbf{X}^T\mathbf{W}(\boldsymbol{\lambda})\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{W}(\boldsymbol{\lambda})\tilde{\mathbf{w}}\end{aligned}$$

where $\tilde{\mathbf{w}}$ is the vector such that $\mathbf{w}^* = \mathbf{W}(\boldsymbol{\lambda})\tilde{\mathbf{w}}$. The premise then follows with (ii). □

Theorem II.2

If $\mathbf{W}(\boldsymbol{\lambda})$ *is pd, if the calibration constraints are consistent, and if* $\boldsymbol{\lambda}^{(0)} = \mathbf{0}$, *then at each step in the iteration the updated estimates of calibrated and g-weights are invariant for both the choice of the g-inverse used in the updating formula, and for the choice of the calibration design matrix* $\mathbf{X}$, *for fixed simplified calibration problem.*
If moreover the iterative procedure based on (II.10) *converges to* $\boldsymbol{\lambda}^*$, *then the solution* $\mathbf{w}\left(\boldsymbol{\lambda}^*\right) = \mathbf{w}^*$ *is invariant as well.*

We can write, from equation (II.8) and the updating equations (II.10), and from proposition II.4:

$$\begin{aligned}\mathbf{X}\boldsymbol{\lambda}^{(l+1)} &= \mathbf{X}\boldsymbol{\lambda}^{(l)} - \mathbf{X}\left(\boldsymbol{\Phi}'\left(\boldsymbol{\lambda}^{(l)}\right)\right)^{-}\boldsymbol{\Phi}\left(\boldsymbol{\lambda}^{(l)}\right) \\ &= \mathbf{X}\boldsymbol{\lambda}^{(l)} - \mathbf{X}\left(\mathbf{X}^T\mathbf{W}\left(\boldsymbol{\lambda}^{(l)}\right)\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{w}\left(\boldsymbol{\lambda}^{(l)}\right) + \mathbf{X}\left(\mathbf{X}^T\mathbf{W}\left(\boldsymbol{\lambda}^{(l)}\right)\mathbf{X}\right)^{-}\mathbf{t}.\end{aligned}$$

Notice that $\mathbf{X}\boldsymbol{\lambda}^{(0)} = \mathbf{0}$. The terms in the right hand side are invariant for $l = 0$, because of theorem II.1 (i) and (iii). So $\mathbf{X}\boldsymbol{\lambda}^{(1)}$ is invariant. Assuming then that $\mathbf{X}\boldsymbol{\lambda}^{(l)}$ is invariant, theorem II.1 (ii) and (iii) imply that the right hand side in the expression for $\mathbf{X}\boldsymbol{\lambda}^{(l+1)}$ is invariant, and so is $\mathbf{X}\boldsymbol{\lambda}^{(l+1)}$. Finally, $\mathbf{g}^{(l+1)} = \mathbf{1}_n + \mathbf{X}\boldsymbol{\lambda}^{(l+1)}$ and $\mathbf{w}^{(l+1)} = \mathbf{D}\left(\mathbf{1}_n + \mathbf{X}\boldsymbol{\lambda}^{(l+1)}\right)$ are invariant too. □

The linear method is covered by theorem II.2: it suffices to stop at $l = 1$, and therefore only $\mathbf{W}\left(\boldsymbol{\lambda}^{(0)}\right) = \mathbf{D}$ need to be pd, which indeed follows from the assumption of positive initial weights.

It follows from theorem II.2 that we have the computationally interesting result that properties of the iterative procedure (convergence rate, the solution, …) do not depend on the way a g-inverse is calculated or on the calibration design matrix chosen to represent the simplified calibration problem. One can use therefore the Moore-Penrose inverse, which is available in the matrix language of SPSS® 9.0. Most packages that include matrix manipulation allow calculating g-inverses. SAS/IML® is another example. From a practical point of view it is interesting to be able to work with a design matrix with linear dependencies between the columns. It simplifies either the preparation of these matrices (if the software does not construct it automatically), or the implementation of its construction from originally observed variables. The fact that the calibration design matrix may have many equivalent representations is exploited utmost in our implementation of the generalised calibration methodology. More about this in the next chapters.

II.E.5 The general calibration problem: the Extended Algorithm

We now deal with the solution of the general calibration problem $\left\{\min D(\mathbf{d},\mathbf{w}); \mathbf{X}^T\mathbf{w} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}$, or $\left\{\min \mathbf{d}^T G(\mathbf{g}); \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}$, of which existence and uniqueness of a solution was already discussed in section II.C. We present here below the *Extended Algorithm* that is implemented in our software g-CALIB-S, and discuss invariance problems related to the use of g-inverses and the choice of the design matrix **X**. The algorithm is an adaptation of the Basic Algorithm in section II.D. "Truncation" of calibrated weights means that, if some updated values w_k are not between $d_k L$ and $d_k U$, where L and U follow from the specification of Ω_B, then, if $w_k < d_k L$ we set w_k equal to $d_k L$, and if $w_k > d_k U$ we set w_k equal to $d_k U$. We assume here that $L > 0$.

EXTENDED ALGORITHM

Step i	As in the Basic Algorithm.
Step ii	As in the Basic Algorithm.
Step iii	If necessary, truncate the update $\mathbf{w}^{(l)}$. The result is denoted as $\tilde{\mathbf{w}}^{(l)}$. Then $\tilde{\mathbf{w}}^{(l)} \in \Omega_B$. If convergence is attained, then go to Step vi, otherwise continue with Step iv.
Step iv	Let $\tilde{\mathbf{W}}^{(l)} = diag\left(\tilde{\mathbf{w}}^{(l)}\right)$. Calculate $\boldsymbol{\Phi}^{(l)} = \mathbf{X}^T\tilde{\mathbf{w}}^{(l)} - \mathbf{t}$ and $\boldsymbol{\Phi}'^{(l)} = \mathbf{X}^T\tilde{\mathbf{W}}^{(l)}\mathbf{X}$.
Step v	As in the Basic Algorithm.
Step vi	The final solution is $\mathbf{w}^* = \tilde{\mathbf{w}}^{(l)}$.

Converge is now attained if absolute change in successive truncated updates of the g-weights is smaller than a pre-specified *tolerance level* $\varepsilon > 0$, i.e. if $\max_{1\le k\le n}\left|\tilde{g}_k^{(l)} - \tilde{g}_k^{(l-1)}\right| \le \varepsilon$, where $\tilde{g}_k^{(l)} = \dfrac{\tilde{w}_k^{(l)}}{d_k}$ (k = 1, …, n).

We already know that, for $l = 1$, $\mathbf{w}^{(l)} = \mathbf{w}^{(1)}$ in Step ii satisfies the invariance properties. With truncation as in Step iii, $\tilde{\mathbf{W}}^{(l)} = diag\left(\tilde{\mathbf{w}}^{(l)}\right)$ is pd (since $L > 0$). Therefore, the update $\mathbf{w}^{(l)}$ in Step v

will satisfy the invariance properties, by theorem II.1. This completely proves invariance in the Extended Algorithm too.

Notice that, for the linear truncated method (4), the update $\mathbf{w}^{(l)}$ in Step v can be written as

$$\begin{aligned}\mathbf{w}^{(l)} &= \mathbf{D}\left(\mathbf{1}_n + \mathbf{X}\left(\boldsymbol{\lambda}^{(l-1)} - \left(\mathbf{X}^T\tilde{\mathbf{W}}^{(l-1)}\mathbf{X}\right)^{-}\left(\mathbf{X}^T\tilde{\mathbf{w}}^{(l-1)} - \mathbf{t}\right)\right)\right)\\ &= \mathbf{D}\left(\mathbf{1}_n + \mathbf{X}\boldsymbol{\lambda}^{(l-1)} - \mathbf{X}\left(\mathbf{X}^T\tilde{\mathbf{W}}^{(l-1)}\mathbf{X}\right)^{-}\left(\mathbf{X}^T\tilde{\mathbf{w}}^{(l-1)} - \mathbf{t}\right)\right)\\ &= \mathbf{w}^{(l-1)} - \mathbf{D}\mathbf{X}\left(\mathbf{X}^T\tilde{\mathbf{W}}^{(l-1)}\mathbf{X}\right)^{-}\left(\mathbf{X}^T\tilde{\mathbf{w}}^{(l-1)} - \mathbf{t}\right).\end{aligned}$$

Hence calculation of $\mathbf{w}^{(l)}$ involves both the non-truncated $\mathbf{w}^{(l-1)}$ and the truncated $\tilde{\mathbf{w}}^{(l-1)}$, or $\tilde{\mathbf{W}}^{(l-1)}$.

Chapter III

Generalised calibration in practice,

or

Back to the world of statistics

III.A REFINEMENTS OF THE GENERALISED CALIBRATION MODEL

III.A.1 Model formulae – Calibration strata

Wilkinson and Rogers (1973) have introduced a very convenient symbolic notation for describing the *linear structure* in regression models. Their model language and algebra is, for instance, extensively used in the software package GLIM, for fitting *generalised linear models* (GLMs) (Francis *et al*, 1993). See also Lindsey (1997) for systematic use of this language in applications. It was already stated that calibration models have a lot in common with linear regression models, which are a special class of GLMs, so that it turns out that the same symbolic language can be used to specify unambiguously the *linear structure*, i.e. the structure of the design matrix, *of the calibration model*. Such a specification in the language of Wilkinson and Rogers (1973) will be called a *model formula*. Notice that the structure of the design matrix is describing the structure of the calibration equations.

It must be noticed, however, that such a model formula does not specify how ultimately the design matrix is (to be) constructed. Rather, a *class of equivalent design matrices* (section II.E.2), or a class of equivalent systems of calibration equations, is defined by any model formula.

Consider qualitative calibration variables *A*, *B*, … and a quantitative calibration variable *Z*. The symbol « 1 » is used to denote the constant calibration variable (with value 1, or any other constant value). Suppose that calibration is on the marginal distribution of both *A* and *B* (in the population), then the linear structure of the calibration model can be described by the model formula *A* + *B*. It can easily be seen that the constant variable 1 may be included in the design matrix of such a calibration model. Hence *A* + *B* is equivalent to 1 + *A* + *B*. We prefer to include a column with constant value 1 in the design matrix, whenever it is possible, and so we also prefer to include the constant term 1 in the model formula, although this is not strictly necessary.

If we wish to calibrate on the joint distribution of *A* and *B* (in the population), then the appropriate model formula is *A***B*, or 1 + *A***B*. An equivalent model formula is 1 + *A* + *B* + *A*.*B*. This formula indicates that calibration is on the total population (size), on the population (size) in the categories of *A* (i.e. calibration on the marginal distribution of *A*), on the population (size) in the categories of *B* (i.e. calibration on the marginal distribution of *B*), and on the population (size) in the cells of a cross-classification of *A* with *B* (i.e. on the joint distribution of *A* and *B*). It is because of the fact that if calibration is on the joint distribution of *A* and *B*, then calibration is implicitly also on the marginal distribution of both *A* and *B*, and, moreover, also on the total population size, that the model formulae 1 + *A* + *B* + *A*.*B*, 1 + *A***B*, and *A***B* are equivalent. The notation *A*.*B* was used so far to symbolise the joint – and only the joint! – distribution of *A* and *B*, but, because of the above clarification, *A*.*B* and *A***B* can, from now on, be used interchangeably. This means that we can say that our calibration models are *hierarchical*: if higher order interaction terms are included, then all corresponding lower order interaction terms are included too.

Each *model term*, be it explicit or not, in a model formula, implies an *effect* on the g-weights $\mathbf{g} = F(\mathbf{X}\boldsymbol{\lambda})$ in the calibration model. Following linear regression (or GLM) terminology, we may say that the model 1 + *A***B* = 1 + *A* + *B* + *A*.*B* implies an *overall effect* (corresponding to the constant term), *main effects* for each of the variables *A* and *B* (corresponding to the model terms *A* and *B*), and an *interaction effect* of *A* and *B* (corresponding to the model term *A*.*B*) on the g-weights.

Having explained the basics of the symbolic model language, it will now be straightforward to interpret the following model formula (and model calculus!), wherein *C* is a third qualitative calibration variable:

- $1 + A*B*C = 1 + A + B + C + A.B + A.C + B.C + A.B.C$
- $1 + A + B*C = 1 + A + B + C + B.C$
- $1 + (A + B)*C = 1 + A*C + B*C = 1 + A + B + C + A.C + B.C = (1 + A + B)*C$

The very last expression, $(1 + A + B)*C$, is extremely useful in the sequel. It implies that the additive model $1 + A + B$ is to be applied in each category of the third variable C. We wil say that the variable C is *out-factorised*. If a model formula can be rewritten, such that one or more qualitative variables are out-factorised, then calibration can be performed separately with a simpler model in each category or cell determined by the out-factorised variables: simultaneous calibration of the entire sample is then equivalent, i.e. results into the same set of g-weigths, to separate calibration with the simpler model in the C-categories. The latter categories or cells are called *calibration strata*. Thus, in our example, $(1 + A + B)*C$, the simple model $1 + A + B$ is applied separately to the sub-samples, or calibration strata, corresponding to the categories of C. Calibration strata will play an important role in our calibration software; see section IV.B for details.

Quantitative variables may be involved too in the calibration model, and the above model language can be easily extended to incorporate such variables too. One peculiarity related to the constant term, however, should be taken care off, as we will soon explain.

We start with the simple model Z, where Z is a quantitative calibration variable. This simple model is meant to imply that calibration is on the total t_z of a numerical variable, z say, in the population. The formula for the model wherein calibration is not only on the total of z in the entire population, but also on the marginal totals of z in sub-populations corresponding to the categories of a quantitative calibration variable A, is $Z*A$, or $Z + Z.A$. Notice again the hierarchical structure of our calibration models; however, $Z*A$ does not imply a main effect of A, or an additional term A in the model formula. If only the totals of z within A-categories, and henceforth also the total of z in the population, are used as calibration benchmarks, then it would not be correct to include the constant term 1 in the model formula. Hence: $Z*A \neq 1 + Z*A$. The model formula $1 + Z*A$, however, does make sense: it implies that, again, the totals of z within A-categories, and henceforth also the total of z in the population, are used as calibration benchmarks, but that also the total population (size) is a benchmark value. More complex model formula can now easily be constructed, taking more than one qualitative and/or quantitative variable into account. However, "products" of quantitative variables should be avoided; we then suggest to create a new variable first. The reader will now be ready to understand the following expressions:

- $A + B*Z = 1 + A + Z + B.Z \neq 1 + A + B + Z + B.Z$
- $A*B*Z = Z + A.Z + B.Z + A.B.Z$
- $(A + B*C)*Z = Z + A.Z + B.Z + C.Z + B.C.Z$

It should be noticed that a quantitative variable cannot be a calibration stratum variable.

III.A.2 The scale parameter

In section II.A we have already discussed the necessity of introducing a *scale parameter*: it can then be argued that, for appropriate choice of the value of the scale, the g-weights (with respect to the scaled initial weights) are scattered around 1, which justifies that the special target point $\mathbf{1}_n$ is included in the set Ω_B; see II.A and II.C. We now discuss this parameter, ϕ, say, in more detail.

The scale must be strictly positive: $\phi > 0$. The modified (*scaled*) model (in standard notation) is defined as follows:

$$\left\{\min \mathbf{d}^T G(\mathbf{g});\ \phi\breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}. \qquad \text{(III.1)}$$

The calibration constraints can be written alternatively as $\phi\mathbf{X}^T\mathbf{D}\mathbf{g} = \mathbf{t}$, or $\mathbf{X}^T(\phi\mathbf{D})\mathbf{g} = \mathbf{t}$. Hence, introducing the scale parameter can be seen as the multiplication of the initial (sampling) weights $d_k\,(=1/\pi_k)$ with the scale ϕ. The calibration problem can then be solved starting from new initial weights, the *scaled* (*initial*) *weights* $\phi\mathbf{d}$. It will be clear that finally the solution, in terms of the calibrated weights **w**, will not change, for given data. However, the g-weights are altered: if the solution is **g** if no scale parameter is present (or $\phi = 1$), then the new weights, after introducing the scale, will be $\mathbf{g}/\phi$. And then, we have indeed: $\mathbf{w} = \phi\mathbf{D}\,\mathbf{g}/\phi = \mathbf{D}\mathbf{g}$. Notice that we could replace the objective function in (III.1) with $\phi\mathbf{d}^T G(\mathbf{g})$, but this does not change the calibration problem at all.

Hence the value of ϕ is theoretically immaterial. Why then complicating things by introducing one more parameter ϕ? The reason is rather of a practical, numerical, nature. We have experienced that our software behaves better if a value of ϕ is carefully chosen (or calculated). This is particularly true for the truncated linear and for the logit method. For some data sets, g-CALIB-S is likely to fail easier if g-weights tend to be large and if either the truncated linear or the logit method is chosen. For those methods it seems to be better to assure in advance that the g-weights are centred on 1. This in fact means that it should be possible to choose the region Ω_B such that it contains the point $\mathbf{1}_n$. But this now indicates that our previous statement that "ϕ *is theoretically immaterial*" is not completely right! The truth is that, strictly speaking, the basic problem $\left\{\min \mathbf{d}^T G(\mathbf{g});\ \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}$ should be extended to $\left\{\min \mathbf{d}^T G(\mathbf{g});\ \phi\breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \phi\mathbf{g} \in \Omega_B\right\}$, or $\left\{\min \mathbf{d}^T G(\mathbf{g});\ \phi\breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \mathbf{g} \in \phi^{-1}\Omega_B = \Omega_B^*\right\}$. However, for the linear and for the multiplicative method, this essentially doesn't change the region Ω_B, since then either $\Omega_B^* = \phi^{-1}\Omega_B = \phi^{-1}(-\infty,+\infty)^n = (-\infty,+\infty)^n = \Omega_B$ for the linear method, or $\Omega_B^* = \phi^{-1}\Omega_B = \phi^{-1}[0,+\infty)^n = [0,+\infty)^n = \Omega_B$, for the multiplicative method. For the truncated linear and logit method we have: $\Omega_B^* = \phi^{-1}\Omega_B = \phi^{-1}[L,U]^n = \left[\phi^{-1}L,\phi^{-1}U\right]^n \neq \Omega_B$, in general, if $\phi \neq 1$. It then follows that (III.1) is a convenient notation, since in practice one will usually specify a desirable region Ω_B <u>in advance</u>. A value for the scale parameter then has to be specified such that (III.1) is feasible. In other words, it is possible that, for given Ω_B, the problem $\left\{\min \mathbf{d}^T G(\mathbf{g});\ \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}$ has no solution (and that the software fails), but that, for appropriately chosen ϕ, the modified problem (III.1) is feasible (and that the software doesn't fail).

The scale ϕ can be interpreted as a preliminary and overall up-weighting factor to produce a first correction for non-response, given that it is calculated properly. If the constant term is (implicitly or explicitly) present in the model, and if one of the calibration variables is this constant, $x_1 = 1$, say, then a reasonable suggestion is to calculate the scale from the formula: $\hat{\phi} = \dfrac{t_1}{\sum_{k\in s} d_k} = \dfrac{N}{\hat{N}_s}$, where $\hat{N}_s$ is an initial (Horvitz-Thompson) estimate of the total population size *N*, based on the sample *s*. If *s* is the respondent sample, then this $\hat{\phi}$ is indeed the reciprocal of an estimated overall response rate. Of course, the scale can be calculated from any other calibration variable. We discuss in chapter IV section B how the software can be used to fix a value for the scale, or to instruct the program to calculate the scale from the data (for each calibration stratum separately).

III.B GENERALISED CALIBRATION AND POST-STRATIFICATION

III.B.1 Complete post-stratification

Consider the following particular calibration problem:

- A is a qualitative variable with a categories, and for each sample element k the vector $\boldsymbol{\delta}_k^A = (\delta_{k1}^A, ..., \delta_{ka}^A)^T$ indicates to which A-category element k belongs, since $\delta_{kr}^A = 1$ if k belongs to the r-th A-category and $\delta_{kr}^A = 0$ otherwise. The indicator variables δ_r^A (r = 1, …, a) may eventually serve as calibration variables.
- B is a qualitative variable with b categories, and for each sample element k the vector $\boldsymbol{\delta}_k^B = (\delta_{k1}^B, ..., \delta_{kb}^B)^T$ indicates to which B-category element k belongs, since $\delta_{kc}^B = 1$ if k belongs to the c-th B-category and $\delta_{kc}^B = 0$ otherwise. The indicator variables δ_r^B (c = 1, …, b) may eventually serve as calibration variables.
- The Kronecker product $\boldsymbol{\delta}_k^{AB} = \boldsymbol{\delta}_k^A \otimes \boldsymbol{\delta}_k^B$, with $((r-1)b+c)$-th component δ_{krc}^{AB} of this $ab \times 1$ matrix (vector) being equal to $\delta_{kr}^A \times \delta_{kc}^B$, is the vector of cell indicators in the complete cross-classification by A and B. The indicator variables δ_{rc}^{AB} (r = 1, …, a; c = 1, …, b) may eventually serve as calibration variables.
- Let n_{rc}^{AB} be the number of sample elements in cell rc in the complete cross-classification of the sample by variables A and B. The sub-sample corresponding to this cell is denoted as s_{rc}^{AB}. Similarly, we define the sub-samples s_r^A, with n_r^A elements, corresponding to the categories of A (or A-margins in the cross-classification), and s_c^B, with n_c^B elements, corresponding to the categories of B (or B-margins in the cross-classification).
- The calibration vector $\mathbf{x}_k$ for element k is defined by $\mathbf{x}_k^T = \left(1, \left(\boldsymbol{\delta}_k^A\right)^T, \left(\boldsymbol{\delta}_k^B\right)^T, \left(\boldsymbol{\delta}_k^{AB}\right)^T\right)$; notice that a constant calibration variable has been included. Notice further that there are many linear column dependencies in the resulting design matrix $\mathbf{X}$. If each row would be reduced to, for instance, the transposed of the Kronecker product, then an equivalent (full-rank) design matrix would be obtained.
- There is an initial weight vector $\mathbf{d}$. This vector may be specified later.
- The vector of calibration totals, corresponding to the design matrix $\mathbf{X}$, is written $\mathbf{t} = \left(N, N_1^A, ..., N_a^A, N_1^B, ..., N_b^B, N_{11}^{AB}, ..., N_{1b}^{AB}, N_{21}^{AB}, ..., N_{2b}^{AB}, ..., ..., ..., N_{a1}^{AB}, ..., N_{ab}^{AB}\right)^T$. N is the size of the population, N_r^A is the size of the population in A-category r, N_c^B is the size of the population in B-category c, and N_{rc}^{AB} is the size of the population in AB-cell rc. We assume numerical consistency: $N = \sum_{r=1}^{a} N_r^A = \sum_{c=1}^{b} N_c^B$, $N_r^A = \sum_{c=1}^{b} N_{rc}^{AB}$ (r = 1, …, a) and $N_c^B = \sum_{r=1}^{a} N_{rc}^{AB}$ (c = 1, …, b). (If the above-mentioned reduced full-rank design matrix is used, then $\mathbf{t} = \left(N_{11}^{AB}, ..., N_{1b}^{AB}, N_{21}^{AB}, ..., N_{2b}^{AB}, ..., ..., ..., N_{a1}^{AB}, ..., N_{ab}^{AB}\right)^T$ should be used.)
- The distance function G, or the calibration function F, is not specified (yet); Ω_B is set appropriately.

More qualitative variables A, B, C, … may be considered; the generalisation of the present discussion is straightforward.

Obviously, since all sample elements *k* in the same cell *rc* in the cross-classification by *A* and *B* have the same set of values for the calibration variables, they will be assigned the same g-weight, g_{rc} say. Hence, there are only *ab* unknown variables to be calculated from the system of calibration equations. On the other hand, the set of calibration equations can be reduced to a set of only *ab* equations corresponding to the calibration variables δ_{rc}^{AB} or to the calibration totals N_{rc}^{AB} (the right hand sides in these equations). The resulting linear system of *ab* equations in *ab* variables generally has a single solution. Hence $\Omega_C \cap \Omega_B$, if non-empty, generally contains only one point **g**, which must be the solution of the reduced set of calibration constraints, independent of the choice of the distance function. This solution can easily be found: the value of the g-weight g_{rc} follows immediately from the calibration constraint corresponding to the calibration variable δ_{rc}^{AB}. This constraint can be written as follows:

$$\sum_{k=1}^{n} d_k g_k \delta_{krc}^{AB} = \sum_{k \in s_{rc}^{AB}} d_k g_{rc} = g_{rc} \sum_{k \in s_{rc}^{AB}} d_k = N_{rc}^{AB}. \qquad \text{(III.2)}$$

Hence:

$$g_{rc} = \frac{N_{rc}^{AB}}{\sum_{k \in s_{rc}^{AB}} d_k}, \qquad \text{(III.3)}$$

and:

$$w_k = d_k \frac{N_{rc}^{AB}}{\sum_{k \in s_{rc}^{AB}} d_k} \quad \text{for all} \quad k \in s_{rc}^{AB}. \qquad \text{(III.4)}$$

It is, of course, assumed that $\sum_{k \in s_{rc}^{AB}} d_k \neq 0$ for all cells *rc*, i.e. that each cell contains at least one sample element (with positive initial weight).

Notice again that these general results, for the given calibration problem, are completely independent of the choice of the distance function *G*, or calibration function *F*, and of the weight vector **d**. The reader, however, will be more familiar with special forms of the formulae (III.3) and (III.4). These are now presented.

***Application* 1** Initial weights are all equal to 1, i.e. $\mathbf{d} = \mathbf{1}_n$. Then:

$$g_{rc} = \frac{N_{rc}^{AB}}{n_{rc}^{AB}} \quad \text{and} \quad w_k = \frac{N_{rc}^{AB}}{n_{rc}^{AB}} \quad \text{for all} \quad k \in s_{rc}^{AB}. \qquad \text{(III.5)}$$

This result is applicable, for instance, if the sample is exhaustive (a census), if complete post-stratification by variables *A* and *B* is applied, and if non-response $\left(n_{rc}^{AB} \leq N_{rc}^{AB}\right)$ is to be adjusted for.

***Application* 2** A simple random sample (SRS) (without replacement, and with fixed size *n*) is drawn, so that all elements have sampling weight $d_k = \frac{N}{n}$. Then:

$$g_{rc} = \frac{N_{rc}^{AB}}{n_{rc}^{AB}\frac{N}{n}} = \frac{n}{N}\frac{N_{rc}^{AB}}{n_{rc}^{AB}} \quad \text{and} \quad w_k = \frac{N_{rc}^{AB}}{n_{rc}^{AB}} \quad \text{for all} \quad k \in s_{rc}^{AB}. \tag{III.6}$$

Notice that the calibrated weights w_k are exactly as in Application 1.

***Application* 3** An *epsem* (equal probability selection method) is applied to draw a probabilistic sample, so that all elements have the same sampling weight $d_k = d_0$, say. Then:

$$g_{rc} = \frac{N_{rc}^{AB}}{n_{rc}^{AB} d_0} \quad \text{and} \quad w_k = \frac{N_{rc}^{AB}}{n_{rc}^{AB}} \quad \text{for all} \quad k \in s_{rc}^{AB}. \tag{III.7}$$

Again the same formula for the calibrated weights is obtained. SRS (Application 2) is an important special case of *epsem* sampling. An *epsem* is also called a *self-weighting design*. *Epsem* samples may be the result of a (very) complex sampling design, such as some two-stage designs, with PPS (*probability proportional to size*) sampling of PSUs, and some SRS of SSUs in the selected PSUs (see Särndal *et al*, 1992, p.141 for further details).

***Application* 4** Suppose a sample is drawn by (fixed size) stratified simple random sampling (STR-SRS). Let the *a priori* or *sampling strata* be indexed by *h*; $\tilde{s}_h$ is the *h*-th sampling stratum in the sample. The sampling weights are $d_k = \frac{N_h}{n_h}$ for all $k \in \tilde{s}_h$. We distinguish two situations:

- *Post-strata are subsets of (or coincide with) sampling strata*: for any *rc*, $s_{rc}^{AB} \subset \tilde{s}_h$ for some *h*. Then:

$$g_{rc} = \frac{n_h}{N_h}\frac{N_{rc}^{AB}}{n_{rc}^{AB}} \quad \text{and} \quad w_k = \frac{N_{rc}^{AB}}{n_{rc}^{AB}} \quad \text{for all} \quad k \in s_{rc}^{AB} \subset \tilde{s}_h. \tag{III.8}$$

  This is, once again, the same formula for the calibrated weights w_k as for *epsem* sampling.

- *Post-strata are cutting across sampling strata (the general situation)*; let $n_{h,rc}$ be the number of sampled elements in the intersection $\tilde{s}_h \cap s_{rc}^{AB}$. Then:

$$g_{rc} = \frac{N_{rc}^{AB}}{\sum_h \sum_{k \in \tilde{s}_h \cap s_{rc}^{AB}} \frac{N_h}{n_h}} = \frac{N_{rc}^{AB}}{\sum_h \frac{N_h}{n_h} n_{h,rc}},$$

$$w_k = \frac{N_h}{n_h}\frac{N_{rc}^{AB}}{\sum_{h'} \frac{N_{h'}}{n_{h'}} n_{h,rc}} \quad \text{for all} \quad k \in \tilde{s}_h \cap s_{rc}^{AB}. \tag{III.9}$$

Looking carefully at all four applications, we can draw useful conclusions with respect to statistical practice. We start with the formulation of properties of the sampling design:

Cond 1 There is some stratification (in the initial phase) of the sampling method.

Cond 2 The sampling design is self-weighting within each sampling stratum.

And we add a property for the estimation procedure:

Cond 3 Post-stratification is involved in the estimation phase, and each post-stratum is a subset of some sampling stratum.

If *Cond* 1 to 3 are satisfied, then the following practical conclusions can be drawn:

Pc 1 *The calibrated weights are the same for all sample elements in a post-stratum; this common calibrated weight only depends on the size of the population and the size of the sample within the post-stratum.*

Pc 2 *The initial (sampling) weights do not appear explicitly in the formula for the calibrated weights.*

Pc 3 *Hence, ignoring the sampling weights does not affect the calibrated weights.*

Pc 4 *For purposes of calibration (only), the initial weights can conveniently be set to* 1 *(or any other constant) without affecting the final calibrated weights.*

These results are useful in many practical situations. We have already mentioned a complex but self-weighting design in Application 3; there could be one or more (first) stage of stratification in this sampling design. At Statistics Belgium, for several complex designs, conditions *Cond* 1 and *Cond* 2 are (at least approximately) satisfied. If then, in the estimation phase, post-strata can be carefully chosen, in such a way that also *Cond* 3 is satisfied, then extrapolation turns out to be extremely simple. Notice that the statistician would not have to choose a calibration function F. No iteration is involved in the calculation of g-weights and calibrated weights; and therefore general purpose software would suffice the get the results.

This is probably what makes post-stratification a popular technique in many governmental statistical agencies. However, the reader should be aware of the fact that these conclusions are only relevant as far as *point estimation* of parameters of study variables (totals, means, …) is concerned. Variance estimation (for the estimators of the parameters), on the other hand, is a completely different story: aspects of the sampling design cannot that easily be ignored (or hidden by choosing appropriate post-strata) for that purpose. Moreover, it is definitely not an optimal strategy to decide to stick to post-stratification techniques and to ignore the overwhelming existence of very attractive, elegant and efficient techniques and accompanying software, which give the statistician a huge flexibility, hopefully resulting into better weighting schemes. I do not say that the era of post-stratification at Statistics Belgium is finished: statistical practice does not have to be complex in order to be efficient. But the reader will know that, for many practical reasons, forced for instance by fieldwork or cost limitations, a sampling plan often reaches some degree of complexity, and the subsequent estimation procedures may then be a bit tricky too.

Moreover, post-stratification as discussed so far almost always suffers from extreme fragmentation of the sample (and sometimes the population too), a problem that is discussed in the next paragraphs, and to which some solutions are presented in the next sub-section III.B.2 and in section III.C.

In this section we have discussed instances of *complete post-stratification*. Deville *et al* (1993) define complete post-stratification (based on two classifying qualitative variables) as the calibration problem wherein population cell sizes N_{rc}^{AB} are known (and used as benchmark values). Notice that these cell sizes may be replaced by appropriate estimates, obtained from an external source. (In household surveys, the Labour Force Survey (LFS) is often used to deliver estimates of population totals or counts.)

A well-known problem, as already mentioned, with complete post-stratification is that some or all post-strata may become extremely small, or even empty, such that the resulting g-weights and calibrated weights will become unstable, or even undefined. At Statistics Belgium, this problem has been resolved traditionally by regrouping original post-strata (corresponding to cells in a cross-classification) into new post-strata, all of which should have a reasonable size in the sample. The resulting calibration problem is still an instance of a complete post-stratification problem, based on one or more modified classification variables, however.

In the next section III.C in this chapter we discuss the popular method of (generalised) raking, where calibration is no longer on cell counts in a cross-classification of the population, but on marginal counts, corresponding to the categories of the classifying variables. Raking methods generally solve the problem of instability of g- and calibrated weights in complete post-stratification. It also works if there are some empty cells, as long as the marginal categories are non-empty (and not too small). Raking methods deserve our special attention, since we will have interesting applications, to be discussed in chapter 5.

Raking, or calibration on known marginal counts, is just one subclass of methods, within the broader class of *incomplete post-stratification* methods. In the next sub-section III.B.2 we discuss incomplete post-stratification methods that fall, at least in some sense, between complete post-stratification methods as discussed here before, and raking methods that will be discussed in section III.C.

III.B.2 Incomplete post-stratification

Deville *et al* (1993) describe *incomplete post-stratification* as follows: "*Any case for which the auxiliary information is less detailed than a complete knowledge of all cell counts can be described as incomplete post-stratification*". (This is not a rigorous definition.) The same authors equivalently use *generalised raking* to name this class of calibration problems. However, I prefer to reserve the term "generalised raking" for that subclass of incomplete post-stratification methods, where calibration is on margins corresponding to (at least two) classification variables. This terminology, I believe, is closer to the original terminology used by Deming and Stephan (1940), when they proposed the classical raking ratio technique. Generalised raking will be discussed separately in the next section III.C, for reasons that will become clear over there.

The same notation as in the previous sub-section is used. Suppose that initially a complete post-stratification problem is considered, but that some sample cell sizes n_{rc}^{AB} are too small (possibly zero). Suppose, on the other hand, that (1°) marginal sample sizes n_r^A and n_c^B are large enough, and (2°) after collapsing original categories of A and/or collapsing original categories of B the new cell sizes $n_{r'c'}^{A'B'}$ are all large enough. Here A' and B' denote the modified classification variables, derived respectively from A and B by collapsing categories for one or both variables. Categories of A' are indexed $r' = 1,\ldots,a'$ and categories of B' are indexed $c' = 1,\ldots,b'$, where $a' \leq a$ and $b' \leq b$. Let $N_{r'c'}^{A'B'}$ be the population cell sizes in the new cross-classification. One could consider the new complete post-stratification problem based on the classification variables A' and B'. The result would be that more stable g-weights and calibrated weights are obtained, but, formally, nothing new would have been done.

An interesting and elegant calibration problem (Deville *et al*, 1993) can be obtained by considering for each sample element k the new calibration vector $\mathbf{x}'^T_k = \left(1, \left(\boldsymbol{\delta}_k^A\right)^T, \left(\boldsymbol{\delta}_k^B\right)^T, \left(\boldsymbol{\delta}_k^{A'B'}\right)^T\right)$, where $\boldsymbol{\delta}_k^{A'B'}$ is the Kronecker product $\boldsymbol{\delta}_k^{A'B'} = \boldsymbol{\delta}_k^{A'} \otimes \boldsymbol{\delta}_k^{B'}$, with obvious definition of $\boldsymbol{\delta}_k^{A'}$ and $\boldsymbol{\delta}_k^{B'}$. Notice that the

calibration vector does involve (explicitly) neither $\boldsymbol{\delta}_k^{A'}$ nor $\boldsymbol{\delta}_k^{B'}$, but the original $\boldsymbol{\delta}_k^{A}$ and $\boldsymbol{\delta}_k^{B}$. This means that calibration is on <u>original</u> marginal counts N_r^A and N_c^B and, at the same time, on <u>new</u> cell counts $N_{r'c'}^{A'B'}$.

Now, sample elements in a new cell $r'c'$ do not necessarily have the same set of values for the calibration variables, and therefore may have different g-weights. This complex dependence of g-weights on calibration variables through the calibration function F implies that it is not possible anymore to reduce the system of calibration variables to a simple set of linear equations that can be solved algebraically. Thus the system has to be solved iteratively, and the g-weights will generally depend on the choice of the distance function or calibration function.

To make things perfectly clear, we illustrate some aspects by means of a small example. Let's start with a complete post-stratification problem, based on a 2×3 classification, schematically represented (without calibration totals) as follows:

Table 3.1 *A two-dimensional contingency table: complete post-stratification*

	B-category			
A-category	1	2	3	Sample margins
1	(1,1,0,1,0,0,1,0,0,0,0,0) 15	(1,1,0,0,1,0,0,1,0,0,0,0) 0	(1,1,0,0,0,1,0,0,1,0,0,0) 10	25
2	(1,1,0,1,0,0,0,0,0,1,0,0) 25	(1,1,0,0,1,0,0,0,0,0,1,0) 18	(1,1,0,0,0,1,0,0,0,0,0,1) 2	45
Sample margins	40	18	12	70

Each cell contains the calibration vector for each of the sample elements in that cell, and the sample cell count. Post-stratification is not possible from this table, since one cell size is zero (and another one is small). However, collapsing the 2nd and 3rd column, a new incomplete post-stratification problem is obtained, with the following schematic representation:

Table 3.2 *A two-dimensional contingency table: incomplete post-stratification* (1)

	B-category			
	1	2	3	
	B'-category			
A-category	1	2		Sample margins
1	(1,1,0,1,0,0,1,0,0,0) 15	(1,1,0,0,1,0,0,1,0,0) 10	(1,1,0,0,0,1,0,0,1,0) 10	25
2	(1,1,0,1,0,0,0,0,1,0) 25	(1,1,0,0,1,0,0,0,0,1) 20	(1,1,0,0,0,1,0,0,0,1) 20	45
Sample margins	40	18	12	70

Notice that the marginal counts, 18 and 12, have not been collapsed. This is essentially why the new calibration problem is not a complete post-stratification problem anymore.

If we compare the corresponding calibration design matrices $\mathbf{X}$ and $\mathbf{X}'$, then it is noticed that 4 columns of $\mathbf{X}$, corresponding to the calibration variables $\delta_{12}^{AB}, \delta_{13}^{AB}, \delta_{22}^{AB}$ and δ_{23}^{AB}, are transformed into 2 columns of $\mathbf{X}'$, corresponding to the calibration variables $\delta_{12}^{AB'}$ and $\delta_{22}^{AB'}$ as follows:

$$\begin{cases} \delta_{12}^{AB} + \delta_{13}^{AB} = \delta_{12}^{AB'} \\ \delta_{22}^{AB} + \delta_{23}^{AB} = \delta_{22}^{AB'} \end{cases} \qquad \text{(III.10)}$$

Strictly speaking we should also include the relationships $\delta_{11}^{AB} = \delta_{11}^{AB'}$ and $\delta_{21}^{AB} = \delta_{21}^{AB'}$, but these do not change any column in the original matrix $\mathbf{X}$. From a practical point of view, the remarks in this paragraph are useful, as we will demonstrate numerically in chapter 5, for two reasons: (1°) it indicates how the new design matrix $\mathbf{X}'$ can be constructed from $\mathbf{X}$ in a straightforward way, by some simple summations, and (2°) it opens a door to alternative modifications of the original complete post-stratification problem, which are practically treated in a very similar way, as illustrated in the next paragraphs.

Indeed, consider again the above numerical example. If one would decide that (only) 2 elements in a cell is enough for calibration and estimation, then only the cells containing 0 and 10 elements could be collapsed. I.e. $\delta_{11}^{AB} = \delta_{11}^{AB'}$, $\delta_{12}^{AB} + \delta_{13}^{AB} = \delta_{12}^{AB'}$ and $\delta_{2c}^{AB} = \delta_{2c'}^{AB'}$ $(c = c' = 1,\ldots,3)$, or, schematically:

Table 3.3 *A two-dimensional contingency table: incomplete post-stratification* (2)

	B-category			
	1	2	3	
	B'-category			
A-category	1	2		Sample margins
1	(1,1,0,1,0,0,1,0,0,0,0) 15	(1,1,0,0,1,0,0,1,0,0,0) 10	(1,1,0,0,0,1,0,0,1,0,0) 10	25
2	(1,1,0,1,0,0,0,0,1,0,0) 25	(1,1,0,0,1,0,0,0,0,1,0) 18	(1,1,0,0,0,1,0,0,0,0,1) 2	45
Sample margins	40	18	12	70

The latter method allows collapsing cells in a more refined way: an individual cell can be collapsed with any other individual cell. The previous method merely allows collapsing each cell in a column (row) with each corresponding cell in any other column (row). Finally, it may be noticed that cells that are collapsed do not necessarily have to be "neighbouring" cells, although this will be a more convenient approach if the classification variables are ordinal.

III.C THE RAKING RATIO TECHNIQUE FOR CROSS-TABULATED DATA

III.C.1 Introduction: equal g-weights

In generalised calibration the g-weights are $g_k = F\left(\mathbf{x}_k^T\boldsymbol{\lambda}\right)$ $(k=1,...,n)$, or $\mathbf{g}(\boldsymbol{\lambda}) = F(\mathbf{X}\boldsymbol{\lambda})$. Hence, if *k* and *k'* are sample elements with the same auxiliary vector $\mathbf{x}_k = \mathbf{x}_{k'}$, then their g-weights are equal to each other: $g_k = g_{k'}$.

Let the calibration model depend on qualitative calibration variables *A*, *B*, … only. Then within each cell in a cross-classification of the sample *s* by the variables *A*, *B*, …, all sample elements will have the same auxiliary vector, and therefore the same g-weight. Let *c* be indexing the cells in the cross-tabulation ($c=1,...,C$ if there are *C* cells), and let $\tilde{g}_c$ be the common g-weight for all sample elements in cell *c*; let s^c be the sub-sample of elements in cell *c*. Then the *j*-th calibration constraint can be rewritten as

$$\sum_{k\in s} w_k x_{kj} = \sum_{k\in s} d_k g_k x_{kj} = \sum_{k\in s} g_k \breve{x}_{kj} = \sum_c \tilde{g}_c \sum_{k\in s^c} \breve{x}_{kj} = t_j .$$

Define $\sum_{k\in s^c} \breve{x}_{kj} = \breve{z}_{jc}$, i.e. the sum of expanded values of the *j*-th calibration variable in cell *c*, which in fact is the weighted sum of the values of the *j*-th calibration variable in cell *c*, the weights in this sum being the initial (sampling) weights. This shows that the calibration constraints can be written in terms of weighted totals over sample elements in the cells of a cross-tabulation; the individual values and initial individual weights have collapsed into these sums, i.e. $\sum_c \tilde{g}_c \breve{z}_{jc} = t_j$.

A similar argument is used to show that the objective function also can be written in terms of cell totals:

$$\mathbf{d}^T G(\mathbf{g}) = \sum_{k\in s} d_k G(g_k) = \sum_c G(\tilde{g}_c) \sum_{k\in s^c} d_k .$$

The weights in this sum are the totals of the initial weights within the cells, that is: $\sum_{k\in s^c} d_k = d_c^+$.

Hence the *expanded data matrix* $\left(\breve{\mathbf{X}}|\mathbf{d}\right)$ can be collapsed into a matrix of weighted totals, the *collapsed data matrix* $\left(\breve{\mathbf{Z}}|\mathbf{d}^+\right)$, where the $(C\times m)$-matrix $\breve{\mathbf{Z}}$ has element $\breve{z}_{jc}$ in the *c*-th row and *j*-th column, and the $(C\times 1)$-vector $\mathbf{d}^+$ has *c*-th element d_c^+. Also $\tilde{\mathbf{g}} = (\tilde{g}_1,...,\tilde{g}_C)$, and $\tilde{\Omega}_B$ follows immediately from Ω_B, e.g. $\tilde{\Omega}_B = [L,U]^C$ if $\Omega_B = [L,U]^n$. Then the calibration problem $\left\{\min \mathbf{d}^T G(\mathbf{g}); \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \mathbf{g}\in\Omega_B\right\}$ can be reformulated equivalently as

$$\left\{\min \mathbf{d}^{+T} G(\tilde{\mathbf{g}}); \breve{\mathbf{Z}}^T\tilde{\mathbf{g}} = \mathbf{t}, \tilde{\mathbf{g}}\in\tilde{\Omega}_B\right\}. \qquad \text{(III.11)}$$

An equivalent formulation of a general calibration model $\left\{\min \mathbf{d}^T G(\mathbf{g}); \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}, \mathbf{g}\in\Omega_B\right\}$ is $\left\{\min \mathbf{d}^T G(\mathbf{g}); \mathbf{X}^T\mathbf{D}\mathbf{g} = \mathbf{t}, \mathbf{g}\in\Omega_B\right\}$. The latter formulation is useful in relation to our SPSS software modules. Indeed, as we will explain in full detail in chapter IV, the main input for the calibration

module g-CALIB consists of the *data matrix* $(\mathbf{X}|\mathbf{d})$ and the vector **t**, in case of individual data. In order to be able to use our software also for solving calibration models for *grouped data* (examples are collapsed individual data as explained, or cross-classified data), i.e. models of the form (III.11), we rewrite this model as follows:

$$\left\{\min \mathbf{d}^{+T} G(\tilde{\mathbf{g}}); \tilde{\mathbf{Z}}^T \mathbf{D}^+ \tilde{\mathbf{g}} = \mathbf{t}, \tilde{\mathbf{g}} \in \tilde{\Omega}_B \right\}, \tag{III.12}$$

where $\tilde{\mathbf{Z}} = \mathbf{D}^{+^{-1}} \breve{\mathbf{Z}}$, or $\breve{\mathbf{Z}} = \mathbf{D}^+ \tilde{\mathbf{Z}}$, with $\mathbf{D}^+ = diag(\mathbf{d}^+)$. It follows then that for a grouped data problem the data matrix $(\tilde{\mathbf{Z}}|\mathbf{d}^+)$ and the vector **t** (the same as for individual data!) are the primary input for our software g-CALIB.

