WORKING PAPER
METHODOLOGY BRANCH

How to avoid getting all tied up bootstrapping a survey: A walk-through featuring the National Longitudinal Survey of Children and Youth

HSMD-2007-001E

Claude Girard

Household Survey Methods Division
Statistics Canada

February 2007

The work presented in this paper is the responsibility of the author and does not necessarily represent the views or policies of Statistics Canada

Foreword

When I first started investigating the variance issue NLSCY had (described in the introduction below) I knew nothing (at least worth mentioning! ☺) about the bootstrap. As I dug deeper (and deeper and…) into the issue, I explored the bootstrap more and more. This paper contains all the lessons I've learnt (the hard way) as I made my way through the investigation. I've made a serious attempt to make the paper self-sufficient for those starting with the bootstrap (though it most certainly fails in the end due to the scope of the undertaking); it contains bits of theory, how some of the theory is meant to be applied in practical situations, ready-to-submit SAS programs to explore the bootstrap and/or to get you started with its implementation, *etc.* While the paper doesn't require any firsthand knowledge of the bootstrap, it *does* assume in return that the reader is familiar with all the methodological steps underlying the creation of survey estimates. With regard to the subjects treated here and there in the paper, like nonresponse and post-stratification just to name these two, I take for granted that the reader's knowledge on these issues is roughly that of Statistics Canada's *Survey methods and practices* (2003).

The style of writing is not formal. Actually, my natural style for writing is wordy and personal; I like to talk to the reader and explain one thing from several perspectives. (A former supervisor of mine once said that my favourite three words in English were "In other words…" Read the paper and see if that person knew me well or not!) In other words, expect redundancy and lengthy explanations throughout. For example, one formula can be stated algebraically, explained through a paragraph and illustrated with a numerical example. I think this will appeal to those who are just making their beginning with the bootstrap; the experts may find the pace too slow for their own taste. But then, it's not with them in mind that I wrote this paper.

I don't see this paper as static but rather as evolving over time as more people will tackle the bootstrap with it as their companion. Consequently, any feedback, suggestions, *etc.* are welcome and will be taken into consideration as further editions of this paper will be made in the future.

An investigation of this scope couldn't (and shouldn't!) be undertaken by just one person. The work I'm reporting here as greatly benefited from (too-numerous-to-count!) discussions held in many brainstorming sessions involving the following people: Michel Ferland, Sarah Franklin, Yves Lafortune, Scott Meyer, Michelle Simard and Marcelle Tremblay. Many thanks to Jean-François Beaumont who has kindly answered numerous initial queries on my part regarding the bootstrap and read an earlier version of this paper. Also, I'm very grateful to Dr Rao for reading a draft version of this paper and being so enthusiastic about this whole endeavour, making me benefit as the review process went along from his keen insights into the bootstrap.

I'm indebted to Sarah Franklin, Yves Lafortune and Michelle Simard for their unfailing support throughout the time I've spent investigating and writing this. Mille merci à toi Beatrice Chapman for the extensive reviewing and proof-reading that you've done of this paper. And my final thanks go to Owen Phillips, Yves Lafortune and Martin Provost for agreeing to review the final version of this paper as it was turned into a Branch Working Paper.

Abstract

The bootstrap is used in estimating variance in many surveys at Statistics Canada. It is perceived as a versatile, reliable and easy-to-implement approach, hence its growing popularity. From a theoretic perspective, it's actually a difficult and still on-going challenge to successfully adapt the bootstrap as it makes its transition from classical statistics, its nurturing ground, to the survey sampling statistical context. This explains why there are several versions of the bootstrap in the survey setting, one of which is the Rao-Wu rescaled bootstrap. It's the one most used at Statistics Canada and this is why this paper focuses exclusively on it.

But difficulties with the bootstrap are not only theoretic in nature. The practitioner implementing the bootstrap for real in a survey will find that the bootstrap is not as well described for *practical purposes* as one may have been lead to believe. There are indeed numerous pitfalls to avoid, corners to cut (and others not to cut!), *etc.* when implementing the bootstrap, and yet the existing literature is hardly of any help regarding these issues. This paper attempts to close the gap between the theory *and* the practice of the bootstrap, at least the way it's customarily implemented at Statistics Canada. It describes the bootstrap method at great length, provides tips and describes common pitfalls the user needs to stay away from and encloses SAS-based computer programs providing useful details on its implementation.

Résumé

Le bootstrap est employé comme technique d'estimation de la variance due à l'échantillonnage par plusieurs enquêtes à Statistique Canada. On la décrit comme une technique versatile, fiable et facile à mettre en oeuvre, d'où sa popularité grandissante ces dernières années. D'un point de vue théorique, adapter le bootstrap au cadre des enquêtes statistiques, alors qu'il a été conçu au départ pour la statistique classique, représente encore aujourd'hui un défi de taille pour les chercheurs. Cela explique pourquoi il existe plusieurs versions du bootstrap dans le cadre des enquêtes, parmi lesquelles on trouve le Rao-Wu rescaled bootstrap. C'est la forme de bootstrap la plus en vogue à Statistique Canada et cet article porte exclusivement sur cette technique.

Ceci étant dit, les difficultés qui accompagnent le bootstrap ne sont pas que de nature théorique. En effet, le méthodologiste qui fait face au défi d'adapter le bootstrap à son enquête s'apercevra qu'en dépit de sa popularité, le bootstrap ne s'accompagne pas d'une documentation volumineuse qui le guiderait dans sa mise en oeuvre. Il y a de nombreux pièges à éviter, des coins à arrondir (et d'autres qu'on ne doit pas arrondir!), *etc.* lorsqu'on met en oeuvre le bootstrap, et pour ces enjeux la littérature existante n'est guère de quelque utilité. Cet article tente de combler le vide qui sépare la théorie de la pratique en ce qui a trait au bootstrap, du moins de la façon que le bootstrap est communément mis en œuvre à Statistique Canada. Il contient une description détaillée, des conseils, des pièges à éviter ainsi que des programmes SAS qui seront utiles à quiconque souhaitant mettre en œuvre le bootstrap.

Table of Contents

1. INTRODUCTION

With the recent advent of powerful PCs the bootstrap has made its way into many surveys at Statistics Canada, providing methodologists with a seemingly easy-to-implement, flexible and versatile variance estimation methodology. But is it really? As it turns out, there are quite a few pitfalls awaiting the practitioner in his/her efforts to implement the bootstrap in a specific survey context; things can indeed get ugly when one leaves the realm of simple surveys to undertake the bootstrap with complex and/or longitudinal surveys.

While performing routine evaluations in the wake of cycle 5's production on earlier-released variance estimates for the National Longitudinal Survey of Children and Youth (NLSCY), which are based on a bootstrap methodology, a series of weird variance estimates were unearthed. They are the consequences of the pitfalls alluded to above of implementing the bootstrap. To give a flavour of what was noticed, Table 1 contains one such series of weird variance estimates; it's about the proportion of children in the Atlantic provinces who, back at cycle 1 of the NLSCY, had a Person Most Knowledgeable (PMK)[1] who lived *alone*[2].

Cycle	Estimate (%)	S.E. (%)	Domain	Sample	DEFF
1	15.72	1.16	541	3,678	3.73
2	15.86	0.30	471	3,361	0.23
3	15.46	0.97	475	3,322	2.41
4	15.14	1.08	424	2,993	2.71
5	14.81	0.91	383	2,814	1.86

Table 1: Point and variance estimates about the proportion of children in the Atlantic provinces who at cycle 1 had a PMK who lived alone.

In Table 1, *DEFF* is the design effect associated with the point estimate, which is the ratio of the variance calculated for the NLSCY to the variance one would get under a Simple Random Sampling design Without Replacement (SRSWOR) of comparable sample size. It is provided in Table 1 as a means to assess just how less effective NLSCY's design is compared to a SRSWOR design. Also, it shows how the variance of the NLSCY is affected over cycles by attrition *compared* to what we'd observe under a SRSWOR design. *Domain* is the un-weighted count of children whose PMK lived alone at cycle 1; *Sample* indicates the number of individuals there are in the sample from the Atlantic provinces; *S.E.* is the estimated standard error associated with the point estimate, expressed in the same units (here, percentage points).

Looking at Table 1, a few things stand out. For instance, the sampling error estimate (the *S.E.* column) of 0.30 calculated for cycle 2 is plain *wrong* (not to mention nonsensical).

[1] In a survey on children like the NLSCY, the PMK is a key element; the PMK is the one person who's in the best position to answer questions relating to a surveyed child. More often than not the PMK is the mother.

[2] A PMK was deemed to live alone if he/she had either a value of 04 (single – never married), 05 (widowed), 06 (separated) or 07 (divorced) with regard to the cycle 1 collected variable AMMPQ04 enquiring about the marital status of the PMK. A very similar characterization of PMK living alone can be obtained using another collected variable, namely ADMPD06A, which enquires if the spouse of the PMK lives in the household.

Furthermore, across cycles, the sampling error estimates are inconsistent: they do not steadily increase over cycles as one would expect them to as a result of the declining effective sample size.

In the investigation process that ensued, several issues were discovered about the implementation of the bootstrap in the NLSCY which, compounded, have lead to the strange estimates that were obtained. This paper describes the bootstrap most commonly-used at Statistics Canada (the Rao-Wu rescaled bootstrap) to the practitioner who knows little about it. It exposes both the theoretical and practical sides of the bootstrap, with particular emphasis on computer-related issues and pitfalls to avoid when implementing it. The bootstrap gains here from setting the record straight with regard to all the implementation issues we've encountered because practitioners are bound to hear about them one way or another (maybe from firsthand experience in their own efforts implementing the bootstrap). And when they do, these unaddressed issues will jeopardize in due time the coveted status the bootstrap currently enjoys with users as a versatile, easy-to-use variance estimation method.

This paper is structured as follows. Section 2 reviews the inferential framework under which we conduct variance estimation. Section 3 will introduce you to the principle behind the bootstrap. Section 4 describes the aspects of the NLSCY which are relevant for variance estimation purposes. Section 5 and those that follow describe at length the implementation of the bootstrap.

2. A WORD ON VARIANCE ESTIMATION

Before tackling the bootstrap, it's a good thing to revisit the inferential framework we're usually in when we conduct a survey. The best overview of the inference challenge may very well come from Ardilly (2000)[3] p.25; it's translated and adapted here (but the final touch of humour, in italics, is his!).

θ: Of interest to us (it's the parameter we're after) but *unknown*, hence...

$\hat{\theta}$: Great, it's computable! But it's plagued with uncertainty so...

$V(\hat{\theta})$: Of interest to us (tells us how good $\hat{\theta}$ is) but unknown, hence...