We now illustrate the usefulness of the above general idea, that of collapsing individual data, in two situations that are of great interest in practice.

III.C.2 Collapsing the data to reduce the size of the calibration problem

The dimension of the data matrix $(\mathbf{X}|\mathbf{d})$, i.e. *n* rows and *m* + 1 columns, for individual data, essentially determines the size of a calibration problem. It would be useful if this data matrix could be reduced. The above idea of collapsing the data provides an efficient solution for that purpose. The above reasoning fully explains how individual data $(\mathbf{X}|\mathbf{d})$ have to be transformed into grouped data $(\tilde{\mathbf{Z}}|\mathbf{d}^+)$. Briefly, the procedure is as follows:

- Aggregate rows in **X** by qualitative variables *A*, *B*, …, using weights **d**. The result is $\breve{\mathbf{Z}}$, as defined above.
- Aggregate rows (elements) in **d** by qualitative variables *A*, *B*, …. The result is $\mathbf{d}^+$, as defined above.
- Compute $\tilde{\mathbf{Z}} = \mathbf{D}^{+^{-1}} \breve{\mathbf{Z}}$, with $\mathbf{D}^+ = diag(\mathbf{d}^+)$.

The resulting data matrix $(\tilde{\mathbf{Z}}|\mathbf{d}^+)$ has size $C \times (m+1)$, wherein *C* is the number of (non-empty) cells in a cross-classification by the qualitative variables *A*, *B*, …. The reduction factor is roughly 1 over the average number of individual data points in a cell. Notice that the number of columns in the data matrix has not changed, i.e. aggregation is column by column.

After collapsing individual data and calculating g-weights and calibrated weights, it will, in most applications, be necessary to turn back to the individual data. This is true for calibration estimation of totals of survey variables, as well as for estimation of variances of estimators of totals, within the generalised calibration framework (see section III.G).

Notice that elements in $\tilde{\mathbf{Z}}$ are weighted cell averages:

$$\tilde{z}_{jc} = \frac{\breve{z}_{jc}}{d_c^+} = \frac{\sum_{k \in s^c} \breve{x}_{kj}}{\sum_{k \in s^c} d_k} = \frac{\sum_{k \in s^c} d_k x_{kj}}{\sum_{k \in s^c} d_k}. \tag{III.13}$$

As we shall see in section III.D, collapsing data has a lot in common with the clustering technique to impose equal g-weights in clusters.

III.C.3 Classical raking ratio in a contingency table

A 2-dimensional contingency table is a 2-way classification of (observed or expected) frequencies. The classification variables are qualitative. In survey statistics, one may want to adjust the frequencies in the table in order to meet some restrictions. The classical raking ratio technique is used to adjust the cell frequencies such that the margins in the table of adjusted frequencies are equal to fixed values. A common situation is where the observed frequencies are obtained from a sample, and the fixed margins are the marginal frequencies of the classification variables in the (finite) population from which the sample has been drawn. The general form of a contingency table, extended with marginal population frequencies is as in table 3.4.

Table 3.4 *A two-dimensional contingency table, extended with population margins*

		B-category			
A-category	⋯	*c*	⋯	Sample margins	Population margins
⋮		⋮			
r	⋯	n_{rc}	⋯	n_{r+}	N_r^A
⋮		⋮			
Sample margins		n_{+c}		*n*	
Population margins		N_c^B			*N*

Our problem here is to show that the raking ratio problem for a contingency table perfectly fits into the generalised calibration framework, and to outline how the data are to be transformed, in order to be able to solve the raking ratio problem by means of our software g-CALIB-S. For a discussion of the raking ratio method, we refer to Deville *et al* (1993). We start with a small hypothetical example. Consider the data in table 3.5, showing the joint distribution of variables *A* and *B* in a sample of size $n = 22$, and the marginal distribution of variables *A* and *B* in a population of size $N = 100$. Variable *A* has $a = 2$ categories, variable *B* has $b = 3$ categories; the number of cells is $C = a \times b = 6$.

Table 3.5 *An example of a* 2×3 *contingency table, extended with population margins*

	B-categories				
A-categories	1	2	3	Sample margins	Population margins
1	4	5	3	*12*	**50**
2	5	2	3	*10*	**50**
Sample margins	*9*	*7*	*6*	*22*	
Population margins	**30**	**40**	**30**		**100**

We have implicitly assumed that the sample data in table 3.5 are obtained from individual data, as given in the first three columns in table 3.6. Notice that we assume that all individuals have the same weight 1 (column 3).

Table 3.6 *Original individual data, with corresponding design matrix* **X**, *and calibration totals* **t**

Original data			Design matrix **X**					
A	*B*	*d*	X0	A1	A2	B1	B2	B3
1	1	1	1	1	0	1	0	0
1	1	1	1	1	0	1	0	0
1	1	1	1	1	0	1	0	0
1	1	1	1	1	0	1	0	0
1	2	1	1	1	0	0	1	0
1	2	1	1	1	0	0	1	0
1	2	1	1	1	0	0	1	0
1	2	1	1	1	0	0	1	0
1	2	1	1	1	0	0	1	0
1	3	1	1	1	0	0	0	1
1	3	1	1	1	0	0	0	1
1	3	1	1	1	0	0	0	1
2	1	1	1	0	1	1	0	0
2	1	1	1	0	1	1	0	0
2	1	1	1	0	1	1	0	0
2	1	1	1	0	1	1	0	0
2	1	1	1	0	1	1	0	0
2	2	1	1	0	1	0	1	0
2	2	1	1	0	1	0	1	0
2	3	1	1	0	1	0	0	1
2	3	1	1	0	1	0	0	1
2	3	1	1	0	1	0	0	1
$\sum_{k \in s} d_k x_{kj}$			*22*	*12*	*10*	*9*	*7*	*6*
t_j			**100**	**50**	**50**	**30**	**40**	**30**

A model for calibration on margins of qualitative variables *A* and *B* can be written 1 + A + B (section III.A.1). A design matrix for this model is as in the last six columns of table 3.6; the headers of these columns are appropriate names for the six calibration variables. The last row in the table contains the population margins, i.e. the calibration totals t_j; the next to last row contains the initial weighted sample totals of the calibration variables. Following the collapsing procedure as outlined before, we obtain the collapsed data matrix $\left(\breve{\mathbf{Z}}|\mathbf{d}^+\right)$ as in the first 7 columns of table 3.7, and the grouped data matrix $\left(\tilde{\mathbf{Z}}|\mathbf{d}^+\right)$ as in the first and 6 last columns of table 3.7.

Notice that in the constructed individual data, each calibration variable x_j is constant within any cell *c*, from which we derive that

$$\tilde{z}_{cj} = x_{kj} \quad \text{for any} \quad k \in s^c, \text{ and any} \quad j = 1, \ldots, m. \tag{III.14}$$

Table 3.7 *Collapsed data matrix* $\left(\breve{\mathbf{Z}}|\mathbf{d}^{+}\right)$ *and grouped data matrix* $\left(\tilde{\mathbf{Z}}|\mathbf{d}^{+}\right)$*, obtained by transformation of individual data* $(\mathbf{X}|\mathbf{d})$

$\mathbf{d}^{+}$	Collapsed design matrix $\breve{\mathbf{Z}}$						Grouped design matrix $\tilde{\mathbf{Z}}$					
	X0	A1	A2	B1	B2	B3	X0	A1	A2	B1	B2	B3
4	4	4	0	4	0	0	1	1	0	1	0	0
5	5	5	0	0	5	0	1	1	0	0	1	0
3	3	3	0	0	0	3	1	1	0	0	0	1
5	5	0	5	5	0	0	1	0	1	1	0	0
2	2	0	2	0	2	0	1	0	1	0	1	0
3	3	0	3	0	0	3	1	0	1	0	0	1
					$\sum_{c=1}^{C} d_c^{+} \breve{z}_{cj}$		*22*	*12*	*10*	*9*	*7*	*6*
					t_j		**100**	**50**	**50**	**30**	**40**	**30**

Obviously, the calibration results from individual data and from grouped data in this example are numerically the same, given that in both calibration problems the same distance or calibration functions are used. For the raking ratio method, the calibration function is the exponential function. The fact that our algorithm converges to the same solution as the classical alternating method, often called *iterative proportional fitting*, is not shown theoretically in this text.

The above small example demonstrates how, *directly from a contingency table*, the data matrix $\left(\tilde{\mathbf{Z}}|\mathbf{d}^{+}\right)$ has to be constructed. SPSS modules are discussed later; rather than using the general collapsing procedure, we use (III.14) to implement a transformation. The same technique can be used if the cells in a cross-classification do not contain observed frequencies, but general real values. We illustrate this in section V.F. Extension of the above procedures from 2 qualitative variables to 3 or more is straightforward.

III.D IMPOSING EQUALITY OF G-WEIGHTS IN CLUSTERS

In section III.C, we have discussed how the survey data can be collapsed over groups of elements with the same calibration vector. This technique can be used at any time and may be useful to reduce the length of the survey data file (i.e. the number of rows in the design matrix), and therefore the time needed to perform the computations. Thus, collapsing survey data has there been introduced first and foremost for practical purposes. Nevertheless, we discussed a useful practical application of collapsing: adjusting cross-classified data, wherein individual observations are not available, but survey data are only available at an aggregated level. The technique is illustrated in section V.F, using aggregated data on labour volume and labour compensation.

When survey data are collapsed, values of calibration variables are not changed: several elements with the same calibration vector are replaced with a single new (higher level) element with the same calibration vector. The new "element", however, gets an initial weight that is the sum of the initial weights of the original elements that it "represents".

In this section we discuss a technique of aggregating data, which does have something in common with collapsing data, but its rationale is rather of a different nature. Indeed, we now want to impose the same g-weights for elements with different calibration vectors. From our statement in the first paragraph of section III.C.1, it follows that, if an ordinary calibration problem is set up, the original survey data must be transformed, because otherwise one could only be lucky to obtain g-weights that satisfy some pre-specified equality constraints. We now show this need for a transformation of the survey data, starting from an extended calibration problem; the new calibration problem can indeed be formulated in terms of the original data, but adding some additional constraints, as follows:

$$\left\{\min \mathbf{d}^T G(\mathbf{g});\ \breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t},\ \mathbf{g} \in \Omega_B,\ \text{equality restrictions on } \mathbf{g}\right\}. \qquad \text{(III.15)}$$

We will show now that this new calibration problem can be transformed into an ordinary problem of the form $\left\{\min \mathbf{b}^T G(\mathbf{f});\ \breve{\mathbf{Y}}^T\mathbf{f} = \mathbf{s},\ \mathbf{f} \in \Theta_B\right\}$, wherein $\mathbf{b}$, $\mathbf{f}$, $\mathbf{Y}$, $\mathbf{s}$ and Θ_B are to be specified, so that finally our software can still be used to solve the problem. In order to formalise this new modified calibration problem, we first have to introduce some new concepts.

A subset of elements in the sample *s*, for which the g-weights have to be equal, not because of equal calibration vectors, but since these equality constraints are imposed by the modeller, is called a *cluster*. In practice a cluster is determined by one or more qualitative variables. Such a variable cannot be a calibration variable (or a variable from which a set of calibration variables is derived), but is actually one of the sampling and/or survey variables, which naturally follows from the survey context. A typical example of a cluster is, in case of persons as elements in the sample *s*, the household to which the element belongs. In business surveys, a cluster could be an enterprise as a set of different local units (the elements in *s*). In labour cost statistics, a cluster could be an enterprise or a local unit as the set of employees (the elements of *s*) which are employed in that enterprise or local unit. Since the technique discussed in this section is built up from the idea of clusters, it will from now on be called *the clustering technique*, or, simply, *clustering*. Notice that a cluster in this calibration context will often – but not necessarily always! – be nothing more (or less) than a cluster in the sampling design, if some cluster sampling technique is applied.

Suppose that the sample *s* contains *n* elements, and that these elements are members of *L* clusters ($L \leq n$). Let s_l be the *l*-th cluster, $l = 1, \ldots, L$; notice that $s = \bigcup_{l=1}^{L} s_l$. We then construct a $n \times L$ matrix $\mathbf{C}$, the *cluster-membership matrix*, where each row corresponds to an element of the sample *s* and each column corresponds the a cluster, and whose entries c_{kl} $(k = 1,\ldots,n; l = 1,\ldots,L)$ are defined as follows:

$$
\begin{aligned}
c_{kl} \quad &= 1 \quad \text{if } \; k \in s_l \\
&= 0 \quad \text{if } \; k \notin s_l .
\end{aligned}
$$

Such a matrix can easily be constructed with our software; see later. Nieuwenbroek (1997) has introduced exactly the same matrix (but is calling it **L**); if $L = n$, then **C** is the identity matrix of order n. It is easy to see that the matrix product $\mathbf{C}^T\mathbf{d} = \mathbf{d}^+$ is a vector of length L, whose l-th element is the sum of the initial weights for elements in the l-th cluster. Given that the g-weights are constant within all clusters, let $g_k = g_{(l)}$ for all elements $k \in s_l$ and for all clusters s_l; i.e. $g_{(l)}$ is the common g-weight for elements in the l-th cluster. Let $\tilde{\mathbf{g}}$ be the L-vector of g-weights $g_{(l)}$. The objective function in (III.15) can now be rewritten: $\mathbf{d}^T G(\mathbf{g}) = \mathbf{d}^T \mathbf{C} G(\tilde{\mathbf{g}}) = \left(\mathbf{C}^T\mathbf{d}\right)^T G(\tilde{\mathbf{g}}) = \mathbf{d}^{+T} G(\tilde{\mathbf{g}})$. For convenience we assume that the rows of the design matrix **X**, of the clustering matrix **C** and of the initial weight vector **d** are ordered according to the clusters, i.e. sample elements within the same cluster have succeeding rows in these matrices and vector. It then follows that the calibration constraints $\breve{\mathbf{X}}^T\mathbf{g} = \mathbf{t}$ can be written equivalently as $\left(\mathbf{C}^T\breve{\mathbf{X}}\right)^T \tilde{\mathbf{g}} = \mathbf{t}$, or $\breve{\mathbf{H}}^T\tilde{\mathbf{g}} = \mathbf{t}$ where $\breve{\mathbf{H}} = \mathbf{C}^T\breve{\mathbf{X}}$. Notice (1°) that the vector of calibration totals has not been changed, and (2°) that $\breve{\mathbf{H}}$ is the ($L \times m$)-matrix with elements $\breve{h}_{lj} = \sum_{k \in s_l} d_k x_{kj} = \sum_{k \in s_l} \breve{x}_{kj} \quad (l = 1,\dots,L;\, j = 1,\dots,m)$. Finally, let $\tilde{\Omega}_B$ be appropriately modified – e.g. $\tilde{\Omega}_B = [0,+\infty)^L$ if originally it were $[0,+\infty)^n$ – then the calibration problem can be written equivalently as

$$
\left\{ \min \mathbf{d}^{+T} G(\tilde{\mathbf{g}}); \breve{\mathbf{H}}^T\tilde{\mathbf{g}} = \mathbf{t}, \tilde{\mathbf{g}} \in \tilde{\Omega}_B \right\}. \qquad \text{(III.16)}
$$

Thus, this transformed calibration problem is formally an ordinary calibration problem, and it will therefore be possible to solve it with our software, once the data are transformed appropriately.

The result is basically the same as the general result in section III.C.1, but our derivation in this section followed a slightly different path, since clusters are sets of elements with generally different calibration vectors, whereas cells in section III.C.1 are sets of elements with the same calibration vector. Moreover, cells are defined by means of qualitative calibration variables, while clusters are defined by means of qualitative *cluster variables*, which are not treated as calibration variables.

I intend to extend our software, such that a cluster variable is taken automatically into account. This should not be difficult at all: the survey data file (see section IV.B.3) should include a cluster variable, and aggregation by this cluster variable is then a straightforward task in SPSS. As long as the software cannot deal with a cluster variable, one should transform the data him/herself. For that reason, the following reformulation of (III.16) is more appropriate, as it indicates clearly how the data are to be prepared if clustering has to be taken into account:

$$
\left\{ \min \mathbf{d}^{+} G(\tilde{\mathbf{g}}); \tilde{\mathbf{H}}^T\mathbf{D}^+\tilde{\mathbf{g}} = \mathbf{t}, \tilde{\mathbf{g}} \in \tilde{\Omega}_B \right\}, \qquad \text{(III.17)}
$$

where, as in section III.C.1, at least formally, $\tilde{\mathbf{H}}$ is defined as $\tilde{\mathbf{H}} = \mathbf{D}^{+^{-1}}\breve{\mathbf{H}}$, or $\breve{\mathbf{H}} = \mathbf{D}^+\tilde{\mathbf{H}}$, with $\mathbf{D}^+ = diag\left(\mathbf{d}^+\right)$. Notice that the entries of the matrix $\tilde{\mathbf{H}}$ are weighted averages of the values of calibration variables within clusters, i.e. for $l = 1, \dots, L$ and $j = 1, \dots, m$:

$$\tilde{h}_{lj} = \frac{\sum_{k \in s_l} d_k x_{kj}}{\sum_{k \in s_l} d_k} . \qquad \text{(III.18)}$$

Finally, implementation will be rather straightforward, since, as mentioned already, the cluster-membership matrix **C** can easily be constructed (using the macros to be discussed in section IV.B.6), and because of the following relationships: $\mathbf{D}^{+} = \mathbf{C}^{T}\mathbf{D}\mathbf{C} = diag\left(\mathbf{C}^{T}\mathbf{d}\right)$ and $\tilde{\mathbf{H}} = \mathbf{D}^{+^{-1}}\mathbf{C}^{T}\tilde{\mathbf{X}} = \mathbf{D}^{+^{-1}}\mathbf{C}^{T}\mathbf{D}\mathbf{X}$, which show how to find the new data structures for the clustered problem from the original data structures, using the clustering transformation **C**.

III.E SIMULTANEOUS CALIBRATION ON TWO LEVELS OF AUXILIARY INFORMATION

III.E.1 The general problem

This section deals with generalised calibration for the following more complex survey data instance:

(i) Ultimate sampling elements *k*, in the element sample *s* of size *n*, are clustered into *L* ultimate clusters $s_{(l)}$ with respective sizes n_l, where $s = \bigcup_{l=1}^{L} s_{(l)}$.

(ii) Element-level auxiliary information is stored in an element-level calibration design matrix **X**, with dimension $n \times m$, and in a $(n \times 1)$-vector **t** of calibration totals.

(iii) Element-level sampling (or initial) weights are stored in a $(n \times 1)$-vector **d**, with elements d_k. Let $\mathbf{D} = diag(\mathbf{d})$, and $\breve{\mathbf{X}} = \mathbf{DX}$ the expanded design matrix.

(iv) Cluster-level auxiliary information is stored in a cluster-level calibration design matrix **Z**, with dimension $L \times p$, and in a $(L \times 1)$-vector **s** of calibration totals.

(v) Cluster-level sampling (or initial) weights are stored in a $(L \times 1)$-vector $\tilde{\mathbf{d}}$, with elements $d_{(l)}$. Let $\tilde{\mathbf{D}} = diag(\tilde{\mathbf{d}})$, and $\breve{\mathbf{Z}} = \tilde{\mathbf{D}}\mathbf{Z}$ the expanded design matrix.

For many surveys, in practice, we can moreover assume that sampling weights for elements within a cluster are constant and equal to the corresponding cluster-level sampling weight:

(vi) $d_k = d_{(l)}$ if $k \in s_{(l)}$, $l = 1, \ldots, L$ and $k = 1, \ldots, n_l$.

So we will work under this assumption throughout.

As in the previous section, let **C** be the $n \times L$ cluster-membership matrix, and let $\mathbf{d}^+ = \mathbf{C}^T\mathbf{d}$ be the $(L \times 1)$-vector of sums of element-level sampling weights within clusters. Under assumption (vi), we have $\mathbf{d}^+ = \mathbf{C}^T\mathbf{d} = \tilde{\mathbf{D}}\mathbf{C}^T\mathbf{1}_n$, i.e. the *l*-th element of $\mathbf{d}^+$ is $n_l d_{(l)}$.

An example of the above situation is often met in household surveys: clusters are households, and are selected according to possibly a complex sampling design. Within households (or clusters), individuals (the ultimate sampling elements) are all selected for participating in the survey. Then it is indeed well known that (vi) is satisfied. Auxiliary information is usually available at household as well as at individual level (and this information is often already used at the sampling design stage). For instance, the total numbers of individuals and households living in large geographical areas are known, and for each household, and therefore also for each individual, it is known in which area it lives.

To tackle such a complex problem, we can proceed in several ways, depending on what kind of calibrated weights we want to obtain, and which auxiliary data are thought to be of interest for calibration. The ultimate aim of this section is to discuss how element-level and cluster-level data can be integrated into a single calibration problem. In each of the next sub-sections we will also indicate what are the consequences of the applied calibration technique on estimation of totals of element- as well as cluster-level study variables.

III.E.2 Element-level calibration, ignoring cluster-level auxiliary information

This is an 'ordinary' generalised calibration problem, formulated as

$$\left\{\min \mathbf{d}^T G(\mathbf{g}); \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}, \tag{III.19}$$

where $\breve{\mathbf{X}} = \mathbf{DX}$. Solving this problem, we get g-weights $g_k = F\left(\mathbf{x}_k^T \boldsymbol{\lambda}\right)$, where $\boldsymbol{\lambda}$ is an optimal *m*-vector of Lagrange multipliers. These g-weights vary across elements, and need not be the same for elements within the same cluster. For estimation of totals of element-level study variables *y*, we apply the following formula:

. $$\hat{t}_y = \sum_{k \in s} d_k g_k y_k = \sum_{l=1}^{L} d_{(l)} \sum_{k \in s} F\left(\mathbf{x}_k^T \boldsymbol{\lambda}\right) y_k \ . \tag{III.20}$$

Estimation of the total of a cluster-level study variable *y* is not immediately obvious. Consider for instance the variable "household size" (i.e. number of individuals in a household, if clusters are households). An *ad hoc* strategy can be applied: artificially transform the cluster characteristic *y* into an element characteristic, by assigning the value $y_{(l)}/n_l$ to element *k* if $k \in s_{(l)}$. Then:

. $$\hat{t}_y = \sum_{k \in s} d_k g_k \frac{y_{(l)}}{n_l} = \sum_{l=1}^{L} d_{(l)} \frac{\sum_{k \in s} F\left(\mathbf{x}_k^T \boldsymbol{\lambda}\right)}{n_l} y_{(l)} \ . \tag{III.21}$$

This is equivalent to constructing a cluster-level g-weight from the estimated element-level g-weights as an arithmetic average: $g_{(l)} = \dfrac{\sum_{k \in s} F\left(\mathbf{x}_k^T \boldsymbol{\lambda}\right)}{n_l}$. (We'll see later that this technique is applied currently in the Labour Force Survey at Statistics Belgium (section V.B.3).) If the cluster variable is one of the variables z_j, then we get an estimate $\hat{t}_{z_j}$ which is not necessarily equal to the fixed total s_j (j = 1, …, p).

III.E.3 Cluster-level calibration, ignoring element-level auxiliary information

This too is an 'ordinary' generalised calibration problem, formulated as

$$\left\{\min \tilde{\mathbf{d}}^T G(\tilde{\mathbf{g}}); \breve{\mathbf{Z}}^T \tilde{\mathbf{g}} = \mathbf{s}, \tilde{\mathbf{g}} \in \tilde{\Omega}_B\right\}, \tag{III.22}$$

where $\breve{\mathbf{Z}} = \tilde{\mathbf{D}}\mathbf{Z}$. Solving this problem, we get g-weights $g_{(l)} = F\left(\mathbf{z}_l^T \boldsymbol{\gamma}\right)$, where $\boldsymbol{\gamma}$ is an optimal *p*-vector of Lagrange multipliers. These g-weights vary across clusters. For estimation of totals of study variables *y*, we apply the following formulae:

- if *y* is a cluster-level variable:

$$\hat{t}_y = \sum_{l=1}^{L} d_{(l)} g_{(l)} y_l = \sum_{l=1}^{L} d_{(l)} F\left(\mathbf{z}_l^T \boldsymbol{\gamma}\right) y_l \ , \tag{III.23}$$

- if *y* is an element-level variable:

. $$\hat{t}_y = \sum_{k \in s} d_k g_k y_k = \sum_{l=1}^{L} \sum_{k \in s_{(l)}} d_k g_{(l)} y_k = \sum_{l=1}^{L} d_{(l)} F\left(\mathbf{z}_l^T \boldsymbol{\gamma}\right) \left(\sum_{k \in s_{(l)}} y_k \right). \qquad \text{(III.24)}$$

Notice that the element-level calibration variables x_j $(j = 1,...,m)$ can be considered as element-level study variables, but that $\hat{t}_{x_j} = \sum_{k \in s} d_k g_k x_{kj} = \sum_{l=1}^{L} F\left(\mathbf{z}_l^T \boldsymbol{\gamma}\right) \sum_{k \in s_{(l)}} d_k x_{kj} \neq t_j$, generally, where t_j is the initially fixed total for that variable x_j.

III.E.4 Element-level calibration, imposing constant element-weights within clusters, but still ignoring other cluster-level auxiliary information

One of the advantages of formula (III.24), as opposed to formula (III.20), for estimation of totals of element-level study variables, is that elements in the same cluster have the same g-weights, which may be expected to result into more stability on the element-level, and to more stable estimates of totals of element-level variables. This stability is a consequence of calibration on cluster-level auxiliary information only. Stability can also be achieved by clustering (section III.D), when calibration is on element-level auxiliary information only. We know from section III.D that the restricted calibration problem, using element-level auxiliary information only, but imposing equal g-weights within clusters, can be formulated as follows:

$$\left\{ \min \mathbf{d}^{+T} G(\tilde{\mathbf{g}}); \breve{\mathbf{H}}^T \tilde{\mathbf{g}} = \mathbf{t}, \tilde{\mathbf{g}} \in \tilde{\Omega}_B \right\}, \qquad \text{(III.25)}$$

where $\breve{\mathbf{H}} = \mathbf{D}^{+} \tilde{\mathbf{H}}$ and $\tilde{\mathbf{H}} = \mathbf{D}^{+^{-1}} \mathbf{C}^T \mathbf{D} \mathbf{X}$ (section III.D). Notice that the *lj*-th entry of the latter matrix is a weighted average of *x*-values (formula (III.18) in section III.D). Still working under assumption (vi), we get a simple unweighted arithmetic average: $\tilde{h}_{lj} = \dfrac{\sum_{k \in s_{(l)}} x_{kj}}{n_l} = \bar{x}_{(l)j}$. Therefore, the g-weights are $g_{(l)} = F\left(\bar{\mathbf{x}}_{(l)}^T \boldsymbol{\lambda}'\right)$, where $\boldsymbol{\lambda}'$ is a <u>new</u> optimal *m*-vector of Lagrange multipliers, and $\bar{\mathbf{x}}_{(l)}^T = \left(\bar{x}_{(l)1},...,\bar{x}_{(l)m}\right)$. Hence the g-weights indeed vary across clusters, but not across elements within clusters. For estimation of totals of study variables *y*, we apply:

- if *y* is a cluster-level variable:

$$\hat{t}_y = \sum_{l=1}^{L} d_{(l)} g_{(l)} y_l = \sum_{l=1}^{L} d_{(l)} F\left(\bar{\mathbf{x}}_{(l)}^T \boldsymbol{\lambda}'\right) y_l \,, \qquad \text{(III.26)}$$

- if *y* is an element-level variable:

. $$\hat{t}_y = \sum_{k \in s} d_k g_k y_k = \sum_{l=1}^{L} \sum_{k \in s_{(l)}} d_k g_{(l)} y_k = \sum_{l=1}^{L} d_{(l)} F\left(\bar{\mathbf{x}}_{(l)}^T \boldsymbol{\lambda}'\right) \left(\sum_{k \in s_{(l)}} y_k \right). \qquad \text{(III.27)}$$

Notice the similarity between (III.26) and (III.40), and between (III.27) and (III.24). Unfortunately, however, $\hat{t}_{z_j} \neq s_j$ (j = 1, …, L).

III.E.5 Integrated element-cluster-level calibration

The disadvantage of the approaches in the previous sub-sections III.E.2-4 is that if calibration is on element-level auxiliary information only, then results are not necessarily numerically consistent at the cluster-level, while if calibration is on cluster-level auxiliary information only, then results are not necessarily numerically consistent at the element-level. It is however possible to achieve numerical consistency at the two levels simultaneously.

Two-level numerical consistency can only be realised if the g-weights are constant within clusters. This implies that we have to combine the technique for calibration on cluster-level auxiliary information, as outlined in section III.E.3, with the clustering technique for calibration on element-level auxiliary information, as outlined in section III.E.4. Combination of the two sets of calibration constraints is straightforward: we have to consider the system with sub-systems $\breve{\mathbf{Z}}^T\tilde{\mathbf{g}} = \mathbf{s}$ and $\breve{\mathbf{H}}^T\tilde{\mathbf{g}} = \mathbf{t}$. This we can write as $\breve{\mathbf{V}}^T\tilde{\mathbf{g}} = \mathbf{u}$, with $\breve{\mathbf{V}} = \left(\breve{\mathbf{Z}}|\breve{\mathbf{H}}\right)$ and $\mathbf{u} = \begin{pmatrix}\mathbf{s}\\ \mathbf{t}\end{pmatrix}$. Now, from section III.E.3, we have $\breve{\mathbf{Z}} = \tilde{\mathbf{D}}\mathbf{Z}$, and therefore we also define $\mathbf{H} = \tilde{\mathbf{D}}^{-1}\breve{\mathbf{H}}$, such that $\breve{\mathbf{H}} = \tilde{\mathbf{D}}\mathbf{H}$ and $\breve{\mathbf{V}} = \tilde{\mathbf{D}}\mathbf{V}$, with $\mathbf{V} = \left(\mathbf{Z}|\mathbf{H}\right)$. Notice that, from section III.E.4, $\mathbf{H} = \tilde{\mathbf{D}}^{-1}\breve{\mathbf{H}} = \tilde{\mathbf{D}}^{-1}\mathbf{D}^{+}\tilde{\mathbf{H}} = \tilde{\mathbf{D}}^{-1}\mathbf{D}^{+}\mathbf{D}^{+^{-1}}\mathbf{C}^T\mathbf{D}\mathbf{X} = \tilde{\mathbf{D}}^{-1}\mathbf{C}^T\mathbf{D}\mathbf{X}$, i.e. $\mathbf{H}$ is simply the $(L \times m)$-matrix with entries $h_{lj} = \sum_{k \in s_{(l)}} x_{kj} = n_l \bar{x}_{(l)j}$. The system of calibration equations can then be written equivalently as $\mathbf{V}^T\tilde{\mathbf{D}}\tilde{\mathbf{g}} = \mathbf{u}$, which makes the weights explicit, and which also indicates that the objective function to be used is $\tilde{\mathbf{d}}^T G(\tilde{\mathbf{g}})$ (as in section III.E.3, but not as in section III.E.4). The latter in fact means that the integrated element-cluster-level approach is essentially a cluster-level calibration, with individual calibration variables being summed within clusters. Thus, the final integrated element-cluster-level calibration problem is:

$$\left\{\min \tilde{\mathbf{d}}^T G(\tilde{\mathbf{g}}); \breve{\mathbf{V}}^T\tilde{\mathbf{g}} = \mathbf{u}, \tilde{\mathbf{g}} \in \tilde{\Omega}_B\right\}. \tag{III.28}$$

Alternative formulations are:

$$\left\{\min \tilde{\mathbf{d}}^T G(\tilde{\mathbf{g}}); \begin{pmatrix}\breve{\mathbf{Z}}^T\\ \breve{\mathbf{H}}^T\end{pmatrix}\tilde{\mathbf{g}} = \begin{pmatrix}\mathbf{s}\\ \mathbf{t}\end{pmatrix}, \tilde{\mathbf{g}} \in \tilde{\Omega}_B\right\}, \tag{III.29}$$

and:

$$\left\{\min \tilde{\mathbf{d}}^T G(\tilde{\mathbf{g}}); \begin{pmatrix}\mathbf{Z}^T\\ \mathbf{H}^T\end{pmatrix}\tilde{\mathbf{D}}\tilde{\mathbf{g}} = \begin{pmatrix}\mathbf{s}\\ \mathbf{t}\end{pmatrix}, \tilde{\mathbf{g}} \in \tilde{\Omega}_B\right\}. \tag{III.30}$$

The latter indicates clearly how the input files for our calibration software module will have to be constructed in practical applications. Notice that, if the constant variable x = 1 were one of the element-level calibration variables, then this variable would become a cluster-level calibration variable h, with values equal to the cluster sizes n_l.

The resulting g-weights can now be written as:

$$\begin{aligned} g_{(l)} &= F\left(\mathbf{z}_l^T\boldsymbol{\gamma}^* + n_l\bar{\mathbf{x}}_{(l)}^T\boldsymbol{\lambda}^*\right) \\ &= F\left(\mathbf{z}_l^T\boldsymbol{\gamma}^* + \mathbf{h}_l^T\boldsymbol{\lambda}^*\right) \\ &= F\left(\mathbf{v}_l^T\begin{pmatrix}\boldsymbol{\gamma}^* \\ \boldsymbol{\lambda}^*\end{pmatrix}\right), \end{aligned} \qquad \text{(III.31)}$$

where $\boldsymbol{\gamma}^*$ and $\boldsymbol{\lambda}^*$ are new vectors of Lagrange multipliers, with length *p* and *m*, respectively. For estimation of totals of study variables *y*, we have:

- if *y* is a cluster-level variable:

$$\hat{t}_y = \sum_{l=1}^{L} d_{(l)} g_{(l)} y_l = \sum_{l=1}^{L} d_{(l)} F\left(\mathbf{z}_l^T\boldsymbol{\gamma}^* + \mathbf{h}_l^T\boldsymbol{\lambda}^*\right) y_l \,, \qquad \text{(III.32)}$$

- if *y* is an element-level variable:

$$\hat{t}_y = \sum_{k\in s} d_k g_k y_k = \sum_{l=1}^{L}\sum_{k\in s_{(l)}} d_k g_{(l)} y_k = \sum_{l=1}^{L} d_{(l)} F\left(\mathbf{z}_l^T\boldsymbol{\gamma}^* + \mathbf{h}_l^T\boldsymbol{\lambda}^*\right)\left(\sum_{k\in s_{(l)}} y_k\right). \qquad \text{(III.33)}$$

The reader will compare these formulas with (III.26) and (III.23), and with (III.27) and (III.24), respectively. Remember that we now have complete numerical consistency: $\hat{t}_{z_j} = s_j$ (j = 1, …, L) and $\hat{t}_{x_j} = t_j$ (j = 1, …, m).

III.E.6 Discussion

In the previous sub-section we have described a technique for dealing with two levels of auxiliary information. The proposed method is a simultaneous calibration technique that is essentially a cluster-level calibration method. In this study we will present – and briefly discuss – some preliminary results, from application of this technique to the Time Use Survey (1999) at Statistics Belgium (section V.D.2). Other surveys will be considered later for application of this technique.

The technique of simultaneous calibration at several levels of auxiliary is not new, as the literature indicates. Moreover, in the literature on calibration, several alternative methods for dealing with auxiliary information that is available at two or more levels are presented. See for instance Sautory *et al* (1999), Mohadjer (1999), Hidiroglou and Särndal (1998), Dupont (1995), Kalton and Brick (1995), Lemaître and Dufour (1987), among others. Notice that these studies often treat the problem in the context of *multi-phase sampling*, which may lead to slightly different types of formulas for, for instance, the g-weights. In multi-phase sampling, it may seem more natural to calibrate step by step, each step corresponding to one of the phases in the sampling procedure and resulting into auxiliary information related to each phase, and sometimes be built up step by step. Cluster sampling is a particular case of two-phase sampling, as for instance Särndal *et al* (1992, p.344) point out.

Those complex calibration methods are worth to be studied further at Statistics Belgium, both at a theoretical level for thorough understanding and at a more practical level, focussing on implementation of the techniques in different multi-phase or multi-stage sample surveys. Time should

be spent on comparison of several alternatives. The case of the LFS would be a useful starting point for an in-depth study, eventually resulting in a stable, but flexible, weighting scheme, that incorporates auxiliary information at two (or more?) levels simultaneously and therefore produces numerical consistency at several levels.

This discussion has drawn our attention to a "small" problem, related to the clusters' sampling weights (or initial weights) $\tilde{\mathbf{d}}$. If cluster-level auxiliary information is derived from element-level data, then a natural choice seems to be the sum of the element-level sampling weights. This implies an additional weighting factor in the objective function: larger households get a higher weight. If, on the other hand, cluster-level auxiliary information is directly observed for clusters as a whole, then it seems natural to use the clusters' sampling weights, which implies that the size of the household is completely ignored. It would be interesting to study the effect of variability in cluster sizes on the estimated cluster-level g-weights, and the resulting point and variance estimates for totals of study variables.

To close this section, we summarise the results in the table on the next page. It may be noticed that application of the *collapsing* technique, as discussed in section III.C.2 could be considered in each of the four situations summarised in the table. Collapsing is useful for reduction of the size of the calibration problem, but otherwise doesn't add anything new. Collapsing is particularly useful for calibration on qualitative variables, when it may be expected that a significant number of units (elements or clusters) have the same calibration vector, such that a significant reduction in the size of the problem can be realised.

Table 3.8 *Overview of calibration models, if two levels of auxiliary information are available*

Level of auxiliary information used	A priori restriction on g-weights – Special technique	Auxiliary information (B : basic ; D : derived)	Calibration constraints	g-Weights	Numerical consistency $\hat{t}_{x_j} = t_j$?	Numerical consistency $\hat{t}_{z_j} = s_j$?
Element	-	$\mathbf{X}, \mathbf{d}, \mathbf{t}$ (B)	$\breve{\mathbf{X}}^T\mathbf{g} = \mathbf{X}^T\mathbf{D}\mathbf{g} = \mathbf{t}$	$g_k = F\left(\mathbf{x}_k^T\boldsymbol{\lambda}\right)$	Yes	-
Element	$g_k = g_{(l)}$ Clustering	$\tilde{\mathbf{H}}, \mathbf{d}^+, \mathbf{t}$ (D)	$\breve{\mathbf{H}}^T\tilde{\mathbf{g}} = \tilde{\mathbf{H}}^T\mathbf{D}^+\tilde{\mathbf{g}} = \mathbf{t}$	$g_{(l)} = F\left(\bar{\mathbf{x}}_{(l)}^T\boldsymbol{\lambda}'\right)$	Yes	-
Cluster	-	$\mathbf{Z}, \tilde{\mathbf{d}}, \mathbf{s}$ (B)	$\breve{\mathbf{Z}}^T\tilde{\mathbf{g}} = \mathbf{Z}^T\tilde{\mathbf{D}}\tilde{\mathbf{g}} = \mathbf{s}$	$g_{(l)} = F\left(\mathbf{z}_l^T\boldsymbol{\gamma}\right)$	-	Yes
Element + Cluster	$g_k = g_{(l)}$ Clustering Simultaneous	$\begin{pmatrix}\mathbf{Z}^T\\ \mathbf{H}^T\end{pmatrix}, \tilde{\mathbf{d}}, \begin{pmatrix}\mathbf{s}\\ \mathbf{t}\end{pmatrix}$ (B) (D)	$\begin{pmatrix}\breve{\mathbf{Z}}^T\\ \breve{\mathbf{H}}^T\end{pmatrix}\tilde{\mathbf{g}} = \begin{pmatrix}\mathbf{Z}^T\\ \mathbf{H}^T\end{pmatrix}\tilde{\mathbf{D}}\tilde{\mathbf{g}} = \begin{pmatrix}\mathbf{s}\\ \mathbf{t}\end{pmatrix}$	$g_{(l)} = F\left(\mathbf{z}_l^T\boldsymbol{\gamma}^* + n_l\bar{\mathbf{x}}_{(l)}^T\boldsymbol{\lambda}^*\right)$	Yes	Yes

Where: $\mathbf{X} = \left(x_{kj}\right)$ $\quad \mathbf{d} = \left(d_k\right)$ $\quad \mathbf{t} = \left(t_j\right)$

$\mathbf{Z} = \left(z_{lj}\right)$ $\quad \tilde{\mathbf{d}} = \left(d_{(l)}\right)$ $\quad \mathbf{s} = \left(s_j\right)$

$\tilde{\mathbf{H}} = \left(\tilde{h}_{lj}\right) = \left(\bar{x}_{(l)j}\right)$ $\quad \mathbf{d}^+ = \left(\sum_{k \in s_{(l)}} d_k\right) = \left(n_l d_{(l)}\right)$

$\mathbf{H} = \left(h_{lj}\right) = \left(n_l \bar{x}_{(l)j}\right)$

III.F ESTIMATING POPULATION TOTALS OF SURVEY VARIABLES: DEALING WITH MISSING VALUES

The basic formula for the *calibration estimator* of the total $\mathbf{t}_y = \sum_{k \in U} y_k$ of a survey variable *y* is

$$\hat{\mathbf{t}}_y = \sum_{k \in s} w_k y_k = \mathbf{w}^T \mathbf{y}, \tag{III.34}$$

where $\mathbf{w} = (w_1, ..., w_n)^T$ is the vector of calibrated weights and $\mathbf{y} = (y_1, ..., y_n)^T$ is the vector of *y*-values for the sample elements. So far we have assumed that the design matrix **X** is *complete*, and now, in order to have a workable formula (III.34), we must furthermore assume that the data matrix $(\mathbf{X}|\mathbf{y})$ is *complete*, by which we mean that for all *n* sample elements values are available for all auxiliary *x*-variables and for the survey variable *y*. Obviously, the sample *s* might be the *respondent sample*. Thus we have covered so far the situation in which there is *unit non-response* only. Notice that the calibration technique is assumed to correct appropriately for unit non-response.

In practice, the data matrix $(\mathbf{X}|\mathbf{y})$ (for the respondent sample) often is *incomplete* (with the restriction, of course, that a row, or *case*, is never missing completely): we then talk about *item non-response* (in addition to unit non-response). One solution to the problem is to complete first the data matrix, and next to proceed as outlined so far. Filling in the gaps in the data matrix is called *imputation*, and the reader will know that there exists a large collection of methods for imputation, ranging from the simplest mean value imputation to the most sophisticated form of (model-based) regression imputation, or from simple *ad hoc* methods to ingenious statistical methods.

We do not deal with imputation methods in this text for two reasons. Firstly, calibration theory basically ignores whether *x*-values are imputed or observed (or obtained from registers): among other things, the calculation of variances for calibration estimators needs some modification, taking into account the uncertainty in imputed *x*-values. Secondly, imputation should be another topic for investigation at the most general level at Statistics Belgium. Only then imputation and calibration should be combined to further improve in a synergistic way our survey estimates.

Consequently the discussion in the present section is very pragmatic and merely aims to provide practical methods to deal with incomplete data matrices, e.g. caused by item non-response.

We distinguish three situations of incompleteness. Table 3.9 presents these schematically; notice the distinction between *m complete cases* and *n – m incomplete cases*.

Table 3.9 *Types of incomplete data matrices*

Case	Situation 1		Situation 2		Situation 3	
	X	**y**	**X**	**y**	**X**	**y**
1	× × × × ×	×	× × × × ×	×	× × × × ×	×
:		:		:		:
:		:		:		:
k	× × × × ×	×	× × × × ×	×	× × × × ×	×
:		:		:		:
:		:		:		:
m	× × × × ×	×	× × × × ×	×	× × × × ×	×
m+1	× × × × ×	?	× ? × × ×	×	? × ? × ×	×
:		:		:		:
n	× × × × ×	?	× × ? × ?	×	× × ? × ×	?

The solution in each of these three situations is based on the procedures as discussed in the previous sections, occasionally followed by an additional correction. This correction can be such that for at least one of the calibration variables the calibration constraint is still satisfied.

SITUATION 1 Complete **X** and incomplete **y**.

Let s_c be the sub-sample of *m* complete cases. Then

$$\hat{t}_y = \frac{\sum_{k\in s} w_k}{\sum_{k\in s_c} w_k} \sum_{k\in s_c} w_k y_k = \hat{N}\tilde{y}_{s_c} = \sum_{k\in s_c} w_k^* y_k \,, \qquad \text{(III.35)}$$

where $\hat{N} = \sum_{k\in s} w_k$ is the calibration estimator of the population size *N*, $\tilde{y}_{s_c} = \frac{\sum_{k\in s_c} w_k y_k}{\sum_{k\in s_c} w_k}$ is a ratio estimator for the mean of the survey variable based on the sample s_c of complete cases, and $w_k^* = \frac{\hat{N}}{\sum_{k\in s_c} w_k} w_k$ is an *adjusted calibration weight*. If the constant variable $x = 1$ is among the calibration variables, then $\hat{N} = N$. Then the estimator is sometimes called the *expanded sample mean* (Särndal *et al*, 1992, p.258) (strictly speaking only if the weights would be the sampling weights). Alternatively, one can use any other calibration variable x_{j_o}, say, and the estimator would then be:

$$\hat{t}_y = \frac{\sum_{k\in s} w_k x_{kj_o}}{\sum_{k\in s_c} w_k x_{kj_o}} \sum_{k\in s_c} w_k y_k = t_{j_o} \frac{\sum_{k\in s_c} w_k y_k}{\sum_{k\in s_c} w_k x_{kj_o}}. \qquad \text{(III.36)}$$

This is a *ratio estimator* for the total t_y, based on the calibration variable x_{j_o} (Särndal *et al*, 1992, p.180) (strictly speaking only if the weights would be the sampling weights).

It is possible to extend these estimators so that more than one calibration variable is taken into account to make the adjustment. If these calibration variables are the indicator variables corresponding to the categories of some qualitative variable(s), than a regression type estimator as (7.6.1) in Särndal *et al* (1992) is obtained.

SITUATION 2 Incomplete **X** and complete **y**.

Here, calibration weights are calculated from the complete cases sample s_c, and calibration weights are therefore available for the elements in this sample only. Henceforth, only the observed *y*-values for elements in s_c can be used:

$$\hat{t}_y = \sum_{k \in s_c} w_k y_k \, .$$

I.e. the sample *s* is simple replaced by the sample s_c. A drawback of this solution is that some observed values for the survey variable are not used. This can, strictly speaking, only be accepted if non-response is ignorable, i.e. the value of *y* does not depend on whether the *x*-values are observed or not. Otherwise some bias can be introduced.

SITUATION 3 Incomplete **X** and incomplete **y**.

This is similar to situation 2.

III.G VARIANCE ESTIMATION

It is well known that if the linear calibration method is used, then the calibration estimator of totals of study variables *y* is a GREG (*generalised regression*) estimator. The formula for the *asymptotic variance* of the GREG estimator is not complicated, and an estimate is easily computed with the following formula (see Deville and Särndal, 1992):

$$\hat{V}\left(\hat{t}_y\right) = \sum_{k\in s}\sum_{l\in s}\left(1-\frac{\pi_k\pi_l}{\pi_{kl}}\right)\left(w_k e_k\right)\left(w_l e_l\right), \tag{III.37}$$

where π_k $(k \in s)$ are the first order inclusion probabilities, π_{kl} $(k,l \in s)$ are the second-order inclusion probabilities, w_k $(k \in s)$ are the calibrated weights, and e_k $(k \in s)$ are *residuals*, to be calculated from a sample-based *weighted linear regression* of *y* on the calibration variables. In matrix notation, this is written as:

$$\hat{V}\left(\hat{t}_y\right) = \hat{\mathbf{e}}^T\mathbf{W}\breve{\mathbf{\Delta}}\mathbf{W}\hat{\mathbf{e}}, \tag{III.38}$$

where $\breve{\mathbf{\Delta}} = \left(1-\frac{\pi_k\pi_l}{\pi_{kl}}\right)$, $\mathbf{W} = diag\left(w_k\right)$ and $\mathbf{e} = \left(e_1,...,e_n\right)^T$. The residual vector **e** can be computed from the survey data as follows:

$$\hat{\mathbf{e}} = \mathbf{y} - \mathbf{X}\hat{\mathbf{b}} = \mathbf{y} - \mathbf{X}\left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{D}\mathbf{y}, \tag{III.39}$$

where $\hat{\mathbf{b}} = \left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{D}\mathbf{y}$ is an estimator for the vector of regression coefficients in the weighted linear regression of *y* on the calibration variables. Notice again the use of g-inverses.

In calibration methodology, it is also proved (Deville and Särndal, 1992) that any calibration estimator $\hat{t}_y$ is asymptotically equivalent with the GREG-estimator. Therefore, the variance of any calibration estimator $\hat{t}_y$ can be estimated using the above formulae for estimating the variance of the GREG-estimator, at least in large samples.

This solves the basic problem of estimating the variance for calibration estimators for totals t_y. A drawback of the method is that the second-order inclusion probabilities should be available and strictly positive. We are currently working on (approximate) mathematical models for describing the sampling design of complex surveys, such that second-order inclusion probabilities can be (approximately) calculated.

It must be noticed that the above formulae for variance estimation need to be modified in case of (substantial) non-response, when the sample *s* is in fact the respondent sample. Calibration is still applied to adjust for non-response (see Dupont (1994), Skinner (1999), Lundstrom and Särndal (1999)). Variance estimation then becomes more difficult, but a complete discussion of it is beyond the scope of this text.

Finally, we mention here a straightforward extension of the above formulae. Usually totals have to be estimated for several study variables. The vector **y** should then be replaced with a matrix **Y**, of dimension $n \times p$, where *p* is the number of study variables involved. Hence the *j*-th column in **Y** corresponds to the *j*-th study variable. Accordingly, $\hat{t}_y$ becomes a vector, $\hat{\mathbf{t}}_y$ say, of calibration

estimates of totals, $\hat{\mathbf{b}}$ becomes a matrix, $\hat{\mathbf{B}}$ say, of estimates of regression coefficients, and $\hat{\mathbf{e}}$ becomes a matrix, $\hat{\mathbf{E}}$ say, of residuals, where:

$$\hat{\mathbf{E}} = \mathbf{Y} - \mathbf{X}\hat{\mathbf{B}} = \mathbf{Y} - \mathbf{X}\left(\mathbf{X}^T\mathbf{D}\mathbf{X}\right)^{-}\mathbf{X}^T\mathbf{D}\mathbf{Y}\,, \tag{III.40}$$

and, finally,

$$\hat{V}\left(\hat{\mathbf{t}}_y\right) = \hat{\mathbf{E}}^T\mathbf{W}\breve{\mathbf{\Delta}}\mathbf{W}\hat{\mathbf{E}}\,, \tag{III.41}$$

is the estimated $p \times p$ *variance-covariance matrix* for the vector $\hat{\mathbf{t}}_y$ of calibration estimators of the totals of the study variables.

This extension is useful in practice, for several reasons. It shows how to deal with many survey variables at the same time. For instance, if the frequency distribution with respect to a qualitative variable has to be estimated, indicator variables have to be constructed, which then become the study variables, for which simultaneous estimation is possible, following the above formulae (III.40) and (III.41). The formulae are also useful with respect to implementation of (co-) variance estimation; see section IV.C.2.vii.

Covariance estimates might also be useful, for instance, when ratios of estimates of totals are considered. This is an example of a *complex statistic*; variance estimation can be based on Taylor series expansion, which then involves estimates of both variances and co-variances.

Chapter IV

g-CALIB-S

Software for generalised calibration

IV.A SOME OTHER IMPLEMENTATIONS

Before we start discussing our SPSS implementation g-CALIB-S, we briefly mention some other, similar systems, developed at other statistical institutes during the past decade. Two of them, GES and CALMAR, are SAS-based systems; the third one, BASCULA, is developed under Delphi. We do not intend to give a complete discussion of these software packages, firstly because our documentation about them is not complete, and secondly because we have no experience at all in running the programs in practical or hypothetical situations. Nevertheless, the next few sub-sections may give an indication of were our software can be placed in relation to those alternative tools.

IV.A.1 SAS-based system GES (Statistics Canada)

GES stands for *Generalized Estimation System*. According to Estevao *et al* (1995), GES is based on the *generalised regression* (GREG) estimation framework, developed by Särndal *et al* (1992). This framework covers a class of calibration estimators to which many commonly used estimators belong. However, the generalised calibration framework introduced by Deville and Särndal (1992) is larger.

GES is accompanied with another SAS-based system, GSAM, the *Generalised Sampling System*. Both systems cover several simple and more complex sampling designs, which becomes important in variance estimation that is included in GES. The advantage of GES is thus the integration of calibration, estimation (for totals, means, ratios and proportions, for the entire population or for domains) and variance estimation, although only in the GREG framework.

IV.A.2 Calibration in BASCULA (Statistics Netherlands)

The new version of BASCULA is developed under Delphi for Windows 95 (Nieuwenbroek *et al*, 1997). BASCULA, like GES, is based on the GREG estimation framework. Whether the software will also include weighting according to Deville and Särndal (1992), as announced in Nieuwenbroek (1997), is not clear. Variance estimation will be based on resampling techniques, in particular on *balanced repeated sampling* (BRR). As GES, BASCULA is thus another more complete system, integrating point estimation (and weighting) and variance estimation in one stand-alone package.

An interesting peculiarity in BASCULA is the way that the g-weights are bounded in the linear method. Contrary to what is done in g-CALIB-S and in CALMAR, the g-weights are not truncated, but rather rescaled (in a fairly complex way); the procedure is iterative. This seems to be a smoother bounding technique than simple (iterative) truncation. It is worth to compare the methods, especially given our experience that convergence under the truncated liner method in g-CALIB-S may be more "difficult"; see section V.E.4 for an example.

IV.A.3 The SAS module CALMAR (INSEE – France)

The SAS-module CALMAR, for *Calage sur Marges*, is based on the generalised calibration framework introduced by Deville and Särndal (1992); see also Deville *et al* (1993) and Sautory (1993). Our tool g-CALIB-S is very close to CALMAR. They both concentrate on estimation of calibrated weights and g-weights. A central device in both packages is the distance function *G*. From a practical point of view, this is obviously very useful, since it allows the user to restrict the weights

flexibly in various ways. From a theoretical point of view it is interesting to notice that the generalised calibration framework is (much) larger than the GREG framework.

CALMAR in some sense is better than our software g-CALIB-S, at least at present. CALMAR indeed is user-friendlier. This is mainly due to the fact that underlying quantitative and qualitative calibration variables are automatically transformed into an explicit design matrix in CALMAR. This largely reduces preparatory work on the data by the user her/himself. However, if several quantitative as well as qualitative variables are calibrated on, then the user of CALMAR still has to do the required transformations on his original data, in order to obtain a standard format for his input data. For instance, joint-effects for qualitative variables, or between a quantitative variable and one or more qualitative variables, have to be generated through one or more additional variables, which must be constructed by the user. Another point where CALMAR is scoring better than g-CALIB-S is error detection and reporting, definitely useful for the more application-oriented user.

It can be argued however, that our software g-CALIB-S is potentially at least as powerful as CALMAR. This is a consequence of using g-inverse matrices. If some practical problems could be solved appropriately in g-CALIB-S, then this package will deliberately be very competitive too. Of course, the approach outlined in this study can probably easily be implemented in CALMAR. At first glance, looking quickly in the SAS/IML guide for matrix language, it is revealed that SAS has powerful built-in procedures, which are not available in SPSS. Anyway, the experienced statistician, with a little feeling for maths, should be able to use g-CALIB-S efficiently. The applications discussed in chapter 5 might illustrate this. The situation is comparable to a situation in regression modelling: users of GLIM (Francis, 1993) should understand very well the theoretical background of this package. Hence, people who like (statistical) puzzling a little bit, will like, and probably prefer, g-CALIB-S.

Despite its weaknesses, g-CALIB-S will soon become a universal tool at Statistics Belgium. This is because SPSS will stay the basic statistical package, at least for some years. CALMAR will be studied soon, and compared with g-CALIB-S, at the Department of Methodology and Co-ordination at Statistics Belgium, since the members of that department very recently obtained a license for SAS too.

IV.B SPSS IMPLEMENTATION G-CALIB-S

IV.B.1 Introduction

In chapter II we have formulated the generalised calibration problem as a mathematical programming problem, in terms of the calibrated weights **w** :

$$\left\{\min D(\mathbf{d},\mathbf{w}); \mathbf{X}^T\mathbf{w} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}, \tag{IV.1}$$

or in terms of the g-weights **g** :

$$\left\{\min \mathbf{d}^T G(\mathbf{g}); \mathbf{X}^T\mathbf{D}\mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}. \tag{IV.2}$$

Recall that Ω_B, if explicitly specified, takes the form $\left[L,U\right]^n$, with $0 \le L \le U$ (and, often, $\mathbf{1}_n \in \left[L,U\right]^n$, i.e. $L \le 1 \le U$; see section III.A.2). Written as in (IV.2), it becomes immediately clear which are the major data input for our calibration software: the design matrix **X**, the initial weight vector **d**, and the calibration totals vector **t**; apart from this, the distance function *G*, and, occasionally, the lower bound *L* and the upper bound *U* for the g-weights have to be specified.

In section IV.B.2 we describe the core module g-CALIB-S.sps, for which the above-mentioned data are the main input. All input is defined through *program parameters*. There are some other parameters than those mentioned here before, to be set by the user before running the program. These are, for instance, the tolerance ε in the convergence criterion and the maximum number of iterations. In the same section we also discuss the features of g-CALIB-S, such as its output and requirements for adequate functioning of the software. Notice the subtle difference between g-CALIB-S.sps and g-CALIB-S, which may both be called "the core module". The latter refers to the set of syntax files g-PREPARE.sps (see hereafter) and g-CALIB-S.sps. It should be clear from the context, and from inclusion or exclusion of the suffix ".sps", about which of the two we are talking.

Never, in practice, the design matrix is presented as such, but has to be constructed from basic data files (or databases). The form of **X** depends on the (set of) calibration model(s) the statistician wants to apply, searching for an appropriate weighting scheme. In section IV.B.3 it is discussed in what form the design matrix should be available for being read, as input, by the core module g-CALIB-S. In section IV.B.6 we discuss SPSS *macros*, which we have developed for constructing the design matrix. Those macros are stored in the SPSS syntax file g-DESIGN.sps. It has to be noticed that it is up to the user to prepare the data for g-CALIB-S. The macros are meant to facilitate that process; we will illustrate this extensively in chapter V. SPSS *matrix functions* are called from the macros, and calling one of these macros is a syntax command too, so the user should be familiar with the basics of *matrix language* and *macro facilities* in SPSS. Matrix language is just an extension of basic syntax language. Of course, input files for g-CALIB-S could also be prepared by using other software, e.g. spreadsheet software like Microsoft Excel, but the data construction process must always end with the creation of two input files for g-CALIB-S, which are in SPSS data file format.

Section IV.B.4 discusses a small auxiliary module g-PREPARE.sps, to be run just before g-CALIB-S.sps is run. This module merely performs some preparatory data manipulations that are related to the way the software deals with calibration strata (see section III.A.1). In the future, we might consider further developments of g-CALIB-S. Some of these could be included in the module g-PREPARE.sps, as pointed out in section IV.C.2.

In section IV.B.5 we present a rudimentary *interface* for the core module g-CALIB-S. This interface is merely a SPSS Production Facility job, by means of which it becomes easy for the user to specify

values for the input parameters for g-CALIB-S. It also allows storing default values for the program parameters. Different jobs, possibly with the same data input files, but with different sets of values for other program parameters, can easily be stored.

IV.B.2 The core module g-CALIB-S.sps

IV.B.2.i Motivation

The heart of our SPSS software for generalised calibration is the implementation of the iterative algorithm for calculation of the g-weights **g**; see section II.D for the Basic Algorithm and section II.E.5 for the Extended Algorithm. The SPSS syntax is stored in the file g-CALIB-S.sps.

We have built this module with three objectives in mind: (1°) it should be as general as possible, (2°) it should be easy to use, and (3°) it should be well-structured, to facilitate possible extension. Generality should imply flexibility, in the sense that many different calibration techniques, from traditional post-stratification methods to the most sophisticated extrapolation methods, could all be handled in a similar way and with the same tool. This should encourage the statistician to explore alternative extrapolation techniques for the survey for which s/he is responsible. I'm strongly convinced of the fact that a general, unifying framework makes statistical methodology more transparent for the practitioner, and that it creates an environment for more efficient communication between applied statisticians, who might, at first sight, seem to do different things, but after all just apply special variants within the same general framework. Variations on the same theme! Thus, working within a uniform framework, it might be easier to understand what others are doing, and thereof to learn from each other. Generalised calibration methodology provides the unifying theoretical framework; g-CALIB-S is intended be a tool that completely reflects all features of the theory. We believe that our software already reaches this objective to a large extent.

We, methodologists at Statistics Belgium, have chosen to use SPSS for implementation of generalised calibration techniques. Or, more correctly, we barely had a choice, and just started using SPSS as an environment for implementation of calibration methodology. I think this has not at all been a bad choice, since from now on our statisticians are offered a uniform framework and a general tool for doing their job. Because of the simple fact that many statisticians were already SPSS users, it seemed logical to develop more advanced methodology in the same statistical package. By the way, SPSS only a few years ago became the general statistical tool at Statistics Belgium, just a few months before the author of this text has joined the institute mid 1997. It might have been unrealistic, and practically impossible, to switch to another statistical package for general use, with the mere purpose of being able, from thereof, to buy specialised software modules for calibration, variance estimation, etc.

One of our objectives, the second one mentioned here above, i.e. that the software must deliberately be easy to use, has been achieved by *parameterising* the computer program. This has been realised by using the macro facilities in SPSS. We discuss these parameters at length here below. This section IV.B.2 will close with a discussion of informative output (details about the iterative process, and summary statistics) produced by g-CALIB-S, and of the output file (a SPSS data file) that contains the g-weights and calibrated weights (among some other estimated statistics).

Finally, the reader might already have understood why our software has been called g-CALIB-S. "**CALIB**", of course, stands for "calibration". The meaning of "**g-**" is twofold: it stands for "generalised" in "**generalised calibration**", but at the same time also refers to the (extensive) use of "**generalised**" or "**g-inverse matrices**". Finally, "**-S**" refers to "**calibration strata**", and the way these can be dealt with in g-CALIB-S.

IV.B.2.ii Parameters in g-CALIB-S

A *parameter* in a SPSS syntax program is in fact a SPSS macro. A macro has a name, e.g. @MyMacro, and some contents, called the *macro body*. The latter can be either a collection of commands (SPSS syntax commands, in our application) or a string (enclosed or not between quotes); the latter will be called *text macros*. If @MyMacro is a text macro, and if the SPSS syntax interpreter detects, somewhere in a syntax program, the expression @MyMacro, then the contents of @MyMacro are substituted and the SPSS processor continues running (i.e. executing) the syntax program. The occurrence of @MyMacro, i.e. a macro name, in a syntax program is a *macro call*. In this section, we need to understand only the use of text macros; some useful *command macros* are discussed in section IV.B.6.

From now on, we will not distinguish between macro names (@MyMacro) and parameters for g-CALIB-S. Thus we speak simply of "*parameters*". Macro names used as parameters in our calibration modules, are starting with the special character @. This is customary when macros are used in syntax files: the purpose is to distinguish macro names from user-defined variable names, command keywords, or other identifiers. Other special characters may be used, e.g. the exclamation mark (!), as mentioned in the SPSS Syntax Guide.

g-CALIB-S works with 13 parameters. A shortlist is in the table below.

Table 4.1 *List of parameters used by* g-CALIB-S, *with short description*

Parameter (@macro name)	**Description**
@WORKDIR	The *path* defining the location of the input data files @XDATA and @CALTOT; also the path for locating output files and temporary files
@XDATA	The *name* of the SPSS input data file containing the survey data
@CALTOT	The *name* of the SPSS input data file containing the calibration totals
@XVARS	A SPSS *variable list*, defining the names of calibration variables in the files @XDATA and @CALTOT, and used in the current run
@STR_1	The *number* of the first calibration stratum to be processed in the current run
@STR_N	The *number* of the last calibration stratum to be processed in the current run
@TYPE	The calibration *method*, i.e. the distance function *G*
@SCALE	A fixed value for the *scale* parameter ϕ, if positive, or 0 or a negative value to ask the program to calculate ϕ
@L	A fixed *lower bound L* for the g-weights
@U	A fixed *upper bound U* for the g-weights
@TOL	The *tolerance* ε in the convergence criterion
@ITERMAX	The *maximum number of iterations*
@INFO	A parameter to specify whether more *detailed intermediate results* have to be included in the informative output

We know discuss each of these parameters in detail.

@WORKDIR	The *path* defining the location of the input data files @XDATA and @CALTOT; also the path for locating output files and temporary files

This parameter is a string, defining the path for locating the input data files for g-CALIB-S. It should include the drive letter, and a complete directory structure. The general format is

<Drive>:\[<directory>\[<subdirectory>\[<subdirectory>\[…]]]]

Notice that the last character in this string must always be a backslash (\). <Drive>, <directory> and <subdirectory> must be valid user-defined names, obeying general Windows rules.

Example: C:\My Documents\SurveyX\Calibration2001\

The output file (containing g-weights, calibrated weights, etc) is stored in the same directory; see below for more details about this file.

The parameter @WORKDIR is case insensitive.

@XDATA	The *name* of the SPSS input data file containing the survey data

This string parameter sets the name for the (a priori constructed) *survey data input file*, which contains, among other things, the design matrix and the initial weights. A complete description of contents and structure of the survey data file is postponed to section IV.B.3. The general format of this name is

<Filename>[.sav]

<Filename> must be a valid filename; the extension .sav is optional or can be replaced by any other extension. However, the survey data input file must be a SPSS data file.

The parameter @XDATA is case insensitive.

@CALTOT	The *name* of the SPSS input data file containing the calibration totals

This string parameter sets the name for the (a priori constructed) *calibration totals input file*, which contains, among other things, the calibration totals for each calibration stratum. A complete description of contents and structure of the calibration totals file is postponed to section IV.B.3. The general format of this name is

<Filename>[.sav]

<Filename> must be a valid filename; the extension .sav is optional or can be replaced by any other extension. However, the calibration totals input file must be a SPSS data file.

The parameter @CALTOT is case insensitive.

@XVARS	A SPSS *variable list*, defining the names of calibration variables in the files @XDATA and @CALTOT, and used in the current run

This string parameter defines a list of variable names, which g-CALIB-S will try to find in both the survey data file and the calibration totals file. We refer to the SPSS Reference Guide for details about valid variable names in SPSS. Variable lists can be defined by using comma's or blanks to separate variable names, and the keyword "TO" to intrinsically specify a collection of consecutive variables in the working data file.

Example: X0, A1, A2, A3, B1, B2, AB11, AB12, AB21, AB22, AB31, AB32
and X0, A1 to A3, B1, B2, AB11 to AB32
and X0 to AB32

are equivalent variable lists, provided that no other variables are occurring in the input files between the variable X0 and AB32. The is still some flexibility in the ordering of the variables in the input files. For example, B1 and B2 might occur before A1, A2 and A3.

The @XVARS parameter is case insensitive.

@STR_1	The *number* of the first calibration stratum to be processed in the current run

This numeric parameter gives the number of the first calibration stratum to which the calibration model, defined through other parameters, will be applied. Notice that the calibration strata are numbered, using integers (1, 2, …); the numbering of calibration strata is stored in the variable STRATUM, which must be present in both the survey data file and the calibration totals file. See section IV.B.3 for details. The parameter has free format.

@STR_N	The *number* of the last calibration stratum to be processed in the current run

This numeric parameter gives the number of the last calibration stratum to which the calibration model, defined through other parameters, will be applied. Notice that the calibration strata are numbered, using integers (1, 2, …); the numbering of calibration strata is stored in the variable STRATUM, which must be present in both the survey data file and the calibration totals file. See section IV.B.3 for details. The parameter has free format.

If the parameters @STR_1 and @STR_N contain the same stratum number, then only that stratum is processed. If the value set through @STR_N is less than the value set through @STR_1, then only the stratum with number @STR_1 is processed.

@TYPE	The calibration *method*, i.e. the distance function *G*

This numerical parameter sets the calibration method, i.e. the distance function *G* or the calibration function *F*. Four methods are implemented in our software; the corresponding values for @TYPE are:

1 if the *linear* method is applied;
2 if the *multiplicative* or *exponential* method is applied;
3 if the *truncated linear* method is applied;
4 if the *logit* method is applied.

If the user-specified value is not 1, 2, 3 or 4 then the value is reset to 1, the linear method is applied, and a warning message is included in the informative output. The parameter has free format.

@SCALE	A fixed value for the *scale* parameter ϕ, if positive, or 0 or a negative value to ask the program to calculate ϕ

If a strictly positive value is set for this numerical parameter, then the scale parameter ϕ takes this fixed value for each calibration stratum. If zero or a negative value is given, then this value is ignored, and ϕ is calculated for each calibration stratum separately when the data for the stratum are processed. Notice that setting @SCALE equal to 1 is equivalent to no a priori supplementary global adjustment to the initial weights. The resulting value of the scale parameter, if calculated by the program, is linked to the first calibration variable in the variable list defined through the parameter @XVARS; see section III.A.2 for details. The parameter has free format.

@L	A fixed *lower bound L* for the g-weights

This numerical parameter is the value of the user-specified lower bound *L* for the g-weights. The value is ignored in the linear and in the exponential method. Notice that, for some values of *L*, the calibration problem might be infeasible. The user should take care when s/he specifies @L; see section II.C for details. The parameter has free format.

@U	A fixed *upper bound U* for the g-weights

This numerical parameter is the value of the user-specified upper bound *U* for the g-weights. The value is ignored in the linear and in the exponential method. Notice that, for some values of *U*, the calibration problem might be infeasible. The user should take care when s/he specifies @U; see section II.C for details. The parameter has free format.

@TOL	The *tolerance* ε in the convergence criterion

This numerical parameter defines the tolerance level ε in the convergence criterion. Notice that the convergence criterion implemented in the current version of g-CALIB-S is based on the maximum change in the g-weights in successive iterations; see sections II.D and II.E.5. Iteration stops if the tolerance level is not exceeded anymore, or if the maximum number of iterations (set through @ITERMAX, discussed hereafter) is reached, whichever occurs first.

The tolerance level has no effect for the linear method. The parameter has free format.

If the method chosen is not the linear one, but one of the three iterative methods, and if the user erroneously has given a zero or negative value for @TOL, then the program automatically resets @TOL to the default value 10^{-5} and a warning message is included in the informative output.

@ITERMAX	The *maximum number of iterations*

This numerical parameter sets the maximum number of iterations. Iteration stops when the maximum number of iterations is exceeded, or if the tolerance level (set through @TOL, as discussed above) is not exceeded anymore, whichever occurs first. The maximum number of iterations has no effect for the linear method. The parameter has free format.

If the method chosen is not the linear one, but one of the three iterative methods, and if the user erroneously has set @ITERMAX to zero or a negative value, then the program automatically resets the value to 100 and a warning message is included in the informative output.

@INFO	A parameter to specify whether more *detailed intermediate results* have to be included in the informative output

This string parameter specifies whether or not informative output, which always includes some basic information about the iteration process and summary statistics at the end of the iterations, is extended with more detailed results of intermediate calculations. Such supplementary output is produced if the value of @INFO is set to Y (or y); otherwise only the standard informative output is provided. The user must be aware that the output can then be extremely long, since some output is at the level of the observations. It is recommended to use this feature only when a previous run of the module was not completed successfully, and when the user tries to locate where the program has failed.

This parameter @INFO is case insensitive.

IV.B.2.iii The SPSS output data file WEIGHTS.sav

A SPSS data file is created at the end of execution of the syntax program g-CALIB-S.sps. The name of this file is invariably WEIGHTS.sav, and it is stored in the working directory specified through the parameter @WORKDIR, which also contains the survey data file (defined through @XDATA) and the calibration totals file (defined through @CALTOT). The variables stored in the file WEIGHTS.sav are:

- CASE : an identification of the cases, which is a copy of the variable CASE in the survey data input file (see section IV.B.3).
- STRATUM : a numbering of the calibration strata; values should be 1, 2, …, which also is a copy of the variable STRATUM in the survey data input file (see section IV.B.3).
- SCALE : a variable containing the value(s) of the scale parameter ϕ. This value is constant within calibration strata, but may vary across calibration strata. A global positive value is set by the user (through the parameter @SCALE), or a calibration stratum-specific value is calculated by the program (see section III.A.2 and @SCALE in section IV.B.2.ii).
- SCAWEI : the values of the so-called *scaled weights* (see section III.A.2). These are the initial weights d_k multiplied with the scale parameter for the calibration stratum to which case k belongs. The scaled weights are the new initial weights in the calibration problem.

- CALWEI : the values of the final calibrated weights w_k (if the iteration process has converged).
- G_WEIG : the values of the g-weights g_k. This is the ratio between the calibrated weights (in CALWEI) and the scaled weights (i.e. the new initial weights, stored in SCAWEI).

The g-weights with respect to the initial weights d_k can be found by multiplying the variable G_WEIG with the variable SCALE. Notice that the initial weights d_k are not stored in the output data file WEIGHTS.sav. However, since the case identification variable CASE is present (with the same name) in the survey data file too, the two files can be merged, so that input and output values for each case are matched. Both the survey data input file @XDATA and the output file WEIGHTS.sav are sorted by STRATUM and CASE at the end of running g-CALIB-S, in order to facilitate merging of these files. Then, an alternative way to find the g-weights w.r.t. the initial weights d_k is to divide the variable CALWEI (which comes from the output file) by the variable WEIGHT (which comes from the input file). Recall that the calibrated weights are not affected by the value of the scale parameter (see section III.A.2).

The user has to merge the files him/herself. Ones this is done, s/he can explore the results more in-depth by using the appropriate SPSS syntax or menu commands, or a specific syntax program may be constructed to present the results in an appropriate format. For an illustrative example, see section V.F.4. The module g-CALIB-S however produces some summary statistics automatically. This is discussed in the next sub-section.

IV.B.2.iv Informative output

By *informative output* we mean some crucial non-statistical information about each step in the iterative procedure, more statistical information on the calibration variables at the end of the procedure (ideally when convergence is attained), and some tables and box plots presenting summary statistics on the scaled, the calibrated and the g-weights. These summary measures are calculated for each calibration stratum separately. Informative output is stored in a SPSS Viewer file (with extension .spo). The name of this file is the same as that for the SPSS Production Facility job-file, with extension .spp; see section IV.B.5.

Non-statistical information as well as statistical information may help to evaluate the validity of the results, and to detect possible problems caused by badly specified data (in the two input files). Useful features are, for instance:

- A measure for the relative change in the g-weights in successive iterations. It is this relative change that has to be smaller than the tolerance level ε, for "numerical" convergence.
- The number of negative current estimates of the g-weights (or calibrated weights) in each of the iterations.
- The various substitutes for the calibration totals for each calibration variable, including the calibration totals themselves (column labelled "Fixed") and the calibrated estimates of these totals (column labelled "CAL_est"), which should, at convergence, be the same. Totals labelled "INI_est" are calculated using the initial weights in the survey data input file; totals labelled "SCA_est" are calculated using the scaled (initial) weights.
- A relative difference between the fixed calibration total and the final calibrated total is calculated for each calibration variable, and presented in the column labelled "% DIFF". At convergence this relative difference must be near zero for each calibration variable. If these values are non-zero, then the iterative procedure might not have been converged (yet), or there may be problems with the data. Among these possible problems we might have numerical inconsistency for the calibration totals, or a calibration variable with zero values only (within a specific calibration

stratum) in the survey data file, while a non-zero calibration total is stored in the calibration totals file. The present version of g-CALIB-S does not detect such problems at the start of running the program; care from the user who constructs the input files can however avoid easily this kind of problems. Nevertheless, we intend to include more problem checking in new releases of g-CALIB-S.

At the end of the iterative procedure, statistical information on the scaled weights, the calibrated weights and the g-weights includes, for each calibration stratum separately:

- The minimum, mean, median and maximum, the standard deviation and the sum, and the 5^{th}, 25^{th}, 75^{th} and 95^{th} percentiles. The sum of calibrated weights is of particular interest: it is the estimated number of cases in the population in a given calibration stratum. It is equal to a fixed calibration total if a constant calibration variable (with values all equal to 1) is included in the survey data input file.
- Box plots showing the distribution of the scaled weights, the calibrated weights and the g-weights in each calibration stratum.

Finally, SPSS users know that through the menu Edit–Options in SPSS windows one can specify which other output can be included in the Viewer output file. For instance, it may be useful to have all commands in that file (the "log"), or warning messages, etc. Such additional output may be helpful to find out where things went wrong in case of program failure. For more details, the user is referred to the SPSS User's Guide.

In chapter V, output files in the context of several case studies will be discussed. Some complete output files will also be reproduced for illustrative purposes in the Appendices.

IV.B.3 Input files for g-CALIB-S

g-CALIB-S needs two input files, referred to by their equivalent parameter names @XVARS and @CALTOT hereafter (see section IV.B.2.ii). We here discuss the contents and structure of these files; in section IV.B.6 we discuss their construction from basic data files, using some macros developed especially for that purpose.

Given the calibration problem (IV.2), it will be obvious that the design matrix **X** and the weight vector **d**, i.e. the survey data, are stored in the survey data file @XDATA, and that the calibration totals **t** are to be stored in the calibration totals file @CALTOT. Some rules should be strictly followed, as explained hereafter.

The survey data file @XDATA must include three variables with fixed names. These are:

- CASE : a unique identification of the cases. As explained in section IV.B.2.iii, this variable will be copied to the output file WEIGHTS.sav.
- STRATUM : a numbering of the calibration strata; values should be 1, 2, …. This variable too is copied to the output file WEIGHTS.sav (section IV.B.2.iii).
- WEIGHT : the variable containing the initial weight vector **d**.

Apart from these three variables, variables with user-specified names will hold the columns of the design matrix **X**. No specific restrictions apply to these variables, but some recommendations could be taken into account. Variables (or columns in **X**) corresponding to the same term in a calibration model formula, as explained in section III.A.1, should be kept together in @XDATA. This is useful since these variables always need to be used together, in any possible calibration model. It will then reduce the length of the list of variables @XVARS. For the same reason, it is also convenient to keep all

calibration variables together, i.e. not to mix up these variables with the variables CASE, STRATUM and WEIGHT, or with any other set of variables that is stored in @XDATA. The latter variables can be original variables, such as qualitative variables, which are to be transformed into a set of indicator variables (with values 0 and 1) before they can be used as calibration variables. Other (original) variables that could be stored are variables holding an alternative series of initial weights, or an alternative calibration stratum specification.

There is one important restriction on the variables' values: STRATUM, WEIGHT, and the calibration variables should have real values only. Otherwise, the program will fail. Moreover, there should be no missing values in these variables. The occurrence of missing values in any of these variables will cause failure of g-CALIB-S. The user should eliminate cases with missing values in these variables before running the program. (I.e. item non-response must be treated as unit non-response.)

To illustrate this, consider a calibration problem with model formula $A*B = 1 + A + B + A.B$, where A and B are qualitative variables with 2 and 3 categories respectively. Suppose that the user has constructed variables X0 (identically 1), A1 and A2, B1, B2 and B3, and AB11, AB12, AB13, AB21, AB22 and AB23, then the structure of the file @XDATA will be, for instance:

CASE STRATUM WEIGHT X0 A1 A2 B1 B2 B3 AB11 AB12 AB13 AB21 AB22 AB23

possibly followed (or preceded, or …) by other variables (from the basic data file, and prepared for possible use later on). The following variable lists @XVARS, for instance, are then possible:

- X0, A1, A2, B1 to B3, AB11 to AB23 — for the model $1 + A + B + A.B = A*B$
- X0, A1, A2 — for the model $1 + A = A$
- B1 to B3 — for the model $1 + B = B$
- AB11 to AB23 — again for the model $A*B$
- etc

It can be noticed that the names of the calibration variables in the file @XDATA are arbitrary, provided the usual SPSS rules are satisfied. Names should be chosen with care, such that model specification and interpretation of results is not too much complicated. (Once a data file is constructed the user can complete the data dictionary of the input files.)

The second input file, @CALTOT, must have a similar structure as @XDATA. Of course, variables CASE and WEIGHT do not appear in this file. STRATUM and all user-specified variables will be present, and the calibration variables should appear in the same order as they appear in @XDATA. No other variables need to be included, although the user can add some. Hence the structure corresponding to the model formula $A*B$ could (for instance) be:

STRATUM X0 A1 A2 B1 B2 B3 AB11 AB12 AB13 AB21 AB22 AB23

Each row in @CALTOT corresponds to a particular calibration stratum, and contains the calibration totals for all the variables for that stratum. Each row thus contains a stratum-specific vector $\mathbf{t}$. In the same way, all rows in @XDATA that correspond to cases in the same stratum, do contain a stratum-specific design matrix $\mathbf{X}$. Notice that the numbers of rows in each of these matrices are generally different, but corresponding columns are representing the same calibration variable.

All this explains how data for different strata can be separated from each other, and therefore also be treated separately.

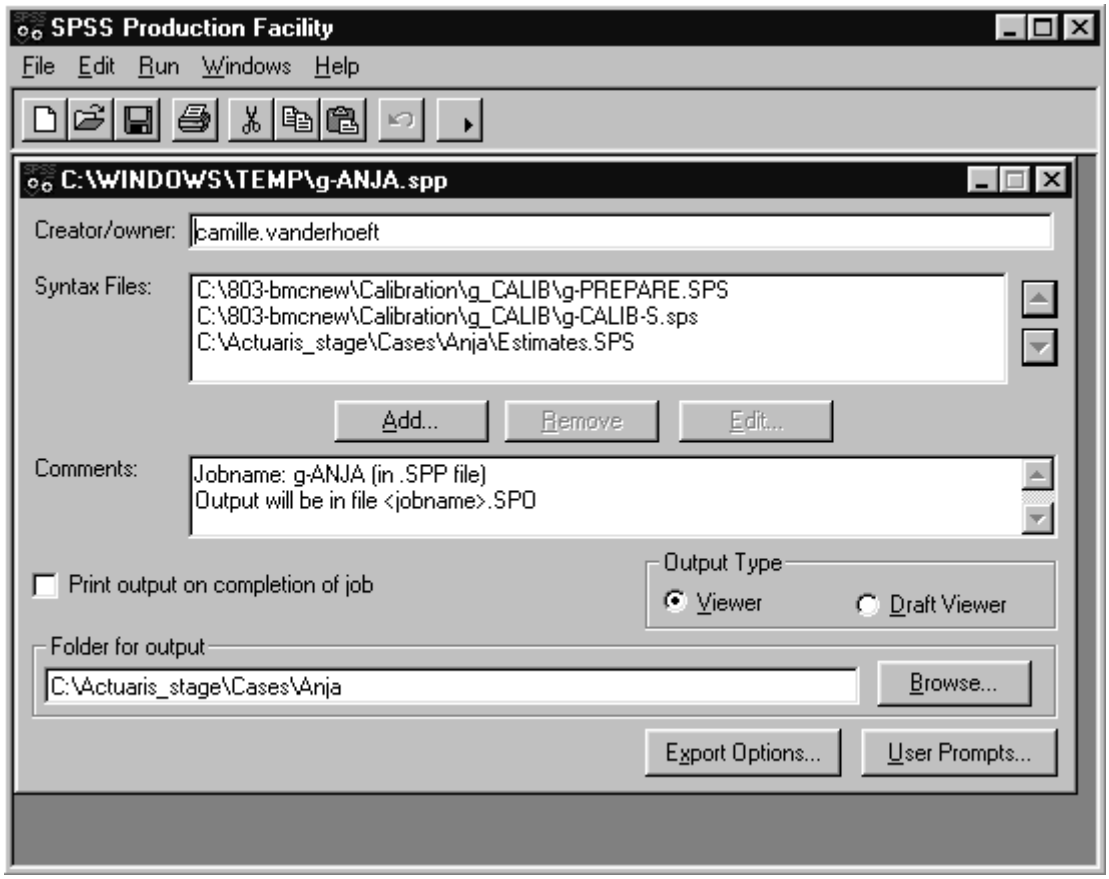
IV.B.4 The auxiliary module g-PREPARE.sps

A small auxiliary syntax module g-PREPARE.sps has to be run before the core module g-CALIB-S.sps is executed. Notice that together, g-PREPARE.sps and g-CALIB-S.sps are constituting the calibration software g-CALIB-S.

The module g-PREPARE.sps merely performs some final preparation on the two input files @XDATA and @CALTOT. The main functionality of g-PREPARE is to sort the files by STRATUM. It further counts the number of cases in each stratum (in @XDATA) and computes the record numbers of the first and last case in each stratum in the sorted @XDATA. This information is temporarily stored in @CALTOT. The module g-CALIB-S.sps itself will remove this information after estimating the g-weights for each stratum. If the program fails, it usually doesn't reach the point where that information is removed. Then the user might have to remove it manually from @CALTOT. However, the presence of that information in @CALTOT at the start of a session should not cause problems, as long as the number of cases per stratum in @XDATA doesn't change.

IV.B.5 The interface: a SPSS Production Facility job

We have used SPSS Production Facility, delivered together with SPSS Base, to create an interface to the calibration software g-CALIB-S. The start-up screen of such a job is shown here below. The name of the job-file (with extension .spp) is free; here it is called g_ANJA.spp.

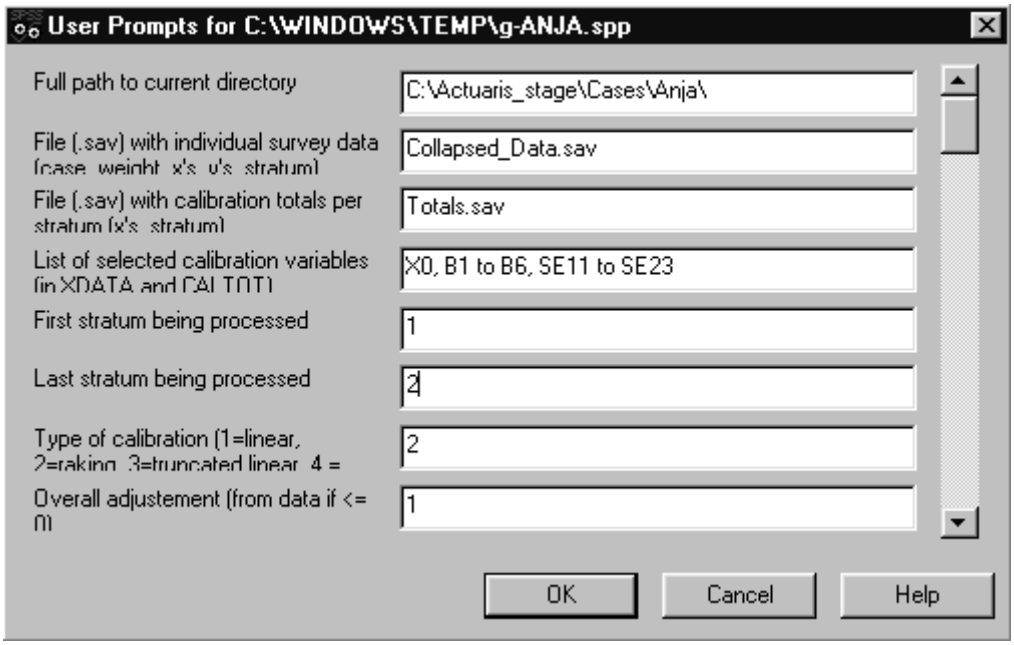


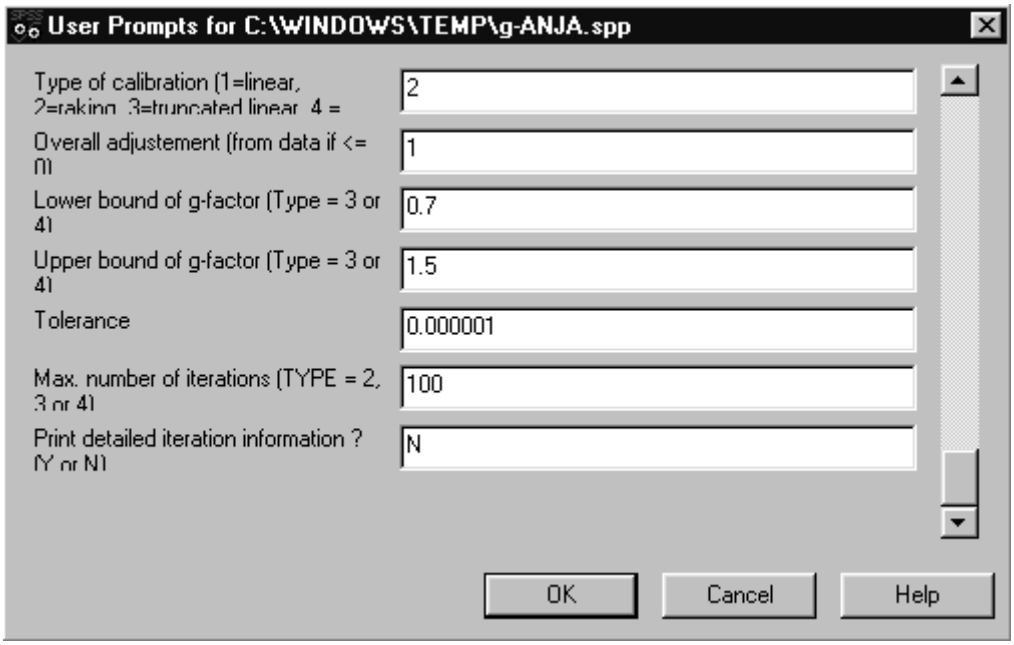
The Syntax Files box shows that two syntax files are called: g-PREPARE.sps and g-CALIB-S.sps (the order is important!). Other syntax files (constructed by the user) may be added either before, to prepare the data, or after, to manipulate the results. The Comments box is useful to put information about the job, e.g. generalities about the calibration model or the data being used. In the Folder for Output box, the user can specify where the program has to store the Viewer output file (.spo), and occasionally other output files that are specified through the Export Options… button (see later). Notice that the folder specified in the Folder for Output box is not the folder where g-CALIB-S will store the output file WEIGHTS.sav.

Recall that g-CALIB-S (i.e. g-PREPARE.sps and g-CALIB-S.sps) is fully parameterised (section IV.B.2.i). SPSS Production Facility translates each parameter, @PAR say, into a text macro named @PAR. These macros are stored in the temporary file SPSSProd.spp, which is stored in SPSS's general folder for temporary files (e.g. C:\windows\temp), and which is automatically "included" before g-PREPARE.sps and g-CALIB-S.sps are "included". Notice that, if the user intends to insert his own syntax files through the Syntax Files box, these syntax files then have to be constructed according to the general SPSS rules for include files (see SPSS Base User's Guide). An advantage of including user-constructed files in the same job, is that the same parameters (@WORKDIR, @XDATA, …) then can be used in the user-constructed syntax files. This allows flexible extension of the basic calibration software with other modules, for instance for estimation of totals of study variables.

Clicking on the User Prompts… button opens the User Prompts window (not shown here). The user should open this window only if standard or default settings for the program parameters (starting with @) have to be changed. So we recommend that the user of g-CALIB-S only changes the entries of the column Default, if necessary. The other columns should only be used by the developer of the interface, and the software behind.