$\hat{V}(\hat{\theta})$: Great, it's computable! But it's plagued with uncertainty so...

[Are you starting to see a pattern emerging here? ☺]

$V\left(V(\hat{\theta})\right)$: Of interest to us (tells us how good $\hat{V}(\hat{\theta})$ is) but unknown, hence...

$\hat{V}\left(\hat{V}(\hat{\theta})\right)$: Great, it's computable! But it's plagued with uncertainty so...

and this can go on and on...

Computing algebraically $V(\hat{\theta})$ is often nothing short of impossible... and $\hat{V}(\hat{\theta})$? We'd rather leave it to God.

There are a few things worth saying about the inferential chain above. First, for surveys, we'd like to have an estimate $\hat{V}(\hat{\theta})$ or a good approximation of it *but* all we'll ever have to work with is *one* observed sample. While we're chiefly interested in computing $\hat{V}(\hat{\theta})$ (and let's admit it: we can't usually *compute* much of anything further down that inferential chain anyway) we *still* need to have a clue somehow if $V\left(\hat{V}(\hat{\theta})\right)$ is indeed so small (compared to $\hat{V}(\hat{\theta})$) as to be ignored; we need such a condition to be (reasonably) fulfilled in order to have any faith whatsoever in practice in our estimate $\hat{V}(\hat{\theta})$. We'll see in later sections that the computer-based environment can get us pretty close to a suitable answer to that question (and it may not be what you're expecting *i.e.*, that it's not always that small).

Also, by focusing solely above on variance, and not on Mean Squared Error (MSE), we've assumed (implicitly at the very least) that the estimator used $\hat{\theta}$ was (nearly) unbiased for θ. While for cross-sectional surveys this is a reasonable assumption to make (in truth, we make that assumption because we usually lack a way to discredit it), it's not that clean-cut in the case of longitudinal surveys. For such surveys an issue arises with

[3] The quote has been taken out from the most recent re-edition of the book.

the Mean Squared Error (MSE) which is probably more conceptual than anything but is worth mentioning here nonetheless. To illustrate what the issue is, suppose we're at cycle 2 of a longitudinal survey and we intend to use cycle 2 weights to estimate a cycle 1 characteristic. (We therefore assume explicitly here that the characteristic is either static-in-time by nature or kept fixed. In relation with Table 1 above, the characteristic *Alone* is an instance of a kept-fixed characteristic since the marital status of an individual may change over time.) Denote by $\hat{\theta}_{C2}$ the cycle 2 weighted estimate of this cycle 1 characteristic and $\hat{\theta}_{C1}$ the original estimate based on cycle 1 weights. This could be of interest as a form of diagnostic on the efficiency of the c1-to-c2-nonresponse model to keep nonresponse bias to a minimum. Suppose that $\hat{\theta}_{C2}$ turns out to be quite different from $\hat{\theta}_{C1}$. Under the assumption that cycle 1's estimate is to be favoured over that of cycle 2's (after all, cycle 2 has fewer respondents than cycle 1's so there can hardly be a gain in information in losing data), we need to compute the variance for cycle 2 accordingly. Since a cycle 2 variance estimate of any kind (let it be under bootstrap or anything else) will hinge on the cycle 2 sample-based estimate, which is off from that of cycle 1, variance and MSE won't match in this case. So if one used cycle 2 to compute the "variance" then one should actually report the computed variance plus a term for the discrepancy between the "pivot" $\hat{\theta}_{C2}$ and the "most suitable" estimate $\hat{\theta}_{C1}$:

$$MSE_{cycle2} = \text{variance} + (\hat{\theta}_{C2} - \hat{\theta}_{C1})^2 \qquad (1)$$

The point of all this is that a longitudinal setting may give rise to issues that are otherwise not met in cross-sectional surveys, those for which methodologists usually owe most of their experience.

There are two grand avenues being offered to us in getting the sought-after variance estimate $\hat{V}(\hat{\theta})$: Taylor linearization and replication methods. The former usually leads to closed-form formulas that appear in survey sampling books. It applies to "well-behaved" estimators like ratios and is best used with simple designs. Taylor linearization is what's behind Statistics Canada's Generalized Estimation System (GES) (when an exact derivation of the variance can't be obtained of course). Taylor linearization will allow you to perform variance estimation for basic estimators like totals and ratios under cluster sampling and simple two-phase designs.

One of the shortcomings of GES as a Taylor-linearization-based tool, though, is that it doesn't carry out variance estimation for the median and some methodologists in the past have turned to the bootstrap solely because of this. If you're in that situation, then consider using Woodruff's method since the only input it really needs is a variance estimate of some weighted mean and, depending on the design used, this may be obtainable through exact methods directly or Taylor linearization (see Annex *B* for further details on Woodruff's method – it's an expanded version of Girard (2005)). This would be greatly beneficial in the case of stratified SRSWOR with large sampling fractions and nonresponse (which are typical conditions in business surveys) as it's shown later that the bootstrap doesn't capture all of the variance in that context.

There are a number of replication methods that are used for variance estimation, among which are the jackknife and the bootstrap. This paper will focus (almost!) exclusively on one of the many existing versions of the latter, the Rao-Wu rescaled bootstrap. (See Section 3 for the "almost" part and more generally Lohr (1999) and Rust and Rao (1996) for a complete account of other methods.)

In a few words…

- Whether it's through Taylor linearization, replication methods or whatever else, in practice variance estimation is conducted using just *one* sample, the one you've observed.

- While the bootstrap (and more generally replication methods) can proved to be a suitable choice for variance estimation in a given context, methods based on Taylor linearization should not be discarded too quickly. Woodruff's method may supply you with the means (!) to circumvent the limitations of Taylor's technique in dealing with the median.

- In a non-conventional setting (*e.g.*, longitudinal surveys for all those methodologists whose experience draws from cross-sectional surveys), revisit basic concepts to make sure nothing gets missed. For example, is everything worth reporting in terms of sampling error well captured by variance alone or is there a need for something like a MSE?

3. THE BOOTSTRAP

This section hardly has anything new to say about the bootstrap; it rather tries to gather in one place a series of facts about the bootstrap that are spread out in the literature, books, oral tradition, *etc.* (and make them somewhat more explicit while we're at it).

Originally, the bootstrap was developed by Bradley Efron (in Efron (1979) and Efron (1982) and presented in extended form in Efron and Tibshirani (1993)) in the framework provided by independently selected observations (a.k.a. classical statistics), and as Särndal *et al.* (1992) remarked (see page 442) this is not without causing problems to survey samplers:

> *So far, the [bootstrap] technique is somewhat unexplored for survey sampling. The bootstrap technique was originally designed for use with independent observations, the standard assumption of traditional statistical theory. One basic problem, not yet definitely answered, is how the technique should be correctly modified to accommodate the special features of survey sampling, including the nonindependence arising in sampling without replacement and other complexities of designs and estimators.*

In other words, as far as classical statistics (*i.e.*, the nurturing ground for the bootstrap) is concerned, just about everything goes wrong here in survey sampling.

What we call the bootstrap in the survey setting actually is a body of techniques which are inspired from the (true) bootstrap and try to achieve the same success in surveys as it has had in classical statistics. Many variants of it thus exist, bearing names like mirror bootstrap and the Rao-Wu rescaled bootstrap for example. Despite the work that has been done since the early 1990s on bridging the gap between classical statistics and survey sampling as far as the bootstrap is concerned, we feel that Särndal *et al.* (1992)'s description of the situation is still of topical interest today and should be kept well in mind by every survey sampler.

First, let's describe a survey setting where the use and implementation of the bootstrap should not cause any problem; it helps see how the bootstrap operates:

1 – The observed sample s of size n is drawn according to a Simple Random Sampling design With Replacement (SRSWR) from a population of N elements, with N/n an integer.

2 - A pseudo-population of N elements is created by stacking N/n copies of the n elements of the observed sample one on top of the other, thus yielding in all $(N/n)*n=N$ elements.

3 - Draw a large number B of samples, called replicates, each of size m and independently from the pseudo-population using SRSWR.

4 – Form B estimates $\hat{\theta}_{boot,1},\ldots,\hat{\theta}_{boot,B}$ from the B replicates in the same manner as the basic estimate $\hat{\theta}$ was obtained from s, the observed sample[4].

The bootstrap variance estimate computed in practice is the following approximation[5]:

$$v_{boot}\left(\hat{\theta}\right)=\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}_{boot,b}-\hat{\theta}\right)^2 \tag{2}$$

Computational Tip #1

When a sampling space (*i.e.*, the set of all possible outcomes from the underlying random process) is so huge that it makes it impossible in practice to work with all possible cases (even with powerful computers), results are obtained using a subset of that space by considering some large number of cases at random; this is known as a Monte Carlo simulation. Given that the space underlying the re-sampling needed to carry out the bootstrap is huge, we work in practice from a randomly selected set of B cases instead as mentioned above. Consequently, that makes the variance estimate (2) a Monte Carlo approximation of the bootstrap variance.

What makes estimator (2) a reasonable one to choose? The answer lies in its unbiasedness for the exact variance $V(\hat{\theta})$ of the estimator $\hat{\theta}$. To express that condition explicitly, we need to realize that variance estimation using the bootstrap actually involves two distinct random sampling mechanisms: the original sampling (which yields the observed sample and provides the framework for design-based inference) *and* the bootstrap *per se*. The unbiasedness condition on $v_{boot}(\hat{\theta})$ spells out as:

$$E_D E_{boot}\left(v_{boot}(\hat{\theta})\middle|s\right)=V(\hat{\theta}) \tag{3}$$

The inner expectation is taken with respect to the re-sampling process (*i.e.*, the bootstrap), *given* the observed sample, while the outer expectation, E_D, is taken with respect to the sampling design. Whenever you wonder whether the bootstrap "works" or "doesn't work" in a given inferential setting (*i.e.*, given sampling design, estimator, *etc.*) you're actually asking whether (3) holds or doesn't hold in that situation.

This pseudo-population form of the bootstrap is intuitive: failing to have in practice the benefit of re-sampling from the whole population (as required, by definition, for design-based variance estimation) because values for the characteristic of interest in its non-sampled portion are unknown to us, we form a makeshift population from what we *do* know *i.e.*, the sample.

[4] This instruction was written here as it's usually told in practice to new users of the bootstrap. We'll see later on that taken literally this can lead to serious problems.
[5] Some authors use B as the denominator instead of our B-1; it doesn't usually matter which one you choose.