After modifying the contents of the start-up screen, the user can save these changes (and rename the job file) if s/he wishes. Next the Run button (▶) can be clicked to start running the job. Then the user is presented the User Prompts for <jobname.spp> window, where s/he can finally set the appropriate values for the parameters, as explained in section IV.B.2.ii. An example of this window is shown below.





(Notice the rather bad quality of the prompts: the second line is not shown completely, which is due to SPSS, and not to the developers of g-CALIB-S; one-line prompts would be better, and the user actually can change the prompts, once s/he understands how the software works.) When all parameter values are set, the OK button is clicked and SPSS then starts running the job.

By default, a job is running in background. However, through the Edit-Options menu in SPSS Production Facility, the user can ask SPSS being shown up while running the job. For this and more functionalities of Production jobs, the user is again referred to the SPSS User's Guide.

One more feature is worth mentioning here. Clicking the Export Options… button in the start-up screen allows to specify that output is also stored in, for instance, a HTML file, which will be saved in the folder specified through the Folder for Output box in the start-up screen. We recommend using this feature, since the HTML file is numbering the syntax commands, which makes it easier to detect where exactly the program has gone wrong, in case of failure. Notice that error messages usually include a command line number.

IV.B.6 The auxiliary module g-DESIGN.sps

In the syntax file g-DESIGN.sps we simply have stored some macros that may help the user to construct the design matrix in a given calibration problem. These macros are:

- DesC1 to create an indicator variable matrix for 1 qualitative calibration variable *A*, i.e. that part of a design matrix **X** that corresponds to the term *A* in a calibration model formula (section III.A.1);
- DesC2 to create an indicator variable matrix for 2 qualitative calibration variables *A* and *B*, i.e. that part of a design matrix **X** that corresponds to the term *A.B* in a calibration model formula;
- DesC3 to create an indicator variable matrix for 3 qualitative calibration variables *A*, *B* and *C*, i.e. that part of a design matrix **X** that corresponds to the term *A.B.C* in a calibration model formula;

- DesC1Z — to create an interaction matrix for 1 qualitative calibration variable *A* and 1 quantitative calibration variable *Z*, i.e. that part of a design matrix **X** that corresponds to the term *A.Z* in a calibration model formula;
- DesC2Z — to create an interaction matrix for 2 qualitative calibration variables *A* and *B*, and 1 quantitative calibration variable *Z*, i.e. that part of a design matrix **X** that corresponds to the term *A.B.Z* in a calibration model formula;
- DesC3Z — to create an interaction matrix for 3 qualitative calibration variables *A*, *B* and *C*, and 1 quantitative calibration variable *Z*, i.e. that part of a design matrix **X** that corresponds to the term *A. B. C.Z* in a calibration model formula.

Hence, up to third-order interaction effects can be included in the calibration models, if the design matrix construction is only based on our 6 macros. It would not be difficult to write new macros for higher-order interaction effects, but before doing this, I intend to rewrite the macros, in order to avoid duplication of calculations. At present, the user should construct higher-order terms her/himself. Notice that the macros in g-DESIGN.sps are not used by the software g-CALIB (i.e. g-PREPARE.sps and g-CALIB-S.sps). Notice that the macro names do not start with a special character (such as @ for the macros discussed in section IV.B.2.ii). Therefore, the user should not use the macro names for other identifiers (variables), when using the macros.

All 6 macros are functioning with input and output arguments. The syntax is as follows:

DesC1	var=*varname* des=*matname* lab=*vecname*
DesC2	var1=*varname1* var2=*varname2* des= *matname* lab=*matname2* p=*num*
DesC3	var1=*varname1* var2=*varname2* var3=*varname3* des= *matname* lab= *vecname* p=*num*
DesC1Z	var=*varname* zet=*quantnam* des= matname lab= *vecname*
DesC2Z	var1=*varname1* var2=*varname2* zet=*quantnam* des= *matname* lab= *vecname* p=*num*
DesC3Z	var1=*varname1* var2=*varname2* var3=*varname3* zet=*quantnam* des= *matname* lab= *vecname* p=*num*

The input arguments are var, var1, var2, var3, zet and p; output parameters are des and lab. *Varname*, *varname1*, *varname2* and *varname3* are user-specified names of qualitative calibration variables (*A*, *B*, …), *quantnam* is a user-specified name of a quantitative calibration variable (*Z*), *matname* is a user-specified name of a matrix that will contain, at exit of the macro, the constructed part of the design matrix; *vecname* is a user-specified name of a row vector that will contain, at exit of the macro, a set of constructed numeric labels for the columns in the matrix *matname*. *Num* should be set such that the computed labels are meaningful; i.e. *num* should be chosen such that $10^{num} - 1$ is at least the largest value assumed by the qualitative variables defined through arguments var or var1 (and var2 (and var3)). I.e. if the values of the qualitative variable(s) vary between 0 and 9, then *num* should be at least 1, if the values of the qualitative variable(s) vary between 0 and 99, then *num* should be at least 2, etc.

Input arguments for the macros should contain only numerical variables, and no missing values should be present. (I work on the software to treat incomplete data, i.e. to deal with missing data.) I recommend to use only integers 1, 2, … for values of qualitative variables, although this is not strictly required for adequate functioning of the software.

To illustrate, consider the following small hypothetical example. Let the original data on 8 cases, in 5 variables or columns, be as follows:

```
      ID WEIGHT STRATUM     A      B      Z
    1.00  10.00    1.00  1.00   1.00  15.00
    2.00  10.00    1.00  1.00   1.00  17.00
    3.00  12.00    1.00  1.00   2.00  12.00
    4.00  10.00    1.00  1.00   2.00  12.80
    5.00   5.00    1.00  2.00   1.00  10.50
    6.00   5.00    1.00  2.00   1.00  12.00
    7.00   5.00    1.00  2.00   2.00  13.40
    8.00   5.00    1.00  2.00   2.00  12.00
```

The column headings are the variable names, stored in the data dictionary of the survey data file (an SPSS data file, say). We assume that *A* and *B* are qualitative variables, while *Z* is a quantitative variable. STRATUM is qualitative too, with, in general, *H* categories. It is important for this variable that all categories 1, 2, …, *H* are represented in the survey data file. In our example, *H* = 1.

Standard SPSS matrix commands (i.e. the GET command) are used to read the case identifiers, the case weights, STRATUM, and the survey variables *A*, *B* and *Z* into vectors ID, WE, STRATUM, A, B and Z, say. (Names are not really important, so far.) Now, suppose that the maximal calibration model is specified through the model formula $1 + A + B + Z + A.B + A.Z + B.Z$. Then g-DESIGN provides the required macros DesC1, DesC2 and DesC1Z to transform the vectors A, B and Z into the corresponding columns of the (maximal) calibration design matrix **X**. The constant term is easy to construct, using matrix functions nrows and make. The matrix commands are:

```
compute X0 = make(nrows(ID),1,1)           /*  Term: 1     */.
DesC1 var=A des=XA lab=LabA                /*  Term: A     */.
DesC1 var=B des=XB lab=LabB                /*  Term: B     */.
DesC2 var1=A var2=B des=XAB lab=LabAB p=1  /*  Term: A.B   */.
DesC1Z var=A zet=Z des=XAZ lab=LabAZ       /*  Term: A.Z   */.
DesC1Z var=B zet=Z des=XBZ lab=LabBZ       /*  Term: B.Z   */.
```

X0, XA, XB, XAB, XAZ, XBZ and also Z are matrices, with, in our example, numbers of columns equal to 1, 2, 2, 4, 2, 2 and 1, respectively; those matrices are the corresponding parts of the design matrix **X**. The survey data input file @XDATA is then easily created by the following SPSS matrix command, which saves the original data and the (maximal) calibration design matrix into that file:

```
save {ID, WE, STRATUM, A, B, Z, X0, XA, XB, XAB, XAZ, XBZ}
  /outfile = 'C:\my documents\calibration\' + @XDATA /format = F3
  /variables = CASE,WEIGHT,STRATUM,A,B,Z,X0,A1,A2,B1,B2,
               AB11,AB12,AB21,AB22,A1Z,A2Z,B1Z,B2Z.
```

The file @XDATA now contains the following data:

```
CASE WEIGHT STRATUM A B  Z X0 A1 A2 B1 B2 AB11 AB12 AB21 AB22 A1Z A2Z B1Z B2Z
   1     10       1 1 1 15  1  1  0  1  0    1    0    0    0  15   0  15   0
   2     10       1 1 1 17  1  1  0  1  0    1    0    0    0  17   0  17   0
   3     12       1 1 2 12  1  1  0  0  1    0    1    0    0  12   0   0  12
   4     10       1 1 2 13  1  1  0  0  1    0    1    0    0  13   0   0  13
   5      5       1 2 1 11  1  0  1  1  0    0    0    1    0   0  11  11   0
   6      5       1 2 1 12  1  0  1  1  0    0    0    1    0   0  12  12   0
   7      5       1 2 2 13  1  0  1  0  1    0    0    0    1   0  13   0  13
   8      5       1 2 2 12  1  0  1  0  1    0    0    0    1   0  12   0  12
```

The top row is just the list of variable names – as defined by the user – stored in the data dictionary of the file. The last 14 columns, from Z to B2Z, constitute the (maximal) design matrix **X**. Notice that *z*-values are displayed as rounded to integer values; the exact values are stored in the file. It is clear that there are several linear dependencies between the columns: the matrix **X** has 14 columns, but its rank is only 7. Constructing the file @XDATA is this way has an important practical consequence: any

sub-model, including the model 1, or the model *Z*, of the maximal model 1 + *A* + *B* + *Z* + *A.B* + *A.Z* + *B.Z* can then easily be applied, just by selecting the appropriate variables from @XDATA. The list of variables to be selected is easily specified through the parameter @XVARS, as explained in section IV.B.2.ii.

Before constructing the above SAVE command, the user can print the contents of the vectors that contain the labels for the columns in the matrices constructed by the macros. An appropriate PRINT command might simply look like:

```
print {LabA, LabB, LabAB, LabAZ, LabBZ}.
```

or, including some information about what is printed:

```
print {LabA} /title "A categories are met in the following order:".
```

Inspecting the output from such commands helps the user to define the list of variable names in the variable's sub-command in the above SAVE command.

Finally, notice that all commands presented in this section are matrix commands. They must therefore be executed only from within a matrix program. A *matrix program* is simply a set of matrix commands in a syntax file, delimited by the MATRIX and END MATRIX commands. The user is referred to, for instance, the SPSS 7.5 *Advanced Model Guide*. Notice that more recent versions of SPSS now include matrix language in the SPSS Base module. This implies that the SPSS Advanced Models module is not needed anymore to run our calibration software (including g-CALIB-S!).

The file g-DESIGN.sps is reproduced in appendix VII.A.2. The core modules g-PREPARE.sps and g-CALIB-S.sps are not reproduced in this report. Interested potential users can contact us: see section VII.A.1 for practical information.

IV.C COMMENTS, AND FUTURE DEVELOPMENTS OF G-CALIB-S

IV.C.1 General comments

IV.C.1.i SPSS as a development environment

SPSS is considered primarily as an application tool for statistical purposes, and that's exactly what it is. But I think it can be more, and in my humble opinion, I think that this work has demonstrated that SPSS is not so bad for development purposes. Generalised calibration is not a trivial statistical problem – or should we say mathematical problem, from the applied discipline of operations research? – and we have deliberately realised a useful tool. At the moment that we started implementing the methodology, we had no idea about all the features with respect to programming (accompanied with all the pitfalls!) offered by SPSS through its basic syntax language, matrix language and macro facilities. SPSS basic syntax language is mainly designed for data manipulation; the main object handled by basic syntax commands is the variable, not the case. This makes basic syntax commands sometimes difficult to understand, especially for the novice, and consequently also difficult for programming. Fortunately, there is the more traditional matrix language (including macro facilities), which is capable of filling in the gaps in the basic syntax language. It is that combination of basic syntax language and matrix language that makes SPSS suitable for development purposes.

True, people are right when they say that SPSS's programming language is old-fashioned, but that's completely out of the question here. Our problem is to have access to a system that allows applying the most modern techniques in survey processing. This problem could be solved in different ways: we could buy another statistical package, so that we could then buy specialised routines that have been developed at other statistical institutes or universities; or we could ask our informatics department to develop a stand-alone package (for calibration) in whatever computer language they want; or we could return to the system of pen and paper (extended with scissors and glue to cut and paste): "keep it simple". Finally, we have opted – or rather: we were forced to choose – for a do-it-yourself solution. The result of this is now available for usage, evaluation, improvement, or depreciation, as you like.

Once more, I want to stress here the fact that we not only constructed just another tool (toy?) statisticians can work with, but that we have built up, in doing all this programming work, some know-how in calibration. And that's finally what we really have to be concerned about.

In the course of developing g-CALIB-S, a process that is not necessarily finished yet, we learned a lot about SPSS syntax language. Looking up things in SPSS reference guides, we detected new features. Sometimes, we were able to exploit these immediately, but more often, we had to postpone their use, since it might take some time to work out new ideas, and since this would have delayed issuing a "final product". One of these features is *scripting*. This might help us to construct a user-friendlier, and more automated, system. We'll see!

The Department of Methodology and Co-ordination at Statistics Belgium will have to decide soon how it will proceed further with respect to implementation of calibration and other survey methodology. It has by now purchased the SAS system. So, advantages and disadvantages of both SPSS and SAS will have to be examined. It is likely that both systems will continue being used at Statistics Belgium, at least for the next few years. This, in my opinion, justifies at least some minor modifications and improvements in g-CALIB-S, provided perhaps that this work will not cost to much time and human resources. Suggestions for such improvements and extension of our software are presented briefly in section IV.C.2.

IV.C.1.ii Performance of the SPSS modules

Persistence of g_CALIB-S will depend largely on its performance. As the software will from now on be used more and more to support calculation of weighting factors, larger data sets will have to be treated soon. This is the ultimate test! So far, we have experienced good results with g-CALIB-S, but sometimes, the processing time is too large for being acceptable in practice. I cannot give much detail yet on evaluation of performance. This is a study of another nature, and I didn't have enough time to work it out for this report. Moreover, as we are running SPSS on PCs, hardware specifications will have to be taken into account. Also, performance of g-CALIB-S should be compared with performance of similar packages, such as CALMAR, which is running under SAS.

It must also be said that performance of SPSS Base will play an important role in this evaluation. We have developed g-CALIB-S under SPSS's version 9.0, although version 10.0 is now available too. The reason is that our modules were not always running as they should, under SPSS 10.0. When trying to understand why failure occurred, we often came to the conclusion that bugs in SPSS itself were responsible. For some problems we could find a "SPSS 10.0 solution", but implementation of such solutions would imply abandoning of what we thought being logical programming.

Finally, we like to mention that our modules are running without problem under SPSS 8.0 (Base and Advanced Statistics modules).

IV.C.2 Future developments

IV.C.2.i Error detection and reporting

More work could be done on error detection and reporting. We have already built in some tricks to capture and solve mistakes in input parameters (@…); this is the easy part.

More difficult, but more important, to detect and solve are problems related to the data. At least two problems deserve special attention: consistency and missing values. In principle, these problems should be easy to solve, exploiting features of matrix language that have not been used so far.

The most difficult problems are related to badly conditioned data. We have already tried to provide measures that indicate why the program fails, or why convergence is slow or not attained at all. We refer to section IV.B.2.iv for details; also refer to the discussion of the program parameter @INFO in section IV.B.2.ii. Apart from inspection of these measures in the Viewer output file, we have to rely on SPSS error messaging. Here, the HTML output file (section IV.B.5) can help to locate the place were the program goes wrong.

After all, the user should take care when constructing the input data files. The subject matter statistician will play an important role in that phase of survey processing. Armed with a good understanding of the specificities of a given survey, and with detailed documentation of the data files and databases, s/he will generally be able to point out quickly how to circumvent data-related problems.

IV.C.2.ii Implementation of determination of the maximum lower bound L^* and minimum upper bound U^*

In section II.C we have proposed a method to detect extreme values for the upper and lower bound on the g-weights in the truncated linear and in the logit method. It was shown that the problem can be found as the solution of a linear programming problem, which in turn can be solved with the simplex

algorithm. Matrix language in SPSS provides the function SWEEP(*matrix*, *k*), which produces a new matrix by pivoting *matrix* on the element in the *k*-th row and *k*-th column. Pivoting is the main operation in the simplex algorithm, whence I believe that implementation of our technique in SPSS is feasible.

The SAS system would simplify the task a lot. Surprisingly (or not?), SAS/IML, the matrix module in SAS, provides a procedure, called *LP*, to solve a linear programming problem of the form (II.6). So, SAS users would not have to implement the simplex algorithm. CALMAR developers should consider it!

IV.C.2.iii Improving the interface: using Scripting

In the SPSS *Base* 9.0 *User's Guide* (p. 675) it is stipulated that scripting allows us to " (1°) *automatically customize output in the Viewer*, (2°) *open and save data files*, (3°) *display and manipulate dialog boxes*, (4°) *run data transformations and statistical procedures using command syntax*, and (5°) *export charts as graphics files in a number of formats* ". The relevance of all these features is not obvious, yet, but some of them might be interesting for further evaluation. It seems that scripting can be used to construct at least a better interface to the core modules. Given this, it might also be possible to add more flexibility and functionality to our software.

I believe that scripting is worth being considered, although I have currently no experience at all. One advantage of script language is that it is based on the Sax Basic language (*op. cit*. p. 675), an object-oriented programming language (and therefore more attractive to contemporary programmers). Combination of script language with syntax language (including matrix language) probably provides a powerful combination for programming in an SPSS environment.

IV.C.2.iv More automated construction of @XDATA and @CALTOT, including management and exploitation of complex files

It would be nice if the input files @XDATA and @CALTOT could be constructed more automatically, i.e. if we could develop modules, probably survey-specific, to transform basic databases into appropriately structured input files for g-CALIB-S. This would require perfect understanding of how basic databases are constructed, on the one hand, and which SPSS features are available for reading these databases, on the other hand. Definitely, more co-operation between methodologists in the information department and methodologists in the statistical department is desirable. Possibly other development tools can be integrated at this point. I believe that the present calibration study is providing indications concerning the construction of databases and files. Each case study definitely provides indications on how to set up an efficient data management system (for a given survey). The result should be a series of transformations of data files, with the restriction that the number and extent of transformation must be kept to a strict minimum, such that in different phases of survey processing appropriate and general tools can be used efficiently. More research has to be done, since we are now only at the beginning of implementation of calibration methodology in daily statistical practice. The literature on calibration methodology is already a step ahead of using basic calibration techniques in a less complex situation, as discussed in this text. The reader is referred to Renssen (1998) for calibration in more complex situations; see also the references in that paper. The ultimate aim is to integrate several surveys in the same system, and to produce thereof estimates that are consistent across the various surveys.

SPSS allows reading and constructing files with a more complex structure than the common rectangular structure. We have recently created such a file, based on tax registers. The final file has records of two types: household records, and records corresponding to the person(s) (usually a single person, or husband and wife) within the same fiscal household. We will try to link this file with a register of administrative households, which is deduced from the register of Belgian citizens. Our goal

is to use indicative information on income, either in the design phase or in the estimation (calibration) phase of household based surveys. SPSS can handle *rectangular files*, *nested files*, *grouped data files* and *mixed files*. The above mentioned data file, with information from the tax register, has a nested (or hierarchical) structure. FILE TYPE and RECORD TYPE commands, followed by a DATA LIST command, allow reading the file as a person file, combining household information with person information. Alternatively, the nested file can also be read as a household file (using only the DATA LIST command), provided each household occupies the same number of records for each household. Otherwise, a rectangular household file can be constructed from the previously constructed person file.

IV.C.2.v More efficient treatment of calibration strata

The notion of calibration strata is a very useful one in practice. Here before we mentioned problems when large data sets are analysed. Introducing calibration strata definitely reduces problems related to the size of the survey data file: calculations are done stratum per stratum, having only the data for the stratum considered in the memory. Currently, however, g-CALIB-S reads the complete data set each time processing of a new stratum starts, next the relevant data is copied to new matrices, while the matrices holding the complete data set are released again. This is likely to be time consuming and should be avoided whenever possible.

To circumvent these problems, I intend to use the *split-file processing* facility in combination with matrix language. However, its feasibility has to be examined further, since the data for one stratum should then be read in a single GET statement. The split-file approach seems to offer more flexibility with respect to the definition of calibration strata. On the other hand, it might require another structuring of data files: remember that currently data are read from two input files (@XDATA and @CALTOT). But rewriting the module g-PREPARE.sps could solve this problem: g-PREPARE.sps could merge @XDATA with @CALTOT, in order to construct a single input file for g-CALIB-S.sps.

Alternatively, the *case selection* feature could be examined for usage with matrix language. Concerning both the *case selection* and the *split file* feature, we refer to the SPSS syntax guides (on matrix language) for more details.

IV.C.2.vi Calculation of calibration estimates for study variables

It would not be difficult to extend our software with a module for calculation of estimates of totals of study variables. Notice (1°) that study variables can be included in the survey data file @XDATA, and (2°) that this file can easily be merged with the output file WEIGHTS.sav, holding the final g-weights and calibrated weights. Treatment of missing values can be considered; the required formulae have been discussed in section III.F.

Basic syntax commands may be used for these purposes. However, matrix language too is an option, especially when (preparation for) variance estimation is considered (see the next sub-section), and/or when simultaneous estimation of totals for several study variables is considered.

IV.C.2.vii Preparing for (co-) variance estimation

In section III.H we briefly discussed variance estimation in the context of generalised calibration. It turns out that this can be based on calculation of (linear) regression residuals. This only involves the design matrix, the (initial/sampling or calibrated) weights, and, of course, the study variables. Conditionally on having read these data structures from the relevant file(s), it would then be easy to implement the calculation of residuals: one matrix command would suffice (in principle). Showing yet again the efficiency of matrix language, both in mathematics and in programming languages!

Calculation of residuals is one very important step in variance estimation. Ultimate calculation of variances, however, involves the second-order inclusion probabilities. For some complex designs, this may need the construction, storage and usage of very large square matrices, of order *n*, holding the second-order inclusion probabilities. For less complex designs, however, one might be able to reduce storage of large second-order inclusion probability matrices to storage of smaller vectors or matrices just holding the parameters necessary for calculation of second-order inclusion probabilities. Appropriate software modules then have to be developed to combine all the components in variance estimation formulae.

IV.C.2.viii Extension of the calibration models with the weighting factors q_k

Our software currently deals with the calibration problem $\left\{\min \mathbf{d}^T G(\mathbf{g}); \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}$. The extension $\left\{\min \mathbf{d}^T \mathbf{Q}^{-1} G(\mathbf{g}); \breve{\mathbf{X}}^T \mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}$ will be implemented soon. Notice that this will imply another variable to be read from the survey data input file @DATA.

A variable holding the factors q_k would be similar to the variables STRATUM, WEIGHT and CASE that have to be present in the survey data input file. It is planned to modify g-CALIB-S such that the names of these variables no longer need to be fixed. It will give the user more flexibility since s/he then can choose more easily a calibration stratum variable, a sampling weight variable, … from the input file.

Chapter V

Applications at Statistics Belgium

V.A REGISTERS AND FRAMES

The power of calibration depends to a large extent on the amount and quality of auxiliary information that is available, hence on external sources of different kinds. It is beyond the scope of this study to discuss the external sources used in the applications hereafter. More work definitely has to be done at Statistics Belgium to set up a good system of databases where good calibration benchmarks and "individual" information can be found, or to make existing sources more readily available for that purpose.

For household and individual surveys, the main source is the *National Register of Physical Persons*. A reduced version of this register is available to the methodological department for sampling and estimation. From this register we have derived a household frame. So far we have used only household characteristics such as the number of members, place of residence, and characteristics copied from the *reference person* (RP) of the household, i.e. its age (in 5-year age classes) and professional status. In the future, we will try to describe the structure of the household in some more detail. So it may be interesting to distinguish 1-parent households from traditional 2-parent ones, households without children from households with children, etc. The National Register contains enough information to derive that kind of household characteristics.

We have also prepared now for linking the National Register, or a derived household frame, with a frame of fiscal households. This is an attempt to get indicative information on income at the level of administrative households.

For business surveys, the main source of information is DBRIS, the *Banque de Données des Redevables de l'Information Statistique*. More about this integrated database is told in section V.G, where an application of calibration to the Structural Business Survey is discussed. Linkage of this database with data from social security services is established.

V.B CALIBRATION OF THE LABOUR FORCE SURVEY (LFS)

V.B.1 Introduction

The *Labour Force Survey* (LFS) deserves special attention in this study, since it is used as a reference for calibration of several, if not all, household-based surveys at Statistics Belgium: the *Household Budget Survey* (HBS), the *Time Use Survey* (TUS), the *Travel Survey* (TS). Moreover, external users often demand results from the LFS, hence the need for careful extrapolation of this survey. On the other hand, we'll see in section V.F that LFS estimates on *labour volume* can be adjusted in turn to meet some relative distribution of labour volume by branch of industry from the *national accounts* (NA). This leads us to the more advanced problem of mutual consistency between results from different surveys, which can be solved generally by repeated calibration.

The initial sample for the LFS is quite large (about 110,000 individuals, or 48,000 households), and participation of selected households and individuals is compulsory, such that very reliable estimates can be obtained, provided that care is taken in all phases of the survey process: starting from sampling design, via collection and coding, editing and imputation, to calibration, estimation and evaluation. Obviously in the present study, we mainly focus on calibration (and estimation). However, as it is important to be aware of the sampling design in calibration too, we also reserve some space for a brief discussion of the design of the LFS.

The sampling design is quite complex, as it will be explained in section V.B.2. This implies that it is not straightforward to calculate first and second order inclusion probabilities, for households and individuals, which are required to derive sampling weights and to perform variance estimation in the context of generalised calibration (section III.G). We mention briefly a mathematical model in order to calculate (approximate) first and second order inclusion probabilities.

Currently, calibration of the LFS seems to ignore the sampling design, at least to some extent. Extrapolation is simply complete post-stratification of individuals; more details are given in section V.B.3. We there comment also on the (possible) effects of ignoring sampling weights in post-stratification. We have not done any numerical comparison yet, since this would take quite some time, and since data files used for calibration should be carefully constructed and well understood. The latter is beyond the author's capabilities at present, since he is just a methodologist (and he likes that!), and not at all a subject matter statistician with regards to the LFS (which he would not like to be!). In other words, we need a more thorough analysis of the LFS on its own, which will be the topic of another paper.

We will see later that estimates from the LFS are used as benchmarks for calibration of several other surveys; household characteristics are sometimes used as calibration variables, while at other instances individual characteristics would be more convenient. The LFS should therefore provide precise estimates both on the household and on the individual level. As an example: the total and regional sizes of the Belgian population of individuals should be accurately estimated – as sums of calibrated individual weights for sampled (and responding) individuals – as well as total and regional sizes of the Belgian population of (private) households – as sums of calibrated household weights for sampled (and responding) households. As we will see in section V.B.3, current post-stratification is using individual characteristics only, and household weights are obtained as averages of individual weights for members of the household. So, one can argue against the appropriateness and precision of totals of household variables. We can do better, through simultaneous calibration of household and individual characteristics, using, henceforth, calibration (or reference) totals for households as well as for individuals. The methodology to achieve this is explained in section V.B.4. An attempt has already been made, but the techniques are not yet implemented in the production process.

We have not reproduced weights for the LFS in this text, since that would not contribute to the completeness of this text. However, estimates from the LFS are used in various other case studies in this report, which justifies at least a brief discussion of the LFS.

V.B.2 The sampling design

We now present briefly the sampling design. The purpose is to show how sampling weights can be calculated for this survey, which is based on a complex sampling design. Notice that the LFS has been restructured since 1999. One of its main features is that it became a rotational survey. I will not give numerical values of the sampling design parameters in this report.

First of all, the Belgian population (of private households) is *geographically stratified* according to the 10 administrative provinces and the Brussels Metropolitan Area. A frame of households is derived from the *National Register* of Belgian citizens.

In the first stage of the sampling procedure (within each stratum separately), *sections* are drawn according to a *probability proportional to size* (PPS) *sampling scheme*. "Sections" are parts of current communities (denoted *c*), but larger than the frequently used *statistical sections*. The size measure used is the number of households in section *c*. The total number of times a section *c* is selected is fixed, and PPS sampling is with replacement. A list-sequential scheme is applied. However, it can be argued that an assumption of *multinomial* PPS *sampling* adequately approximates the sequential scheme. Such a mathematical model is easier to handle, and ultimately provides a(n) (approximate) formula for calculating 2^{nd}-order inclusion probabilities too (based on a generalisation of formula 3.8.3 in Särndal *et al* (1992)), although we do not discuss this further in detail in this report. To each draw of a section corresponds a fixed number (depending on stratum) of households to be selected. Notice that sections are in fact PSUs (primary sampling units) in multi-stage cluster sampling of individuals.

Next, the "sample of sections" (taking into account their multiplicities) is spread "uniformly" over 12 trimesters (3 years). The procedure is such that (1°) approximately the same number of households will be drawn and contacted in each trimester, and (2°) the number of households drawn in each trimester is (approximately) fixed. While sections are drawn for a 3-year period, households in selected sections are drawn each year, using the most recent version of the National Register. Some realistic assumptions result in the fact that allocation of sections to trimesters does not have to be made explicit in formulae for inclusion probabilities.

Within each selected section, we then draw completely at random (i.e. according to SRS) the (random) number of households that has to be selected. Hence sampling of households is a two-stage cluster sampling design within strata. It is therefore relatively easy to obtain (approximate) 1^{st} and 2^{nd}-order inclusion probabilities for households. The 1^{st}-order probabilities, and hence the corresponding sampling weights, do only depend on the total (fixed) number of households to be selected in a given stratum and on the total number of households in the stratum. Hence the sampling design is self-weighting within the 11 geographical strata.

Since all individuals (above 15 years old) in a selected household are interviewed, sampling weights for individuals equal sampling weights of the household to which they belong.
Non-response rates are low for the LFS, because of the compulsory character of the survey. This makes the LFS an interesting survey for a thorough study of calibration techniques, since calibration then almost only corrects for sampling error, for which it has initially been developed, and since there is only limited disturbance due to non-response (given that other non-sampling errors also have a small impact).

V.B.3 Current calibration: post-stratification of individuals

Currently, extrapolation of the LFS is a complete post-stratification technique. Sampling weights can be ignored, since post-strata are subsets of sampling strata: see section III.B.1 and the remarks on the sampling design for the LFS in the previous sub-section.

Calibration, i.e. post-stratification, is at individual level, not at household level. Individuals are post-stratified by 4 variables:

- Gender;
- Age, grouped in 16 classes (0-4 yrs, 4-9, …, 70-74, 75+);
- The geographical stratification variable used in the sampling design (11 categories);
- Reference period, i.e. the trimester for which a household has been selected.

No cell in the resulting four-way classification is empty or considered to be too small, so there was no need to regroup some of the 1408 cells (for one survey year). Calibration totals are obtained from the National Register.

After calculating post-stratification weights for (sampled, and responding) individuals (including children of less than 15 years old that are in the selected households), calibration weights for households are obtained by averaging the weights of the members of the household. This approach has not been justified in any of the previous sections in this report. I strongly recommend a thorough comparison of this technique with the techniques discussed in sections III.D-E, in particular the technique of imposing equality of g-weights within clusters (households), if calibration is on individual characteristics.

V.B.4 Suggestion: calibration on both individual and household characteristics

The LFS data clearly provide a rich basis for the development of a sophisticated weighting scheme, taking individual as well as household variables into account. Hence we have a perfect situation for comparison of different calibration techniques, as outlined in section III.E. The ultimate goal of a more sophisticated calibration strategy is not its complexity, but its efficiency, such that reliability of estimates, which will be used as calibration totals for other surveys, is assured, and such that numerical consistency of estimates across different surveys is improved. Notice that one of the purposes of generalised calibration, other than simple complete post-stratification, is to obtain stability of g-weights and calibrated weights (and estimates based on these). Of course, post-stratification is not necessarily doomed a priori to be of bad quality, especially not for such a large respondent sample.

A study of calibration of the LFS should start with a complete overview of the different uses of the results of the LFS. Notice that the type of variables for which totals have to be estimated, has an impact on the calibration model that might be chosen. Also, the various external sources that are available should be investigated further, and new sources is being looked after.

V.C CALIBRATION OF THE HOUSEHOLD BUDGET SURVEY (HBS)

V.C.1 Introduction

In this section we focus on the *Household Budget Survey* (HBS). Our main purpose here is to summarise the basic principles of the sampling design, and the way to incorporate the design in up-weighting schemes. We do not intend to repeat the calculations for the weights for the HBS, but a discussion of present practice is interesting for at least two reasons: (1°) to show how the nature of the study variables has to be considered when a weighting scheme is set up; (2°) to prepare for a more extensive numerical illustration of weighting for the Time Use Survey (TUS) in section V.D.

A study of calibration models for the HBS is important to be considered, since the HBS has many users. One of them is the *service for calculation of the consumer price index* (CPI). We do not have to deliver the weights to this service, but consideration of the needs for the CPI may shed light on the quality of the data that are collected in the HBS. This, together with lessons drawn from a study of calibration, may give indications about possible needs for improvement of the HBS.

The HBS has been reorganised in 1999. Since then it is a continuous survey. The burden for sampled households and individuals has been reduced: they now have to record details about their expenses during 1 month (the *reference month*) in a household or individual diary, while an additional questionnaire retrospectively collects information on expenses that cannot be covered by the diaries. This questionnaire asks for information on expenses during a period of 4 months: the reference month and the 3 preceding months. This allows detecting and covering expenses with some larger periodicity (e.g. insurance payments, rent for housing, etc), or expenses with an occasional character (e.g. buying a car, travel expenses, etc).

The reformed HBS actually started in November and December 1998, as a pilot survey (e.g. to get an idea about response rates). The data for these two months could be incorporated in the results. In this text, however, we ignore the data collected on households for which the reference month is either November or December 1998.

Response rates are very low for the HBS (less than 10%). A discussion of generalised calibration of the HBS therefore should focus on its potential to correct for non-response. Calibration models should incorporate (non-) response models, or, alternatively, an integrated system of response modelling to correct for differential non-response and calibration modelling to reduce sampling error should be studied. Variance estimation is an important issue in that context. This, however, is beyond the scope of this text. Here (and in section V.D) we use calibration as a method for simultaneous adjustment for non-response (a source of non-sampling error) and sampling error.

V.C.2 The sampling design, and Phase I inclusion probabilities

The sampling design for the HBS is very complex. It is a multi-phase design, of which a very brief (and certainly incomplete) discussion is presented here. Our main interest is to show how first-order inclusion probabilities, and hence sampling weights, can be calculated. These will be incorporated as initial weights in generalised calibration for the TUS in section V.D. The TUS is in fact a continuation of the HBS: responding households (for the HBS) are invited to participate in the TUS too. Therefore, the sampling design for the HBS only needs to be extended by one more phase (Phase IV here below), in order to cover the entire sampling procedure for the TUS. That is why this section also covers the TUS.

Three phases can be considered in the sampling procedure for the HBS; a 4th phase is to be considered for the TUS. However, a preliminary step in fact is the **geographical stratification** of the population (of households or individuals). The strata are the three regions Brussels Metropolitan Area, Flanders and Wallonie; they are numbered h = 1, 2, 3, respectively. Experience has shown that response is lower, for instance, in Brussels. Hence initial sampling fractions are higher for this stratum. Given this stratification, the following multi-phase procedure (within each stratum) is designed.

Phase I **2-stage sampling of households** (within each region h)

Stage I.1 **Probability proportional to size** (PPS) sampling of communities c

- We draw $n_{\mathrm{I}h}$ times, with replacement (WR), a community in region h. $n_{\mathrm{I}h}$ is fixed a priori. $n_{\mathrm{I}hc}$ will denote the number of times community hc has been selected; this number is random. Notice that $n_{\mathrm{I}h} = \sum_{c \in s_{\mathrm{I}h}} n_{\mathrm{I}hc}$, which is fixed.
- The size measure used is the number M_{hc} of households in community hc (i.e. community c in stratum h).

Stage I.2 **Stratified simple random sampling (STR-SRS)** of households in selected communities

- The initial (random) sample size is $m_{hc} = n_{\mathrm{I}hc} G_h$, in a selected community hc, where G_h is a fixed "group" size depending only on region h. Groups are sets of households, to be assigned to the same interviewer. G_h is larger for Brussels (h = 1) then for the other two strata, because of higher non-response in Brussels.
- The stratification variable is household size, with categories 1, 2, 3, 4 and 5+ (numbered k = 1, …, 5).
- Per "group", with size G_h, of households to be selected, the allocation to the strata k is fixed; let G_{hk} be the number of households to be selected *per group* in stratum k, with $G_h = \sum_k G_{hk}$. Then, there will finally be $m_{hck} = n_{\mathrm{I}hc} G_{hk}$ households selected in stratum k in selected community hc, and $m_{hc} = \sum_k m_{hck}$. G_{hk} is relatively larger for k = 1, since 1-person households tend to have lower response rates than others.

Phase II A **response mechanism** reduces the initial household samples to **respondent samples**

- Let $\dot{m}_{hck}$ be the number of responding households, among the m_{hck} initially selected in Phase I.
- The response probability for household i in stratum k in community hc is denoted θ_{hcki}. Then $\dot{m}_{hck} / m_{hck}$ is the observed value for θ_{hcki}, which is actually very small (less then 10%). A response model might be used to smooth observed response probabilities.

Phase III Occasionally, **the size of the respondent samples is limited**.

- Hence the respondent samples are reduced, in order not to exceed a maximum number of respondents *per group*. The main purpose of this additional step is to assure that different interviewers will have more or less the same workload per group of households they have been assigned to. (It's a way to simplify payment of interviewers.)
- The maximum number of respondents *per group* is set to q_h, depending only on region h, so that the maximum number of households interviewed in a selected community hc is $n_{\mathrm{I}hc} q_h$.

- The sizes of the final respondent samples (within communities *hc*) per stratum *k* are $\ddot{m}_{hck}$.

Phase IV A **second response mechanism** reduces the final samples for the HBS to **final TUS samples**

- Let m'_{hck} be the sizes of the final TUS respondent samples.
- $m'_{hck} / \ddot{m}_{hck}$ is an observed dropout rate, measuring willingness of households having participated in the HBS, to continue participation in the TUS. A dropout model might be used to smooth observed dropout rates.

It further has to be noticed that sampled households are allocated to a particular month, which is called the *reference month* for that household. We do not discuss this property of the sampling design, and its consequences, in detail in the present text. For calibration purposes we assume that households allocated to different non-overlapping time periods (reference months, trimesters (as a series of three reference months), etc) are elements of independent samples.

Assuming proper or multinomial PPS sampling in Phase I (see also section V.B), it is possible to derive manageable (approximate) formulae for 1st (and 2nd) order inclusion probabilities, and hence for sampling weights, corresponding to Phase I. The result is the following 1st order inclusion probability for a household *i* in (household size) stratum *k* in community *hc* :

$$\pi_{hcki} = \frac{m_{hk}}{M_{hck}} \frac{M_{hc}}{M_h}, \tag{V.1}$$

where M_{hck} is the number of households in stratum *k* in community *hc*, $M_{hc} = \sum_k M_{hck}$ is the number of households in community *hc*, $M_h = \sum_c M_{hc}$ is the total number of households in stratum *h*, and m_{hc} is the number of sampled households in community *hc* (see Phase I, Stage I.2 here before). We can distinguish between a *calculated value* for m_{hc}, and an *actual value*. The former are following from the above reasoning, and are based on predetermined group sizes; the latter are obtained simply by counting from the sample. I recommend using the actual values when 1st order inclusion probabilities have to be calculated and used as initial weights in calibration.

The Phase I sampling weights for households are then calculated from:

$$d_i = \pi_{hcki}^{-1} = \frac{M_{hck}}{m_{hk}} \frac{M_h}{M_{hc}}, \tag{V.2}$$

for any household *i* in (household size) stratum *k* in community *hc*. The numbers M_{hck}, M_{hc} and M_h are obtained from a frame of households, which is derived from the Belgian National Register of individuals. This sampling weight is also valid for any member of household *i*, since all household members all interviewed for the HBS (and TUS).

One could construct response or dropout models, to include additional corrections in these sampling weights. Our choice however is to use the above Phase I sampling weights as initial weights in generalised calibration. Consequently, generalised calibration is intended to correct for non-response as well as for sampling error.

A model-based study (using logit modelling, for instance) of non-response for the HBS (and dropout for the TUS) would be very useful, since it could give indications about the auxiliary variables to be included in a calibration model that has to correct for non-response. This, however, is not a topic for the present study.

V.C.3 Current post-stratification, and suggested generalised calibration

Current extrapolation for the HBS is still based on a traditional post-stratification technique. Post-stratification for households is based on the following post-stratification variables:

- The geographical stratification variable used in the sampling design (3 categories);
- The size of the household (categories 1, 2, 3, 4+), also used in the sampling design;
- The number of active persons in the household (categories 0-1, 2+);
- Age of RP, combined with professional status of RP (categories *employed*, *self-employed*, *non-active* <60 *yrs old*, *non-active* 60-69 *yrs old*, *non-active* 70+ *yrs old*);
- Reference period, e.g. the month or trimester for which a household has been selected.

A complete cross-classification by these variables is not possible: there are too many cells, given that only 3,745 households have agreed to participate in the HBS in 1999. Hence, many cells had to be regrouped.

Estimation of totals has to be carried out for several types of variables (at household level here only), e.g.

- Monthly expenditure variables y and y_j, with observed values y_i = expenditure by household i in its reference month, and y_{ji} = expenditure in month j by household i;
- Quarterly expenditure variables q and q_j, with observed values q_i = expenditure by household i in the 4-month period (j–3, j), where j is the reference month of household i, and q_{ji} = expenditure in month j by household i, which is not directly observed, but derived from an observed expenditure covering 4 months.

Other periodicities are possible, but considering this here would not contribute to the present discussion. The above variables could also be replaced with a corresponding set of indicator variables, indicating whether there was a strictly positive expenditure or not (for a given period).

If the 12 sub-samples, corresponding to 12 reference months, are treated as being independent, then extrapolation can be done for each month separately, and a total estimated expenditure for the survey year and for the entire population (possibly restricted to a geographical stratum, or to another domain of interest) would be calculated from $\hat{T}_y = \sum_j \hat{T}_{yj} = \sum_j \sum_i w_{ji} y_{ji} = \sum_j \sum_i w_{ji} y_i$, where the summation with index i is over households with reference month j. (Notice that, in fact, 12 variables y_j are reduced to one variable y.) Stated in terms of generalised calibration methodology, it follows that, for each month j, a set of calibrated weights $\mathbf{w}_j = (w_{ji})$ or g-weights $\tilde{\mathbf{g}}_j = (\tilde{g}_{ji})$ for households with reference month j has to be found, based on a calibration model $\{\min \tilde{\mathbf{d}}_j G(\tilde{\mathbf{g}}_j); \mathbf{Z}_j^T \tilde{\mathbf{D}}_j \tilde{\mathbf{g}}_j = \mathbf{s}\}$, where $\tilde{\mathbf{d}}_j$ is a vector of initial weights for households with reference month j, $\mathbf{Z}_j$ is an appropriate design matrix, based on households with reference month j and household level auxiliary information, and $\mathbf{s}$ is a corresponding vector of calibration totals (which does not depend on reference month!). See section III.E.3 for a derivation of this calibration model. The same calibration results can also be used

to estimate the population total (for the entire survey year) for *q*-variables: $\hat{T}_q = \sum_j \hat{T}_{qj} = \sum_j \sum_i w_{ji} \overline{q}_{ji} = \frac{1}{4} \sum_j \sum_i w_{ji} q_i = \sum_j \sum_i w_{ji} q_i^*$, where $q_i^* = \frac{1}{4} q_i$.

Alternatively, calibration (or post-stratification in particular) can be done independently for each trimester, using all households with reference month in the trimester considered. Replacing subscript *j* with subscript *t*, and introducing notation referring to trimesters instead of months, it will be clear that, within the generalised calibration framework, we then consider a calibration model $\left\{\min \tilde{\mathbf{d}}_t G(\tilde{\mathbf{g}}_t); \mathbf{Z}_t^T \tilde{\mathbf{D}}_t \tilde{\mathbf{g}}_t = \mathbf{s}\right\}$ for each trimester *t*. Notice that the vector **s** of calibration totals has not changed. Moreover, time (e.g. month *j*) can be reintroduced as a calibration stratum variable; the same vector **s** is then repeated for each calibration stratum. Let $\mathbf{w}_t^*$ be a solution to $\left\{\min \tilde{\mathbf{d}}_t G(\tilde{\mathbf{g}}_t); \mathbf{Z}_t^T \tilde{\mathbf{D}}_t \tilde{\mathbf{g}}_t = \mathbf{s}\right\}$, *t* = 1, …, 4. The above-mentioned estimators $\hat{T}_y$ and $\hat{T}_q$ will then be replaced with estimators $\hat{T}_y^* = \sum_t \sum_i w_{ti}^* y_i$ and $\hat{T}_q^* = \sum_t \sum_i w_{ti}^* q_i^*$, respectively, with $q_i^* = \frac{1}{4} q_i$ as before.

Currently, extrapolation is done for each trimester separately, as explained in the previous paragraph. The extrapolation coefficients thus obtained are transmitted to Eurostat, together with the observed (edited) data. The calibrated weights are considered to be stable.

The advantage of a period-based calibration method (i.e. to use time as a calibration stratification) is that period-specific estimates are obtained. Provided that these are precise, they might be used to study trends in expenditure, and to detect possible seasonal effects.

A comparative study of weighting schemes for the HBS has to be carried out. Notice that here too two-level calibration can be considered. Moreover, it follows from the continuity of the sample that time will be an important calibration variable (possibly the calibration stratum variable), as explained in the previous paragraphs.