There are practical difficulties though with this first form of the bootstrap. First, it's expressed solely in terms of SRSWR and cannot be easily generalized to other designs (how to appropriately stack up the sample to create the pseudo-population when the units have unequal weights?). Also, but to a lesser extent, *N/n* has to be an integer. Nonetheless, work has been carried out in the literature to extend that form of the bootstrap and to develop close siblings which overcome the difficulties we've alluded to; see Sitter (1992) for further details.

As we mentioned in Section 2, any variance estimation method, the bootstrap included, has to work in practice from one sample (and one only), and that's quite a challenge when you come to think of it. Indeed, how can we possibly hope to have something meaningful to say about a quantity like the sampling variance using *just one sample* when by definition it takes *all* of the possible samples to construct it? And amazingly we often can do as much with apparently so little! Actually, the name "bootstrap" does capture that eerie-feeling of getting away with something when at first there seemed to have been no way out. Indeed, apparently the word is taken from the expression "to pull oneself up by one's bootstraps" which is a line from a story describing how one of the characters pulled himself out of a bad spot by using his bootstraps when there were actually no other "conventional" way out (Efron and Tibshirani (1998) p.5.).

Even though this paper is all about the bootstrap, it's probably good to differentiate it here from the jackknife method since it seems the two are often mistaken one for the other. The (delete-1) jackknife can be described as forming all possible sub-samples from the observed sample of units obtained by removing one (sampling) unit at a time. The remaining units in a given sub-sample are re-weighted appropriately to account for the loss of the removed unit and an estimate is built from such a sub-sample the same way the sample-based estimate was obtained. A key feature of the jackknife is that it does not usually give rise to a huge sampling space as with the bootstrap so in implementing it one can actually exhaust it. Consequently, there's *no* error component due to re-sampling in estimating variance with the jackknife since the entire re-sampling space is used, not just a subset of it as with the bootstrap. Despite this, the bootstrap is often preferred to the jackknife (again, see Lohr (1999) or Rust and Rao (1996) for further discussion).

3.1 Bootstrap and with-replacement designs

The method of implementation of the bootstrap which is in vogue at Statistics Canada, and was the one used for the NLSCY, is known as the Rao-Wu rescaled bootstrap. We describe it here in the case of a stratified multi-stage with-replacement sampling design.

1 – Within each stratum *h*, draw directly from the observed sample of n_h Primary Sampling Units (PSUs) some large number *B* of sub-samples of size $n_h - 1$, hereafter called *replicates*, according to SRSWR (using SAS *proc surveyselect*, say).

2 – Compute the initial b^{th} bootstrap weight[6] for all units k according to

$$w_{k,b}^{initial} = \left(\frac{n_h}{n_h - 1}\right) \times mult_{k,b} \times w_k \qquad (4)$$

where: - w_k is the design weight of unit k;

- $mult_{k,b}$ is the multiplicity of unit k in replicate b *i.e.*, the number of times it was selected under the SRSWR scheme.

3- Use the set of B weights to obtain B bootstrap estimates $\hat{\theta}_{boot,1},\dots,\hat{\theta}_{boot,B}$ in the same manner as the final estimate was obtained from the observed sample s (*i.e.*, have the replicates go through nonresponse and post-stratification methodologies, for instance, should these apply to the survey under study – we'll have more to say on what this precisely means later on) and compute the following:

$$v_{boot} = \frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}_{boot,b} - \hat{\theta}\right)^2 \qquad (5)$$

Notes:

- You may find the use of a stratified multi-stage design as the framework to introduce the bootstrap a bit too general for your own taste. If so, then you'll enjoy Simulation A which is all about the bootstrap in the context of good old SRSWR (actually, Simulation A exploits SRSWOR but the sampling fraction used there is so small that it amounts to conducting SRSWR).

- When sampling with replacement *n-1* elements from a set of *n distinct* identifiers $\{id_1,\dots,id_n\}$ (*i.e.*, the sample) according to Step 1 above, what is obtained as a result is a set of *n-1* survey identifiers $\{id_{(1)}^*,\dots,id_{(n-1)}^*\}$ which are (quite possibly) no longer all distinct. Indeed, some units from the sample may be selected more than once under the with-replacement scheme to form a given replicate. In practice, though, we prefer to work again from a set of distinct identifiers $\{id_{(1)},\dots,id_{(j)}\}$ made of the j distinct identifiers making the given replicate. But in order to do that *and* not lose all of the information originally contained in $\{id_{(1)}^*,\dots,id_{(n-1)}^*\}$ we need to keep track of each distinct identifier's *multiplicity*. In other words, in practice instead of working from the set $\{id_{(1)}^*,\dots,id_{(n-1)}^*\}$ we prefer the set of pairs $\{(id_{(1)}, mult_{(1)}),\dots,(id_{(j)}, mult_{(j)})\}$ and this is precisely what we exploit in (4).

- Strictly speaking, the rescaled bootstrap originated in Rao and Wu (1988) but the "weight-based" presentation of it that we use, *i.e.*, (3), is taken from Rao, Wu and Yue (1992).

[6] To avoid over-crowding equations with too much notation, let's keep the stratum subscript out of them whenever the context allows. For example, despite the missing subscript h, the design weight is considered to be dependent upon the stratum the unit is in.

- You may find it odd (at least now that we're about to mention it! ☺) that nothing is said of Secondary Sampling Units (SSUs) that arise in two-stage sampling; the only re-sampling we've described here is solely in terms of PSUs. We'll see in Section 7.1 what bearing SSUs have on variance estimation and what the bootstrap, when constructed the way that we've just described, can (and can't) do for us.

- You may wonder why, in Step 1, we picked $n_h - 1$ PSUs with replacement and not, say, $n_h - 2$? The paragraph here is too narrow to contain the beautiful argument behind that one choice so we refer you to the box *From Efron's bootstrap to Rao-Wu bootstrap*, below, to get the full story on this.

- And while we're at it, how does one come to even *propose* that the bootstrap weights should take the form (4) to begin with? In short, the bootstrap weights precisely take that form so that the bootstrap variance estimate of the mean coincides with the usual exact variance estimate. Here again, you'll have to refer to the box *From Efron's bootstrap to Rao-Wu bootstrap*, below, to learn more on this.

Computational Tip #1 (continued)

To drive home the point made in Computational Tip #1 about the size of sampling spaces, let's see just how big the one arising from the Rao-Wu rescaled bootstrap actually is. For example, bootstrapping a sample made of just 50 units will generate a sampling space of some 50^{49} replicates. In practice, we never pick more than *B*=1,000 replicates. Even if *B* appears large, it falls considerably short of the 50^{49} figure. To appreciate how big a number 50^{49} is, notice that while there are 4 digits in "1,000", $[\log(50^{49})]+1=[49\log(50)]+1=84$ digits are needed to write "50^{49}" in decimal notation! (Note: the brackets stand for the floor function).

So, an important implication of this is: when we choose to work with the bootstrap from *B* replicates instead of the whole truckload of replicates there exist, what we'll get in the end is a Monte Carlo approximation of the bootstrap variance estimate. In other words, don't get the idea from Simulation A that "Monte Carlo" and "bootstrap" are two "competing" methods; they're not comparable at all, the same way an adjective is not to be mistaken for a noun. Something is qualified as being "Monte Carlo" when it exploits only a portion of the whole space defined by some random process. So the bootstrap can be Monte Carlo (and will always be in practice actually) if the number of replicates used does not exhaust all possibilities. Monte Carlo is not in itself a random process like sampling from the population or re-sampling from an observed sample are; it's merely the acknowledgement that not all of the possibilities generated by the underlying process are being exploited.

Before digging any further into the bootstrap let's illustrate all this with a SAS-supported example – all figures and estimates quoted in Simulation A below are taken from the SAS code provided in Program A. The SAS code may actually help you "see" what the bootstrap is all about.

Simulation A

A population of 10,000 units is created with a normal variable of interest Y randomly generated. The population mean, $\bar{y}$, is 49.85981 and the idea here is to estimate it using the usual Horvitz-Thompson estimator $\hat{\bar{y}}$ using an observed sample of size 100 drawn from the population under SRSWOR:

$$\hat{\bar{y}} = \frac{\sum_{k \in s} \frac{y_k}{\pi_k}}{N}$$

where π_k is the (first-order) inclusion probability of the k^{th} unit.

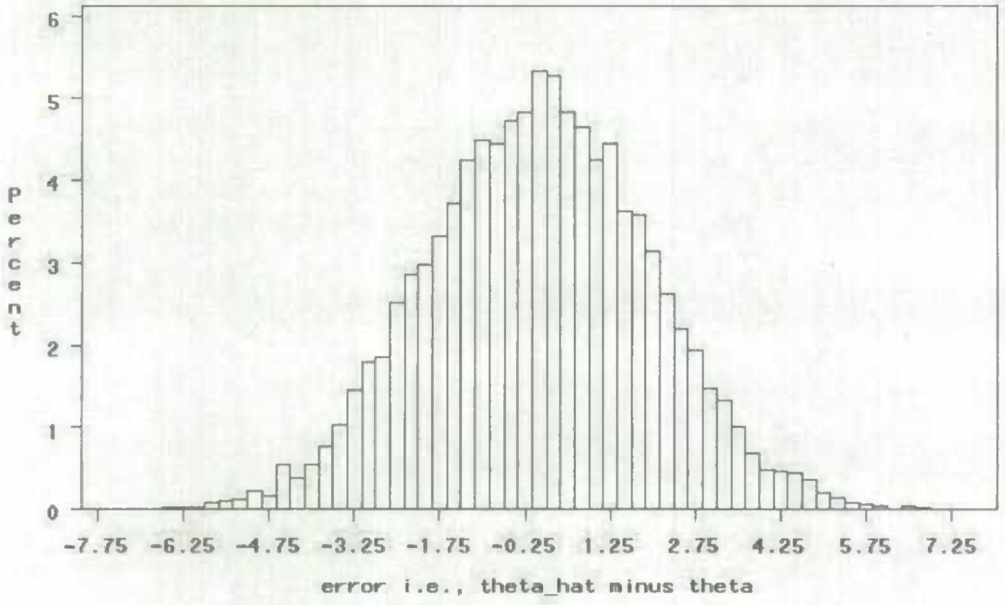
The (exact) variance of that estimator is calculated to be (see for instance Result 3.3.2, in Särndal *et al.* (1992) on page 68):

$$V(\hat{\bar{y}}) = (1 - n/N)\frac{S_U^2}{n} = (1 - 100/10000)\frac{388.2015}{100} = 3.843195$$

An approximation of that variance was obtained by a Monte Carlo simulation where 10,000 samples were drawn and the empirical variance of the ensuing estimates $\hat{\theta}_{MC,1},\ldots,\hat{\theta}_{MC,10000}$ calculated; we obtained:

$$v_{MC}(\hat{\bar{y}}) = 3.837265$$

Of course, a Monte Carlo approximation *really* is useful when, contrarily to our setting, the exact variance is *not* known beforehand (*i.e.*, cannot be computed from a formula, which would have been the case had we been interested here, say, in the median instead of the mean). But even then the Monte Carlo estimates tell us, as a whole, something valuable about the estimator used: the histogram we plot from them approximates the sampling distribution of the estimator. For reasons that will become apparent later we didn't plot the Monte-Carlo estimates directly but rather the differences or errors $(\hat{\theta}_{MC,i} - \theta)$, for $i = 1,\ldots,10{,}000$, between each Monte-Carlo estimate and the true mean.