Current post-stratification techniques have to be compared with alternative calibration techniques. By the way, post-stratification as outlined in the beginning of this sub-section can, strictly speaking, not be justified, given the general discussion in section III.B.1. Indeed, it follows from formula (V.2) for the Phase I sampling weights that the sampling design is not self-weighting within post-strata: these sampling weights do depend on community *c*, while that level of geographical stratification is not appearing in the above list of post-stratification variables. We will have to investigate this further in order to understand possible effects of post-stratification in a situation where application of post-stratification is theoretically not allowed.

V.C.4 Calculation of Phase I sampling weights

This section V.C is closed with some further comments and results with respect to the calculation of Phase I 1st order inclusion probabilities and sampling weights.

As mentioned before, we have not used the pilot survey data carried out in November and December 1998, but only data from households with reference month in 1999. The weights are at household level, and therefore also valid at individual level, although no individual auxiliary data is used. Actual sample sizes m_{hc} (see section V.C.2) are thus obtained from the 1999 initial HBS sample. Population counts of households, i.e. M_{hck}, M_{hc} and M_h, are obtained from a frame of households, derived from the National Register of individuals. The 1998 register has been used, since sampling was also based on this register. Two series of sampling weights have been calculated: the “calculated” series

based on calculated values for the sample sizes m_{hc}, and the "actual" series based on the sample itself. A graphical comparison has revealed that the two series are very close to each other. This is also reflected in the table here below. After calculating the two series of weights and assigning the right pair of weights to each sampled household, it is possible to estimate the number of households per region (stratum) as the sum of the sampling weights for households within the region. "Valid N" refers to the number of households in the initial sample.

Table 5.1 *Estimated numbers of households, based on two series of Phase I sampling weights, by region and by reference month*

		Phase I sampling weights with ACTUAL sample size	Phase I sampling weights with CALCULATED sample size	
		Sum	Sum	Valid N
Region	Brus-Brux	469,830.3	476,188.8	14,664
	Vlaanderen	2,369,250.2	2,419,729.4	26,813
	Wallonie	1,367,248.8	1,396,778.2	19,981
Table Total		*4,206,329.3*	*4,292,696.4*	*61,458*
Reference month	1	440,797.6	453,879.1	5,569
	2	328,153.9	335,932.2	4,664
	3	339,771.9	347,040.9	5,074
	4	344,016.9	350,714.0	5,131
	5	345,287.6	352,191.6	5,102
	6	343,612.2	350,522.0	5,131
	7	348,455.3	354,942.2	5,131
	8	342,241.1	348,223.3	5,131
	9	345,362.9	350,951.2	5,132
	10	341,117.4	348,223.3	5,131
	11	343,563.1	349,890.6	5,131
	12	343,949.4	350,185.9	5,131
Table Total		*4,206,329.3*	*4,292,696.4*	*61,458*

The numbers of households in the 1998 sampling frame, per region, are: 467,860 for *Brus-Brux*, 2,353,864 for *Vlaanderen* and 1,356,956 for *Wallonie*. Both series of sampling weights are over-estimating the frame-based counts of households in the 3 regions. However, the series of weights based on actual sample sizes provides less over-estimating values, and, more importantly, provides a relative distribution of households across regions, which is closer to the frame-based one. Over-estimation of absolute counts is not that important, provided the relative bias is similar in different sub-populations. Further calibration will automatically adjust such a bias.

No further numerical results have been obtained for the HBS, since the respondents sample (of households) was not available to me at the moment of dealing with this survey in the context of writing this text on calibration. Notice, however, that the respondent sample for the HBS is close to the respondent sample for the TUS. The latter has been explored extensively, as it will be demonstrated in the next section.

V.D CALIBRATION FOR THE TIME USE SURVEY (TUS)

V.D.1 Introduction – Preparing input files for g-CALIB-S

The *Time Use Survey* (TUS) 1999 has been organised as a continuation of the HBS 1999. Details about the sampling design, non-response problems, and general issues on calibration, together with some more background information on this survey may therefore be found in section V.C on the HBS 1999.

In the next sub-sections, we will use the Phase I sampling weights, discussed at length in section V.C, as initial weights for calibration of the TUS. It is not the purpose of the present study to examine in detail the different weighting schemes that will be obtained, nor to provide a comparison. Rather this text is focussing on technical aspects of calibration. One important issue in this report is that, based on the TUS 1999, an illustration is given of the two-level simultaneous calibration technique discussed in section III.E, as well as of the related basic and derived techniques for one-level calibration (e.g. the clustering technique in sections III.D and III.E.4). Thus, the TUS is used to illustrate how sophisticated calibration tools could be used to find optimal calibration weights for, for instance, the LFS (see section V.B.4), and obviously also for other household-based surveys, and, why not, probably also for business surveys.

The TUS is being analysed by specialists at the Vrije Universiteit Brussel, i.e. by Prof. Dr. I. Glorieux and his assistant Ms. J. Vandeweyer. Several discussions of up-weighting of the TUS have already taken place between those people and myself. I have tried to incorporate their ideas and their wishes in the present study. However, this work is not completely finished yet. The main purpose of my own work on calibration so far is to provide the tools (SPSS syntax programs), not only for calibration, given the required input files are ready, but also to construct these input files from various other sources. A lot of time has in fact been spent on bringing different files together. More work has to be done, but I believe a good starting point has now been reached, making creation of input files for calibration more easy, more reliable, and faster, in the future.

Consequently, we do not much concentrate on numerical results in this section, but more on the procedures to obtain those results. Numerical results shown hereafter should therefore primarily be considered as illustrations of generalised calibration methodology.

A syntax program (PREPARE_TUSDATA.sav) that prepares the basic TUS data for different types of calibration is presented in appendix VII.B.1; output files from this syntax program are: TUS99-XD DATA.sav, TUS99-HD DATA.sav, TUS99-ZD DATA.sav and TUS99-VD DATA.sav. The program is still a bit messy, but can be a good starting point for future refinements and improvement of the survey data files that will be used as input files for calibration. The program itself provides an outline of the steps that have been, and that will always have to be taken to prepare the required files. No further discussion is included at this place; the interested reader can have a look at the syntax for more details. A guide to understand more easily the data transformation procedures is the summary table 3.8 in section III.E.6. One lesson can be drawn from this: more work has to be done with respect to preparation of basic files, and integration of various files into a single (application-specific) system.

Section 2 in PREPARE_TUSDATA.sav shows that the calibration strata (in the variable STRATUM, see section IV.B.3) are the three geographical regions *Brus-Brux*, *Vlaanderen* and *Wallonie*. We have aggregated the above mentioned data files by STRATUM. The aggregated files are useful in two ways: (1°) a zero sample total (per calibration stratum) for a particular calibration indicates that this variable should be omitted from the data files, or at least not be included in the calibration variable list @XVARS; (2°) the sample totals for these calibration variables can be replaced with the corresponding (estimated) population totals, i.e. the calibration totals. The latter is exactly what we have done to prepare the calibration totals files; the way to do this was quasi manually; calibration

totals are estimates from the 1999 LFS. The resulting files are called TUS99-XD TOTALS t.sav, TUS99-HD TOTALS t.sav, TUS99-ZD TOTALS s.sav and TUS99-VD TOTALS u.sav. Notice the similarity between the names of the 8 files that have been prepared, and the matrix notation in section III.E (especially in table 3.8 in section III.E.6).

We are then ready to apply different calibration models to each pair of data files, e.g. TUS99-XD DATA.sav and TUS99-XD TOTALS t.sav. The content of each pair of files is briefly discussed in the next sub-sections V.D.2.i-iv, although the calibration variables actually used are not our main concern in this study. Further research may result in the incorporation of other calibration variables.

V.D.2 One-level and two-level calibration of the TUS

V.D.2.i Individual-level calibration using type $(\mathbf{X}, \mathbf{d}, \mathbf{t})$ *data*

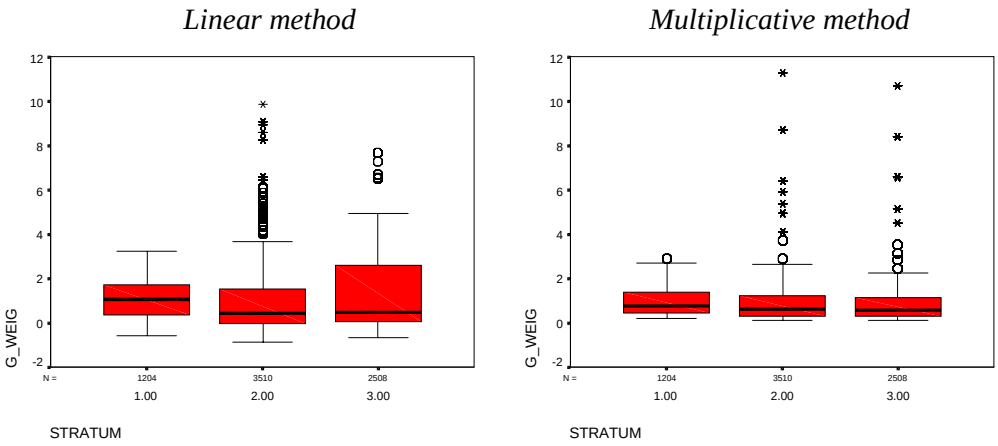
Three original qualitative variables, representing individual characteristics of responding people in the 1999 TUS, are transformed into calibration variables (i.e. indicator variables):

- Gender (male, female; calibration variables: S1, S2);
- Age (<30, 30-39, 40-49, 50-59, 60-69, 70+; calibration variables: A1 to A6);
- Education (LO or unknown, LMO, HMO/VS, HOBU, UNI; calibration variables: E1 to E5).

Hence, we have variables S1, S2, A1 to A6 and E1 to E5 in TUS99-XD DATA.sav and TUS99-XD TOTALS t.sav; a constant variable X0 is included too, as usual. The calibration stratum variable STRATUM distinguishes the three major Belgian regions, as mentioned already; the file TUS99-XD DATA.sav includes a variable PROV, which can be used as an alternative calibration stratification variable (PROV should then be renamed as STRATUM; the file TUS99-XD TOTALS t.sav then has to be redefined appropriately). The initial weight variable WEIGHT contains the Phase I sampling weights (based on actual counts m_{hc}, as discussed in section V.C; the series based on calculated m_{hc} is stored too in the survey data file). A household identification variable (CLUSTER) is present in TUS99-XD DATA.sav, but is not used here.

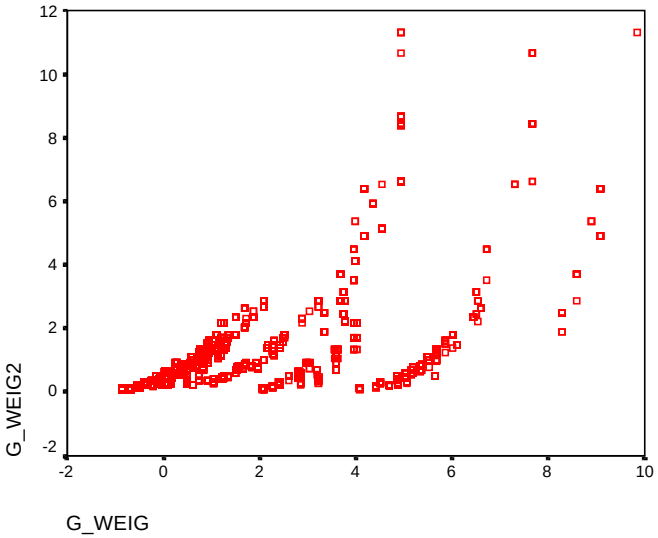
Two calibration models $\left\{\min \mathbf{d}^T G(\mathbf{g}); \mathbf{X}^T \mathbf{D}\mathbf{g} = \mathbf{t}, \mathbf{g} \in \Omega_B\right\}$ have been applied: the linear method and the multiplicative method, both with model formula 1 + Gender + Age + Education. Notice that the linear method will then produce g-weights that are additive in the calibration variables, while the multiplicative method produces g-weights that are additive in the calibration variables on a logarithmic scale. The scale parameter is always estimated from the data, using the constant variable X0; the g-weights are with respect to the scaled weights. The two series of g-weights are shown in figure 5.1. With the multiplicative method, the g-weights are less distributed, and there are less extreme values, at least for stratum 2 (*Vlaanderen*). A practical advantage of the g-weights under the multiplicative method is that are all positive. This truncation from below is partly responsible for the smaller dispersion in the g-weights too.

Fig 5.1 *Comparison of 2 series of g-weights from individual-level calibration of TUS 1999 data, both with model formula* 1 + Gender + Age + Education



The survey data file TUS99-XD DATA.sav can easily be merged with the two output files (both called WEIGHTS.sav at exit of g-CALIB-S, but renamed immediately), since all files are sorted by CASE on exit of g-CALIB-S. This allows constructing various scatter diagrams, plotting the series of weights (G_WEIG for the linear method, and G_WEIG2 for the multiplicative method) against each other. A simple scatter diagram is in figure 5.2. Calibration variables can be used to constructs separate scatter diagram for different subpopulations. This might help to identify the "clusters" that appear to be present in figure 5.2. Obviously, many other data exploration techniques may be used to try to understand the data and the calibration results. This, however, is not the purpose of the present text.

Fig 5.2 *Comparison of 2 series of g-weights from individual-level calibration of TUS 1999 data, both with model formula* 1 + Gender + Age + Education, *by means of a scatter diagram*



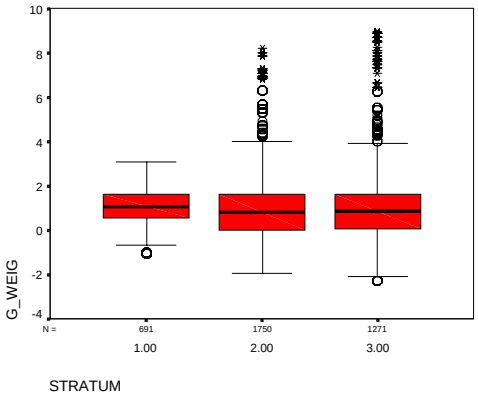
V.D.2.ii Clustering: individual-level calibration using type $\left(\tilde{\mathbf{H}}, \mathbf{d}^{+}, \mathbf{t}\right)$ *data*

With the approach of the previous section, individuals belonging to the same household have different g-weights and calibrated weights, because they generally have different individual characteristics. In this section we impose the same estimated weight for individuals in the same household, irrespective of their individual characteristics. The clustering technique is applied to that end (section III.E.4). Hence, we have to solve calibration models of the type $\left\{\min \mathbf{d}^{+T} G(\tilde{\mathbf{g}}); \tilde{\mathbf{H}}^{T}\mathbf{D}^{+}\tilde{\mathbf{g}} = \mathbf{t}, \tilde{\mathbf{g}} \in \tilde{\Omega}_{B}\right\}$. Recall that the elements of the design matrix are averages of individual values within households; see table 3.8 in section III.E.6.

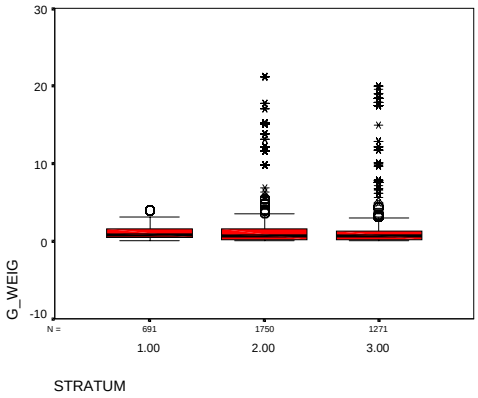
We have again tried the linear and the multiplicative method, with model formula 1 + Gender + Age + Education, as in the previous section. The results are in figure 5.3. The linear method still results into negative weights. The multiplicative method now gives quite extreme g-weights, compared with the results in the previous section. We therefore tried to restrict the g-weights even more, by using the logit method. Unfortunately, the method to find extreme lower and upper bounds (section II.C) has not been implemented yet. Some trial and error resulted into acceptable results for the logit method with lower bound of 0.0 and upper bound of 8.0. The results are also shown in figure 5.3.

Fig 5.3 *Comparison of 3 series of g-weights from individual-level calibration of TUS 1999 data, all with model formula* 1 + Gender + Age + Education, *using clustering to impose equal g-weights within households*

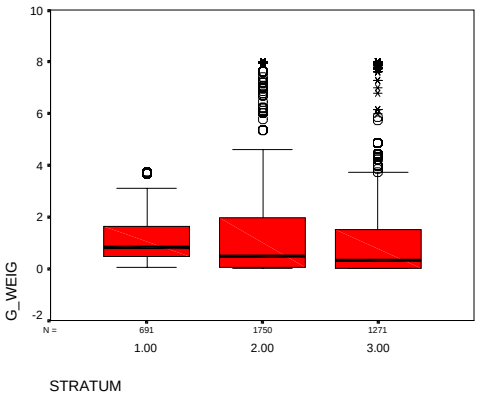
Linear method



Multiplicative method



Logit method with L = 0.0 *and U* = 8.0



V.D.2.iii Household-level calibration using type $\left(\mathbf{Z},\tilde{\mathbf{d}},\mathbf{s}\right)$ *data*

Household characteristics used for calibration at household-level are:

- Household size, or HHsize (1, 2, 3, 4, 5+; calibration variables ZS1 to ZS5);
- Age of RP, or RPage (<30, 30-39, 40-49, 50-59, 60-69, 70+; calibration variables: ZA1 to ZA6);
- Education of RP, or RPeduc (LO or unknown, LMO, HMO/VS, HOBU, UNI; calibration variables: ZE1 to ZE5).

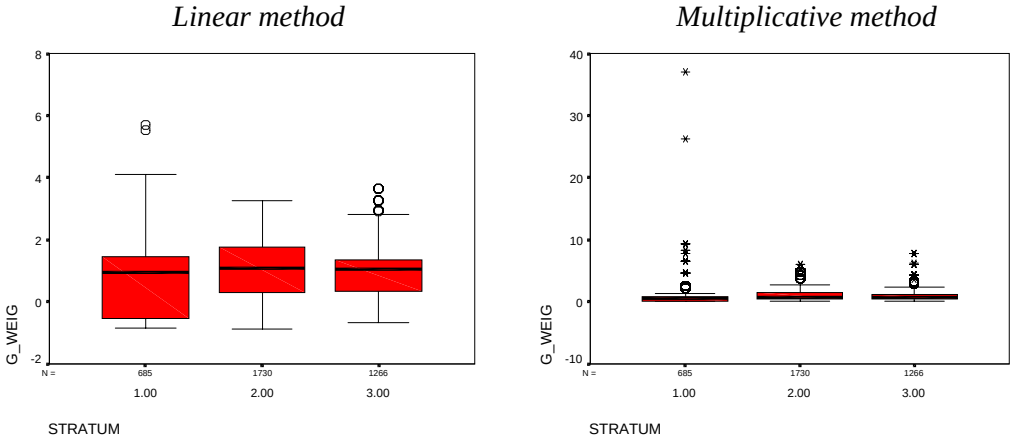
Calibration variables in TUS99-ZD DATA.sav and TUS99-ZD TOTALS s.sav are: ZX0, ZS1 to ZS5, ZA1 to ZA6 and ZE1 to ZE5; no variables representing joint effects of the above three variables are constructed. Again geographical stratification is in STRATUM; initial Phase I sampling weights are in WEIGHT; and some more variables are present in TUS99-ZD DATA.sav, such as PROV and MONTH which could both be used to define calibration strata (or calibration variables).

The calibration models being applied are of the form $\left\{\min \tilde{\mathbf{d}}^T G(\tilde{\mathbf{g}}); \mathbf{Z}^T\tilde{\mathbf{D}}\tilde{\mathbf{g}} = \mathbf{s}, \tilde{\mathbf{g}} \in \tilde{\Omega}_B\right\}$. The linear, the multiplicative and the linear truncated method have been applied. The bounds in the latter method were L = 0.01 and U = 5.0. The model formula in all three models was 1 + HHsize + RPage; RPeduc had to be omitted because of missing values for the corresponding calibration variables. The distributions of the resulting series of g-weights are shown in figure 5.4.

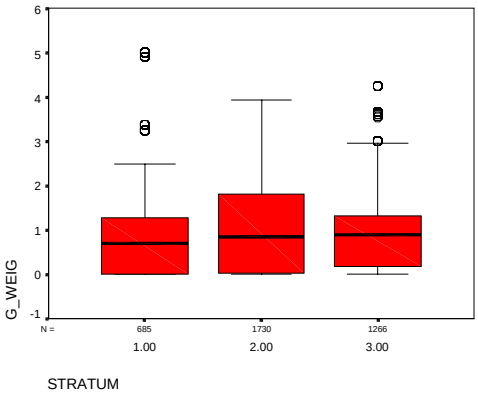
The linear method again results into negative weights, which is undesirable. The multiplicative method has apparently given bad results: there are (surprisingly) many extreme g-weights, which was already seen in the previous section, but which is here even more striking. Surprisingly, the truncated linear method allowed restricting the range of g-weights to the interval [0.01, 5.0]. It can also be noticed that convergence of the truncated linear method was very slow: respectively 24, 38 and 34 iterations were needed for calibration strata (regions) 1 to 3.

The logit method with g-weight range [0.01, 5.0], failed for stratum 1, but converged in only 6 iterations for strata 2 and 3. The results are not that much different from those resulting from the truncated linear method (strata 2 and 3).

Fig 5.4 *Comparison of 3 series of g-weights from household-level calibration of TUS 1999 data, all with model formula* 1 + HHsize + RPage



Truncated linear method with L = 0.01 and U = 5.0



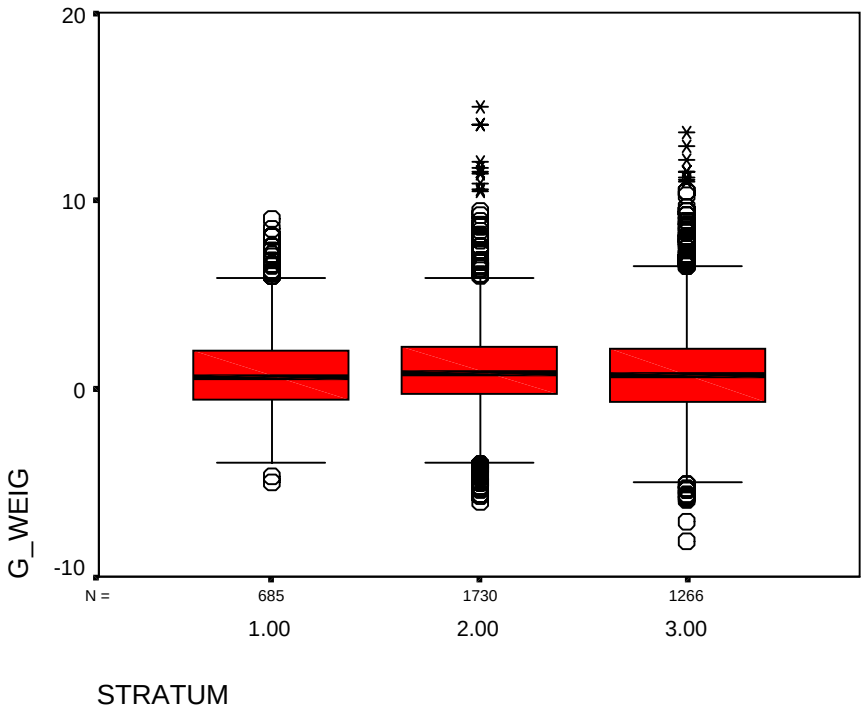
V.D.2.iv Simultaneous two-level calibration using type $\left(\mathbf{V},\tilde{\mathbf{d}},\mathbf{u}\right)$ *data*

Finally, calibration in this section is using both individual-level and household-level auxiliary information. The calibration variables are, combining the situations in the previous three sub-sections: the constant variable ZX0, HHsize indicators ZS1 to ZS5, RP's age indicators ZA1 to ZA6, and variables X0, S1 to S2, A1 to A6, and E1 to E6, which are similar to the variables with the same names in sections V.D.2.i-ii, but instead of being indicator variables (for individuals) they are now counting the number of household members in the corresponding categories of the underlying qualitative calibration variables. The calibration models applied in this section are of the form $\left\{\min \tilde{\mathbf{d}}^T G(\tilde{\mathbf{g}}); \begin{pmatrix}\mathbf{Z}^T \\ \mathbf{H}^T\end{pmatrix}\tilde{\mathbf{D}}\tilde{\mathbf{g}} = \begin{pmatrix}\mathbf{s} \\ \mathbf{t}\end{pmatrix}, \tilde{\mathbf{g}} \in \tilde{\Omega}_B\right\}$; see also table 3.8 for more details, and for comparison with the methods of the previous sections.

Henceforth, we are now ready to calibrate on individual and household information simultaneously. We start with the linear method and model formula 1 + HHsize + RPage + X0 + Gender +Age + Education. (We cannot write 1 for X0, since the term considered is the number of members for each household; Gender, Age and Education have an similarly modified interpretation, as already stated in the preceding paragraph. This, actually, is the difference between design matrices $\tilde{\mathbf{H}}$ and $\mathbf{H}$; see table 3.8.) Figure 5.5 shows the results. These are clearly not useful, as there are too many negative g-weights (and calibrated weights).

Several alternative models can be tried, in order to improve the pattern of estimated g-weights in figure 5.5. Of course, there are the multiplicative, the logit and the truncated linear method to force positive estimates for weights. But other model formulae too may work in finding weighting schemes with less negative weights. Figure 5.6 shows the estimated g-weights corresponding to 4 calibration models, all with model formula 1 + HHsize + X0 + Gender. This is done for illustration here; no other argument justifies these calibration models. Again slower convergence of the truncated linear method for stratum 1 may be notified.

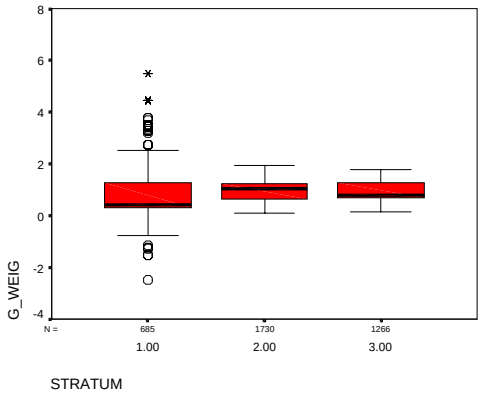
Fig 5.5 *g-Weights from two-level calibration of TUS 1999 data, using the linear method with model formula*
1 + HHsize + RPage + X0 + Gender +Age + Education



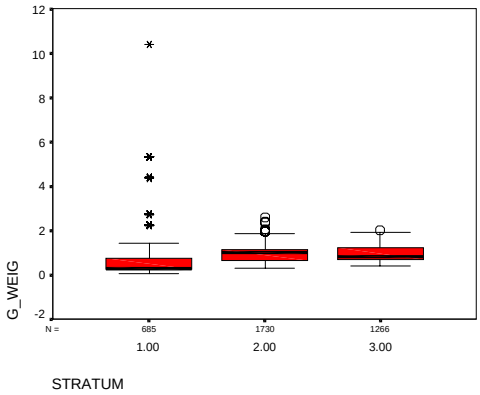
The model formula 1 + HHsize + X0 + Gender has reduced auxiliary information to 1 household characteristic and 1 individual characteristic. Obviously, only household characteristics, or only individual characteristics could be used within the same two-level calibration framework. If only household characteristics are used, the results will be exactly as in section V.D.2.iii. On the other hand, if only individual characteristics would be used, then we would not get the same results as in section V.D.2.ii. This is true because the objective function in the mathematical programming problem is different: the weights in the weighted sum of distance measures are different in the two approaches; the calibration constraints however are equivalent. Results are not presented here to illustrate these features.

Fig 5.6 *Comparison of 4 series of g-weights from two-level calibration of TUS 1999 data, all with model formula* 1 + HHsize + X0 + Gender

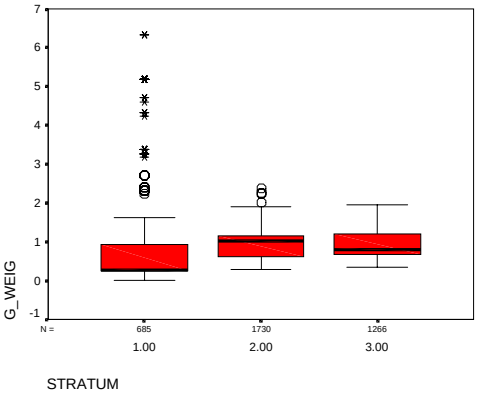
Linear method



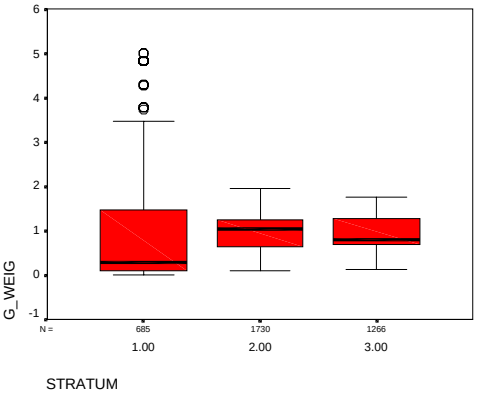
Multiplicative method



Logit method
with $L = 0.0$ *and* $U = 7.0$



Truncated linear method
with $L = 0.01$ *and* $U = 5.0$



V.E CALIBRATION OF THE TRAVEL SURVEY (TS)

V.E.1 Introduction

We can use the *Travel Survey* (TS) 2000 to illustrate not only the efficiency of g-CALIB-S, but also several other aspects related to calibration. It is clear that an extensive calibration study for this survey can be carried out, similar to what has been initiated here before for the *Time Use Survey* (TUS). Indeed, data and tools are now ready to try out one-level and two-level calibration, using many individual-level and household-level characteristics. This, however, is not what will be focussed on in the next sub-sections.

During the last 12 months, other aspects of the TS have been investigated, and a lot of time and energy has been put in establishing an integrated system to process this survey. This work is not finished yet, but advantage can be already taken now from at least two features of this integrated system:

- When the sample is designed, sampling information such as stratification variables and sampling fractions are stored in a systematic way in SPSS data files; the definition of strata is archived in SPSS syntax files.
- A text file contains enough information for all sampled individuals and households about their willingness to participate, and effective participation, in the TS, together with basic characteristics, asked for in the questionnaire. This file is available at any time, once data collection and data coding have started. Its name for the 2nd trimester is **t.i11.B23A.T0002.D071200**.

In the next sub-section we will see that a sample of households is selected for each trimester. In this study we restrict ourselves to calibration for the 2nd trimester. We will illustrate how the input files for g-CALIB-S can efficiently be prepared, taking full advantage of the above mentioned features of the current survey process for the TS at Statistics Belgium. Calibration itself, and discussion of results from different calibration models, is limited for this survey; we want to stress the fact that a lot of flexibility (with respect to calibration) is reached once an efficient integrated system is set up.

One thing that is currently still slowing down the production of calibrated weights and calibration estimates of totals of survey variables for the TS, is access to external sources. This problem needs to be tackled soon. A solution for the TS will clearly also be useful for other surveys where calibration has to be done. Generally speaking, it is a problem of integrating various databases – both registers and survey databases – in a single reference database.

V.E.2 The sampling design

The sampling design for the TS 2000 has deliberately been kept very simple. This is because, at the end of 1999, we felt the need for a thorough study of all aspects of this survey. Simplicity makes it possible to estimate variances within the generalised calibration framework. This in turn allows to evaluate the precision of calibration estimators of study variables, and hence to evaluate the quality of the survey. Decisions could then be drawn regarding a possible redefinition of the sampling design, if necessary. In this study, we do not go that far, but I believe the fundamentals now have been established to start that kind of quality study. Calibration is definitely an important intermediate step in the production of high-quality estimates.

It was decided to draw 4 samples of households, corresponding to the 4 trimesters. Households selected for a particular trimester receive the questionnaire by ordinary mail right after the trimester is

finished. They are kindly asked to complete the form about their travel experiences (both for leisure and for business) in the preceding trimester.

In this report, I have only used data for the 2nd trimester to illustrate various aspect of – or related to – calibration. The sampling design is STR-SRS, or stratified simple random sampling, of households. The sampling frame is a list of administrative private households, constructed from the National Register dated 1/1/2000. The stratification variables for the 2nd trimester are:

- REGION: distinguishing the major geographical areas *Brus-Brux*, *Vlaanderen* and *Wallonie*;
- PROV: distinguishing the 10 Belgian *provinces* and the Brussels Metropolitan Area (or *Brus-Brux*); PROV is a refinement of REGION;
- AGESIZ: a combination of *age of reference person* (RP) of a household and *household size*, with the following 17 categories:
  - 1 = "<25 & 1p" 2 = "<25 & 2+"
  - 3 = "25-39 & 1p" 4 = "25-39 & 2p" 5 = "25-39 & 3p" 6 = "25-39 & 4+"
  - 7 = "40-54 & 1p" 8 = "40-54 & 2p" 9 = "40-54 & 3p" 10 = "40-54 & 4+"
  - 11 = "55-64 & 1p" 12 = "55-64 & 2p" 13 = "55-64 & 3p" 14 = "55-64 & 4+"
  - 15 = "65+ & 1p" 16 = "65+ & 2p" 17 = "65+ & 3+"

A priori sampling fractions for the $11 \times 17 = 187$ sampling strata are ranging from 1/1000 to 1/200; the overall sampling fraction is 1/524, corresponding to about 8204 initially sampled households. Sampling fractions do not vary by PROV within REGIONs. Larger non-response rates (from experience in the 1st trimester) in some strata (e.g. in *Brus-Brux*, for smaller households and for younger RPs) are taken into account.

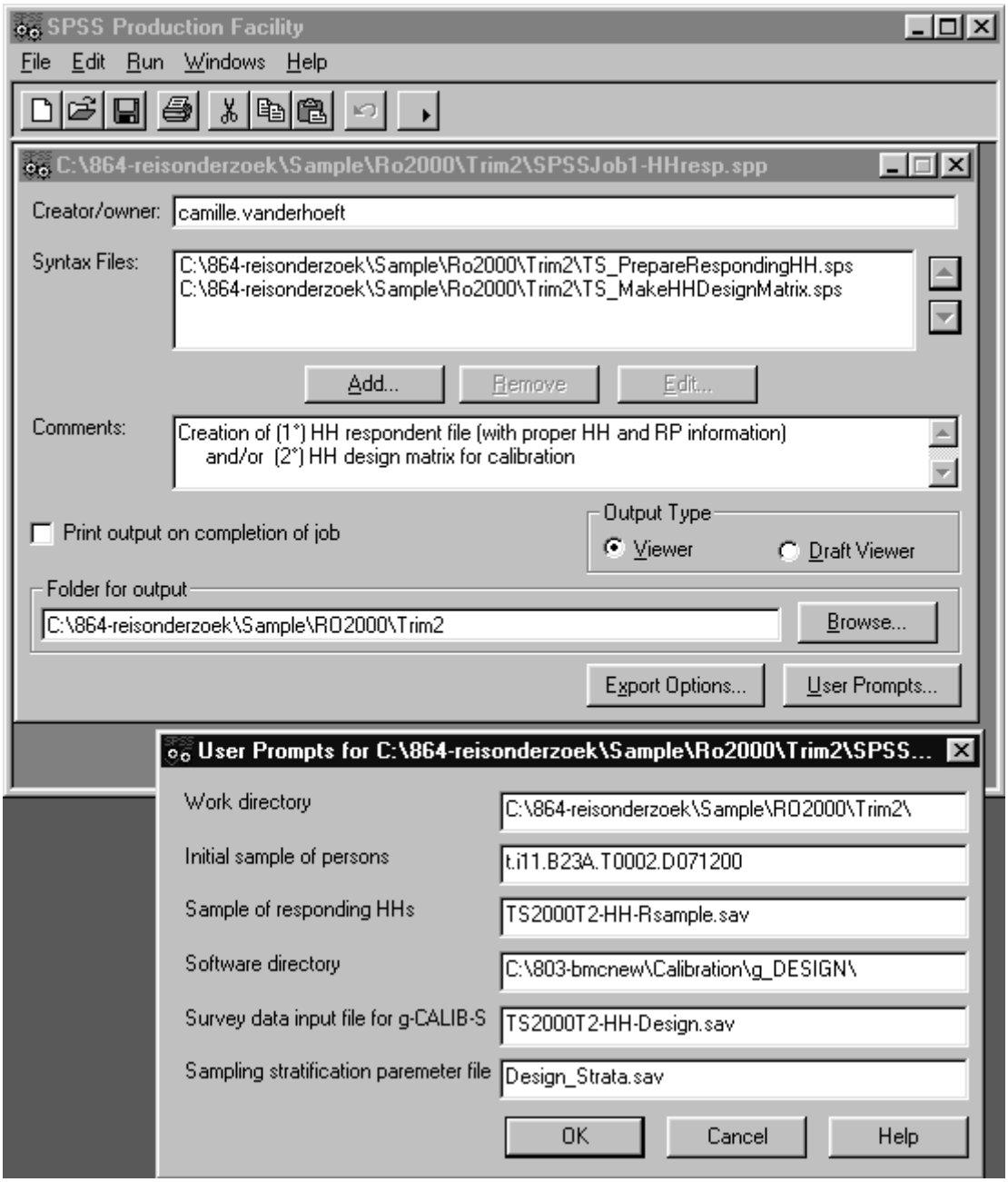
The definition of stratification variables, from basic variables in the household sampling frame is stored as SPSS syntax. A SPSS data file, called **DESIGN_STRATA.sav** (187 records), contains the stratification variables, a stratum identification, and for each stratum or record the number of households in the sampling frame and the number of households selected from the frame. From the latter two variables the sampling fractions, and hence the sampling weights, can be recalculated.

V.E.3 Preparing input files for g-CALIB-S

Two SPSS syntax files are constructed to transform basic data on sampled (responding and non-responding) individuals/households in the above mentioned text file **t.i11.B23A.T0002.D071200** and from the sampling information file **DESIGN_STRATA.sav** into a household respondent file, and finally into a survey data input file for g-CALIB-S. The two syntax programs, **TS_PrepareRespondingHH.sps** and **TS_MakeHHDesignMatrix.sps**, are reproduced in appendix VII.B. I believe these are very interesting files, showing how from basic data g-CALIB-S's input files can be prepared. The programs are important study material, for potential users of g-CALIB-S. It is demonstrated in these syntax programs how the construction of the survey data input file can be stored in well-structured and well-documented computer programs. Once such programs exist, it is relatively easy to modify the syntax when the calibration model has to be changed. This may be necessary for several reasons: new auxiliary information may become available, other calibration variables will be used, one wants to switch from household-level calibration to individual-level calibration, or calibration will be on both household and individual level simultaneously. Last but not least, and this is exactly my proper recent experience, the programs may have to be adapted to available series of calibration totals. The latter is still the weakest point in these programs, but if in the future external

databases (from registers or surveys) will be more easily accessible, then we will certainly be able to improve the programs, probably setting up finally a more standardised and complete set of data transformation programs.

These programs are called in a SPSS Production Facility job. The starting window and prompts window are as below. The input files are the files as mentioned earlier. Notice that the macros in g-DESIGN.sps are used (in the second program) to construct the design matrix for calibration. The output files are indicated as the "Sample of responding HHs" and the "Survey data input file for g-CALIB-S". Notice that the second file also contains the sampling weights.



At the end of the job, an additional file named TS-CalTotStructure.sav is created, by aggregating the survey data input file for g-CALIB-S on the calibration stratum variable STRATUM. Sample totals of calibration variables in the latter file are produced this way. This information is useful to detect empty

classes, which should then have to be treated further manually, in order to avoid problems when calibration is performed. More importantly, the sample totals in TS-CalTotStructure.sav can simply be replaced with appropriate calibration totals. The new file is then the calibration totals input file for g-CALIB-S. Interestingly, we prepared a lot of calibration variables in the survey data input file, and consequently also had these calibration variables in TS-CalTotStructure.sav. However, only for 2 underlying calibration variables the calibration totals were easily available (actually from DESIGN_STRATA.sav!), so that only a minority of cells could be filled with a calibration total; the other cells had to be emptied! This doesn't cause a problem for g-CALIB-S, as long as calibration variables with empty cells are not selected in the calibration model.

Another interesting feature of the data transformation programs is that (detailed) response information can easily be produced from now on. As an example, consider the table below, produced by the first program when sampling information from DESIGN_STRATA.sav is included and statistics are printed to file in order to be able to inspect the results of the first part of the data transformation.

Province – Brussels = Antwerpen

RP-age - HH-size combination	Nbr. of HHs in the HH's population stratum		Nbr. of HHs in the HH's sample stratum		STR-SRS Sampling weight of the HH	
	Mean	Valid N	Mean	Valid N	Mean	Valid N
0-24 1p	9678	N=1	23	N=1	420.78	N=1
0-24 2+	6512	N=2	9	N=2	723.56	N=2
25-39 1p	50328	N=33	118	N=33	426.51	N=33
25-39 2p	39179	N=8	37	N=8	1058.89	N=8
25-39 3p	35428	N=17	33	N=17	1073.58	N=17
25-39 4+	56770	N=21	53	N=21	1071.13	N=21
40-54 1p	40856	N=21	57	N=21	716.77	N=21
40-54 2p	43434	N=21	43	N=21	1010.09	N=21
40-54 3p	43423	N=23	43	N=23	1009.84	N=23
40-54 4+	72351	N=38	72	N=38	1004.88	N=38
55-64 1p	26035	N=17	46	N=17	565.98	N=17
55-64 2p	51049	N=39	89	N=39	573.58	N=39
55-64 3p	16866	N=21	30	N=21	562.20	N=21
55-64 4+	9188	N=7	16	N=7	574.25	N=7
65+ 1p	81786	N=47	82	N=47	997.39	N=47
65+ 2p	83586	N=78	146	N=78	572.51	N=78
65+ 3+	15702	N=17	28	N=17	560.79	N=17

This is only one out of 11 such tables, for the province of **Antwerpen**, but it shows clearly that response rates can easily be produced now; "Valid N" stands for the number of responding households. A bit more work on the programs is required to include the production of response information in the calibration preparation step. Thus, the programs might become useful (data mining) tools for the survey manager who needs to follow up response behaviour and data collection. It simply implies that the text file mentioned earlier (t.i11.B23A.T0002.D071200 for the 2nd trimester**)** should be available and be consulted frequently.

The calibration variables that can finally be used, because the corresponding calibration totals are in the calibration totals file, are X0, AGE1 to AGE5 (for age classes of RP, called AgeRP5 hereafter) and HHS1 to HHS2 (1 resp. 2 or more persons in the household, called HHsize2 hereafter). Notice that the

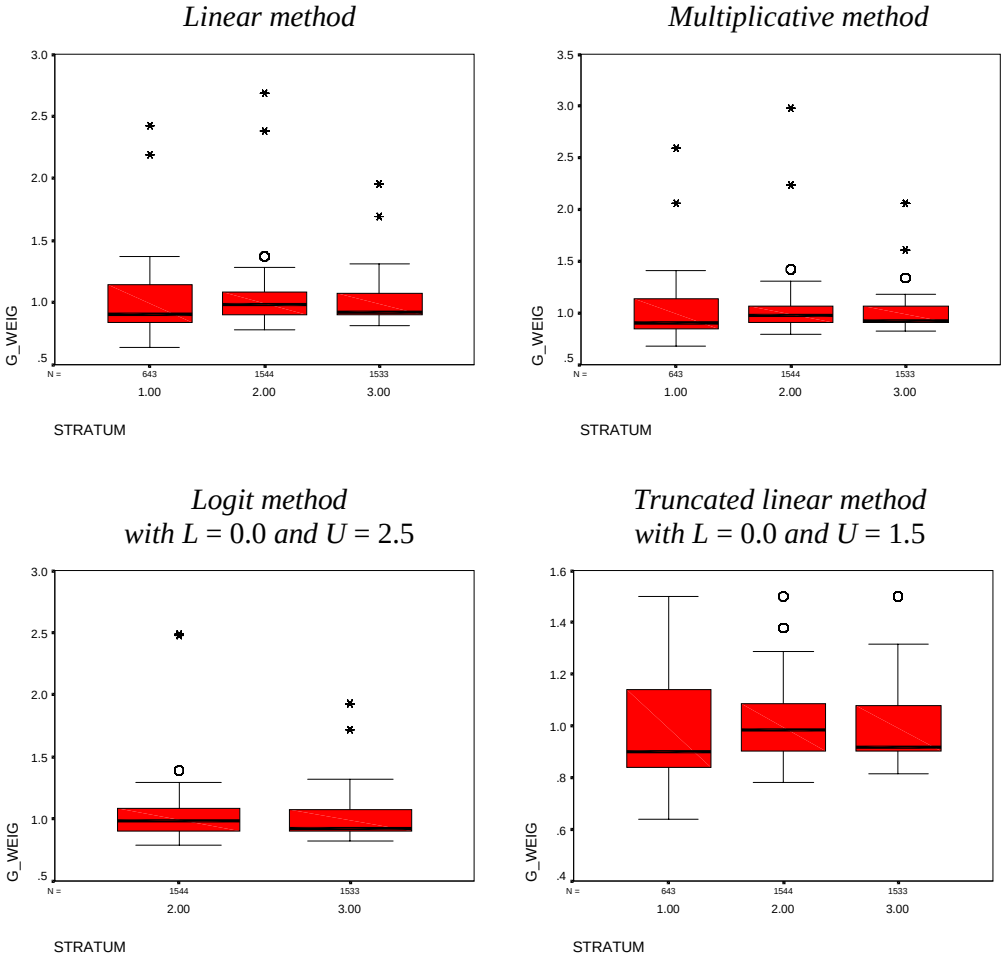
data are of type $\left(\mathbf{Z}, \tilde{\mathbf{d}}, \mathbf{s}\right)$, as in section III.E.3. The calibration models are of type $\left\{\min \tilde{\mathbf{d}}^T G(\tilde{\mathbf{g}}); \mathbf{Z}^T \tilde{\mathbf{D}} \tilde{\mathbf{g}} = \mathbf{s}, \tilde{\mathbf{g}} \in \tilde{\Omega}_B\right\}$. See table 3.8. The maximal design matrix corresponds to the model formula 1 + AgeRP5 + HHsize2.

V.E.4 Calibration results

For illustration we have retained 6 models:

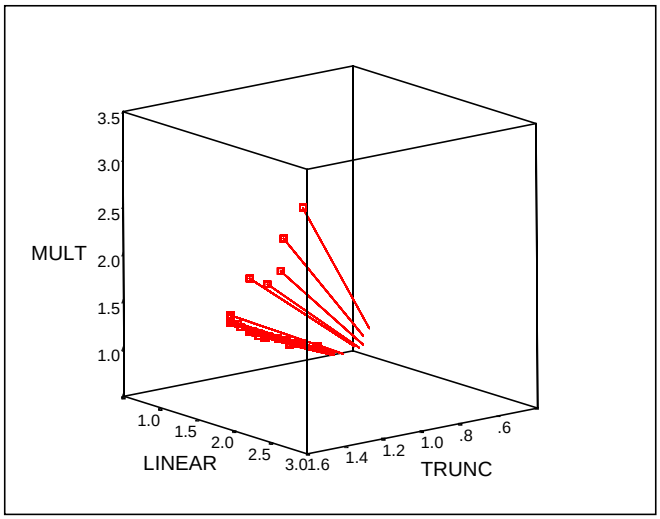
- Model 1 : linear method and model structure 1 + AgeRP5 + HHsize2
- Model 2 : multiplicative method and model structure 1 + AgeRP5 + HHsize2
- Model 3 : truncated linear method, with L = 0.0 and U = 1.5, and model structure 1 + AgeRP5 + HHsize2
- Model 4 : logit method, with L = 0.0 and U = 2.5, and model structure 1 + AgeRP5 + HHsize2
- Model 5 : multiplicative method and model structure 1 + AgeRP5
- Model 6 : multiplicative method and model structure 1 + HHsize2

Fig 5.7 *Comparison of 4 series of g-weights from household-level calibration of TS 2000 data (trimester 2), all with model formula* 1 + AgeRP5 + HHsize2



The results (estimated g-weights) are in figures 5.7-9. The results for models 1 to 3 are very similar. This is confirmed by the 3D scatter in figure 5.8, which could be constructed easily after merging the survey data input file with the three (renamed) output files WEIGHTS.sav for models 1 to 3. The line segments in this graph are called "spikes" and connect each (*x*,*y*,*z*) data point (corresponding to a household) in the g-weight space with the origin. The fact that many spikes coincide indicates that most data points are on the line $x = y = z$ of equal g-weights from the three models.

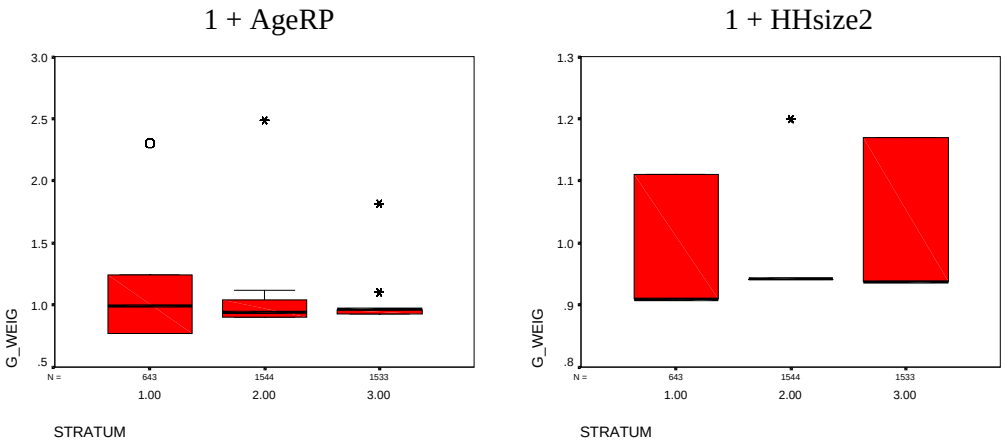
Fig 5.8 *Comparison of 3 series of g-weights from household-level calibration of TS 2000 data, all with model formula* 1 + AgeRP5 + HHsize2, *by means of a 3D scatter diagram*



Some problems were encountered with the logit method (model 4). First, several different pairs (*L*, *U*) were tried, but the calibration failed for calibration stratum 1 (REGION *Brus-Brux*). We don't see why, at this moment. Second, it is strange that the *U*-value had to be increased from 1.5 under the truncated linear method to 2.5 under the logit method. The theory in section II.C, however, indicated that extreme values for *L* and *U* are independent of the distance or calibration function. We have currently no explanation, so that further research is necessary, possibly resulting in improvements to our software g_CALIB-S.

We then omitted one of the terms in the model formula (and used the multiplicative method invariably): the new models are models 5 and 6, for which the estimated g-weights are plotted in figure 5.9. Comparing models 5 and 6 with model 2 seems to indicate that HHsize2 has minor effect on the g-weights in calibration stratum 3 (REGION *Wallonie*), while AgeRP5 has minor effect only in calibration stratum 2 (REGION *Vlaanderen*). This kind of calibration model comparisons has to be studied further, and formal statistical tests should be implemented in calibration practice.

Fig 5.9 *Comparison of 2 series of g-weights from household-level calibration of TS 2000 data (trimester 2); multiplicative method*



Finally, for all models 1 to 6, the scale parameters ϕ for the three calibration strata were estimated from the data (variable X0). Hence the g-weights are always with respect to the scaled sampling weights. Therefore, the g-weights do not reflect to correction for non-response, but merely the effect of sampling error. The estimates for ϕ however can be interpreted as estimated reciprocal response rates, globally within the three calibration strata. This is true because the initial weights are the true sampling weights. The respective values are 2.24841, 2.02310 and 2.23724. These are indeed comparable with the reciprocal overall response rate of 8,204/3,720 = 2.20538, or a response rate of 45.34%.

V.F GENERALISED RAKING OF CROSS-CLASSIFICATIONS OF LABOUR VOLUME AND LABOUR COMPENSATION

V.F.1 The problem

In this section, we discuss generalised calibration for cross-tabulations of *labour volume of employees*, *labour volume of self-employed people* and *compensation of employees* by *branch of industry* (BRANCH), *gender* (SEX) and *level of education* (EDUC).

Our aim in this report is merely to illustrate generalised calibration technology, so the reader is asked not to consider the figures here below as being final.

The *National Accounts* (NA-1997) are providing an appropriate breakdown by BRANCH (NACE classification), but a breakdown by SEX and EDUC is not available from this source. The labour force survey (LFS) carried out by Statistics Belgium, however, provides a breakdown by all three variables. The data on labour volume of employees and of self-employed people are shown in tables 5.2 and 5.3. The row (SEX-EDUC) totals, from the LFS, will be calibration totals. The NA's labour volume distribution by BRANCH, adjusted for the LFS's total labour volume, will serve as a series of calibration totals by BRANCH; the figures are in the last row in each table.

Table 5.2 *Labour volume of employees* ($\times 10^3$) *by* BRANCH, SEX *and* EDUC; *breakdown from LFS, with breakdown by* BRANCH *from NA*

SEX / **EDUC**	**BRANCH** A+B	C+D+E	F	G+H+I	J+K	other	*Total*
Male							
Lower	6.63	270.65	107.75	245.45	56.87	180.24	*867.59*
Middle	4.69	192.13	59.18	125.24	24.26	77.67	*483.17*
Higher	1.08	112.66	14.17	65.42	92.29	214.96	*500.58*
Female							
Lower	2.74	92.71	5.14	142.5	65.04	239.45	*547.58*
Middle	1.57	43.9	1.61	68.78	22.11	119.98	*257.95*
Higher	0.17	46.42	3.48	50.72	59.19	361.41	*521.39*
Total	*16.88*	*758.47*	*191.33*	*698.11*	*319.76*	*1193.7*	*3178.3*
% from NA	0.64%	21.69%	5.66%	21.74%	10.39%	39.88%	100%
Adjust. NA	*20.21*	*689.44*	*179.93*	*690.89*	*330.32*	*1267.47*	*3178.3*

Table 5.3 *Labour volume of self-employed* ($\times 10^3$) *by* BRANCH, SEX *and* EDUC; *breakdown from LFS, with breakdown by* BRANCH *from NA*

SEX / **EDUC**	**BRANCH** A+B	C+D+E	F	G+H+I	J+K	other	*Total*
Male							
Lower	35.72	15.37	31.66	81.18	12.45	10.45	*186.82*
Middle	18.15	10.99	22.92	49.14	4.12	6.98	*112.29*
Higher	3.16	8.08	5.06	28.21	43.92	38.13	*126.56*
Female							
Lower	19.07	4.6	1.65	63.17	7.52	15.39	*111.39*
Middle	6.25	2.5	0.64	22.61	1.8	13.82	*47.62*
Higher	2.73	3	1.1	21.45	15.6	32.26	*76.13*
Total	*85.07*	*44.54*	*63.02*	*265.75*	*85.42*	*117.02*	*660.82*
% from NA	8.71%	4.53%	6.66%	28.63%	34.04%	17.43%	100%
Adjust. NA	*57.53*	*29.95*	*44.02*	*189.22*	*224.93*	*115.16*	*660.82*

Data on "compensation" of employees are found in table 5.4. For compensation, the situation concerning availability of data is a bit more difficult then for labour volume. As for labour volume, (percentage) breakdown of compensation by BRANCH is still available in the NA 1997 : relative and absolute figures are in the last two rows of table 5.4. The European *Structure of Earnings Survey* (SES-1995) provides a breakdown of wages, which are only one component of compensation, by BRANCH, SEX and EDUC, but restricted to NACE C to K branch of industry. To complete table 5.4, i.e. to fill in the initially empty columns "A+B" and "other", and to transform wages into compensation, the following procedure is followed.

SES 1995 also gives the corresponding (restricted) breakdown of *average wage* (wage per employee). For each class of industry (C+D+E, F, G+H+I, J+K) we calculate the relative average wage, and an un-weighted average of these 4 relative distributions of wage per employee is then obtained. For classes "A+B" and "other" of BRANCH we find overall average compensation by dividing total compensation of employees in these classes (from NA, i.e. 12.0 and 1525.0, resp.) by the corresponding total labour volumes (as found from LFS's total labour volume and NA's percentage distribution of labour volume, i.e. $3178.3 \times 0.64\ \% = 20.21$ for "A+B" and $3178.3 \times 39.88\ \% = 1267.47$ for "other"; see table 5.2). The average distribution from the other 4 classes was applied to these two figures, in order to complete the cross-classification of average wages in BRANCH classes "C+D+E", "F", "G+H+I" and "J+K" and (imputed) average compensation in BRANCH classes "A+B" and "other" by all three classification variables. Finally, averages are transformed into (estimated) totals through multiplication with labour volume estimates as found in table 5.5 panel A (first part: estimated labour volume for employees). These are the data in table 5.4.

Calibration totals by BRANCH are in the last row of table 5.4. There are no calibration totals corresponding to SEX and EDUC for these data on labour compensation. Hence the marginal column in table 5.4 is not completed.

Table 5.4 *Compensation*[a] *of employees* ($\times 10^9$ BEF) *by* BRANCH, SEX *and* EDUC; *breakdown from LFS, with percentage breakdown by* BRANCH *from NA*

SEX / **EDUC**	**BRANCH** A+B	C+D+E	F	G+H+I	J+K	other	*Total*
Male							
Lower	3.79	236.82	70.26	195.92	55.98	192.29	-
Middle	3.18	189.40	45.10	118.73	33.87	97.92	-
Higher	1.10	166.30	17.20	94.85	161.31	403.14	-
Female							
Lower	0.99	52.85	2.95	76.01	34.85	159.10	-
Middle	0.73	30.64	1.01	43.98	20.64	104.26	-
Higher	0.11	42.30	2.47	48.03	67.38	434.02	-
Total	*9.90*	*718.30*	*138.98*	*577.52*	*374.02*	*1390.73*	-
% from NA	0.27%	24.73%	4.93%	21.01%	14.87%	34.19%	100%
Abs. (NA)	12.00	1103.00	220.00	937.00	663.00	1525.00	4460.00

[a] "Compensation" means: wage for BRANCH categories C-K, and (imputed) compensation for other BRANCH categories.

V.F.2 Preparing the data

Section III.C (sub-section III.C.3 in particular) points out how the cross-tabulations in tables 5.2-4 must be transformed in order to create the input files for g-CALIB-S. We first constructed a SPSS file, called CROSSTABS.sav, as follows. Each record corresponds to a cell in one of the three tables. There are 5 columns (variables), called SEX, EDUC, BRANCH, LABOUR and TABLE. The first three variables are numerical versions of the three classification variables in the tables (with obvious integer-valued coding). The variable TABLE contains the table number (1, 2 or 3, resp. for tables 5.2, 5.3 and 5.4). The variable LABOUR contains labour volumes, as in tables 5.2 and 5.3, or compensation, as in table 5.4. For example, the first record contains the data vector (1, 1, 1, 6.63, 1), and represents the first cell in table 5.2; similarly for the $36 \times 3 - 1 = 107$ other records. Hence, CROSSTABS.sav contains the above tables in appropriate column format.

A SPSS syntax program, TRANSFORM CROSSTABS.sps, is reproduced in appendix VII.D.1: it uses the macros in g-DESIGN to transform the cross-tabulation in column format (CROSSTABS.sav) into an appropriate survey data input file for g-CALIB-S. Notice that the variable TABLE becomes the calibration stratum variable STRATUM, and LABOUR becomes the initial weights variable WEIGHT. A case identification (CASE) is constructed from the variables TABLE, EDUC, SEX and BRANCH. The output file is called COLLAPSED_DATA.sav, and will serve as survey data input file for g-CALIB-S.

The calibration totals are stored in TOTALS.sav, which has a structure that is similar to that of COLLAPSED_DATA.sav. The values were copied from an Excel workbook (which was used for preliminary exploration of the data) into TOTALS.sav. There are missing values in this calibration totals file: the third record, corresponding to TABLE=3, or calibration STRATUM=3, contains no values for the calibration variables corresponding to SEX and EDUC, but only for the calibration variables corresponding to BRANCH. For the other two tables (or calibration strata) the records are complete. This peculiarity can easily be dealt with by g-CALIB-S : in one run tables 1 and 2 will be calibrated, and in another run table 3 is calibrated.

V.F.3 Application of generalised raking

We then applied the raking method to our data, i.e. a generalised calibration model with an exponential calibration function *F* (the "multiplicative" method). The appropriate model formula for the data on labour volume in tables 5.2 and 5.3 is 1 + BRANCH + SEX*EDUC (or 1 + BRANCH + SEX + EDUC + SEX.EDUC), and the model formula for the data on labour compensation is simply 1 + BRANCH. As explained in the previous section, this needs two separate runs of g-CALIB-S. We have instructed the calibration software to calculate the scale parameter, separately for each table (or calibration stratum), from the data, using the calibration variable X0 with constant value 1. In the first run, the input parameters for g-CALIB-S are set as follows:

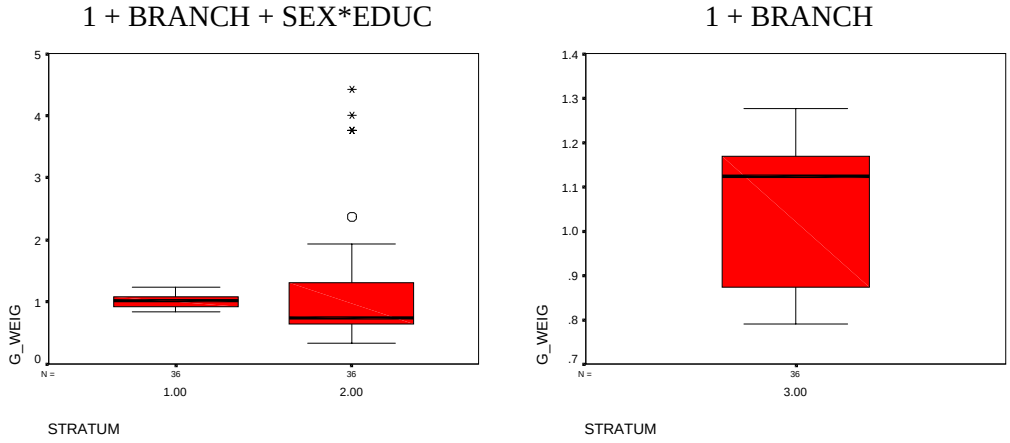
Parameter (macro name)	**Value**	**Comment**
@WORKDIR	C:\Actuaris_stage\Cases\Anja\	
@XDATA	Collapsed_Data.sav	
@CALTOT	Totals.sav	
@XVARS	X0, B1 to B6, SE11 to SE23	1 + BRANCH + SEX*EDUC
@STR_1	1	All 2 tables are estimated …
@STR_N	2	… with this model formula
@TYPE	2	i.e. Classical raking ratio
@SCALE	0	Scale from the data (X0)
@L	0.7	Not used since @TYPE = 2
@U	1.5	Not used since @TYPE = 2
@TOL	0.000001	
@ITERMAX	100	
@INFO	N	

In the second run, the input parameters for g-CALIB-S are set as follows:

Parameter (macro name)	**Value**	**Comment**
@WORKDIR	C:\Actuaris_stage\Cases\Anja\	
@XDATA	Collapsed_Data.sav	
@CALTOT	Totals.sav	
@XVARS	X0, B1 to B6	1 + BRANCH
@STR_1	3	Only table 3 is estimated …
@STR_N	3	… with this model formula
@TYPE	2	i.e. Classical raking ratio
@SCALE	0	Scale from the data (X0)
@L	0.7	Not used since @TYPE = 2
@U	1.5	Not used since @TYPE = 2
@TOL	0.000001	
@ITERMAX	100	
@INFO	N	

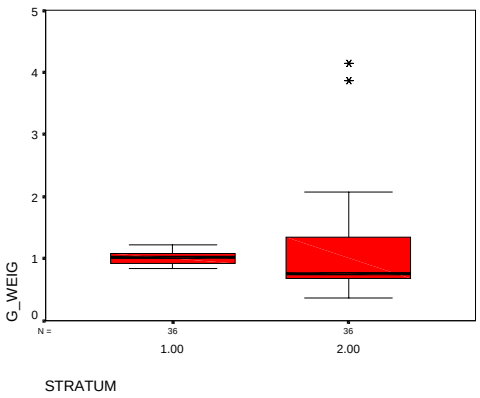
The estimated scale for each of the first two tables is 1.0, as expected; the estimated scale equals 136.27 for TABLE 3. The numbers of iterations were 4, 6 and 4, resp. The estimated g-weights are shown graphically in figure 5.10.

Fig 5.10 *Estimated g-weights for labour volume of employees ("stratum" 1) and of self-employed ("stratum" 2), in left panel, and for labour compensation of employees, in right panel; generalised raking method*; *different model formulae*



Next, for data on labour volume (strata 1 and 2), we applied the raking method with model formula 1 + BRANCH + SEX + EDUC, i.e. omitting the joint effect from SEX and EDUC. Applying this model is equivalent to classical raking in a 3-way cross-classification. The estimated g-weights for this slightly simplified model are in figure 5.11. Comparison of this graph with the one in the left panel of figure 5.10 may give some indication about the significance of the joint effect of SEX and EDUC on the estimated g-weights. Our software g-CALIB-S provides tables with summary statistics for the distributions of g-weights and calibrated weights within each calibration stratum, but these are not reproduced here. At first glance, there seems to be no significant joint effect from SEX and EDUC. We should perform formal (statistical) tests to draw firm conclusions with respect to significance of various terms in a calibration model formula. This, however, is not a topic of this study.

Fig 5.11 *Estimated g-weights for labour volume of employees ("stratum" 1) and of self-employed ("stratum" 2); generalised raking method; model formula*
1 + BRANCH + SEX + EDUC



V.F.4 Presentation of the results in cross-classification tables

g-CALIB-S delivers the results of calibration in standard output files; see sections IV.B.2.iii-iv. For the present application on labour volume and labour compensation we have written a SPSS syntax program to produce cross-tabulations of g-weights and calibrated weights. The latter are the adjusted labour volume and labour compensation statistics the user will be interested in. The tables have the usual format; we present these on the next pages, without changing the layout. The program, ESTIMATES.sps is reproduced in appendix VII.D.2, for illustrative purposes; the program was included in the job calling g-PREPARE.sps and g-CALIB-S.sps (section IV.B.5). ESTIMATES.sps demonstrates how g-CALIB-S's input and output files have to be merged to prepare output in a readable format.

Comparison of tables 5.5 panel A and 5.6 confirms that the joint effect of SEX and EDUC is very small for the data on labour volume ("strata" 1 and 2).

This application (of g-CALIB-S) was an unusual one: the study variable (here: labour volume/compensation) is involved in the calibration through the (initial) weights, and cannot be separated from the auxiliary variables (gender, education, branch of industry). Otherwise stated: generalised raking in cross-classifications can be solved by means of our software g-CALIB-S, but it does fit merely artificially into the generalised calibration framework based on (aggregated) individual observations. As illustrated here, the power of software like g-CALIB-S partly lies in the flexibility to change easily from one model (formula) to another.

Finally, it is noteworthy that percentage distributions were involved as calibration benchmarks. Our software does not (yet) allow working with relative distributions directly, so that absolute distributions are to be prepared beforehand by the user, and stored in the calibration totals input file. The SAS macro CALMAR can work directly with percentage distributions. This is related to the fact that the original variables are automatically transformed into indicator variables in CALMAR.

Table 5.5 **Panel A** *Estimated labour volume* ($\times 10^3$ BEF) *for employees ("stratum" 1), labour volume* ($\times 10^3$ BEF) *for self-employed ("stratum" 2); generalized raking method; model formula*: **1 + BRANCH + SEX*EDUC**

Estimated contingency table (CALWEI in table format)
STRATUM 1.00

					BRANCH 1.00 Sum	2.00 Sum	3.00 Sum	4.00 Sum	5.00 Sum	6.00 Sum	Table Total Sum
SEX	1.00	EDUC	1.00	CALWEI	8.06	246.95	101.84	248.88	61.23	200.64	867.59
			2.00	CALWEI	5.78	177.76	56.72	128.76	26.48	87.67	483.17
			3.00	CALWEI	1.26	98.49	12.83	63.55	95.19	229.26	500.58
	2.00	EDUC	1.00	CALWEI	3.18	80.72	4.64	137.88	66.82	254.35	547.58
			2.00	CALWEI	1.82	38.18	1.45	66.48	22.69	127.32	257.95
			3.00		.19	39.20	3.04	47.60	58.98	372.37	521.39
Table Total		CALWEI			20.28	681.30	180.52	693.15	331.39	1271.62	3178.26

Estimated contingency table (CALWEI in table format)
STRATUM 2.00

					BRANCH 1.00 Sum	2.00 Sum	3.00 Sum	4.00 Sum	5.00 Sum	6.00 Sum	Table Total Sum
SEX	1.00	EDUC	1.00	CALWEI	24.88	11.57	22.42	63.48	49.96	14.52	186.83
			2.00	CALWEI	13.95	9.12	17.91	42.39	18.24	10.70	112.30
			3.00	CALWEI	1.06	2.92	1.72	10.61	84.77	25.48	126.56
	2.00	EDUC	1.00	CALWEI	12.45	3.24	1.10	46.29	28.28	20.04	111.40
			2.00	CALWEI	4.08	1.76	.42	16.58	6.77	18.00	47.62
			3.00		1.12	1.33	.46	9.89	36.91	26.43	76.14
Table Total		CALWEI			57.54	29.95	44.03	189.23	224.94	115.16	660.85

Table 5.5 **Panel B** *Estimated labour compensation* ($\times 10^9$ BEF) *for employees ("stratum" 3); generalized raking method; model formula*: **1 + BRANCH**

Estimated contingency table (CALWEI in table format)
STRATUM 3.00

					BRANCH						Table Total
					1.00 Sum	2.00 Sum	3.00 Sum	4.00 Sum	5.00 Sum	6.00 Sum	Sum
SEX	1.00	EDUC	1.00	CALWEI	4.59	363.65	111.21	317.87	99.23	210.85	1107.41
			2.00	CALWEI	3.85	290.83	71.39	192.63	60.04	107.37	726.12
			3.00	CALWEI	1.33	255.36	27.22	153.89	285.94	442.06	1165.81
	2.00	EDUC	1.00	CALWEI	1.20	81.15	4.67	123.32	61.77	174.46	446.58
			2.00	CALWEI	.88	47.05	1.60	71.36	36.59	114.33	271.80
			3.00		.13	64.95	3.91	77.93	119.44	475.92	742.28
Table Total		CALWEI			12.00	1103.00	220.00	937.00	663.00	1525.00	4460.00

a STRATUM = 3.00

Table 5.6 *Estimated labour volume* ($\times 10^3$ BEF) *for employees ("stratum" 1), labour volume* ($\times 10^3$ BEF) *for self-employed ("stratum" 2); generalized raking method; model formula*: **1 + BRANCH + SEX + EDUC**

Estimated contingency table (CALWEI in table format)
STRATUM 1.00

					BRANCH 1.00 Sum	2.00 Sum	3.00 Sum	4.00 Sum	5.00 Sum	6.00 Sum	Table Total Sum
SEX	1.00	EDUC	1.00	CALWEI	8.06	246.94	101.96	248.74	61.13	200.50	867.34
			2.00	CALWEI	5.75	176.83	56.49	128.02	26.30	87.15	480.54
			3.00	CALWEI	1.27	99.11	12.93	63.92	95.66	230.57	503.46
	2.00	EDUC	1.00	CALWEI	3.18	80.81	4.65	137.95	66.79	254.46	547.83
			2.00	CALWEI	1.84	38.60	1.47	67.16	22.90	128.61	260.58
			3.00		.19	39.01	3.03	47.34	58.61	370.32	518.51
Table Total		CALWEI			20.28	681.30	180.52	693.15	331.39	1271.62	3178.26

Estimated contingency table (CALWEI in table format)
STRATUM 2.00

					BRANCH 1.00 Sum	2.00 Sum	3.00 Sum	4.00 Sum	5.00 Sum	6.00 Sum	Table Total Sum
SEX	1.00	EDUC	1.00	CALWEI	24.45	11.41	22.53	62.02	48.22	14.05	182.68
			2.00	CALWEI	13.32	8.75	17.49	40.26	17.11	10.06	107.00
			3.00	CALWEI	1.16	3.20	1.92	11.51	90.84	27.37	136.01
	2.00	EDUC	1.00	CALWEI	13.03	3.41	1.17	48.19	29.08	20.66	115.55
			2.00	CALWEI	4.58	1.99	.49	18.50	7.47	19.89	52.92
			3.00		1.00	1.19	.42	8.74	32.22	23.13	66.69
Table Total		CALWEI			57.54	29.95	44.03	189.23	224.94	115.16	660.85

V.G GROSSING-UP THE STRUCTURAL BUSINESS SURVEY (SBS) ON ENTERPRISES

V.G.1 The problem

The main purpose of the present section is to study the currently used extrapolation technique for the *Structural Business Survey* (SBS) on enterprises, carried out at Statistics Belgium since 1995. Our ultimate goal is to verify whether this technique can be treated as a special application of the generalised calibration framework. We'll see in section V.G.3 that this indeed is possible, which then opens the way to more sophisticated calibration methods and amelioration of the weighting scheme, as proposed in section V.G.4. Interestingly, this application deserves some special attention, due to the presence of over-coverage, as explained in sections V.G.3-4. Care should therefore be taken, especially when several calibration models are being compared.

We have chosen the 1998 SBS to work out theoretical aspects, as well as to illustrate our findings numerically. However, numerical results as such should not be used, or compared with published figures. So we have reduced the presentation of statistics on SBS variables in this text to an absolute minimum.

The central statistical unit in our study is the *enterprise*; see Communautés Européennes (1993).

Statistics Belgium has set up a database on business activity, wherein the legal unit is the basic entity. This database is called DBRIS, for *Banque de Données des Redevables de l'Information Statistique,* and is up-dated regularly using information from tax registers (i.e. VAT register, and IPCALART = *Impôt des Personnes Physiques – Calcul des Articles*) and information from the social security services (i.e. ONSS-RSZ or *Office National de la Sécurité Social – Rijksdienst voor Sociale Zekerheid*, and INASTI-RSVZ or *Institut National d'Assurances Sociales des Travailleurs Indépendants - Rijksdienst voor Sociale Verzekering van Zelfstandigen*). Thus, DBRIS is an *integrated database,* based on several external databases. The SBS too, although being a sample-based survey, is used to correct and extend information in DBRIS, whenever this is considered necessary. From DBRIS is constructed yearly a *population of enterprises,* which is used as the *sampling frame* for the SBS. This is a complex matter, and important to understand the specificities of the SBS, but the description of it is beyond the scope of this text.

The sampling frame is definitely not the *target population* in the SBS. Under- as well as over-coverage is present, and appropriate means are to be taken to correct for the resulting non-sampling errors. We'll see that this implies a quite special supplementary computational step, to be carried out before generalised calibration can be applied.

I had to work on two tables, one for the sampling frame (called UNIVERS98), and one for the sample (called ECHANTILLON), stored in an MS Access database. Only a few variables, relevant for the present study, were selected for these tables from the original data files. An overview is given in table 5.7. The third column gives the new name I have assigned to the variables in SPSS syntax programs and data files.

Table 5.7 *Variables used in this study, and included in the MS Access tables*

Field in MS Access tables	Description	New SPSS name
ECHANTILLON		
NBR_ENT_TOTAL	Number of enterprises in the stratum (*h*) to which the enterprise (*k*) belongs, from the sampling frame (i.e. N_h for all $k \in F_h$; see elsewhere for notations)	NBR_FRAM
NBR_ENT_ECHAN	Number of enterprises in the stratum (*h*) to which the enterprise (*k*) belongs, from the sample (i.e. n_h for all $k \in s_h$)	NBR_SAMP
POIDS_ENT	The final weight (w_k), as currently calculated	POIDS_EN
NACEBEL	NACE 4 code (4 digits)	NACE4
CLASSE_III	Classification by TO and ONSS size class (see tables 6.* and 6.$)	CLSS_III
STATUT_SUIVI	4-digitcode containing information about response / non-response, activity, why activity stopped, … on sampled units	STAT_SUI
UNIVERS98		
NACEBEL_REGROUPE	4 or 5-digit codes for NACE category	NACE_GR
CLASSE_IMPORTANCE	ONSS size classification	CLSS_IMP
CA_DECL_TVA	Turnover, from tax register	
REVENU_INASTI	Turnover, from IPCALART, as reported to INASTI	REV_INAS
CLASSE_III	Classification by TO and ONSS size class	CLSS_III

A variable TO is derived from CA_TVA and REV_INAS, and measures *turnover* of the enterprise. TO is used to construct one of the (post-) stratification variables.

CLSS_IMP is a classification variable, based on the size of the enterprise, measured as the *number of salaried employees*. It is a classification as used by ONSS. Table 5.8 presents the definition of CLSS_IMP.

From TO (turnover) and CLSS_IMP (ONSS size classification) is constructed the variable CLSS_III, as given in table 5.9. CLSS_III is a (post-) stratification variable.

The second (post-) stratification variable is NACE4, which, in the sample, is derived from NACE_GR. The values of NACE4 are not presented here.

Table 5.8 *ONSS size classification* CLSS_IMP

CLSS_IMP	Number of salaried employees
0	Enterprise is not in the ONSS data base (there are no salaried employees)
1	1 – 4
2	5 – 9
3	10 – 19
4	20 – 49
5	50 – 99
6	100 – 199
7	200 – 499
8	500 – 999
9	1000 - ∞

Table 5.9 *The variable* CLSS_III, *as constructed from* TO *and* CLSS_IMP

CLSS _III	TO (MIO Bef)				
CLSS_IMP	< 20	20 - 50	50 - 100	100 - 200	200 +
0	**0**	**1**	**2**	**3**	**4**
1	**1**	**1**	**2**	**3**	**4**
2	**2**	**2**	**2**	**3**	**4**
3	**3**	**3**	**3**	**3**	**4**
4	**4**	**4**	**4**	**4**	**4**
5	**5**	**5**	**5**	**5**	**5**
6	**5**	**5**	**5**	**5**	**5**
7	**5**	**5**	**5**	**5**	**5**
8	**5**	**5**	**5**	**5**	**5**
9	**5**	**5**	**5**	**5**	**5**

Based on the variable STAT_SUI, sampled enterprises are classified into 5 categories. The resulting variable is called NRStatus, since it has to do much with the status of the enterprise as respondent or non-respondent. NRStatus is described (not defined) in table 5.10. By “full activity” is meant that the enterprise existed during the entire survey year (1998).

Table 5.10 *Classification* NRStatus *of sampled enterprises*

NRStatus	Description
A	The sampled enterprise belongs to the *target* population, and is a respondent with full activity in the survey year
B	The sampled enterprise belongs to the *target* population, but is a non-respondent, since no or too bad information is available, although there was full activity in the survey year
C	The sampled enterprise does *not* belong to the *target* population, but did respond
D	The sampled enterprise could not be contacted (no information about activity in the survey year is available)
E	The sampled enterprise had incomplete activity in the survey year (and belongs to the *target* population), but did not respond

This classification is splitting up the sample into five (mutually exclusive and exhaustive) sub-samples: $s = s^A \cup s^B \cup s^C \cup s^D \cup s^E$. The situation is schematically clarified by means of figure 5.12 on the next page.

V.G.2 The sampling design

Stratified simple random sampling (STR-SRS) is used for the SBS. The sampling strata are denoted *h* $(h = 1,...,H)$, and are the (non-empty) cells in a complete cross-classification of enterprises by the stratification variables CLSS_III and NACE4. This stratification can be considered in the target population *U* and in the sampling frame *F*, as well as in the initial sample, its 5 sub-samples, and the respondent sample. The table below introduces appropriate notation:

	Target population	Sampling frame	Initial sample	Respondent sample
Stratum *h*	U_h	F_h	s_h	r_h
Stratum size	N_h^U	N_h	n_h	m_h
Union and total	U and N^U	F and N	s and n	r and m

Notice that the *respondent sample r* is simply the sub-sample s^A.

If sub-samples, such as s^A, are restricted to a particular stratum *h*, then we write s_h^A. To simplify notation we write s_h^{AB} for $s_h^A \cup s_h^B$, etc.

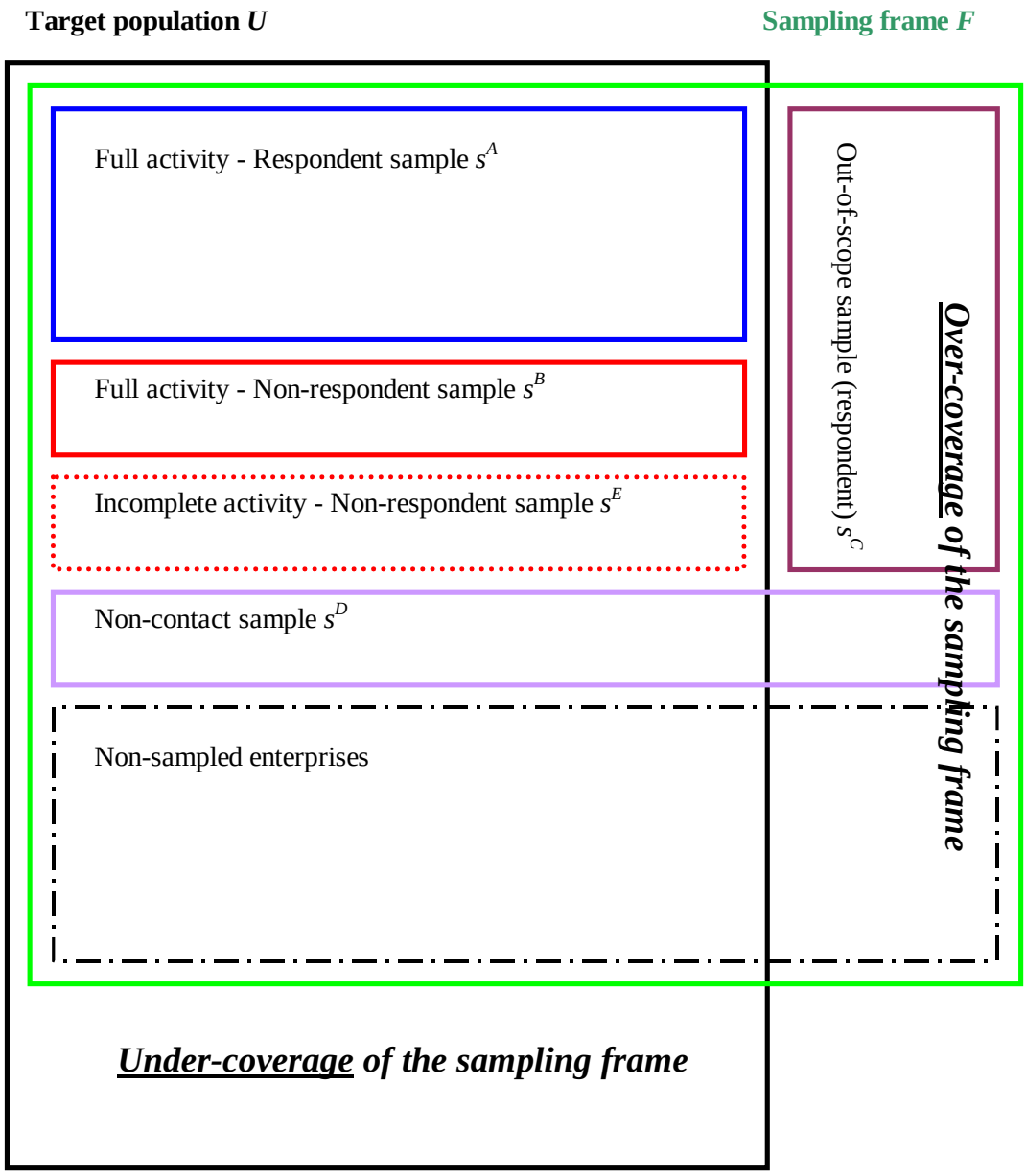
For STR-SRS, it is easy to derive the first- and second-order inclusion probabilities (based on the sampling frame counts!):

$$\pi_k = \frac{n_h}{N_h} \quad \text{for} \quad k \in F_h \qquad \text{and} \qquad \pi_{kl} = \frac{n_h}{N_h}\frac{n_{h'}}{N_{h'}} \quad \text{for} \quad k \in F_h, l \in F_{h'}, h \neq h'$$

$$= \frac{n_h}{N_h}\frac{n_h - 1}{N_h - 1} \quad \text{for} \quad k, l \in F_h$$

Therefore, the *sampling weights* are: $d_k = \pi_k^{-1} = \frac{N_h}{n_h}$ for $k \in F_h$. The simple formula for the second-order inclusion probabilities is useful to calculate variance estimates, as discussed in section III.G; see also section IV.C.2.vii.

Fig 5.12 *The relationship between the SBS sample s, with its 5 sub-samples, and the target population U and sampling frame F*



Initial sample sizes are calculated from sampling frame sizes and sampling fractions f_h as follows: $n_h = N_h \times f_h$. Some sampling fractions, for large enterprises (i.e. in CLSS_III categories 4 and 5) are set equal to one, such that sampling is *exhaustive* in sampling strata that contain large enterprises. Other sampling fractions are between 1/100 and ½, depending on the size of the strata and the

variability of turnover in the strata. Notice that the final values for the sample sizes are obtained by rounding calculated sizes $N_h \times f_h$.

We have retained N = 385,931 enterprises from the frame for our illustrative analysis. There should be n = 37,673 enterprises in the initial sample; there are finally m = 32,918 enterprises in the respondent sample $r = s^A$. Those enterprises are distributed over H = 2,011 strata, in the data files we are presently using.

V.G.3 Understanding current extrapolation practice

In many surveys, study variables are of different types, and the SBS is no exception. So we should first specify for which type of variables we want to get calibration estimates of totals. This is usually having an impact on the formulae for extrapolation. We restrict ourselves in this study to variables y such as turnover, various costs, etc; the relevance of this becomes clear when we explain the formula for current extrapolation. The total $t_y = \sum_{k \in U} y_k$ has to be estimated. Notice that we consider the total for the target population, not for the sampling frame. Only values y_k, observed for responding enterprises $k \in s^A$, can be used , so that grossing-up is from the respondent sample s^A to the target population U. The calibration estimator is: $\hat{t}_y = \sum_{k \in s^A} w_k y_k$, where w_k is the calibrated weight, to be estimated from a suitable calibration model. The form of the w_k will depend on the type of study variable chosen.

Up to now, calibration for the SBS is essentially traditional *complete post-stratification* (with an additional correction for over-coverage). The post-strata are simply equal to the sampling strata. Conditions *Cond* 1-3 in section III.B.1 are thus satisfied, so that the practical conclusions *Pc* 1-4 can be taken into account in our discussion of extrapolation practice for the SBS.

The post-strata are equal to the sampling strata. This has two major disadvantages: (1°) no estimates can be obtained if the sample doesn't contain responding enterprises; (2°) the resulting weights are likely to be unstable. One tries to avoid (1°) by making the SBS compulsory, but non-empty respondent samples within all post-strata can still not be guaranteed. Notice that, given that *Cond* 1-3 are satisfied, the post-stratified (or calibration estimator based on post-stratification) can be written as a sum of independent estimators within post-strata: $\hat{t}_y = \sum_h \hat{t}_{yh} = \sum_h \sum_{k \in s_h^A} w_k y_k$.

The formula used to calculate the weights w_k for enterprises k in the restriction $r_h = s_h^A$ of the respondent sample to the post-stratum h is:

$$w_k = \frac{N_h}{n_h^A} \frac{n_h^A + n_h^B + \frac{n_h^E}{2}}{n_h^{ABCE}} = d_k \frac{n_h}{n_h^A} \frac{n_h^A + n_h^B + \frac{n_h^E}{2}}{n_h^{ABCE}}. \qquad \text{(V.3)}$$

The following reasoning can be built up to justify this formula.

Think first of the two-step grossing-up procedure in the simple situation where $r \subset s \subset F = U$, where s is a SRS from F and no coverage problems are met. Grossing up of the observed total $\sum_{k \in r} y_k$ (non-weighted because of SRS) is first from the respondent sample r to the initial sample s by

multiplication with the factor $\frac{n}{m}$ and next from the initial sample to the target population U by multiplication with $\frac{N}{n}$. Hence: $\hat{t}_y = \sum_{k \in r} y_k = \frac{N}{n}\frac{n}{m}\sum_{k \in r} y_k$. Notice the similarity with (V.3), but the latter formula includes an additional correction factor, which must be due to the complexity of the SBS situation. Indeed the additional factor is due to coverage problems, as we now explain step by step.

Step 0 Hypothesis: under-coverage exists, but has a small (ignorable) effect on estimates (negative bias on estimates of totals of non-negative variables). Hence we assume that the sampling frame F completely covers the target population U, or $U \subset F$. And no correction for under-coverage has to be included therefore.

Step 1 Calculate the (non-weighted) average of y over the sub-sample s_h^A: the result is $\bar{y}_h^A$. The total of y for s_h^A is $n_h^A \bar{y}_h^A$, simply the observed total.

Step 2 Hypothesis: within the population of enterprises with activity during the whole year, non-response is completely at random, i.e. $\bar{y}_h^A$ also applies to s_h^B. Hence the total for s_h^{AB} is estimated as $\left(n_h^A + n_h^B\right)\bar{y}_h^A$.

Step 3 Hypothesis: a reasonable estimate for the average of y for enterprises that have not been active during the whole survey year is $\bar{y}_h^A/2$. Hence the total for s_h^{ABE} is estimated as $\left(n_h^A + n_h^B\right)\bar{y}_h^A + n_h^E \bar{y}_h^A/2 = \left(n_h^A + n_h^B + \frac{n_h^E}{2}\right)\bar{y}_h^A$. Notice that s_h^{ABE} is the *contact sample* in the target population.

Step 4 Hypothesis: the *contact rate* among enterprises in the target population is equal to the contact rate among enterprises which are *out-of-scope*, and therefore equal to the contact rate among all enterprises, either inside or outside the target population. This contact rate (within the target population) is therefore estimated as $\frac{n_h - n_h^D}{n_h} = \frac{n_h^{ABCE}}{n_h}$. The reciprocal of this estimated contact rate is used to up-weight the total in step 3 to the set $s_h \cap U_h = s_h \setminus \left(F_h \setminus U_h\right)$, which is that part of the initial sample s_h that is contained in the target population. The resulting estimated total for $s_h \cap U_h$ thus equals $n_h \frac{n_h^A + n_h^B + \frac{n_h^E}{2}}{n_h^{ABCE}} \bar{y}_h^A$.

Step 5 Now, the inclusion probability for sampled elements in the target population is the same for those outside the target population, i.e. $\frac{n_h}{N_h}$. Therefore up-weighting the total for $s_h \cap U_h$ in step 4 to the total for the target population U_h results into $N_h \frac{n_h^A + n_h^B + \frac{n_h^E}{2}}{n_h^{ABCE}} \bar{y}_h^A$.

The nature of the study variable y has become important in step 3. It is clear that the reasoning cannot be applied to, for instance, the variable $y \equiv 1$, whose total to be estimated is the number N_h^U of enterprises in the target population U_h .