Graph 1: Sampling distribution of the estimator of the mean obtained through a Monte Carlo simulation.

An unbiased estimator of that variance using the observed sample s can be used to get an estimate of the variance:

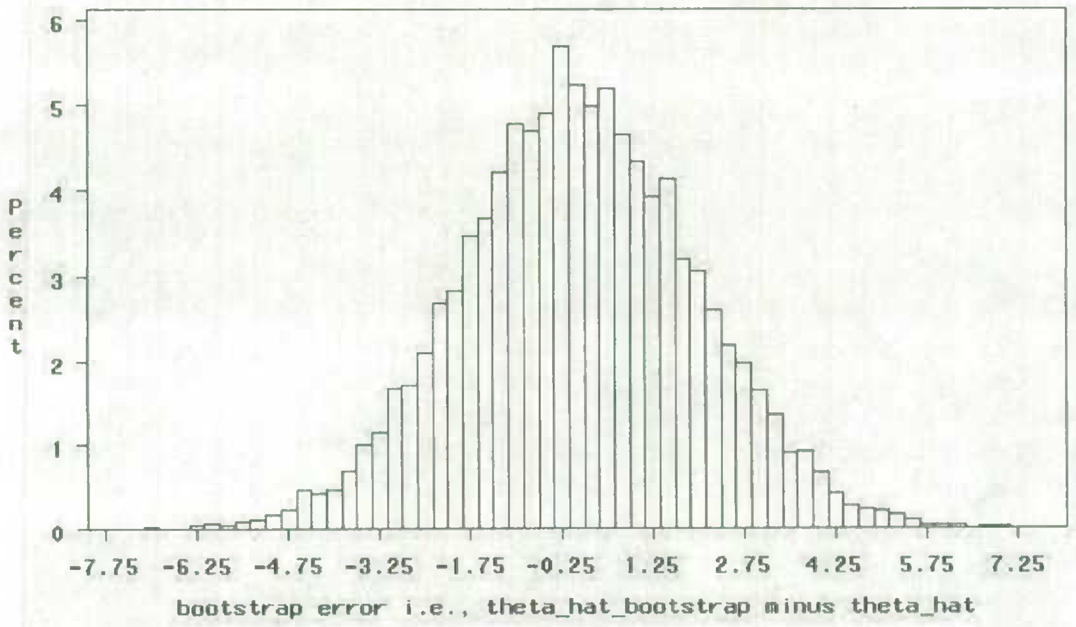
$$\hat{V}(\hat{\bar{y}}) = (1 - n / N)\frac{S_s^2}{n} = (1 - 100/10{,}000)\frac{381.2359}{100} = 3.774235$$

Bootstrapping this sample with one set of 1,000 replicates we get, using (5):

$$v_{boot}(\hat{y}) = 3.722582$$

More precisely, 1,000 sub-samples (*i.e.*, the replicates), each of size 99, were first drawn under SRSWR from the observed sample. Afterwards, bootstrap weights were assigned to units according to (4) and the ensuing bootstrap-weighted means were computed for each of the 1,000 replicates in order to compute (5).

It's interesting to draw the histogram (using the same scale as the one above depicting Monte Carlo estimates) of the bootstrap estimates:



Graph 2: Sampling distribution of the estimator of the mean obtained under bootstrap.

As with the Monte Carlo histogram, differences rather than the estimates themselves were plotted. It's important to note that the differences *here with the bootstrap* are taken between the bootstrap estimates and the *sample-based* estimate of the mean (and not the true mean). This is because the bootstrap estimates are distributed about the sample-based estimate "the same way" as the Monte Carlo estimates are distributed about the true mean; consequently, to make distributions directly comparable, differences must be computed accordingly.

It's quite striking how the two histograms are alike. This is actually no accident because both the Monte Carlo simulation, as it's used here, and the bootstrap work to approximate the (exact) sampling distribution of the estimator of the *error* $\hat{\bar{y}}-\bar{y}$. (That distribution of errors and the sampling distribution of the original estimator $\hat{\bar{y}}$ of $\bar{y}$ are just a translation factor away from one another. We paraphrase this by saying that while the original sampling has the population value in its sight, the bootstrap rather has the sample-based estimate in its sight. So, our discussion above can be summarized as follows: both sampling processes behave the same way with regard to their *respective* targets.) So, the bootstrap does not get to estimate the exact sampling variance just because it's conceived to "track it" but rather because it works to match the *whole* sampling distribution of the estimator of the error $\hat{\bar{y}}-\bar{y}$. And once that match has (reasonably) been achieved, *any* characteristic of the sampling distribution can then at once be estimated from the bootstrap distribution whether it's a given moment, its median, its inter-quartile range, *etc.* Of course, in practice, we're merely interested only in its variance. The point is that the bootstrap does not estimate the sampling variance *per se* but provide the means for us to estimate it as a by-product of its efforts estimating the whole sampling distribution.

Furthermore, as Rao and Wu (1988) observe (see page 232, below equation 2.6), in the case of the mean, the variance estimate under bootstrap (not its Monte Carlo approximation) *is* the usual variance estimate. (Actually, this is where condition (3) described above comes from.) In other words, if we had the benefit of computing the bootstrap variance estimate from *all* possible replicates (and not just rely on a Monte Carlo approximation using B of them as we always do in practice), then the numerical value would be that of the usual variance estimate. So our Monte Carole bootstrap variance approximation, obtained from bootstrapping an observed sample, gets closer and closer, with the increasing number of replicates used, to the usual sample-based estimate of variance, *not* to the exact one. It goes to the exact variance only when we further take expectation with respect to the sampling design (*i.e.*, by averaging the bootstrap approximations obtained, arising from bootstrapping a large number of observed samples).

This is as good a place as any to address the following issue that may have come up to your mind: How do we get from Efron's bootstrap to Rao-Wu rescaled bootstrap? In other words, where do the bootstrap weights given by (4) come from anyway? If you feel like learning about the genesis of the Rao-Wu rescaled bootstrap then the next box is for you; otherwise you can skip it without compromising the sequel.

From Efron's bootstrap to Rao-Wu's bootstrap

To help introduce Rao-Wu rescaled bootstrap we'll follow very closely the presentation given in sections 2, 2.1 and 2.2 of Rao and Wu (1988) and provide additional details as we go along.

Let's consider a sample s which was obtained from a stratified design with an unequal probability selection of n_h (>1) PSUs with replacement within strata and that we're interested in estimating the variance of the usual estimator of the population mean. Suppose the sample in stratum h is the set $s_h = \{y_{(k)} | k = 1,\ldots,n_h\}$. (The index notation used to describe s_h is to remind the reader that the values $y_{(k)}$ need not be all distinct due to the with-replacement feature of the design.) An unbiased estimator of the mean for stratum h is then

$$\bar{y}_h = \frac{\frac{1}{n_h}\sum_{k=1}^{n_h}\frac{y_{(k)}}{p_{(k)}}}{N_h}$$

where $p_{(k)}$ is the probability of selection of unit (k).

The estimator for the overall mean is

$$\bar{y} = \sum_{h=1}^{H} W_h \bar{y}_h$$

where W_h is the weight assigned to the stratum h with $\sum_h W_h = 1$.

Adapting equations 2.9.7 and 2.9.9 from Särndal *et al.* (1992) to our situation we find that an unbiased estimator of the variance of $\bar{y}_h$ in stratum h is given by:

$$\hat{V}_h(\bar{y}_h) = \hat{V}_h\left(\frac{1}{N_h} \frac{\sum_{k=1}^{n_h} y_{(k)}}{n_h p_{(k)}} \right) = \frac{1}{N_h^2} \frac{1}{n_h(n_h-1)} \sum_{k=1}^{n_h} \left(\frac{y_{(k)}}{p_{(k)}} - \frac{1}{n_h} \sum_{l=1}^{n_h} \frac{y_{(l)}}{p_{(l)}} \right)^2$$

Let's now see what we get by bootstrapping the sample. For starters, Efron's bootstrap requires that we perform simple random sampling of size n_h with replacement from s_h within each stratum and independently across strata; this yields within stratum h the set of PSUs $\{y_j^*\}_{j=1}^{n_h}$, which is customarily called a replicate (or a bootstrap sample). Again, due to the with-replacement feature of the bootstrap this time, any given y-value $y_{(k)}$ of s_h may be chosen more than once in a replicate; this is reflected here by the asterisk "*". (In other words, it may very well be that $y_{j_1}^* = y_{j_2}^* = y_{(k)}$ for some k and $j_1 \neq j_2$.) Then form the bootstrap estimator $\hat{\theta}_{b,h}$ from the replicate drawn the same way the estimator $\bar{y}_h$ was built from the sample s_h; by doing so we get:

$$\hat{\theta}_{b,h} = \frac{\sum_{j=1}^{n_h} \frac{y_j^*}{n_h p_j}}{N_h}$$

The theory behind SRSWR ensures that $\hat{\theta}_{b,h}$ is unbiased for $\bar{y}_h$:

$$E_{*h}\left(\hat{\theta}_{b,h}\right) = E_{SRSWR,h}\left(\frac{\sum_{j=1}^{n_h} y_j^*}{N_h n_h p_j} \right) = \frac{\sum_{k=1}^{n_h} y_{(k)}}{N_h n_h p_{(k)}} = \bar{y}_h$$

Indeed, under SRSWR an estimator of the total of a variable γ, $\sum_{i \in s_h} \gamma_i$, in the "population" s_h of size n_h is $\sum_{i \in s_{SRSWR}} \gamma_{(i)}$, the latter being built from a SRSWR sample of size n_h. (Be vigilant here: n_h is both the population size and the sample size in our SRSWR design, so whenever it enters a formula make sure you know if it's as the "sample size" or as the "population size".)

(Note: E_{*h} is the expectation with respect to the bootstrap in stratum h and is the notation used in Rao-Wu (1988); it corresponds, by definition of the bootstrap, to the expectation taken with respect to SRSWR in stratum h.)

We now need to evaluate the (exact) variance with respect to the bootstrap sampling of $\hat{\theta}_b$ within a given stratum h, $V_{*h}\left(\hat{\theta}_{b,h}\right)$ and see if we get exactly $\hat{V}_h(\bar{y}_h)$ or not.

To this end, let's rewrite our bootstrap estimator in terms of Särndal *et al.* (1992)'s notation in their Section 3.3.2 with the intent of using their equation (3.3.23) later on:

$$\hat{\theta}_{b,h} = \frac{\sum_{j=1}^{n_h} \frac{y_j^*}{n_h p_j}}{N_h} = \frac{1}{N_h}\frac{1}{n_h}\sum_{j=1}^{n_h}\frac{y_j^*}{p_j} = \frac{1}{N_h}\bar{z}_{os}$$

What their equation (3.3.23) says in terms of our context and $\bar{z}_{os}$ is:

$$V_{SRSWR}(\bar{z}_{os}) = \frac{n_h(n_h-1)}{n_h^2}\frac{S_{Uz}^2}{n_h}$$

where:

$$S_{Uz}^2 = \frac{1}{n_h-1}\sum_{i=1}^{n_h}(z_i-\bar{z})^2$$

To see that, you need to make the following equivalences between their notation and ours: $U = s_h$, $m = n_h$, $N = n_h$. This yields

$$V_{*h}(\hat{\theta}_{b,h}) = \frac{1}{N_h^2}V_{SRSWR}(\bar{z}_{os}) = \frac{1}{N_h^2}\frac{n_h-1}{n_h}\frac{s_{h,z}^2}{n_h}$$

Explicitly,

$$V_{*h}(\hat{\theta}_{bh}) = \frac{1}{N_h^2}\frac{n_h-1}{n_h}\frac{\sum_{k=1}^{n_h}\left(\frac{y_{(k)}}{p_{(k)}} - \frac{1}{n_h}\sum_{l=1}^{n_h}\frac{y_{(l)}}{p_{(l)}}\right)^2}{n_h(n_h-1)} = \frac{n_h-1}{n_h}\hat{V}_h(\bar{y}_h)$$

The bootstrap variance estimate in stratum h thus differs from the corresponding standard variance estimate of the estimator of the mean under unequal probability with replacement sampling computed earlier, $\hat{V}_h(\bar{y}_h)$, simply by the multiplicative factor $\frac{n_h - 1}{n_h}$.

The journey up to here has been somewhat treacherous so let's summarize our findings so far. We've noticed that the expectation and variance under bootstrap of $\hat{\theta}_{b,h}$ were:

$$E_{*h}\left(\hat{\theta}_{b,h}\right) = E_{SRSWR}\left(\hat{\theta}_{b,h}\right) = \bar{y}_h$$

$$V_{*h}\left(\hat{\theta}_{b,h}\right) = \frac{n_h - 1}{n_h}\hat{V}_h(\bar{y}_h)$$

where, again, $\hat{V}_h(\bar{y}_h)$ is the estimate of the variance under the (original) sampling design.

What does it all mean? It means that while Efron's bootstrap leads to a (point-) bootstrap estimator which is unbiased for the estimated mean within any given stratum h (which is a desirable feature), it *doesn't* quite yield the usual variance estimate. So what would? We need to *rescale* the bootstrap estimator so it would fit *both* criteria above. But how? We can first propose an alternative estimator of the form $\alpha\hat{\theta}_{b,h}$ for some α yet to be determined. While this fix would allow us to retrieve the variance estimate with the proper choice of α, it would also lead to a bootstrap estimator biased for $\bar{y}$. Indeed, $E_{*}\left(\alpha\hat{\theta}_{b,h}\right) \neq \bar{y}$ unless we make the getting-us-nowhere choice $\alpha = 1$.

What if we propose instead an alternative estimator of the form $\tilde{\theta}_{b,h} = \alpha\hat{\theta}_{b,h} + \beta$ for suitable constants α and β? Its expectation and variance with respect to the bootstrap would be respectively:

$$E_{*h}\left(\alpha\hat{\theta}_{b,h} + \beta\right) = \alpha\bar{y}_h + \beta$$

$$V_{*h}\left(\alpha\hat{\theta}_{b,h} + \beta\right) = \alpha^2\frac{n_h - 1}{n_h}\hat{V}_h(\bar{y}_h)$$

The latter equation forces our choice for α:

$$V_{*h}\left(\alpha\hat{\theta}_{b,h} + \beta\right) = \hat{V}_h(\bar{y}_h) \Leftrightarrow \alpha = \sqrt{\frac{n_h}{n_h - 1}}$$

Consequently, in order to get a bootstrap estimator in stratum h which is unbiased for $\bar{y}_h$ we need to choose β as:

$$E_*\left(\alpha\hat{\theta}_{b,h}+\beta\right)=\sqrt{\frac{n_h}{n_h-1}}\bar{y}_h+\beta=\bar{y}_h \Leftrightarrow \beta=\left(1-\sqrt{\frac{n_h}{n_h-1}}\right)\bar{y}_h$$

So, the grand conclusion we've reached here is this: a linear function $\hat{\theta}_{b,h}$ of Efron's bootstrap estimator in stratum h which is unbiased for $\bar{y}_h$ *and* whose variance under bootstrap sampling is the usual sampling variance estimate is:

$$\tilde{\theta}_{b,h}=\sqrt{\frac{n_h}{n_h-1}}\hat{\theta}_{b,h}+\left(1-\sqrt{\frac{n_h}{n_h-1}}\right)\bar{y}_h=\bar{y}_h+\sqrt{\frac{n_h}{n_h-1}}\left(\hat{\theta}_{b,h}-\bar{y}_h\right)$$

To keep the derivation of Rao-Wu bootstrap as close as possible to Efron's we've restricted ourselves here to re-sampling of size n_h from the sample in stratum h of size n_h. If we allow from the beginning the re-sampling to yield bootstrap samples of arbitrary size m_h in stratum h then we'll end up proposing as the Rao-Wu rescaled bootstrap estimator:

$$\tilde{\theta}_{b,h}=\sqrt{\frac{n_h}{n_h-1}}\hat{\theta}_{b,h}+\left(1-\sqrt{\frac{m_h}{n_h-1}}\right)\bar{y}_h=\bar{y}_h+\sqrt{\frac{m_h}{n_h-1}}\left(\hat{\theta}_{b,h}-\bar{y}_h\right)$$

$$\text{where: } \hat{\theta}_{b,h}=\frac{\sum_{j=1}^{m_h}\frac{y_j^*}{m_h p_j}}{N_h}$$

This is precisely the estimator obtained by Rao and Wu (1988); see their equation (2.4).

Now, this being said, how do we get something like (4)? In other words, *if we chose* to write the estimator $\bar{y}_h$ in the following form:

$$\bar{y}_h=\frac{\sum_{k=1}^{n_h}y_{(k)}w_{(k)}}{N_h}$$

for some "weights" $w_{(k)}$, *then* what would be the ensuing weights $w_{boot,(k)}$ for the corresponding bootstrap estimator $\tilde{\theta}_{b,h}$? To answer that, we first need to express Rao-Wu rescaled bootstrap estimator in terms of the units of the sample itself, rather than in terms of the units in the bootstrap sample as it's currently in. The trick is that the latter may contain several "copies" of any given sampled unit. We achieve that transition without losing information by introducing bootstrap multiplicities $mult_b(k)$ which simply count how many times a given sampled unit was selected by the bootstrap:

$$\hat{\theta}_{b,h} = \frac{\sum_{j=1}^{m_h} \frac{y_j^*}{m_h p_j}}{N_h} = \frac{\sum_{k=1}^{n_h} \frac{y_{(k)}^* mult_b(k)}{m_h p_k}}{N_h}$$

Then,

$$\tilde{\theta}_{b,h} = \frac{\sum_{k=1}^{n_h} w_{(k)} y_{(k)}}{N_h} + \sqrt{\frac{m_h}{n_h - 1}} \left(\frac{\sum_{k=1}^{n_h} \frac{n_h}{m_h} w_{(k)} y_{(k)}}{N_h} - \frac{\sum_{k=1}^{n_h} w_{(k)} y_{(k)}}{N_h} \right)$$

$$= \frac{\sum_{k=1}^{n_h} y_{(k)} w_{(k)} \left(1 - \sqrt{\frac{m_h}{n_h - 1}} + \sqrt{\frac{m_h}{n_h - 1}} \frac{n_h}{m_h} mult_{(k)} \right)}{N_h}$$

This suggests to define the bootstrap weights $w_{boot,(k)}$ when m_h are taken with replacement from the n_h sampled units as

$$w_{boot,(k)} = w_{(k)} \left(1 - \sqrt{\frac{m_h}{n_h - 1}} + \sqrt{\frac{m_h}{n_h - 1}} \frac{n_h}{m_h} mult_{(k)} \right)$$

In the (important) special case resulting from the choice of $m_h = n_h - 1$ the expression of the bootstrap weights can be seen to simplify to (4).

What values should you take for m_h? Why $m_h = n_h - 1$ above all other choices? There's no completely satisfactory answer to this riddle; let's nonetheless explore some avenues. The choice $m_h = n_h - 1$ sure leads to the simplest of all forms for the bootstrap weights since the term $1 - \sqrt{\frac{m_h}{n_h}}$ vanishes for that choice of m. Actually there's more to it: $m_h = n_h - 1$ is then the only choice of m_h which results in zero weights for units not selected in the replicate. A choice of $m_h > n_h - 1$ will lead to negative weights for those units which are not selected in the replicate. Even though this doesn't invalidate in any way the resulting bootstrap, it's an annoyance to have negative weights. Any given choice of $m_h < n_h - 1$ sure presents the advantage of leading to replicates smaller in size (*i.e.*, the number of units they would contain) over the choice $m_h = n_h - 1$ though improvements on bootstrap performance appear dubious to us. Preston and Chipperfield (2002) describe a bootstrap which hinges on the choice $m_h = {n_h}/{2}$; simulations we've performed with the Rao-Wu rescaled bootstrap has not yielded any noticeable advantage from choosing $m_h = {n_h}/{2}$ over $m_h = n_h - 1$. While there doesn't seem to be any real gain in bootstrap performance resulting from a choice of $m<n$-1 we suspect such a choice to lead to greater instability in the variance estimates as m gets smaller but this we didn't explore. Rao and Wu (1988) initially built a theoretic case in favour of $m_h = n_h - 3$ from matching the third moment of the bootstrap distribution with the corresponding moment of the sampling distribution but this choice didn't translate into any performance gain when scrutinized under simulations.

If we redo this exercise all over again but this time for a without-replacement design then we'll recover equation 4.2 from Rao and Wu (1988) as the chosen rescaled bootstrap estimator in this case.