Steps 4 and 5 are quite subtle. It may be easier, and less confusing, to explain the up-weighting procedure in terms of $\frac{\tilde{n}_h^{ABCE}}{\tilde{n}_h}$ instead of $\frac{n_h^{ABCE}}{n_h}$, and $\frac{\tilde{N}_h}{\tilde{n}_h}$ instead of $\frac{N_h}{n_h}$, where the tilde (~) denotes restriction to the target population. However, $\tilde{n}_h$, for instance, cannot be determined, since elements in the non-contact sample s_h^D cannot be identified as being inside or outside the target population. In other words, we don't know the size of $s_h^D \cap U_h$. That's why we have to make the assumptions in steps 4 and 5, so that not the "tilded" sizes themselves, but the ratios of "tilded" sizes can be replaced by estimated ratios based on observable (sub-) sample (and sub-population) sizes.

The resulting estimator for the total t_{yh} of the variable *y* in stratum *h* can be written in several ways, in order to make the step by step up-weighting explicit, and finally in order to write it in a "generalised-calibration-like" form:

$$\begin{aligned}
\hat{t}_{yh} &= N_h \frac{n_h^A + n_h^B + \frac{n_h^E}{2}}{n_h^{ABCE}} \bar{y}_h^A \\
&= \frac{N_h}{n_h} \frac{n_h}{n_h^{ABCE}} \left(n_h^A + n_h^B + \frac{n_h^E}{2} \right) \bar{y}_h^A \\
&= \frac{N_h}{n_h} \frac{n_h}{n_h^A} \frac{n_h^A + n_h^B + \frac{n_h^E}{2}}{n_h^{ABCE}} \sum_{k \in s_h^A} y_k \\
&= \sum_{k \in s_h^A} d_k g_k^* y_k
\end{aligned} \tag{V.4}$$

The last expression justifies formula (V.3), if

$$g_k^* = \frac{n_h}{n_h^A} \frac{n_h^A + n_h^B + \frac{n_h^E}{2}}{n_h^{ABCE}}, \quad \text{for all } k \in s_h^A. \tag{V.5}$$

V.G.4 Toward generalised calibration

Now, consider another study variable *z*, and suppose that *z* is constant within post-stratum *h*, i.e. assume $z_k = 1$ for all $k \in s_h^A$. Assuming further that *z* is a variable of the same type as variable *y* in the previous section, we get:

$$\hat{t}_{zh} = \sum_{k \in s_h^A} d_k g_k^* z_k = \sum_{k \in s_h^A} d_k g_k^* = N_h \frac{n_h^A + n_h^B + \frac{n_h^E}{2}}{n_h^{ABCE}} \overset{(N)}{=} N_h^*. \tag{V.6}$$

It might happen that $z_k = 1$ for all $k \in s_h^A$, not just for a particular post-stratum *h*, but for all post-strata $h = 1, \ldots, H$. Then:

$$\hat{t}_z = \sum_{h=1}^{H} \hat{t}_{zh} = \sum_{h=1}^{H} N_h \frac{n_h^A + n_h^B + \frac{n_h^E}{2}}{n_h^{ABCE}} = \sum_{h=1}^{H} N_h^* \,. \qquad \text{(V.7)}$$

However, (V.6) and (V.7) should not be misunderstood as estimates for N_h^U and N_h, respectively, as argued already in the previous sub-section. The only purpose of introducing the constant variable *z* and setting its value equal to 1, is to move on to the generalised calibration model here below.

A possible correct interpretation follows from rewriting (V.6) as $\frac{N_h}{n_h^{ABCE}}\left(n_h^A + n_h^B + \frac{n_h^E}{2}\right)$, interpreting $n_h^A + n_h^B + \frac{n_h^E}{2}$ as a measure for total "relevant activity" in s_h^{ABCE}, which is then up-weighted to the sampling frame (in post-stratum *h*) through multiplication with the factor $\frac{N_h}{n_h^{ABCE}}$. By "relevant" we mean that activity of enterprises that are in the sampling frame but not in the target population is not considered, and that enterprises (in the target population) with incomplete activity are considered as enterprises with only half as much activity as enterprises with full activity. Hence $N_h^* = \frac{N_h}{n_h^{ABCE}}\left(n_h^A + n_h^B + \frac{n_h^E}{2}\right)$ will be interpreted as an estimate of total activity in the sampling frame (restricted to post-stratum *h*). It is important to remark that all this is justifiable only in the context of estimating study variables of the type as considered in the previous sub-section.

In the sequel we call the measures N_h^* the *adjusted (population) post-stratum sizes* (as adjustments of the sampling frame post-stratum sizes N_h, in the context of estimating study variables of the type as considered in the previous sub-section).

We now consider the following generalised calibration problem:

- The sub-sample of elements for which observations on *y* are available, and which can therefore be up-weighted, is the respondent sample s^A.
- Calibration variables are the post-stratum indicator variables: δ_h (*h* = 1, …, *H*), with values $\delta_{kh} = 1$ if $k \in s_h^A$ and 0 otherwise. (Such indicator variables resemble the variable *z* considered here before!) The calibration vector for enterprise *k* is thus $\mathbf{x}_k = (\delta_{k1},...,\delta_{kH})^T$.
- Calibration totals are the adjusted post-stratum sizes $N_h^* = N_h \frac{n_h^A + n_h^B + \frac{n_h^E}{2}}{n_h^{ABCE}}$ ($h = 1,...,H$).
- Initial weights are $d_k = \frac{N_h}{n_h}$ for $k \in s_h^A$, and for all *h* = 1, …, *H*.
- Then, one has to find g-weights g_k ($k \in s_h^A$), such that $\sum_{k \in s^A} d_k g_k \delta_{kh} = N_h^*$. We also add the additional constraint that the g-weights are constant within each post-stratum.

The solution to this generalised calibration problem is exactly as in (V.5). For the above-mentioned class of study variables *y* the target population total t_y can then be estimated as follows:

$$\hat{t}_y = \sum_{h=1}^{H} \hat{t}_{yh} = \sum_{h=1}^{H} \sum_{k \in s_h^A} d_k g_k^* y_k = \sum_{k \in s^A} d_k g_k^* y_k \ . \qquad \text{(V.8)}$$

This proves that the extrapolation technique currently used in the SBS at Statistics Belgium can be formulated as a generalised calibration problem. Notice that we have a very special calibration problem here: complete post-stratification, with post-strata that coincide with the sampling strata. No distance function had to be chosen, but if we want to use our software g-CALIB-S, then any distance function *G* (or calibration function *F*) may be used. The linear method is the most economical choice, since then no iteration is required.

V.G.5 Generalised calibration for the SBS in the future

It should now immediately be clear that other auxiliary information could be integrated into the calibration problem, i.e. that the current complete post-stratification problem can, quite straightforward, be extended to a generalised calibration problem. A more general form of the calibration vector, still incorporating the post-stratum indicator variables, for element k is $\left(1, \delta_{k1}, ..., \delta_{kH}, x_1, ..., x_p\right)^T$, with corresponding vector of calibration totals $\left(N^*, N_1^*, ..., N_H^*, t_1, ..., t_p\right)^T$, where $N^* = \sum_h N_h^*$, and known *x*-totals t_j (for the target population). The user should however have our discussion in the previous paragraphs in mind, and calculate carefully meaningful calibration totals: the type of study variables for which ultimately the total has to be estimated is of utmost importance.

Of course, if the calibration problem is not post-stratification anymore, then the sampling weights should be taken into account.

To demonstrate that calibration totals should be prepared carefully, just consider the above formula $N^* = \sum_h N_h^*$, for computing the calibration total corresponding to the constant calibration variable 1. One might suggest to calculate this total as $N \dfrac{n^A + n^B + \dfrac{n^E}{2}}{n^{ABCE}} = N^{**}$, similar to the way of computing the calibration benchmarks N_h^*. However, generally N^{**} will not be equal to N^*, so that, if N^{**} were used together with the N_h^*, consistency of the calibration equations would be lost. Equality $N^* = \sum_h N_h^*$ precisely implies consistency when N^* is used together with the N_h^*.

As seen before, complete post-stratification corresponds to the cross-tabulation of enterprises by variables NACE4 (which has a lot of categories!) and CLSS_III (with 6 categories). Alternative calibration models will then immediately be clear: calibration could be on the margins in the cross-classification (together with calibration on other *x*-totals), or on the same margins and on counts corresponding to regrouped cells in the table (together with calibration on other *x*-totals), etc. See section III.B.2 for a discussion of incomplete post-stratification techniques. Some of these models are applied now to the SBS data for survey year 1998.

In order to prepare for application of the complete post-stratification model NACE4*CLSS_III = 1 + NACE4 + CLSS_III + NACE4.CLSS_III, but at the same time also for lower-order models 1, 1 + NACE4 and 1 + CLSS_III (see section III.A.1 for interpretation of these model formulae), one should calculate all calibration benchmarks as sums of those for the cells (*h*) in the complete cross-classification by NACE4 and CLSS_III. This is to satisfy numerical consistency.

The choice of the distance function is completely up to the statistician, with the consequence that this choice will affect the g-weights.

All this shows that generalised calibration is a very efficient framework, by means of which alternative, and probably more sophisticated calibration techniques become fairly obvious. The result is that auxiliary information can be used in an optimal way; finally pointing to an efficient extrapolation scheme that provides stable calibration weights (and g-weights) for the SBS. An in-depth study of weighting schemes SBS based on generalised calibration methodology will probably be the topic of a forthcoming paper.

V.G.6 Some results

For numerical illustration, I have decided to select a small, manageable, part of the data. Only enterprises in NACE1 category 4 are selected. In our respondent sample s^A we then have only 3806 enterprises, classified within 21 NACE4 categories and 6 CLSS_III categories, as shown in table 5.11. Notice that some NACE4 $\times$ CLSS_III cells (post-strata) are empty in the respondent sample s^A.

It took a significant amount of time to prepare, from the Access tables discussed in section V.G.1, the input files for g-CALIB-S. The last part of the data transformation procedure only is interesting in the context of this study. This last part is mainly performed by the syntax program SBS_g-DESIGN.sps, which can be found in section VII.E.1. Preparation of the input file Ultim Resp (A) Sample.sav was in fact more time consuming then writing the program SBS_g-DESIGN.sps and doing (manually) the very last data preparations. The file Ultim Resp (A) Sample.sav could be considered a (standard, at least for the SBS) survey data file, from which preparatory work for creating the input files for g-CALIB-S would start. This survey data file contains enough information for each enterprise in a single record. Population information is included too, e.g. the number of enterprises in the sampling strata to which the enterprise considered belongs. Such information allows calculating sampling design parameters, such as the sampling fractions, sampling weights, etc. Moreover, based on survey variables it is possible to classify sampled enterprises immediately in one of the sub-samples s^A, … or s^E, so that in each record we can also include the numbers n_h^A, … and n_h^E, where h is the sampling (or post-) stratum to which the enterprise belongs. Given the presence of all this information, it is then finally possible to calculate the adjusted frame sizes, i.e. the calibration totals needed for our analysis.

The purpose of SBS_g-DESIGN.sps is to transform the "basic" data file Ultim Resp (A) Sample.sav into appropriate input files for g-CALIB-S. The reader is asked to study the program in appendix (section VII.E.1). It is included for illustration, and can be useful as a starting point in new applications. In particular, it shows how the macros for constructing the design matrix are used. The last part of the program (after the matrix module) will be more confusion at first sight. As indicated in comments in the syntax itself, a table containing the calibration totals will have to be constructed. This table, extended with marginal calibration totals, is reproduced here as table 5.12. It contains the calibration totals for the post-stratification model 1 + NACE4 + CLSS_III + NACE4. CLSS_III, as well as for all its sub-models (e.g. 1 + NACE4). Given the structure of the survey data input file SBS Survey data.sav, which is completely constructed by the syntax itself, it is then not too difficult to construct also the calibration totals input file SBS Cal-totals.sav. Some commands at the end of the syntax program are intended to be helpful in that respect. Notice that the calibration totals in table 5.12 satisfy numerical consistency.

Table 5.11 *Cross-classification of 3806 enterprises in* s^A, *by* NACE4 *and* CLSS_III

Numbering of NACE4 categories	**NACE4**	**CLSS_III** 0	1	2	3	4	5	*Total*
1	4010					31	2	*33*
2	4020		1	1	1	7	1	*11*
3	4030	1	1		1			*3*
4	4100			1		19	2	*22*
5	4511	21	21	16	26	29	6	*119*
6	4512	12	8	1	5	4		*30*
7	4521	62	159	106	122	466	218	*1133*
8	4522	13	62	27	28	42	7	*179*
9	4523	9	15	14	31	105	47	*221*
10	4524	2		1	2	12	7	*24*
11	4525	44	53	19	24	43	22	*205*
12	4531	68	89	33	33	105	40	*368*
13	4532	6	4	3	5	15	10	*43*
14	4533	54	116	43	43	103	24	*383*
15	4534	10	3	2	3	10	6	*34*
16	4541	12	42	15	7	22	1	*99*
17	4542	66	141	58	56	96	16	*433*
18	4543	34	35	11	16	32	2	*130*
19	4544	34	82	27	25	72	18	*258*
20	4545	22	9	5	6	12	3	*57*
21	4550	5	3	2	3	6	2	*21*
Total		*475*	*844*	*385*	*437*	*1231*	*434*	***3806***

Table 5.12 *Calibration totals, i.e. adjusted frame sizes, appropriate for applying the maximal model* NACE4 * CLSS_III

Numbering of NACE4 categories	**NACE4**	**CLSS_III** 0	1	2	3	4	5	*Total*
1	4010					31.0	2.0	33.0
2	4020		1.0	3.0	1.0	7.0	1.0	13.0
3	4030	2.0	2.0		2.0			6.0
4	4100			7.0		19.0	2.0	28.0
5	4511	748.0	393.0	109.0	59.9	29.0	6.0	1344.9
6	4512	32.0	19.0	4.0	7.0	4.0		66.0
7	4521	3685.6	3486.5	1136.3	554.4	473.0	219.0	9554.8
8	4522	1043.0	1327.8	245.0	114.0	44.0	7.0	2780.8
9	4523	466.8	297.8	126.0	139.0	107.0	49.0	1185.6
10	4524	29.5		18.0	13.0	12.0	7.0	79.5
11	4525	2952.1	1214.7	191.8	111.0	46.0	22.0	4537.6
12	4531	4194.4	1776.5	353.0	145.0	108.5	41.0	6618.4
13	4532	459.5	97.3	35.0	22.0	15.0	10.0	638.8
14	4533	3776.8	2394.5	451.4	182.0	104.0	25.0	6933.7
15	4534	779.0	105.0	35.0	14.0	10.0	6.0	949.0
16	4541	843.0	808.0	127.0	26.0	22.0	1.0	1827.0
17	4542	4215.3	2777.8	532.0	228.0	97.0	16.0	7866.1
18	4543	2030.9	790.9	144.0	66.0	34.0	2.0	3067.8
19	4544	2324.3	1596.0	249.7	116.0	74.0	18.0	4378.0
20	4545	1086.6	192.0	43.0	13.5	12.5	3.0	1350.6
21	4550	9.0	8.0	6.0	6.0	6.0	2.0	37.0
Total		28677.8	17287.6	3816.3	1819.8	1255.0	439.0	53295.5

We are then ready to apply g-CALIB-S. The complete post-stratification model can be applied by setting the parameters through the interface of g-CALIB-S as follows:

Parameter (macro name)	**Value**	**Comment**
@WORKDIR	C:\Actuaris_stage\Cases\JeanMarie\	
@XDATA	SBS Survey data.sav	
@CALTOT	SBS Cal-totals.sav	
@XVARS	X0, NACE4_01 to NACE4_21, CLAS_0, CLAS_1, CLAS_2, CLAS_3, CLAS_4, CLAS_4, N4CL001 to N4CL126	1 + NACE4 + CLSS_III + NACE4 * CLSS_III
@STR_1	1	There no calibration strata
@STR_N	1	(i.e. STRATUM ≡ 1)
@TYPE	1	Irrelevant for complete post-stratification
@SCALE	0	Estimated (from X0)
@L	0.7	Not used since @TYPE = 1
@U	1.5	Not used since @TYPE = 1
@TOL	0.000001	
@ITERMAX	100	
@INFO	N	

With @XVARS as in the table, there are 141 calibration variables; for 3806 cases (enterprises); this results into a large data matrix. But g-CALIB-S did not have a problem calculating the g-weights etc. The scale was calculated as $\phi = 1.245680120$, and the distribution of the g-weights is shown graphically in figure 5.13. (The graphs are automatically produced by g-CALIB-S.) The (few) extreme g-weights could be removed by application of a simpler model. The model without interaction effects between NACE4 and CLSS_III has been applied; the linear method was chosen. The results are displayed in figure 5.14. There are no negative weights, although the linear method was used, and no weights are to extreme. Hence there is no need to try another method, e.g. the multiplicative method to make weights positive. Nevertheless, we have applied the classical raking method, i.e. the multiplicative method with model formula 1 + NACE4 + CLSS_III, for illustration. The results for the raking model are displayed in figure 5.15. As expected, the results of the linear method and the multiplicative method, when the model formula is 1 + NACE4 + CLSS_III, are barely different.

Fig 5.13 *Distribution of g-weights and calibrated weights from application of the complete post-stratification model* NACE4 * CLSS_III

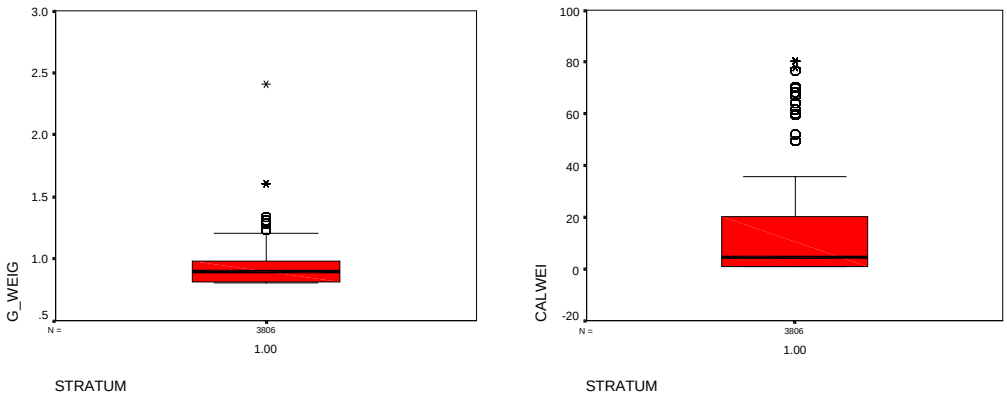


Fig 5.14 *Distribution of g-weights and calibrated weights from application of the LINEAR calibration method, with additive model formula* 1 + NACE4 + CLSS_III

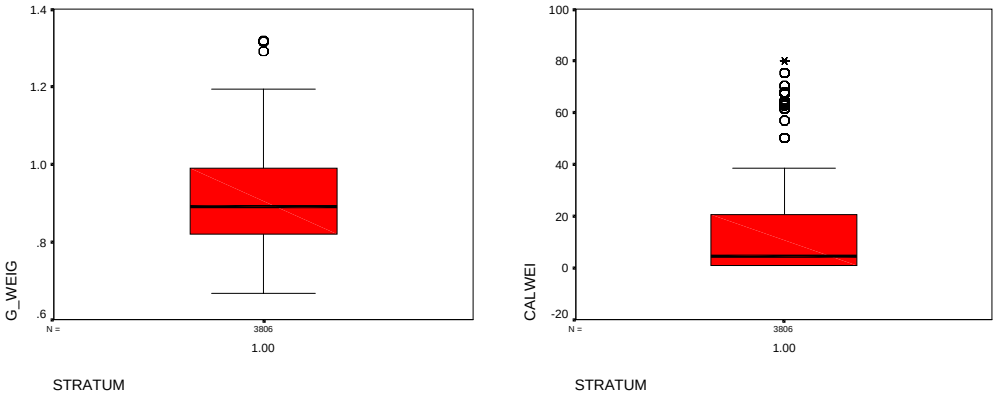
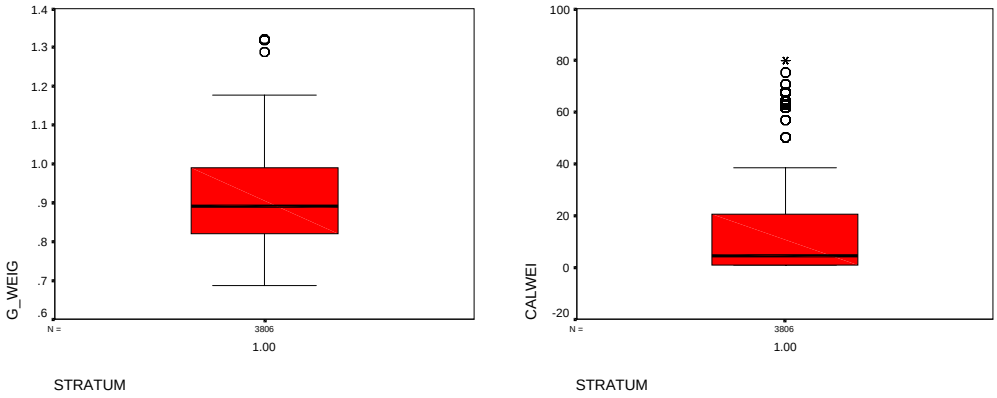


Fig 5.15 *Distribution of g-weights and calibrated weights from application of the MULTIPLICATIVE calibration method, with additive model formula* 1 + NACE4 + CLSS_III *(i.e. the raking method)*



In table 5.13 we give summary statistics for the scaled weights – which are, of course, independent of the calibration method –, the g-weights and the calibrated weights, for the three models that have been applied. (The table is constructed from tables that are produced by g-CALIB-S; only minor work on layout was necessary, in order to bring various tables together.) Clearly, the calibration method (here: linear versus multiplicative) has virtually no effect on the (right) tail(s) of the distribution of the g- and calibrated weights. Exclusion of interaction effects from the model formula tends to have only a moderate effect on the distributions, for the data treated in this application. The main conclusion is that there is no need at all to stick with the complete post-stratification model for these data; and only slightly more stability is obtained by dropping the interaction effects from the model. The classical raking method thus seems to be an excellent technique for getting extrapolation coefficients for the SBS data.

Table 5.13 *Summary statistics for scaled weights, g-weights and calibrated weights under three calibration models, for SBS 1998 data (restricted to NACE1 category 4)*

Statistic	*Method*	*Model formula*	*Min*	*Perc 25*	*Median*	*Mean*	*Perc 75*	*Max*	*Std Dev.*
SCAWEI			1.25	1.25	4.95	14.00	21.13	66.77	19.21
CALWEI	Linear	NACE4 * CLSS_III	1.00	1.02	4.39	14.00	19.96	80.23	19.83
	Linear	1 + NACE4 + CLSS_III	.83	1.02	4.42	14.00	20.48	79.78	19.74
	Multipl.	1 + NACE4 + CLSS_III	.86	1.02	4.42	14.00	20.48	79.95	19.74
G_WEIG	Linear	NACE4 * CLSS_III	.80	.81	.90	.91	.98	2.41	.11
	Linear	1 + NACE4 + CLSS_III	.67	.82	.89	.91	.99	1.32	.10
	Multipl.	1 + NACE4 + CLSS_III	.69	.82	.89	.91	.99	1.32	.10

Finally, notice that g-CALIB-S's various output files, containing estimated weights, can easily be merged, so that detailed graphical displays can be produced, allowing more detailed comparison of different weighting schemes. I believe it is worth searching for a bit more automation in producing such results. It would offer more flexibility – and more speed – in the process of studying different potential weighting methods. In other words, g-CALIB-S, together with such complementary tools, would provide a lot of support to the subject matter statistician who has to come up finally with a reliable weighting scheme, which is only possible after thorough evaluation and comparison of (many) possible alternatives.

Chapter VI

General comments

This work (on calibration) is not finished! I believe that a good start has been made in several ways, two of which are: understanding generalised calibration methodology, and setting up a powerful and flexible statistical tool and environment for calibration. However, calibration is only just one relatively small part of the entire survey process, and several other issues should be studied in a similar way.

There were several reasons for writing this document on generalised calibration, such as:

- Documenting current state-of-the-art in survey calibration and estimation methodology, and, at the same time, providing a sound starting point for an in-depth methodological study of each of our surveys at Statistics Belgium.
- Bringing practice-oriented issues in generalised calibration together in a single document, which hopefully will serve as a guide in that phase of the survey process that has to do with extrapolation and estimation.
- Providing a common framework for calibration, for all (or many?) kinds of surveys at Statistics Belgium; providing a common language and a common tool for all statisticians.
- Initiation to our software g-CALIB-S. Since many people currently work in completely different ways and with different tools, working on its implementation will automatically result into an extremely critical evaluation of the software and, more generally, the methodology.

Research in our statistical office should primarily be application-oriented, since our ultimate task is simply the production of figures. However, high-quality statistics necessitate a critical evaluation of all aspects over and over again. Appropriate statistical tools must therefore be introduced, such that the high demands for quality can be met in an optimal and contemporary way. We have to be prepared for step-by-step integration of more complex systems in our daily work, and this requires a good understanding of these tools and the methods behind them. So it will be clear that qualified people have to contribute in managing the implementation and use of such systems. This requires research and reflection!

It is hoped that subject matter specialist and methodologists will co-operate closely in the near future, and elaborate on the implementation of generalised calibration methodology in a uniform way. Experience so far indicates that this is not the easiest part, since, among other things, databases are not immediately designed with calibration in mind, or since preceding phases of the surveys are not automated in such a way that directly useable data files are available. Software, system and database designers too will play an important role in successful implementation of the methodology.

People outside our institute have shown their interest in these SPSS modules. Therefore, we decided to elaborate further on the development of these modules. We hope to be ready with a slightly extended version in the next few months. Our first goal is the implementation of a data management module that allows flexible integration of either individual-level or cluster-level auxiliary information, or both. Besides this, a user friendlier interface is envisaged too.

I believe that this text demonstrates the power of mathematical formalism and abstraction. As soon as a basic, but sound mathematical framework is defined or discovered, new results will follow more easily, and existing techniques are likely to be more easily recognized merely as special instances of the same general method or model. A formal mathematical language (of calibration methodology) turns out to be a very efficient way of communication, (un)fortunately only possible to be “spoken” in written documents. I was often seduced by the attractiveness, excitement and efficiency of mathematical formulation and derivation. This partly explains why the numerical illustrations are rather limited in number and in length and thoroughness.

This work might give indications of future developments in survey methodology – especially its implementation – at Statistics Belgium. At the end the entire survey process should be an integrated system, wherein each step takes into account what has been or what will be going on in preceding or in subsequent steps. While working on different practical applications, I was always impeded in producing results, due to inconsistencies between results in one step and things that should be done in the light of calibration. One seemingly trivial, but no less fundamental problem is that variables used in different steps are not always compatible with variables to be used in subsequent steps, or that variables used in preceding steps are often not easily understood. This is true for the variable's values, but also for their description. Consequently, we should strive working on (the same) basic files as much as possible, in each step of the survey process. In other words, the number of basic files should be reduced to an absolute minimum. Transformations of basic data files and databases, be they survey step specific or subject matter specific, should be archived as well-documented computer programs (SQL, SPSS syntax, …), rather than as supplementary data files.

Chapter VII

Appendices : SPSS syntax programs

VII.A SPSS® GENERALISED CALIBRATION MODULES

VII.A.1 The core modules g-PREPARE.sps and g-CALIB-S.sps

These syntax modules can be obtained from the author at the following address:

Dr. Camille VANDERHOEFT
Statistics Belgium
Rue de Louvain, 44
B-1000 Brussels
BELGIUM

e-mail: camille.vanderhoeft@statbel.mineco.fgov.be

The software is developed under SPSS® 9.0 for Windows Base. We've run it also, with success, under SPSS® 8.0 for Windows Base, extended with the Advanced Statistics module that provides the matrix language.

Write or mail to the above address for support too.

VII.A.2 The auxiliary module g-DESIGN.sps

```
************************************************************************.
* g_DESIGN.sps                                                         *.
*                                                                      *.
* Macros to construct the design matrix for a calibration model        *.
*                                                                      *.
* C. VANDERHOEFT & E. WAEYTENS  28/06/00                               *.
************************************************************************.
* Available macros :                                                   *.
*  DesC1, DesC2, DesC3, DesC1Z, DesC2Z, DesC3Z                         *.
************************************************************************.

*     DesC1 : creating indicator variable matrix for 1 categorical var.

define DesC1(var=!tokens(1) / des=!tokens(1) / lab=!tokens(1)).
compute !des = design(!var).
compute !lab = make(1,ncol(!des),0).
loop #J = 1 to ncol(!des).
compute !lab(1,#J) = mmax(!var &* !des(:,#J)).
end loop.
print ncol(!des) / title='Macro DesC1 executed; number of columns created :'.
!end define.

*     DesC2 : creating indicator variable matrix for 2 categorical vars.

define DesC2(var1=!tokens(1) / var2=!tokens(1)
           / des=!tokens(1) / lab=!tokens(1) / p=!tokens(1)).
compute #des1 = design(!var1).
compute #des2 = design(!var2).
compute !des = make(n,ncol(#des1)*ncol(#des2),0).
compute !lab = make(1,ncol(!des),0).
loop #I = 1 to nrow(!var1).
compute !des(#I,:) = kroneker(#des1(#I,:),#des2(#I,:)).
end loop.
```

```
loop #J = 1 to ncol(#des1).
loop #K = 1 to ncol(#des2).
compute #M = (#J-1)*ncol(#des2)+#K.
compute !lab(1,#M) = mmax(10**!p * !var1 &* #des1(:,#J)
                        +          !var2 &* #des2(:,#K)).
end loop.
end loop.
print ncol(!des) / title='Macro DesC2 executed; number of columns created :'.
!end define.

*     DesC3 : creating indicator variable matrix for 3 categorical vars.

define DesC3(var1=!tokens(1) / var2=!tokens(1) / var3=!tokens(1)
           / des=!tokens(1) / lab=!tokens(1) / p=!tokens(1)).
compute #des1 = design(!var1).
compute #des2 = design(!var2).
compute #des3 = design(!var3).
compute !des = make(n,ncol(#des1)*ncol(#des2)*ncol(#des3),0).
compute !lab = make(1,ncol(!des),0).
loop #I = 1 to nrow(!var1).
compute !des(#I,:) = kroneker(kroneker(#des1(#I,:),#des2(#I,:)),#des3(#I,:)).
end loop.
loop #J = 1 to ncol(#des1).
loop #K = 1 to ncol(#des2).
loop #L = 1 to ncol(#des3).
compute #C = (#J-1)*ncol(#des2)*ncol(#des3)
            +(#K-1)*ncol(#des3)
            + #L.
compute !lab(1,#C) = mmax(10**(2*!p)  * !var1 &* #des1(:,#J)
                        + 10**!p      * !var2 &* #des2(:,#K)
                        +               !var3 &* #des3(:,#L)).
end loop.
end loop.
end loop.
print ncol(!des) / title='Macro DesC3 executed; number of columns created :'.
!end define.
```

```
*     DesC1Z : splitting up a quantitative var over categories of 1 var.

define DesC1Z(var=!tokens(1) / zet=!tokens(1) / des=!tokens(1) / lab=!tokens(1)).
compute !des = design(!var).
compute !lab = make(1,ncol(!des),0).
loop #J = 1 to ncol(!des).
compute !lab(1,#J) = mmax(!var &* !des(:,#J) * 10).
compute !des(:,#J) = !des(:,#J) &* !zet.
end loop.
print ncol(!des) / title='Macro DesC1Z executed; number of columns created :'.
!end define.

*     DesC2Z : splitting up a quantitative var over cells of 2 vars.

define DesC2Z(var1=!tokens(1) / var2=!tokens(1) / zet=!tokens(1)
           / des=!tokens(1) / lab=!tokens(1) / p=!tokens(1)).
compute #des1 = design(!var1).
compute #des2 = design(!var2).
compute !des = make(n,ncol(#des1)*ncol(#des2),0).
compute !lab = make(1,ncol(!des),0).
loop #I = 1 to nrow(!var1).
compute !des(#I,:) = kroneker(#des1(#I,:),#des2(#I,:)).
end loop.
loop #J = 1 to ncol(#des1).
loop #K = 1 to ncol(#des2).
compute #M = (#J-1)*ncol(#des2)+#K.
compute !lab(1,#M) = mmax((10**!p * !var1 &* #des1(:,#J) + !var2 &* #des2(:,#K)) * 10).
compute !des(:,#M) = !des(:,#M) &* !zet.
end loop.
end loop.
print ncol(!des) / title='Macro DesC2Z executed; number of columns created :'.
!end define.

*     DesC3Z : splitting up a quantitative var over cells of 3 vars.
```

```
define DesC3Z(var1=!tokens(1) / var2=!tokens(1) / var3=!tokens(1) / zet=!tokens(1)
            / des=!tokens(1) / lab=!tokens(1) / p=!tokens(1)).
compute #des1 = design(!var1).
compute #des2 = design(!var2).
compute #des3 = design(!var3).
compute !des = make(n,ncol(#des1)*ncol(#des2)*ncol(#des3),0).
compute !lab = make(1,ncol(!des),0).
loop #I = 1 to nrow(!var1).
compute !des(#I,:) = kroneker(kroneker(#des1(#I,:),#des2(#I,:)),#des3(#I,:)).
end loop.
loop #J = 1 to ncol(#des1).
loop #K = 1 to ncol(#des2).
loop #L = 1 to ncol(#des3).
compute #C = (#J-1)*ncol(#des2)*ncol(#des3)
           +(#K-1)*ncol(#des3)
           + #L.
compute !lab(1,#C) = mmax(10**(2*!p)  * !var1 &* #des1(:,#J)
                        + 10**!p      * !var2 &* #des2(:,#K)
                        +               !var3 &* #des3(:,#L)).
compute !des(:,#C) = !des(:,#C) &* !zet.
end loop.
end loop.
end loop.
print ncol(!des) / title='Macro DesC3Z executed; number of columns created :'.
!end define.
```

VII.B SPSS SYNTAX FILES FOR THE HBS AND TUS

VII.B.1 Syntax to prepare for calibration of TUS at individual, household, or integrated individual and household level

```
************************************************************************.
* Prepare_TUSdata.sps                                                  *.
************************************************************************.
* This syntax program prepares TUS 1999 data for calibration, taking Phase I  *.
*  sampling weights (HBS 1999) into account.                           *.
*                                                                      *.
* Survey data are prepared for different types of calibration:         *.
*  - using individual auxiliary data (X,d,t) -> ind. g-weights         *.
*  - using individual auxiliary data (H~,d+,t) -> restricted ind. g-weights   *.
*  - using household auxiliary information (Z,d~,s) -> HH g-weights    *.
*  - using ind. and HH aux. information ((Z|H),d~,(s'|t')') -> new HH g-wghts *.
* (Calibration totals s and t are not created here.)                   *.
*                                                                      *.
* Two series (approximations) of sampling weights are available, but results  *.
*  should be very close, since the series are close.                   *.
*                                                                      *.
* The sampling weights can be ignored!                                 *.
************************************************************************.
* Input files:  TBS-extractie.sav     (TUS 1998 and 1999 individual data)     *.
*               SampleWeights.sav     (Phase I sampling weights HBS 1999)     *.
* Auxil. files: TUS-1999.sav          (interm.: reduced TBS-extractie.sav)    *.
*               SWeights-HBS99.sav    (red. SampleWeights., keyed and sorted) *.
*               Duplicates.sav        (intermediate)                          *.
*               TUS99- to Xd data.sav (ready to create design matrix)         *.
*               TUS99- to Zd data.sav (ready to create design matrix)         *.
*               TUS99-TEMP_H          (H,d~; erased)                          *.
* Output files: TUS99-BasicInd.sav    (BASIC DATA: Ind. and HH information)   *.
*               TUS99-Xd data.sav     (X,d   : for cal. at Ind. level)        *.
*               TUS99-Hd data.sav     (H~,d+ : clustering, restricted cal.)   *.
*               TUS99-Zd data.sav     (Z,d~  : for cal. at HH level)          *.
```

```
*               TUS99-Vd data.sav     (V,d~  : for cal. at Ind. + HH level)   *.
*************************************************************************.


*     Store current environment variables.
preserve.
*     Reset environment variables.
set mxmemory = 500000.
set workspace = 200000.
set mxloops = 35000.        /* Should be at least the number of observations.
*     Load the macros for creation of design matrix.
INCLUDE FILE = 'C:\803-bmcnew\Calibration\g_DESIGN\g-DESIGN.SPS'.


*$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  SECTION 1  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*.


*************************************************************************.
* Start preparing basic individual auxiliary data TUS99-BasicInd.sav          *.
*************************************************************************.

*   Explore TUS data file.

GET FILE "C:\Actuaris_stage\Cases\TUS\TBS-extractie.sav".

*** Year of interview (weekday and weekendday).

STRING year1 year2 (A4).
COMPUTE year1 = substr(agdatse,1,4).
COMPUTE year2 = substr(agdatwe,1,4).
FREQUENCIES VARIABLES=year1 year2.

COMPUTE yrequal=0.
IF (year1 = year2) yrequal=1.
VARIABLE LABLES yrequal 'Are interview years equal?'.
FREQUENCIES VARIABLES=yrequal.
```

```
*** Eliminate if one inteview day felt in 1998.

SELECT IF (year1<>'1998' AND year2<>'1998').
CROSSTABS
  /TABLES=year1  BY year2
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT .

*** Check interview months.

STRING month1 month2 (A2).
COMPUTE month1 = substr(agdatse,6,2).
COMPUTE month2 = substr(agdatwe,6,2).
FREQUENCIES VARIABLES=month1 month2.
CROSSTABS
  /TABLES=month1  BY month2
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT .


*   Prepare TUS 1999 respondent file for merging and save (renamed).

SORT CASES BY men (A).
SAVE OUTFILE="C:\Actuaris_stage\Cases\TUS\TUS-1999.sav" /COMPRESSED.


*   Prepare file with sampling weights: remove HHs with duplicate key (men).

GET FILE="C:\Actuaris_stage\Cases\HBS\SampleWeights.sav".

*** Construct HH number, as in TUS data file (of respondents).

COMPUTE men = TRUNC(HBEno/1000).
FORMATS men (F6).
SORT CASES BY men (A).
```

```
SAVE OUTFILE="C:\Actuaris_stage\Cases\HBS\SWeights-HBS99.sav"
    /COMPRESSED.

*** Identify duplicates in key variable 'men'.

COMPUTE duplic1 = (men=LAG(men,1)).
VALUE LABELS duplic1 1 'Duplicate key values in MEN (HH number)'.
FREQUENCIES VARIABLES=duplic1.

SELECT IF (duplic1=1).
SORT CASES BY men (A).
SAVE OUTFILE="C:\Actuaris_stage\Cases\HBS\Duplicates.sav"
    /KEEP=men duplic1 /COMPRESSED.

MATCH FILES /FILE="C:\Actuaris_stage\Cases\HBS\SWeights-HBS99.sav"
 /TABLE='C:\Actuaris_stage\Cases\HBS\Duplicates.sav'
 /BY men.
VALUE LABELS duplic1 1 'Excluded cases (HHs with duplicate key)'.
FREQUENCIES duplic1.

*** Delete cases with duplicate key MEN.

SELECT IF (sysmis(duplic1)).


*   Add sampling weights to TUS 1999 respondents.

MATCH FILES /FILE="C:\Actuaris_stage\Cases\TUS\TUS-1999.sav"
            /TABLE=*
            /BY men.

*** Eliminate HHs selected for HBS pilot survey (nov-dec 1998).

SELECT IF (hbeno >= 300000).
CROSSTABS
  /TABLES=year1  BY year2
```

```
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT .
FREQUENCIES VARIABLES=month1 month2.
CROSSTABS
  /TABLES=month1  BY month2
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT .

*** Save final TUS Basic file, with responding individuals.
*** Define overaal labels.

VALUE LABELS oplgraad 1 "LO" 2 "LMO" 3 "HMO/VS" 4 "HNUO" 5 "UNI".
SAVE OUTFILE="C:\Actuaris_stage\Cases\TUS\TUS99-BasicInd.sav"
    /DROP=agdatse,wcodeday,wd1 TO wd11,agdatwe,zcodeday,zd1 TO zd11,
          year1,year2,yrequal,month1,month2,sector,taalrol,duplic1
    /COMPRESSED.


*$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  SECTION 2  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*.


************************************************************************.
* Start preparing individual auxiliary data: TUS99- to Xd data.sav      *.
************************************************************************.

GET FILE="C:\Actuaris_stage\Cases\TUS\TUS99-BasicInd.sav".

*** Create final auxiliary variables and complete dictionary.

VARIABLE LABELS respnr "ID of Ind in HBS/TUS"
                region "Region of residence of Ind's HH"
                men    "HH ID in HBS/TUS (=cluster)"
                sexe   "Sex of Ind".

***** Month of participation (for HBS).
COMPUTE month = TRUNC(men/10000)-2.
```

```
VARIABLE LABELS month "Month of interview (for HBS)".
VALUE LABLES month 1 "Jan" 2 "Feb" 3 "Mar" 4 "Apr" 5 "May" 6 "Jun"
                   7 "Jul" 8 "Aug" 9 "Sep" 10 "Okt" 11 "Nov" 12 "Dec".

***** Age in 6 categories: AGE6.
RECODE age (LOWEST thru 29 = 1) (30 thru 39 = 2) (40 thru 49 = 3)
           (50 thru 59 = 4) (60 thru 69 = 5) (70 thru HIGHEST = 6) INTO age6.
VARIABLE LABELS age6 "Age of Ind in 6 ctgrs".
VALUE LABELS age6 1 "< 30" 2 "30-39" 3 "40-49" 4 "50-59" 5 "60-69" 6 "70 +".

***** Rename education variable.
RENAME VARIABLES (oplgraad = educ5).
VARIABLE LABELS educ5 "Highest degree of education of Ind".

*** Save what is strictly needed.

SAVE OUTFILE="C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav"
    /KEEP=respnr, region, month, men, sexe, age6, educ5, sw_i_s, sw_i
    /COMPRESSED.
SYSFILE INFO 'C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav'.

*** Statistics for the new file.

NEW FILE.
GET FILE="C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav".
FREQUENCIES VARIABLES=region,month,sexe,age6,educ5.
WEIGHT BY sw_i_s.
FREQUENCIES VARIABLES=region,month,sexe,age6,educ5.
WEIGHT OFF.

***** Matrix program to create a desired design matrix X, and
***** finally the survey data input file for g-CALIB-S.

NEW FILE.
MATRIX.
```

```
***** Read data from variables into vectors.

get case /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav' /variables=respnr.
get str1 /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav' /variables = region.
get str2 /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav' /variables = month.
get clus /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav' /variables = men.
get sex /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav' /variables = sexe.
get age /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav' /variables = age6.
* 11 Missing values of EDUC5 are classified as LO (=1).
* They could also be ignored completely.
get edu /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav'
        /variables = educ5 /missing = ACCEPT /sysmis = 1.
get wei1 /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav' /variables = sw_i_s.
get wei2 /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Xd data.sav' /variables = sw_i.

compute N = nrow(case).
print N / title = 'Number of cases processed :'.

***** Compute stratum variable if combination of Region and Month,
***** or if other period than Month is used.

***** Prepare terms of maximal calibration model
*     1 + sex + age6 + educ5 + 3rd order interactions.
*     (region and/or month as calibration stratum variables).

compute X0 = make(N,1,1)                          /* Term: 1             */.
DesC1 var=sex des=XS lab=LabS                     /* Term: Sex           */.
DesC1 var=age des=XA lab=LabA                     /* Term: Age6          */.
DesC1 var=edu des=XE lab=LabE                     /* Term: Educ5         */.

***** Store design matrix, assigning appropriate names.
***** 'Default' STRATUM is Region, Month is included.
***** 'Default' WEIGHT is Sw_i_s, Sw_i is included.

save {case,str1,str2,clus,wei1,wei2,X0,XS,XA,XE}
     /outfile = 'C:\Actuaris_stage\Cases\TUS\TUS99-Xd data.sav' /variables =
```

```
       CASE,STRATUM,month,CLUSTER,WEIGHT,sw_i, X0, S1,S2, A1 to A6, E1 to E5.

***** End of matrix module.
END MATRIX.

NEW FILE.

*************************************************************************.
* End of preparing individual auxiliary data (X,d)                      *.
*************************************************************************.


*$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  SECTION 3  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*.


*************************************************************************.
* Start preparing individual auxiliary data (H~,d+): clustering         *.
*   The DESIGN matrix X must be fully specified first                   *.
*   This could be ommitted if clustering would be integrated in g-PREPARE     *.
*************************************************************************.

GET FILE="C:\Actuaris_stage\Cases\TUS\TUS99-Xd data.sav".
AGGREGATE
  /OUTFILE=* /BREAK=cluster
  /case = MIN(case) /stratum = MEAN(stratum) /month = MEAN(month)
  /weight = SUM(weight) /sw_i = SUM(sw_i)
  /X0, S1,S2, A1 to A6, E1 to E5 = MEAN(X0, S1,S2, A1 to A6, E1 to E5).
SAVE OUTFILE="C:\Actuaris_stage\Cases\TUS\TUS99-Hd data.sav" /COMPRESSED.

NEW FILE.

*************************************************************************.
* End of preparing individual auxiliary data (H~,d+)                    *.
*************************************************************************.
```

```
*$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  SECTION 4  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$*.


*********************************************************************************.
* Start preparing household auxiliary data (Z,d~)                                *.
*********************************************************************************.

*   Select RPs from basic individual data file.

GET FILE='C:\Actuaris_stage\Cases\TUS\TUS99-BasicInd.sav'.
SELECT IF (suite = 1).

*** Create final auxiliary variables and complete dictionary.

VARIABLE LABELS respnr "ID of RP in HBS/TUS"
                region "Region of residence of HH"
                men    "HH ID in HBS/TUS".

***** Month of participation (for HBS).
COMPUTE month = TRUNC(men/10000)-2.
VARIABLE LABELS month "Month of interview (for HBS)".
VALUE LABLES month 1 "Jan" 2 "Feb" 3 "Mar" 4 "Apr" 5 "May" 6 "Jun"
                   7 "Jul" 8 "Aug" 9 "Sep" 10 "Okt" 11 "Nov" 12 "Dec".

***** Age in 6 categories: RPAGE6.
RECODE age (LOWEST thru 29 = 1) (30 thru 39 = 2) (40 thru 49 = 3)
           (50 thru 59 = 4) (60 thru 69 = 5) (70 thru HIGHEST = 6) INTO RPage6.
VARIABLE LABELS RPage6 "Age of RP in 6 ctgrs".
VALUE LABELS RPage6 1 "< 30" 2 "30-39" 3 "40-49" 4 "50-59" 5 "60-69" 6 "70 +".

***** Rename education variable.
RENAME VARIABLES (oplgraad = RPeduc5).
VARIABLE LABELS RPeduc5 "Highest degree of education of RP".

*** Save what is strictly needed.
```

```
SAVE OUTFILE="C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav"
    /KEEP=respnr, region, month, men, HHsize, RPage6, RPeduc5, sw_i_s, sw_i
    /COMPRESSED.
SYSFILE INFO 'C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav'.

*** Statistics for the new file.

NEW FILE.
GET FILE="C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav".
FREQUENCIES VARIABLES=region,month,hhsize,rpage6,rpeduc5.
WEIGHT BY sw_i_s.
FREQUENCIES VARIABLES=region,month,hhsize,rpage6,rpeduc5.
WEIGHT OFF.

***** Matrix program to create a desired design matrix Z, and
***** finally the survey data input file for g-CALIB-S.

NEW FILE.
MATRIX.

***** Read data from variables into vectors.

get case /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav' /variables=respnr.
get str1 /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav' /variables = region.
get str2 /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav' /variables = month.
get clus /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav' /variables = men.
get siz /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav' /variables = HHsize.
get age /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav' /variables = RPage6.
* 3 Missing values of RPEDUC5 are classified as LO (=1).
* They could also be ignored completely.
get edu /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav'
        /variables = rpeduc5 /missing = ACCEPT /sysmis = 1.
get wei1 /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav' /variables = sw_i_s.
get wei2 /file='C:\Actuaris_stage\Cases\TUS\TUS99- to Zd data.sav' /variables = sw_i.

compute N = nrow(case).
```

```
print N / title = 'Number of cases processed :'.

***** Compute stratum variable if combination of Region and Month,
***** or if other period than Month is used.

***** Prepare terms of maximal calibration model
*     1 + HHsize + RPage6 + RPeduc5 + 3rd order interactions.
*     (region and/or month as calibration stratum variables).

compute X0 = make(N,1,1)                          /* Term: 1             */.
DesC1 var=siz des=XS lab=LabS                     /* Term: HHSIZE        */.
DesC1 var=age des=XA lab=LabA                     /* Term: RPAge6        */.
DesC1 var=edu des=XE lab=LabE                     /* Term: RPEduc5       */.

***** Store design matrix, assigning appropriate names.
***** 'Default' STRATUM is Region, Month is included.
***** 'Default' WEIGHT is Sw_i_s, Sw_i is included.

save {case,str1,str2,clus,wei1,wei2,X0,XS,XA,XE}
     /outfile = 'C:\Actuaris_stage\Cases\TUS\TUS99-Zd data.sav' /variables =
       CASE,STRATUM,month,CLUSTER,WEIGHT,sw_i,
       ZX0, ZS1 to ZS5, ZA1 to ZA6, ZE1 to ZE5.

***** End of matrix module.
END MATRIX.

NEW FILE.

*************************************************************************.
* End of preparing individual auxiliary data (Z,d~)                      *.
*************************************************************************.


*$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$  SECTION 5  $$$$$$$$$$$$$$$$$$$$$$$$$$$$$*.
```

```
*******************************************************************************.
* Start preparing individual-household auxiliary data (V,d~)                  *.
*******************************************************************************.

*** Create data matrix H,d~ (NOT H~ !!).

GET FILE="C:\Actuaris_stage\Cases\TUS\TUS99-Xd data.sav".
AGGREGATE
  /OUTFILE=* /BREAK=cluster
  /case = MIN(case) /stratum = MEAN(stratum) /month = MEAN(month)
  /weight = MEAN(weight) /sw_i = MEAN(sw_i)
  /X0, S1,S2, A1 to A6, E1 to E5 = SUM(X0, S1,S2, A1 to A6, E1 to E5).
SAVE OUTFILE="C:\Actuaris_stage\Cases\TUS\TUS99-TEMP_H.sav" /COMPRESSED.

*** Merge individual with household data.

NEW FILE.
MATCH FILES
 /FILE='C:\Actuaris_stage\Cases\TUS\TUS99-Zd data.sav'
 /FILE="C:\Actuaris_stage\Cases\TUS\TUS99-TEMP_H.SAV"
   /RENAME  sw_i=drp1 weight=drp2 cluster=zclust month=zmonth stratum=zstrat
 /BY case
 /DROP=drp1 drp2.

*   Test validity of matches.

CROSSTABS
  /TABLES=stratum BY zstrat
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT.
CROSSTABS
  /TABLES=month BY stratum
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT.
CROSSTABS
  /TABLES=month BY zmonth
```

```
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT.
COMPUTE dif=0.
COMPUTE dif=cluster-zclust.
FREQUENCIES dif.

*   Save merged design matrix.

SAVE OUTFILE="C:\Actuaris_stage\Cases\TUS\TUS99-Vd data.SAV"
    /DROP zclust zmonth zstrat dif /COMPRESSED.

*************************************************************************.
* End of preparing individual-household auxiliary data (V,d~)           *.
*************************************************************************.

ERASE FILE="C:\Actuaris_stage\Cases\TUS\TUS99-TEMP_H.SAV".

NEW FILE.

*     Restore current environment variables.
restore.
```

VII.C SYNTAX FILES FOR THE TS

VII.C.1 From a basic individual data file to a household respondents file

```
************************************************************************.
* TS_PrepareRespondingHH.sps                                           *.
************************************************************************.
* Purpose: create HH respondent file                                   *.
*          = file from which survey data input file for g_CALIB-S can be*.
*            constructed                                               *.
************************************************************************.
* Input files : @INPDATA = initial TS sample of individuals            *.
*               @SDESIGN = sampling stratification information         *.
* Output file : @OUTDATA = responding HHs, with proper HH and RP's char's*.
* Working file(s) : AGG-HH.sav                                         *.
************************************************************************.
* Other program parameters (macros, set through Prod. Fac. job)        *.
*    @WORKDIR = directory where everything is stored                   *.
************************************************************************.

SET MXLOOP=100000.
SET MXMEMORY=200000.

DEFINE @message(!positional !tokens(1)).
print /title="***** MESSAGE *****".
print /title=!1.
!END DEFINE.

***** Here we go!!.

matrix.
@MESSAGE "The initial sample of persons is read ...".
@MESSAGE "Haven't you forgotten the position of STAT_RSP ?".
end matrix.
```

```
***** From ASCII file to SPSS data file.
*     Read ASCII file with complete intial sample of individuals.
*     Only variables needed for calibration are read.
*   ! Position for STAT_RSP depends on the TRIMESTER.

data list file=@WORKDIR + @INPDATA fixed records=1
  /1 TSHHNo  1- 8(F)   NISCOM 14-18(F)   AGE_CY   21-22(F)
     SEX    23-23(F)   EDUC   24-24(F)   PROF     25-26(F)
     NN_IND 29-39(F)   NN_RP  40-50(F)   STAT_RSP 52-52(A).

*     Add information to DD (of working file).

*     Check contents of working file.

matrix.
@MESSAGE "Inspect distribution of variables read from input file ...".
end matrix.
FREQUENCIES /VARIABLES=ALL /FORMAT=CONDENSE LIMIT(75).
matrix.
@MESSAGE "Persons w/o NN of RP cannot be assigned to a HH; hence exclude ...".
end matrix.

***** Construct HH file.

*  A. Proper HH characteristics:
*       - Size of HH, and response at HH level
*       - Residence
*       - Number of active members
*       - Age of youngest and oldest HH member
*     (stored in temporary file).

matrix.
@MESSAGE "Proper HH information is created ...".
end matrix.
NUMERIC s.isRP (F11) /s.active (F1) /HHresp (F2).
```

```
COMPUTE s.isRP = 0.
IF (nn_ind = nn_rp) s.isRP =1.
RECODE stat_rsp (" "=0) ("R"=0) ("Y"=1) INTO HHresp.
RECODE prof (1 thru 5=1) (6 thru 11=0) (ELSE=0)  INTO s.active.
AGGREGATE OUTFILE=@WORKDIR + 'AGG-HH.sav' /BREAK TShhno
  /HHsize   'Nbr of HH members (in sample file)' = N
  /n_HHresp 'Nbr of responding members in HH'    = SUM(HHresp)
  /is_RP    'RP identified (by NN)'              = MAX(s.isRP)
  /niscom   'NIS code of residence'              = MAX(niscom)
  /n_active 'Min nbr of active members in HH'    = SUM(s.active)
  /minage   'Age of youngest member'             = MIN(age_cy)
  /maxage   'Age of oldest member'               = MAX(age_cy).

*  B. HH characteristics from RP:
*       - Age, sex, educational level, and prof. status of RP
*       - NN of RP.

matrix.
@MESSAGE "RP's characteristics are prepared as HH information ...".
end matrix.
SELECT IF (s.isRP = 1).
RENAME VARIABLES (sex=RP_sex) (age_cy=RP_age) (educ=RP_educ)
                 (prof=RP_prof) (s.active=RP_act).
SORT CASES BY TShhno.

***** Merge RP characteristics (in *) with proper HH characteristics.

matrix.
@MESSAGE "Proper HH and RP information are joined together ...".
end matrix.
MATCH FILES FILE=* /TABLE=@WORKDIR + 'AGG-HH.sav' /BY TShhno.

*     Check contents of working file.

matrix.
@MESSAGE "Inspect distribution of characteristics in INITIAL HH sample ...".
```

```
end matrix.
FREQUENCIES /VARIABLES=ALL /FORMAT=CONDENSE LIMIT(75).

***** Select responding HHs.

SELECT IF (n_HHresp > 0).
matrix.
@MESSAGE "Responding HHs are selected ...".
end matrix.

***** Include SAMPLING INFORMATION in the HH respondent file.
matrix.
@MESSAGE "Sampling strat. information is included in HH RESPONDENT file ...".
@MESSAGE "Sampling parameters: from PSIZE and SSIZE ...".
@MESSAGE "An extra variable REGION is 'automatically' included this way".
end matrix.

*     Sampling stratification PROV : Province + Brussels (11).
COMPUTE Prov = TRUNC(niscom/10000).
IF (TRUNC(niscom/1000) = 21) Prov = 2.
IF (TRUNC(niscom/1000) = 23 OR TRUNC(niscom/1000) = 24) Prov = 10.
IF (TRUNC(niscom/1000) = 25) Prov = 11.
FORMATS Prov (F2).
VARIABLE LABELS Prov "Province - Brussels".
VALUE LABELS Prov
 1 "Antwerpen" 2 "Brus-Brux" 3 "West-Vlaanderen" 4 "Oost-Vlaanderen"
 5 "Hainaut" 6 "Liège" 7 "Limburg" 8 "Luxembourg" 9 "Namur"
 10 "Vlaams-Brabant" 11 "Brabant Wallon".

*     Sampling stratification AgeSize : RPage-HHsize classes (17).
RECODE RP_age (Lowest thru 24 = 1) (25 thru 39 = 2) (40 thru 54 = 3)
              (55 thru 64 = 4) (65 thru Highest = 5) INTO nagecl5.
IF (nagecl5 = 1 AND n_HHresp = 1) AgeSize = 1.
IF (nagecl5 = 1 AND n_HHresp > 1) AgeSize = 2.
IF (nagecl5 = 2 AND n_HHresp = 1) AgeSize = 3.
IF (nagecl5 = 2 AND n_HHresp = 2) AgeSize = 4.
```

```
IF (nagecl5 = 2 AND n_HHresp = 3) AgeSize = 5.
IF (nagecl5 = 2 AND n_HHresp > 3) AgeSize = 6.
IF (nagecl5 = 3 AND n_HHresp = 1) AgeSize = 7.
IF (nagecl5 = 3 AND n_HHresp = 2) AgeSize = 8.
IF (nagecl5 = 3 AND n_HHresp = 3) AgeSize = 9.
IF (nagecl5 = 3 AND n_HHresp > 3) AgeSize = 10.
IF (nagecl5 = 4 AND n_HHresp = 1) AgeSize = 11.
IF (nagecl5 = 4 AND n_HHresp = 2) AgeSize = 12.
IF (nagecl5 = 4 AND n_HHresp = 3) AgeSize = 13.
IF (nagecl5 = 4 AND n_HHresp > 3) AgeSize = 14.
IF (nagecl5 = 5 AND n_HHresp = 1) AgeSize = 15.
IF (nagecl5 = 5 AND n_HHresp = 2) AgeSize = 16.
IF (nagecl5 = 5 AND n_HHresp > 2) AgeSize = 17.
VARIABLE LABELS AgeSize 'RP-age - HH-size combination' .
VALUE LABELS AgeSize
  1 "0-24  1p"  2 "0-24  2+"
  3 "25-39 1p"  4 "25-39 2p"  5 "25-39 3p"  6 "25-39 4+"
  7 "40-54 1p"  8 "40-54 2p"  9 "40-54 3p" 10 "40-54 4+"
 11 "55-64 1p" 12 "55-64 2p" 13 "55-64 3p" 14 "55-64 4+"
 15 "65+   1p" 16 "65+   2p" 17 "65+   3+".

*     Merge HH RESPONDENT file with transformation table
*     and compute the sampling weights.
SORT CASES BY Prov (A) AgeSize (A).
MATCH FILES /FILE=* /TABLE=@WORKDIR + @SDESIGN /BY Prov AgeSize.
COMPUTE SampWei = Psize / Ssize.
VARIABLE LABELS Psize   "Nbr. of HHs in the HH's population stratum"
               /Ssize   "Nbr. of HHs in the HH's sample stratum"
               /SampWei "STR-SRS Sampling weight of the HH".
matrix.
@MESSAGE "The SAMPLING WEIGHTS have finally be computed and included ...".
end matrix.

*     Check contents of working file before saving.

matrix.
```

```
@MESSAGE "Inspect distribution of characteristics in HH RESPONDENT sample ...".
end matrix.
FREQUENCIES /VARIABLES=ALL /FORMAT=CONDENSE LIMIT(75).
SORT CASES BY prov.
SPLIT FILE SEPARATE BY prov.
TABLES
  /FORMAT BLANK MISSING('.')
  /OBSERVATION psize ssize sampwei
  /TABLES agesize BY (psize + ssize + sampwei) BY prov
  /STATISTICS mean( ) validn( ( NEQUAL5.0 )).
SPLIT FILE OFF.

***** Response analyis.

matrix.
@MESSAGE "More information on RESPONSE/NONRESPONSE can be produced ...".
@MESSAGE "The only thing I need to do this is TIME (not much!)".
end matrix.

***** Sort and save HH respondent sample.

matrix.
@MESSAGE "Saving the HH RESPONDENT sample ...".
@MESSAGE "The file is sorted on TShhno".
end matrix.
SORT CASES BY TShhno.
SAVE OUTFILE=@WORKDIR + @OUTDATA
  /DROP=s.isrp nagecl5 stratum /COMPRESSED.

***** Clean up.
ERASE FILE=@WORKDIR + 'AGG-HH.sav'.

*************************************************************************.
* End of creation of HH respondent file for TS                          *.
*************************************************************************.
```

VII.C.2 Creation of a design matrix for household-level calibration

```
************************************************************************.
* TS_MakeHHDesignMatrix.sps                                            *.
************************************************************************.
* Purpose: create design matrix for HH respondent file                 *.
*          = the survey data input file for g_CALIB-S                  *.
************************************************************************.
* Input file  : @OUTDATA = HH respondent sample                        *.
* Output files: @DESMAT  = design matrix for HH respondent sample      *.
*               TS-CalTotStructure.sav = aggregated version of @DESMAT *.
* Working file(s) :                                                    *.
************************************************************************.
* Other program parameters (macros)                                    *.
*    @WORKDIR = directory where everything is stored                   *.
************************************************************************.

SET MXLOOP=100000.
SET MXMEMORY=200000.

DEFINE @message(!positional !tokens(1)).
print /title="***** MESSAGE *****".
print /title=!1.
!END DEFINE.

***** Here we go!!.

matrix.
@MESSAGE "The HH respondent sample file is read ...".
end matrix.

GET FILE=@WORKDIR + @OUTDATA.

***** Choose and recode calibration variables (must be numeric).
```

```
matrix.
@MESSAGE "Prepare potential variables for calibration ...".
@MESSAGE "(Geographic, Socio-demographic, HH composition)".
end matrix.

***   Geografic variables : Arrond, Prov, Region.

*     ARROND : Arrondissement (43).
COMPUTE arrond = TRUNC(niscom/1000).
FORMATS arrond (F2).
VARIABLE WIDTH arrond(7).
VARIABLE LABELS arrond 'Arrondissement'.
VALUE LABELS arrond 21 'Brussel' 11 'Antwerpen' 12 'Mechelen' 13 'Turnhout'
 23 'Halle-Vilvoorde' 24 'Leuven' 71 'Hasselt' 72 'Maaseik' 73 'Tongeren'
 41 'Aalst' 42 'Dendermonde' 43 'Eeklo' 44 'Gent' 45 'Oudenaarde'
 46 'Sint-Niklaas' 31 'Brugge' 32 'Diksmuide' 33 'Ieper' 34 'Kortrijk'
 35 'Oostende' 36 'Roeselare' 37 'Tielt' 38 'Veurne' 25 'Nivelles'
 51 'Ath' 52 'Charleroi' 53 'Mons' 54 'Mouscron' 55 'Soignies' 56 'Thuin'
 57 'Tournai' 61 'Huy' 62 'Liège' 63 'Verviers' 64 'Waremme' 81 'Arlon'
 82 'Bastogne' 83 'Marche-en-Famenne' 84 'Neufchâteau' 85 'Virton'
 91 'Dinant' 92 'Namur' 93 'Philippeville'.

*     PROV : Province + Brussels (11).
*COMPUTE prov = TRUNC(niscom/10000).
*IF (arrond = 21) prov = 2.
*IF (arrond = 23 OR arrond = 24) prov = 10.
*IF (arrond = 25) prov = 11.
*FORMATS prov (F2).
*VARIABLE LABELS prov "10 Provinces + Brussels".
*VALUE LABELS prov
* 1 "Antwerpen" 2 "Brussel" 3 "West-Vlaanderen" 4 "Oost-Vlaanderen"
* 5 "Hainnaut" 6 "Liège" 7 "Limburg" 8 "Luxembourg" 9 "Namur"
* 10 "Vlaams-Brabant" 11 "Brabant Wallon".

*     REGION : Region (3).
*RECODE prov (2=1) (1=2) (7=2) (10=2) (3 thru 4=2) (ELSE=3) INTO region.
```

```
*FORMATS region (F1).
*VARIABLE LABELS region "Region".
*VALUE LABELS region 1 "Brus-Brux" 2 "Vlaanderen" 3 "Wallonie".

***   Socio-demographic variables : Agecat, Profcat, ProfAge.

RECODE rp_prof (1=2) (2=1) (3 thru 5=3) (6 thru HIGHEST=4) INTO profcat.
RECODE rp_age (LOWEST thru 24=1) (25 thru 39=2) (40 thru 54=3)
              (55 thru 64=4) (65 thru HIGHEST=5) INTO agecat.
NUMERIC profage (F1).
IF (profcat <= 3) profage = profcat.
IF (profcat = 4 AND agecat <= 4) profage = 4.
IF (profcat = 4 AND agecat  = 5) profage = 5.
VARIABLE LABELS profcat "Professional category of RP"
               /agecat  "Age class of RP"
               /profage "Combination Prof-Age (old post-strat. var.) (RP)".
VALUE LABELS
   profcat 1 "E+F" 2 "0" 3 "I" 4 "NR"
  /agecat  1 "<=24" 2 "25-39" 3 "40-54" 4 "55-64" 5 "65+"
  /profage 1 "E+F" 2 "0" 3 "I" 4 "NR <64" 5 "NR 65+".

***   HH composition variables : HHsize2, Active2, Adol.

RECODE HHsize (1=1) (ELSE = 2) INTO HHsize2.
RECODE n_active (0 = 1) (ELSE = 2) INTO Active2.
VARIABLE LABELS HHsize2 "HH size class"
                Active2 "Presence of active HH members".
VALUE LABELS
   HHsize5 1 "single" 2 "2+ pers"
  /Active2 1 "No" 2 "Yes".

***  Inspection of new variables.

FREQUENCIES VARIABLES=region arrond prov agecat profcat profage HHsize2 active2
  /FORMAT=CONDENSE.
```

```
matrix.
@MESSAGE "Find macros for creating design matrix ...".
end matrix.

INCLUDE FILE=@SOFTDIR + 'g-DESIGN.sps'.

matrix.
@MESSAGE "Start creation of design matrix ...".
end matrix.

matrix.

@MESSAGE "Design matrix will be stored with other potential cal. vars. ...".

get ID   /file=* /variables=TShhno.
get SW   /file=* /variables=SampWei.
get STR  /file=* /variables=region.
get ST2  /file=* /variables=prov.
get AGE  /file=* /variables=agecat.
get PRO  /file=* /variables=profcat.
get COM  /file=* /variables=profage.
get HHS  /file=* /variables=hhsize2.
get ACT  /file=* /variables=active2.

compute N = nrow(ID).
print N /title='Number of cases processed:'.

@MESSAGE "Preparing terms of calibration model ...".
compute X0 = make(N,1,1)                          /*  Term: 1              */.
DesC1 var=AGE des=XAge lab=Labage                 /*  Term: Agecat  (5)    */.
DesC1 var=PRO des=XPro lab=LabPro                 /*  Term: Profcat (4)    */.
DesC1 var=COM des=XCom lab=LabCom                 /*  Term: ComPrAg (6)    */.
DesC1 var=HHS des=XHHs lab=LabHHs                 /*  Term: HHsize  (2)    */.
DesC1 var=ACT des=XAct lab=LabAct                 /*  Term: Active  (2)    */.
DesC2 var1=AGE var2=PRO des=XAgPr lab=LabAgPr p=2 /*  Term: Agecat.Profcat */.
DesC2 var1=HHS var2=ACT des=XHsAc lab=LabHsAc p=2 /*  Term: HHsize.Active  */.
```

```
@MESSAGE "Sugg. max. mod. 1 : (1 + Agecat*Profcat + HHsize*Active) * STRATUM".
@MESSAGE "Sugg. max. mod. 2 : (1 + ComPrAg + HHsize*Active) * STRATUM".
@MESSAGE "STRATUM is set to Region; alternative stored as Prov".

@MESSAGE "Pay attention to order of labels (Later: calibration totals!!) ...".

print {Labage}.
print {LabPro}.
print {LabCom}.
print {LabHHs}.
print {LabAct}.
print {LabAgPr}.
print {LabHsAc}.

@MESSAGE "Original data, design matrix and other vars. are stored ...".

*     Info      Original vars           Terms in DM.
save {ID,SW,STR,ST2,AGE,PRO,COM,HHS,ACT,X0,XAge,XPro,XCom,XHHs,XAct,XAgPr,XHsAc}
     /outfile= @WORKDIR + @DESMAT /variables =
      CASE, WEIGHT, STRATUM,                        /* Fixed names           */
      prov,agecat,profcat,profage,hhsize5,active2,  /* Original variables    */
      x0, age1 to age5, pro1 to pro4, com1 to com5, /* Choose names for ...  */
      hhs1 to hhs2, act1 to act2,                   /* ... 0-1 columns in DM */
      ap1 to ap20, ha1 to ha4.

@MESSAGE "Columns of DM are summed OVER ALL STRATA ...".
compute X0    = csum(X0).
compute XAge  = csum(XAge).
compute XPro  = csum(XPro).
compute XCom  = csum(XCom).
compute XHHs  = csum(XHHs).
compute XAct  = csum(XAct).
compute XAgPr = csum(XAgPr).
compute XHsAc = csum(XHsAc).
```

```
@MESSAGE "Inspect labels and overall sample totals! ...".
print { 1,LabAge,LabPro,LabCom,LabHHs,LabAct,LabAgPr,LabHsAc;
       X0,XAge,  XPro,  XCom,  XHHs,  XAct,  XAgPr,  XHsAc}.

end matrix.

matrix.
@MESSAGE "Load file with calibration design matrix ...".
end matrix.
get file= @WORKDIR + @DESMAT.

matrix.
@MESSAGE "Aggregating by STRATUM; then inspect sample totals of cal.vars. ...".
end matrix.
AGGREGATE
  /OUTFILE=* /BREAK=stratum
  /x0, age1 to age5, pro1 to pro4, com1 to com5, hhs1 to hhs2,
   act1 to act2, ap1 to ap20, ha1 to ha4
   = SUM(x0, age1 to age5, pro1 to pro4, com1 to com5, hhs1 to hhs2,
         act1 to act2, ap1 to ap20, ha1 to ha4).

matrix.
@MESSAGE "From the aggregated survey input file can the calibration totals ...".
@MESSAGE "... file be easily created; this is saved as TS-CalTotStructure.sav".
end matrix.

SAVE OUTFILE=@WORKDIR + "TS-CalTotStructure.sav" /COMPRESSED.

matrix.
@MESSAGE "Finally, find calibrtion totals, and put them in TS-CalTotStructure.sav".
end matrix.
```

VII.D SYNTAX FILES FOR AN APPLICATION ON LABOUR VOLUME AND LABOUR COMPENSATION

VII.D.1 Transformation of cross-tabulated data on labour volume and labour compensation

```
************************************************************************.
* Transform crosstabs.sps                                              *.
*                                                                      *.
* C. VANDERHOEFT              November 2000                            *.
*                                                                      *.
* Data provided by ANJA TERMOTE, statistician                          *.
*                                                                      *.
************************************************************************.
* This program illustrates how to use the macros from g-DESIGN.sps for     *.
* constructing the calibration design matrix, and how to create the survey *.
* data file that is one of the input files for the calibration module      *.
* g-CALIB-S.sps.                                                           *.
*                                                                          *.
* The original data are a cross-tabulation of (weighted) totals of a       *.
* quantitative variable, stored in column format in the SPSS data file     *.
* C:\Actuaris_Stage\Cases\Anja\Crosstabs.sav                               *.
*                                                                          *.
* We recommend to sort this file by the variable 'Table' and by all other  *.
* qualitative calibration variables used later.                            *.
*                                                                          *.
****************************************************************************.

***** Set: @DRIVE   = drive letter (e.g. 'C:\') (for software and data)
*          @SOFTDIR = location of software
*          @WORKDIR = work directory
*          @INPDATA = name of input file
*          @DESMAT  = and name of output file.

DEFINE @DRIVE ()
  'C:\'.
```

```
!END DEFINE.

DEFINE @SOFTDIR ()
  '803-bmcnew\Calibration\g_DESIGN\'.
!END DEFINE.

DEFINE @WORKDIR ()
  'Actuaris_Stage\Cases\Anja\'.
!END DEFINE.

DEFINE @INPDATA ()
  'Crosstabs.sav'.
!END DEFINE.

DEFINE @DESMAT ()
  'Collapsed_Data.sav'.
!END DEFINE.

***** Load the macros.

INCLUDE FILE = @DRIVE + @SOFTDIR + 'g-DESIGN.SPS'.

***** Clear working data file.

NEW FILE.

***** Start matrix module.

matrix.

***** Read data from variables into vectors.
*     (Vector names are arbitrary).

get se /file=@DRIVE + @WORKDIR + @INPDATA /variables = sex.
get ed /file=@DRIVE + @WORKDIR + @INPDATA /variables = educ.
get br /file=@DRIVE + @WORKDIR + @INPDATA /variables = branch.
```

```
get cv /file=@DRIVE + @WORKDIR + @INPDATA /variables = lab_vol.
get st /file=@DRIVE + @WORKDIR + @INPDATA /variables = table.

compute N = nrow(st).
print N / title = 'Number of CELLS processed :'.

***** Prepare terms of maximal calibration model
*     1 + Branch + Sex*Educ = 1 + Sex + Educ + Branch + Sex*Educ.
*     It will allow to apply many alternative calibration models.

compute X0 = make(N,1,1)                     /*  Term: 1          */.
DesC1 var=se des=XS lab=LabS                 /*  Term: Sex        */.
DesC1 var=ed des=XE lab=LabE                 /*  Term: Educ       */.
DesC1 var=br des=XB lab=LabB                 /*  Term: Branch     */.
DesC2 var1=se var2=ed des=XSE lab=LabSE p=1  /*  Term: Sex*Educ   */.

***** Prepare remaining input variable(s) for the calibration module.

compute id = st*1000 + se*100 + ed*10 + br.

***** Print original variables, components of design matrix and labels.

print { 99,nrow(se),nrow(ed),nrow(br),99,nrow(st), 1,LabS,LabE,LabB,LabSE ;
        id,     se ,     ed ,     br ,cv,     st ,X0,  XS,  XE,  XB,  XSE }
      /formats F4.

***** Store original data and design matrix, assigning appropriate names.

save {  id,    cv,     st, se,  ed,    br,
        X0,   XS,      XE,               XB,                              XSE }
     /outfile = @DRIVE + @WORKDIR + @DESMAT /variables =
      CASE,WEIGHT,STRATUM,sex,educ,branch,
        X0,S1,S2,E1,E2,E3,B1,B2,B3,B4,B5,B6,SE11,SE12,SE13,SE21,SE22,SE23.

***** End of matrix module.
end matrix.
```

```
***** Load file with calibration design matrix (for inspection).
*     This is one of the input files for g-CALIB-S.

get file = @DRIVE + @WORKDIR + @DESMAT.
```

VII.D.2 Presentation of the results in cross-tabulations

```
************************************************************************.
* Estimates.sps                                                        *.
*                                                                      *.
* C. VANDERHOEFT              November 2000                            *.
*                                                                      *.
************************************************************************.

NEW FILE.
GET FILE=@WORKDIR + 'WEIGHTS.SAV'.
SORT CASES BY case.
SAVE OUTFILE=@WORKDIR + 'WEIGHTS.SAV' /COMPRESSED.

NEW FILE.
GET FILE=@WORKDIR + @XDATA.
SORT CASES BY case.

MATCH FILES /FILE=* /TABLE=@WORKDIR + 'Weights.sav'
 /RENAME (stratum = d0) /BY case /DROP= d0.

SELECT IF (~ SYSMIS(g_weig)).

SPLIT FILE SEPARAT BY stratum.

WEIGHT OFF.

TEMPORARY.
NUMERIC T0000000.
LEAVE T0000000.
VARIABLE LABEL T0000000 'Table Total'.
VALUE LABELS T0000000 0 ' '.
TABLES
  /FORMAT BLANK MISSING('.')
  /OBSERVATION calwei
```

```
  /TABLES (sex > (educ) + T0000000) > calwei
  BY (branch + T0000000) > (STATISTICS)
  BY (stratum +  T0000000 )
  /TITLE 'Estimated contingency table (CALWEI in table format)'
  /STATISTICS
  sum( ).

TEMPORARY.
TABLES
  /FORMAT BLANK MISSING('.')
  /OBSERVATION g_weig
  /TABLES (sex > (educ)) > g_weig
  BY branch > (STATISTICS)
  BY stratum
  /TITLE 'Estimated g-weights (G-WEIG in table format)'
  /STATISTICS
  sum( ).

NEW FILE.

EXECUTE.
```

VII.E SPSS SYNTAX FILES FOR APPLICATION ON SBS

VII.E.1 Syntax to prepare input files for g-CALIB-S, from a file containing survey and sampling frame data

```
************************************************************************.
* SBS_g-DESIGN.sps                                                     *.
*                                                                      *.
* C. VANDERHOEFT                December 2000                          *.
************************************************************************.
* This program illustrates how to use the macros from g-DESIGN.sps for *.
* constructing the calibration design matrix, and how to create the survey *.
* data file that is one of the input files for the calibration module  *.
* g-CALIB-S.sps.                                                       *.
************************************************************************.
* Input file  :                                                        *.
*    @INPDATA : Ultim Resp (A) Sample.sav  (Complete respondent sample sA) *.
* Output files :                                                       *.
*    @BASIC  : SBS-Sample 4.sav    (Final, possibly reduced sample)    *.
*    @DESMAT : SBS Survey data.sav (Survey data file, with design matrix)  *.
************************************************************************.

SET MXLOOP = 100000.
SET MXMEMORY = 200000.

***** Set: @SOFTDIR = location of software,
*          @WORKDIR = work directory,
*          @INPDATA = name of input file,
*          @DESMAT  = and name of output file.

DEFINE @SOFTDIR ()
  'C:\803-bmcnew\Calibration\g_DESIGN\'.
!END DEFINE.

DEFINE @WORKDIR ()
```

```
  'C:\Actuaris_Stage\Cases\JeanMarie\'.
!END DEFINE.

DEFINE @INPDATA ()
  'Ultim Resp (A) Sample.sav'.
!END DEFINE.

DEFINE @BASIC ()
  'SBS-Sample 4.sav'.
!END DEFINE.

DEFINE @DESMAT ()
  'SBS Survey data.sav'.
!END DEFINE.

DEFINE @ADJFRAM ()
  'SBS Cal-totals.sav'.
!END DEFINE.

***** Prepare data file: Select enterprises in NACE1 category 4.
*                       Number NACE4 categories.
*                       Compute enterprise identification ID_ENT.

NEW FILE.
GET FILE = @WORKDIR + @INPDATA.
SELECT IF nace1="4".                    /* Omit if complete sample is processed*/
SORT CASES BY nace4.
SAVE OUTFILE=@WORKDIR + @BASIC /COMPRESSED.
AGGREGATE /OUTFILE=* /BREAK=nace4 /N_BREAK=N.
COMPUTE no_nace4 = $CASENUM.
MATCH FILES /TABLE=* /FILE=@WORKDIR + @BASIC /BY nace4.
SORT CASES BY no_nace4 clss_iii.
COMPUTE id_ent = $casenum.
SAVE OUTFILE=@WORKDIR + @BASIC /COMPRESSED.

***** Load the macros for constructing the design matrix.
```

```
INCLUDE FILE = @SOFTDIR + 'g-DESIGN.SPS'.

***** Start matrix module.
matrix.

***** Read data from variables into vectors.
get ID /file=* /variables = id_ent.
get SW /file=* /variables = weight.        /* The sampling weights are ready */
get A  /file=* /variables = no_nace4.
get B  /file=* /variables = clss_iii.

compute N = nrow(ID).
print N / title = 'Number of cases processed :'.

***** Prepare terms of calibration model.
compute X0 = make(N,1,1)                          /*  Term: 1              */.
DesC1 var=A des=XA lab=LabA                       /*  Term: NACE4          */.
DesC1 var=B des=XB lab=LabB                       /*  Term: CLSS_III       */.
DesC2 var1=A var2=B des=XAB lab=LabAB p=2         /*  Term: NACE4.CLSS_III */.
compute STR = make(N,1,1).

***** Inspect labels, to assign names to calibration variables.
print {LabA} /title "NACE4 categories are met in the following order:".
print {LabB} /title "NACE4 categories are met in the following order:".
print {LabAB} /title "NACE4 categories are met in the following order:".


***** Store the design matrix, etc, assigning appropriate names. Inspection
*     of the NACE4 x CLSS_III table and the labels has suggested the ordering
*     of the calibration variables for the margins of CLSS_III. Similar
*     reordering is possible for the cell indicators, but for 126 variables,
*     this is a lot of unnecessary work.
save { ID, SW, STR, A, B, X0, XA, XB, XAB }
     /outfile = @WORKDIR + @DESMAT /variables = CASE, WEIGHT, STRATUM,
      NO_NACE4, CLSS_III, X0, NACE4_01 to NACE4_21,
```

```
                          CLAS_4, CLAS_5, CLAS_1, CLAS_2, CLAS_3, CLAS_0,
                          N4CL001 to N4CL126.

print /title="**** THE REMAINDER OF THIS PROGRAM IS TO HELP THE USER PREPARING".
print /title="**** THE CALIBRATION TOTALS FILE".
print /title="**** READ THE SYNTAX, OR OPEN LOGS IN THE VIEWER FOR INSTRUCTIONS".

end matrix.
***** End of matrix module.

***** Load file with calibration design matrix.
***** Aggregate by STRATUM, and inspect sample totals of cal.vars.

get file = @WORKDIR + @DESMAT.
AGGREGATE
  /OUTFILE='Agg_Sample.sav' /BREAK=stratum
  /nace4_01 TO nace4_21 = SUM(nace4_01 TO nace4_21)
  /clas_0 = SUM(clas_0) /clas_1 = SUM(clas_1) /clas_2 = SUM(clas_2)
  /clas_3 = SUM(clas_3) /clas_4 = SUM(clas_4) /clas_5 = SUM(clas_5)
  /n4cl001 TO n4cl126 = SUM(n4cl001 TO n4cl126).

*   IF SOME CAL.VARS. HAVE ZERO SAMPLE TOTAL, THEN DELETE THESE VARIABLES !!!.

***** Reload file with calibration design matrix, and merge with basic file.
***** The required population information is then available for calculation
***** of the "adjusted post-stratum population sizes", and for preparation
***** of the calibration totals file.
GET FILE = @WORKDIR + @DESMAT.
MATCH FILES /FILE=*
 /FILE=@WORKDIR + @BASIC
 /RENAME (clss_iii no_nace4 stratum weight = d0 d1 d2 d3)
 /DROP= d0 d1 d2 d3.
*   There is a perfect match if the variable MATCH has only the value zero.
COMPUTE match = case-id_ent.
FREQUENCIES /VARIABLES=match.
*   Calibration totals file: adjusted frame sizes.
```

```
*SELECT IF (no_nace4 = 9 and clss_iii=0).                /* For testing only */

TABLES
  /FORMAT BLANK MISSING('.')
  /OBSERVATION caltot
  /TABLES no_nace4 > caltot
  BY clss_iii > (STATISTICS)
  /STATISTICS
  mean( ).

***** ALL CALIBRATION TOTALS CAN BE DERIVED FROM THIS TABLE: IT IS
***** SUGGESTED TO COPY THE TABLE TO A SPREADSHEET, AND FINDING THE
***** MARGINAL TOTALS IN ORDER TO PRODUCE THE CALIBRATION TOTALS
***** CORRESPONDING TO THE TERMS NACE4 AND CLSS_III IN THE MODEL FORMULA.
***** UNFORTUNATELY, THE CELL ENTRIES ARE NOT IN THE 'RIGHT' ORDER,
***** i.e. NOT IN THE SAME ORDER AS THE CELL INDICATOR VARIABLES (N4CL###)
***** IN THE SURVEY DATA INPUT FILE.
***** THEREFORE ...
WEIGHT BY caltot.
AGGREGATE
  /OUTFILE=@WORKDIR + 'Aggr_Partial.sav' /BREAK=stratum
  /n4cl001 TO n4cl126 = SUM(n4cl001 TO n4cl126).

***** WE HAVE THE CELL BENCHMARKS, AFTER DIVIDING BY THE SAMPLE SIZES,
***** IN THE RIGHT ORDER NOW.
***** SOME MORE MANUAL WORK IS REQUIRED TO CONSTRUCT THE BENCHMARKS FILE.
***** WE ALREADY PREPARE THE STRUCTURE ...
GET FILE=@WORKDIR + 'Agg_Sample.sav'.
SAVE OUTFILE=@WORKDIR + @ADJFRAM /COMPRESSED.

***** JUST FILL IN NOW THE RIGHT NUMBERS.
***** DON'T FORGET TO DELETE VARIABLES FOR WHICH SAMPLE
***** TOTAL OR BENCHMARK IS ZERO.
```

References

Brackstone, G.J. (1999) Organisation d'un Service de Méthodes d'Enquêtes, IN: Brossier, G. et Dussaix A.-M. (ed.) *Enquêts et Songades – Methodes, Modèles, Applications, Nouvelles Approches*, Paris : Dunod, pp.118-134.

Barnett, V. (1991) *Sample Survey Principles & Methods*, London: Arnold.

Brickman, L. (1998) *Mathematical Introduction to Linear Programming and Game Theory*, New York: Springer-Verlag.

Cameron, N. (1985) *Introduction to Linear and Convex Programming*, Cambridge: Cambridge University Press.

Chambers, R.L., Skinner, C.J. and Wang, S. (1999) Intelligent Calibration, *Proceedings of the 52nd Session of the International Statistical Institute*, Tome LVIII, Book 2, pp.321-324.

Cochran, W.G. (1977) *Sampling Techniques*, 3rd ed., New York: Wiley.

Communautés Européennes (1993) Rêglement (CEE) N° 696/93 du Conseil du 15 mars 1993 relatif aux unités statistiques d'observation et d'analyse du système productif dans la Communauté. Dans : Journal officiel des Communautés Européennes N° L76 /1, 30 mars 1993.

Deming, W.E. and Stephan, F.F. (1940) On a Least Squares Adjustment of a Sampled Frequency Table when the Expected Marginal Totals are Known, *Annals of Mathematical Statistics*, Vol. 11, pp.427-444.

Deville, J.-C. (2000) Generalized Calibration and Application to Weighting for Non-response, IN: Bethlehem, J.G. and van der Heijden, P.G.M. (eds.) *Proceedings of the 14th Compstat Symposium in Computational Statistics*, Utrecht, Physica-Verlag.

Deville, J.-C. and Särndal, C.-E. (1992) Calibration Estimators in Survey Sampling, *Journal of the American Statistical Association*, Vol. 87, No. 418, pp.376-382.

Deville, J.-C., Särndal, C.-E. and Sautory, O. (1993) Generalized Raking Procedures in Survey Sampling, *JASA*, Vol. 88, No. 423, pp.1013-1020.

Dupont, F. (1994) Calibration Used as a Nonresponse Adjustment, IN: Diday, E. (ed.) *New Approaches in Classification and Data Analysis*, Springer Verlag, pp.539-548.

Dupont, F. (1995) Alternative Adjustments where there are Several Levels of Auxiliary Information, *Survey Methodology*, Vol. 21, No. 2, pp.125-135.

Elliot, D. (1999) Report of the Task Force on Weighting and Estimation, *GSS Methodological Series*, No. 16, London: Office for National Statistics.

Estevao, V., Hidiroglou, M.A.. and Särndal, C.-E. (1995) Methodological Principles for a Generalized Estimation System at Statistics Canada, *Journal of Official Statistics*, Vol. 11, No. 2, pp.181-204.

Francis, B., Green, M. and Payne, C. (1993) *The GLIM System – Release 4 Manual*, Oxford: Clarendon Press.

Hidiroglou, M.A. and Särndal, C.-E. (1998) Use of Auxiliary Information for Two-Phase Sampling, *Survey Methodology*, Vol. 24, No. 1, pp.11-20.

Kalton, G. and Brick, J.M. (1995) Weighting Schemes for Household Panel Surveys, *Survey Methodology*, Vol. 13, pp.199-207.

Lemaître, G. and Dufour, J. (1987) An Integrated Method for Weighting Persons and Families, *Survey Methodology*, Vol. 21, No. 1, pp.33-44.

Lindsey, J.K. (1997) *Applying Generalized Linear Models*, New-York: Springer-Verlag.

Lundström, S. and Särndal, C.-E. (1999) Calibration as a Standard Method for Treatment of Nonresponse, *JOS*, Vol. 15, No. 2, pp.305-327.

Mohadjer, L. (1999) Sample Weights for Households with Multiphase Data Collection Approaches, *Proceedings of the 52nd Session of the International Statistical Institute*, Tome LVIII, Book 2, pp.317-320.

Nieuwenbroek, N. (1997) General Regression Estimator in Bascula 3.0: Theoretical Background, *CBS Research Paper*, No. 9737.

Nieuwenbroek, N., Renssen R., Slootbeek, G. and Veugen, T. (1997) A General Weighting Package Including Estimates for Population Totals and Corresponding Variances: Extended Version, *CBS Research Paper*, No. 9745.

Rao, C.R. (1973) *Linear Statistical Inference and its Applications*, 2nd ed., London: Wiley.

Renssen, R.H. (1998) Use of Statistical Matching Techniques in Calibration Estimation, *Survey Methodology*, Vol. 24, No. 2, pp.171-183.

Skinner, C. (1999) Calibration Weighting and Non-Sampling Errors, *Research in Official Statistics*, No. 1, pp.33-43.

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

Sautory, O. (1993) La macro CALMAR: Redressement d'un Echantillon par Calage sur Marges, *Document de travail de la Direction des Statistiques Démographiques et Sociales*, no. F9310.

Sautory, O. *et al* (1999) Use of Auxiliary Information in Sampling Surveys, *Document de travail de la Direction des Statistiques Démographiques et Sociales*, TES Course DAT-202-E, *TES Intitute*, Luxembourg.

SPSS Inc. (1999a) *SPSS® Base 9.0 User's Guide*, SPSS Inc.

SPSS Inc. (1999b) *SPSS® Base 9.0 Syntax Reference Guide*, SPSS Inc.

SAS Institute Inc. (1990) *SAS/IML® Software; Usage and Reference*, Version 6, 1st ed., SAS Institute Inc.

Vanderhoeft, C., Waeytens, E. and Museux J.-M. (2000a) Generalised Calibration with SPSS® 9.0 for Windows Base. Paper presented at the *2ième Colloque Francophone sur les Sondages*, Brussels. To appear in the proceedings.

Vanderhoeft, C., Museux J.-M. and Waeytens, E. (2000b) g-DESIGN and g-CALIB-S : SPSS® modules for Generalised Calibration. *The Survey Statistician*, IASS Newsletter, No. 43.

Wilkinson, G.N. and Rogers, C.E. (1973) Symbolic Description of Factorial Models for Analysis of Variance, *Journal of the Royal Statistical Society, series C (Applied Statistics)*, Vol. 22, pp.181-191.