Computational Tip #2:

Some people will themselves compute in SAS the bootstrap variance estimate (5) rather than rely on some existing statistical software to compute it for them. Unless they're very careful, what people working in SAS with *proc summary* (say) will end up calculating actually is:

$$v_{boot}^{SAS}\left(\hat{\theta}\right) = \frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}_{boot,b} - \bar{\theta}_{boot}\right)^2 \qquad (6)$$

where: $\bar{\theta}_{boot} = \dfrac{\sum_{b=1}^{B}\hat{\theta}_{boot,b}}{B}$.

So, while (5) calls for squared deviations to be computed with respect to the sample-based estimate $\hat{\theta}$, (6) actually computes the square deviations with respect to the average of the bootstrap estimates $\bar{\theta}_{boot}$. While in many situations both computations will nearly (numerically) agree, one must not resort to this shortcut blindly. In the program provided as Program A about Simulation A, we computed at its very end (5) and (6) and got 3.722582 and 3.72225, respectively. (While this is not a difference to get excited about, it remains that Simulation A hardly qualifies as an exciting framework to begin with either! ☺) It's interesting to observe that the variance estimate $v_{boot}^{SAS}\left(\hat{\theta}\right) = E_{boot}\left(\hat{\theta}_{boot,b} - \bar{\theta}_{boot}\right)^2$ is smaller than the bootstrap variance estimate $v_{boot}(\hat{\theta}) = E_{boot}(\hat{\theta}_{boot,b} - \hat{\theta})^2$; this follows from the general observation that $E(X-a)^2$ is minimize for the choice $a = E(X)$.

Softwares like SUDAAN compute the bootstrap variance estimates along the lines of (5), and not (6), with the already-mentioned exception that SUDAAN uses *B* as the denominator instead of our *B*-1. This is incidentally why SUDAAN asks of the user both the bootstrap weights *and* the design weight; the latter is used by SUDAAN in the background to compute the point estimate needed in the squared term of (5).

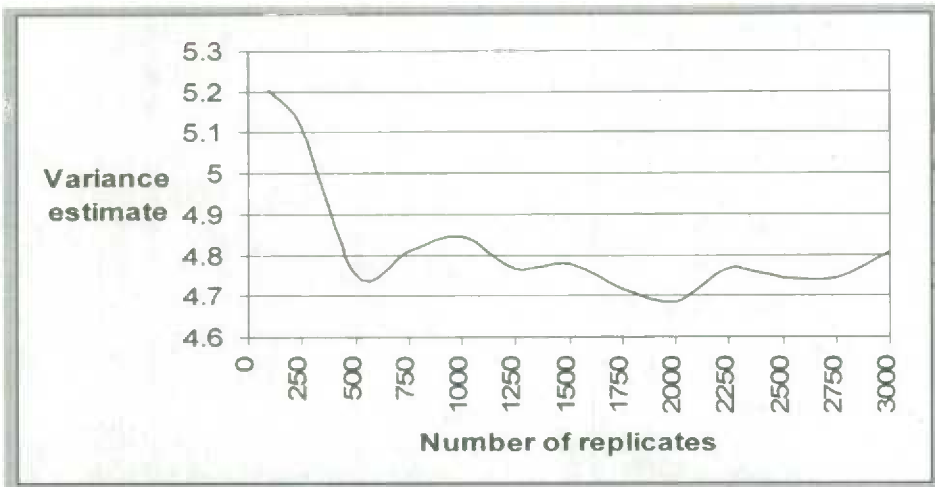
One inconvenience of SUDAAN (and many other such softwares) is that intermediate results needed to get (5) are not accessible afterwards to the user. For instance, one cannot access the *B* replicate estimates SUDAAN used to compute the variance. These are useful for plotting purposes (to run diagnostics, better understand the bootstrap, *etc.*).

Strictly speaking, SUDAAN (which is better known to many users through its SAS-callable version) does not support the bootstrap method *per se*. What it does support is the balanced repeated replication (BRR) whose variance estimate is computed using an expression identical to our (5) using BRR replicates in lieu of bootstrap replicates. So, if bootstrap replicates are fed directly into SUDAAN *as if* they were replicates obtained under BRR, then SUDAAN will correctly compute for us the ensuing bootstrap variance estimate.

In Simulation A, *B*=1,000 was used and this happens to be the number of replicates NLSCY uses also. There are surveys whose bootstrap relies on 500, and there are specific settings where one could get away with variance estimation just fine using as little as 250 replicates. The generally accepted rule is that you should consider having as many replicates as your computer system will allow: the bootstrap should not freeze all resources nor take forever to run. You'll find in Section 9.1 heuristic results on the stability of variance estimates as a function of the number of replicates that may be of some assistance; but in the end the number of replicates to use in a given setting is a call the practitioner has to make. In the meantime, here's a trick some like to use:

Computational Tip #3

Some methodologists like in practice to compute bootstrap variance approximations arising from ever-increasing numbers of replicates used and plot them (such a strategy is described in Program B):



Graph 3: Variance estimates as a function of the number of replicates used.

They then eye-ball the graph for some threshold number of replicates beyond which variance approximations seem to stabilize and go on to use it as the survey's number of replicates. The question is: looking at Graph 3, where would you draw that line?

Since the discussion so far has hinged on simulations, a word of warning is in order. Since in a simulation framework the value of the population parameter we're after, θ, is *known*, it's quite tempting to substitute this value for the estimate $\hat{\theta}$ in (5) and thus compute instead

$$v_{boot}^{alt}\left(\hat{\theta}\right)=\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}_{boot,b}-\theta\right)^{2} \tag{7}$$

After all, why resort to the sample-based estimate $\hat{\theta}$ when we can actually do "better" and use the true value θ instead? Tempting as it is, it's *wrong*. Indeed, rearranging (7) will show the difference arising from using θ instead of $\hat{\theta}$ in the variance computation:

$$v_{boot}^{alt}\left(\hat{\theta}\right)=\frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}_{boot,b}-\hat{\theta}+\hat{\theta}-\theta\right)^{2}$$

$$=\frac{1}{B-1}\left[\sum_{b=1}^{B}(\hat{\theta}_{boot,b}-\hat{\theta})^{2}+2(\hat{\theta}-\theta)\sum_{b=1}^{B}(\hat{\theta}_{boot,b}-\hat{\theta})+\sum_{b=1}^{B}(\hat{\theta}-\theta)^{2}\right]$$

$$=v_{boot}(\hat{\theta})+2(\hat{\theta}-\theta)\frac{B}{B-1}\frac{\sum_{b=1}^{B}(\hat{\theta}_{boot,b}-\hat{\theta})}{B}+\frac{B}{B-1}(\hat{\theta}-\theta)^{2} \tag{8}$$

Consequently,

$$E_D E_{boot}\left(v_{boot}^{alt}(\hat{\theta})\middle|s\right) \cong V + 0 + V = 2V \text{ !!}$$

(The rightmost term of (8) is a constant with respect with the inner-expectation *i.e.*, it's independent of the replication process, so all that remains is the expectation taken with respect to the design and this is by definition the sampling variance.)

So, with the alternate approximation (7) which uses θ directly, you end up calculating the variance twice! Therefore, the point that was made right after Simulation A is all the more important: the bootstrap, as a re-sampling method, hinges on the observed sample and not on the population directly as the usual survey sampling procedure does. In other words, bootstrap estimates vary about the sample estimate the same way the sample estimates vary about the true value; therefore, if the bootstrap estimates are compared directly to the true value, then the variability of the bootstrap estimates assessed in such a way involves "double counting".

Computational Tip #4

This computational tip is about saving space in storing bootstrap files.

There are usually two files produced by the bootstrap sampling process: a file of multiplicities and a file of bootstrap weights. Both files will likely require insane amounts of storage space on a server. A bootstrap file can easily be several mega-bytes big. While the sheer size of these files isn't in itself an issue, their ever-growing number on a given server *is* due to surveys cumulating many cycles worth of data over time.

Two things conspire in making bootstrap files huge: a by-default allocation in SAS of too much memory to store a bootstrap weight variable and the fact that there's "too much air" in the file. Thus, to save space all we need is to refrain ourselves from using the space-consuming memory allocation provided to variables by SAS as the default and compress the ensuing output SAS file.

First, the default length for a numeric variable in SAS is 8 bytes. In most (if not all) practical settings this is overkill: the level of precision this provides far exceeds what is typically needed. For example, if a numeric variable can only contain an integer, as it's the case in a file of multiplicities, then using 3 bytes in SAS (which is the minimum) through the *length* statement will save lots of storage space and *won't* result in any loss of precision. On the other hand, if a variable contains real numbers, then it may very well be that 4 bytes will capture all the precision that's relevant for computation purposes.

We can on top of that compress the file to take the "air" out. While this can be done using WinZip, compression can also be handled by SAS directly as it creates the permanent output file. The big advantage of this over zipping the file is that the SAS-compressed file looks like any regular SAS file. It doesn't require any specific manipulation when it's read back into SAS: the de-compression is operated by SAS automatically without the user having to bother with it in any way.

Described this way it's pretty esoteric but it's really quite simple as the provided Program I illustrates. The program shows how to assign a variable a given number of bytes for storage through the *length* statement and how the compression is implemented. By running the program you'll find that the size of the file of multiplicities (which is full of integers) can be reduced by some 80% while the file containing real numbers is reduced in size by about 60%.

What about running time when dealing with compressed files? While SAS' documentation mentions that processing a compressed SAS file may take more time, we didn't notice any significant difference in reading time between compressed and corresponding un-compressed files. Another potential issue has to do with the "truncation" which results in using fewer than 8 bytes for a variable having as value real numbers (like a bootstrap weight variable): does it impact on the results? For example, will the estimated variance using the truncated bootstrap file of weights be different than the one relying on all 8 bytes? Tests performed using NLSCY's data show that with 4 bytes the farther apart the ensuing variance estimate has been from the corresponding estimate based on 8-bytes variables was one thousandth of one percent!

Bottom line: you should evaluate in light of the uses of the data what precision level for numeric variables is more than enough; this will greatly contribute in keeping the size of files to a minimum.

If you're *still* worried about truncating a numeric variable, compare the "error" in the ensuing estimates that you introduce by truncating a weight variable to 3 decimals, say, as opposed to using all 8 bytes to that of the sampling error of the estimate itself. You'll find that the truncation will affect digits in the estimates which are plagued with much greater uncertainty due to the sampling variance. In other words, the error due to truncation is expected to be several orders of magnitude *smaller* than the sampling error. To illustrate, suppose an estimated value of 10,000 based on truncated weights differs by its last digit with the 8-bytes-based estimate: if the sampling error is as low as 1%, then this means that our 8-bytes-based estimate was uncertain as far as the hundreds go. So why differentiate two estimates which are identical except, due to truncation, for their last digit then?

3.2 Bootstrap and without-replacement designs

In Simulation A above, the sampling fraction of 100/10,000=1% was deemed low enough to conduct the bootstrap there ignoring the without-replacement nature of the initial SRSWOR design used. It's as good a time as any to talk about the bootstrap in the case where the sampling fraction in a stratum *h* is too large to ignore; this topic in the context of complex surveys is the main focus of Mach *et al*. (to appear). (Actually, it's instructive to keep in mind while reading this the issues they raise in their conclusion and see to what extent the points covered here and there in this paper address them.) One proceeds as above except that the following bootstrap weight is calculated instead of (4) in a given replicate b and stratum h:

$$w_{k,b}^{fpc,initial} = \left[1-\sqrt{1-f_h}+\sqrt{1-f_h}\left(\frac{n_h}{n_h-1}\right)mult_{k,b}\right]w_k \tag{9}$$

Note: You get (9) from Rao and Wu (1988)'s equation 4.1 the same way we've obtained (4) in the box *From Efron's bootstrap to Rao-Wu bootstrap*, above, from their equation 2.4.

The bootstrap weight obtained through (9) contains a factor which is a function of the sampling fraction f_h in stratum h which allows us to recover for a total (or a mean) the finite population correction factor (fpc) $1-f_h$ that distinguishes its variance estimator under SRSWOR from that obtained under SRSWR. In other words, if in the context of SRSWOR we compute in parallel bootstrap variance estimates using (4) and (9) with ever growing sampling fractions, then we'll see the two variance estimates drift further and further apart. This is what Simulation C illustrates.

Simulation C

In Program C you'll find the code for a limited simulation that shows how bootstrap variance estimates based on (4) and (9) differ when the sampling fraction is no longer negligible; it was used to obtain the figures of Table 2. The idea here is to address an ambiguous statement often heard about the bootstrap and large sampling fractions; it goes something like this: "The bootstrap doesn't work when the sampling fraction is large." This is true only if by "bootstrap" we mean the one which does not integrate the finite population correction factor *i.e.*, the bootstrap implemented using (4). But then, when the design is with replacement this is not the form of the bootstrap one should be using! If one uses in such situations the bootstrap defined by the weights (9) (as one should) then the bootstrap does work for large sampling fractions. This simulation is used to show that each form of the bootstrap "tracks down" correctly the variance with respect to the design it was built for: with-replacement bootstrap goes with SRSWR and without-replacement bootstrap goes with SRSWOR. It's when people start to carelessly mix with-replacement bootstrap and without-replacement designs that things go awry; if it backfires, then don't blame it on the (with-replacement) bootstrap.

For each of the SRSWOR and SRSWR designs, the exact variance, its usual unbiased estimator and the corresponding bootstrap variance approximation (with or without the finite population correction factor, accordingly) are calculated. BS_NO_FPC is the variance approximation obtained using (4) and BS_FPC is the variance approximation obtained using (9).

		SRSWOR			SRSWR		
n	f (%)	Exact	Estimated	BS_FPC	Exact	Estimated	BS_NO_FPC
100	1	3.8432	3.7742	3.8567	3.8820	3.8124	3.8956
500	5	0.7376	0.7067	0.7157	0.7764	0.7439	0.7533
1,000	10	0.3494	0.3532	0.3499	0.3882	0.3925	0.3888
2,000	20	0.1553	0.1507	0.1514	0.1941	0.1883	0.1892
3,000	30	0.0906	0.0913	0.0919	0.1294	0.1304	0.1313
4,000	40	0.0582	0.0568	0.0507	0.0971	0.0946	0.0845
5,000	50	0.0388	0.0389	0.0394	0.0776	0.0778	0.0788

Table 2: Bootstrap variance approximations (9) and (5) as a function of the sampling fraction.

We know from the exact variance formulas that the variance of the Horvitz-Thompson estimator of the mean under SRSWR is $100\times(1/f-1)^{-1}\%$ greater than that under SRSWOR. Indeed,

$$V_{SRSWOR}(\hat{\mu}) = (1-n/N)\frac{S_U^2}{n} \tag{10}$$

$$V_{SRSWR}(\hat{\mu}) = \frac{S_U^2}{n} \tag{11}$$

The relative difference between the two is computed to be:

$$\text{relative difference} = \frac{100\times\left|V_{SRSWR}(\hat{\mu})-V_{SRSWOR}(\hat{\mu})\right|}{V_{SRSWOR}(\hat{\mu})} = 100\times(N/n-1)^{-1} \tag{12}$$

So, with a sampling fraction *f* of 25%, the variance under SRSWR is 30% larger than that under SRSWOR; it gets to 100% when the sampling fraction *f* is 50%. Our simulation's findings agree (as they should!) with these results which stem from the theory of survey sampling. The simulation does show that in presence of large sampling fractions the bootstrap conducted from (4) still is on track (and doesn't go awry as some have been lead to believe) *provided* you're working from a SRSWR design. But if you're in fact dealing with SRSWOR, as it's usually the case, then you have to switch to (9) to conduct your bootstrap; otherwise you'll be estimating variance as if you were under SRSWR, not SRSWOR.

It's important to note a peculiar feature of the bootstrap weights obtained through (9) which *may* have repercussions on just how appropriate a variance estimate obtained using them will be in practice. Going back to (4), we notice that if a unit *k* is not selected in a given replicate (*i.e.*, its multiplicity term is 0) then its corresponding bootstrap weight is 0 as we expect it to be (we expect units not selected by the bootstrap to form a given replicate not to contribute to the (replicate) point-estimate). But the same doesn't arise in (9): units with a multiplicity term of 0 *will* inherit a non-zero bootstrap weight; it will actually be equal to $w_{k,h}^{fpc,initial} = \left[1-\sqrt{1-f_h}\right]w_k$. Even though this weight is usually not big in itself (you can actually show it's smaller than 1 in the case of SRSWOR), there are often *many* non-selected-in-the-replicate units out there so that the whole contribution of non-selected units is non-negligible. So, all the units in the observed sample get to contribute in each of the replicates, *regardless* of whether they were selected in the replicate or not. This has an impact on how we conduct domain estimation. Usually, we expect that for a given domain and replicate, only the units that we've selected in the replicate *and* belong to the domain would contribute. But here, we get a contribution from two classes of units: units belonging to the domain, regardless of whether they were selected in the replicate or not. But these units belonging to the domain have had their "total" weight shared with everyone. So, are these two contributions enough to recover all that is due for the domain? Quite possibly, but still the user has to be on the lookout for anything fishy in the estimates that could point out to specific issues with the "shared" weights.

Computational Tip #5

As we've just pointed out, conducting bootstrap variance estimation using the weights (9) requires special care. Since units with zero multiplicity nonetheless get a non-zero weight, the output of *proc surveyselect* is no longer *enough* to conduct the inference. Indeed, those sampled units not selected in a given replicate (*i.e.*, those with multiplicity zero) are *not* included in the output file by the procedure. (And if you forget that, your bootstrap variance approximations will not hold water.)

The procedure has an option *outall* that can be specified which forces *proc surveyselect*'s output to be based on all units that have been sampled; this option cannot be used, though, with all designs.

A way to assemble all that is needed which will work when the option *outall* is not available is to first stack the sample on top of itself as many times as there are replicates while taking care to index the copies from 1 to the number of replicates using a variable called *replicate*. One then proceeds to merge this file of copies of the sample to the replicated units on the variables "replicate" and "unit identifier". In the resulting file, those units with a missing multiplicity (*i.e.*, those in sample but not in the corresponding replicate) are then assigned a multiplicity of zero before actually computing (9).

If the without-replacement scheme is not (stratified) SRSWOR than things get quite difficult. We don't cover here those cases because we didn't experiment them at all – this was not required for the NLSCY; we refer the reader directly to the sources, *e.g.*, Rao and Wu (1988), Sitter (1992), for available options.

At this point we've seen how basically the bootstrap operates; it remains to see how it's implemented in real survey situations. The NLSCY will serve here to illustrate many key features of the bootstrap implementation; it is briefly introduced in the next section.

In a few words…

- The distribution of all bootstrap estimates matches (up to a translation factor) the sampling distribution of the estimator. We can avoid talking about that translation factor if we express ourselves in terms of errors rather than estimates: the distribution of errors of the bootstrap with respect to the sample-based estimate matches that of the distribution of errors of the estimator with respect to the population parameter of interest. From this fundamental correspondence follows the equivalence of any corresponding moments, for example, one of which is the sought-after variance of the estimator.

- The Rao-Wu rescaled bootstrap is attractive to survey samplers because, among other things, there's a weight-based form to it (see Rao, Wu and Yue (1992), first paragraph of Section 3.3).

- Rao-Wu rescaled bootstrap is designed in such a way as to yield a bootstrap estimate which coincides with the usual variance estimate in the case of a mean.

- Use the form of the bootstrap that suits best the design, depending on whether the latter is with or without replacement.

- Rao-Wu rescaled bootstrap must be implemented with care in SAS depending on whether the sampling fraction is negligible or not. Furthermore, when the weights (9) are computed, extra care must be taken to ensure that the SAS file driving the computations contains a record for everyone sampled.

- The set of all possible replicates that can be drawn is usually so huge that only *B* of them can be investigated in practice. Consequently, the variance estimate we get in practice should rather be described as a Monte Carlo approximation of the bootstrap variance estimate.

- Use as many replicates *B* as possible without monopolizing all resources of a PC. There is usually little benefit from drawing more than one thousand replicates.

- Considerable savings can be made in terms of storage space of the bootstrap files if the precision needed about the weights is scrutinized (*i.e.*, 8 bytes is overkill) *and* the files are compressed *within* SAS as they're outputted to their storage locations.

4. OVERVIEW OF THE NLSCY

In this section we briefly discuss aspects of the National Longitudinal Survey of Children and Youth (NLSCY) which are of particular importance for variance estimation. The idea here is not to become an expert of the NLSCY but rather to get a feel for the methodological context that give the issues addressed here all their flavor.

Two aspects of the NLSCY drive its variance methodology: it's a longitudinal survey and it's based on the LFS. (If you have little interest in longitudinal or LFS supplement surveys then you may want to skim through this section and just read the summary at the end.) Some of the earlier work done in relation with variance estimation and the NLSCY can be gathered from Laflamme (2002). A more complete account of the LFS as far as NLSCY is concerned can be obtained from Chapter 5 of NLSCY's Cycle 6 UserGuide.

The NLSCY has existed since 1994. Over the years, it gave birth to several cohorts, one of which was born when the survey itself was first introduced. Throughout the life of a NLSCY cohort (some cohorts have a shorter life span than others), a new cycle of collection is undertaken every two years; as these words are written, the original cohort is in the field for its seventh cycle. A new cohort, called an Early Childhood Development cohort (ECD) is freshly introduced at each cycle. It starts up with a sample of 0-1 year-olds taken from some of the LFS rotation groups which cover the age period of interest.

While the NLSCY has drawn babies at one point in its history from the Birth Registry, it chiefly obtained for its still-active cohorts its children from the rotating samples of the LFS. Thus the focus here is on parts of the LFS design that are relevant to the NLSCY and leave issues specifically related to the Birth Registry out. Furthermore, only generalities about the LFS are presented; exceptions to what is presented here exist and the reader is referred to LFS' documentation for an exhaustive description of the LFS design.

The LFS is based on a stratified two-stage design. Within LFS strata, which are geographic areas, clusters of households are formed and we can think of these as city blocks. The clusters are assigned randomly to one of 6 entities called a rotation group. Each rotation group is assigned to 2 months: Rotation 1 is associated to January and July; Rotation 2 is associated to February and July; and so on. At any given time the LFS sample is made of households from all 6 rotation groups. Each month a sub-sample of households, taken under a systematic selection scheme from the rotation group associated to the month, enters the LFS and will remain in the survey for 6 months; it replaces a sub-sample of households which are due out, now that their 6-month period has come to an end. For example, in July of a given year, the sub-sample taken from Rotation 1 which has entered the LFS back in January is due to exit and is to be replaced by a sub-sample taken from Rotation 1. A given rotation group thus contributes a sub-sample to the LFS every six months. When a given rotation group is exhausted, which may take 2 to 4 years, it's replaced by a fresh cluster from which upcoming sub-samples will now be taken. Each rotation group indexed by the date it's rotating in is thus a PSU for the NLSCY and these are the units we'll re-sample in NLSCY to carry out the bootstrap.

The following diagram illustrates all this in 3 steps. The large rectangle is a stratum and the 6 upward rectangles are the selected clusters; they're numbered from 1 to 6. Each has a light-gray top portion representing its contribution to the first sample that has rotated in. The darker-gray portion below it represents the part of the clusters that rotates in as the corresponding light-gray portion rotates out, and so on.

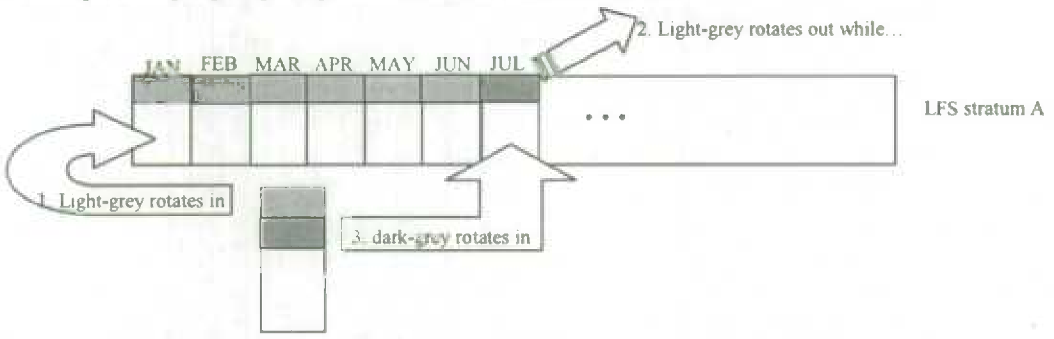
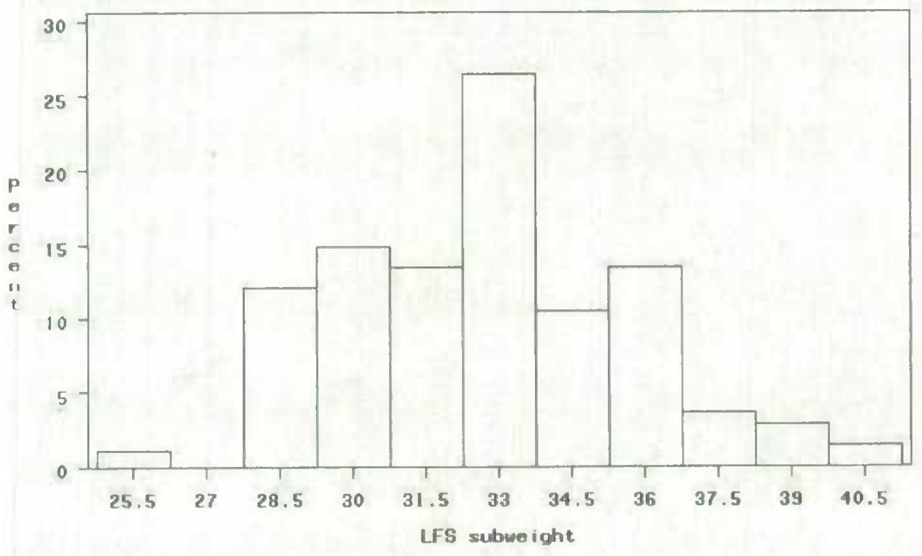


Diagram 1: Illustrating how a sub-sample of rotation group 1 enters a LFS stratum in January, remains for six months before exiting only to be replaced by another sub-sample from rotation group 1.

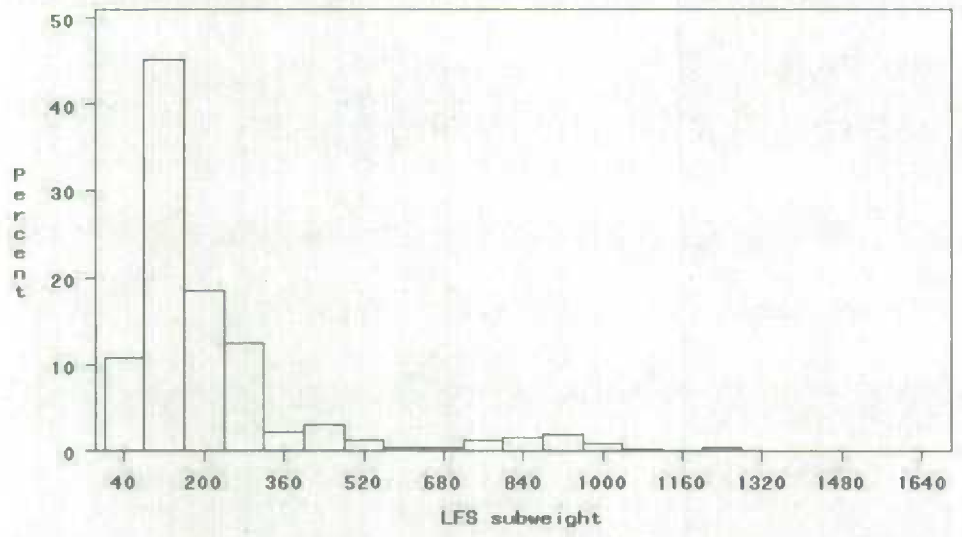
Since the NLSCY is interested in sampling 0-1 year-olds, which is a rare domain of the LFS, a large number of rotation groups must be surveyed; depending on the cycle and the province of residence, NLSCY typically uses from 12 to 15 rotation groups. Why not more? Two reasons: 1) there are not that many other rotation groups which cover a given reference period of 0-1 year-olds; 2) as exciting and important the NLSCY is, it's not the sole survey resting on the LFS so overlap must be controlled, and thus rotation groups left entirely for others to use.

While the sub-samples of any six successive months are independent one from another, this is not true when more than six are visited: two sub-samples will unavoidably have come from the same rotation group. So, as a result of sampling more than 6 successive-in-time PSUs, there's dependence among the PSUs taken from a same rotation period at an interval of 6 months. This inner-cluster dependence of sub-samples is not documented in the NLSCY and presumably has gone unnoticed. The Youth In Transition Survey (YITS) was aware of it and actually revised LFS design weights to account for it. In upcoming months we'll investigate whether we should adjust the LFS weights in a similar fashion. We already know that often a sub-sample taken from a cluster yields no eligible kids for the NLSCY so pairs of dependent sub-samples may not exist at all. In other words, given NLSCY's target population is rare with respect to the LFS, are there enough "opportunities" in the contributing sub-samples for the dependence to manifest itself? This will require further investigation. In the short note Beaumont (2000) you'll find a way to account for the inner-cluster dependence into the weights; this is what YITS has relied on for its LFS-based weighting strategy.

The weight of a household as inherited from the LFS is passed on to the children. Within NLSCY the adjustments such as nonresponse and post-stratification are child-based. One important feature of the LFS which has important repercussions in NLSCY is its unequal weights. LFS is by no stretch of the imagination close to relying on a self-weighted design. Since the LFS stratification has a very strong geographical component, the distribution of the LFS weights of the households entering the NLSCY at cycle 1 were plotted in PEI and Ontario. Notice just how the scales involved with these two distributions are different. Having to deal in variance estimation with such hugely dissimilar weights is bound to have repercussions in the whole (point and variance) estimation process. We can certainly expect variance estimates to be unstable *i.e.*, vary a lot when inferential circumstances are quite similar, more than they'd be under an equal weights design for instance.



Graph 4: Distribution of LFS weights in cycle 1 of the NLSCY in Prince-Edward-Island



Graph 5: Distribution of LFS weights in cycle 1 of the NLSCY in Ontario

In the NLSCY there exist two types of longitudinal inferences depending on who is considered a respondent at a given cycle. One approach is to consider a kid a (longitudinal) respondent if and only if the kid was a respondent to all previous cycles; the kid is not allowed to have known one episode of nonresponse. This is called the funnel approach. On the other hand we allow to some extent episodes of past nonresponse; this is called the swiss-cheese approach. (If you think of the "holes" in the file such episodic nonresponse creates, then you'll see where the terminology used comes from.)

The NLSCY also supports, as other longitudinal surveys often do, some cross-sectional analyses. When a cross-sectional analysis is made using a longitudinal cohort, several samples are usually involved. For example, assume there's interest in some common characteristic of 0-5 year-olds at cycle 3 of the NLSCY. To infer on that cross-sectional population we rely on samples taken from 3 different cycles. Indeed, the 4-5 year-olds at cycle 3 entered the NLSCY as 0-1 year-olds back at cycle1; the 2-3 year-olds at cycle 3 were actually introduced as 0-1 year-olds in cycle 2; the 0-1 year-olds at cycle 3 were just introduced (*i.e.*, cycle 3). To simplify, assume that a given group of 0-1 year-olds entering the NLSCY (*i.e.*, at a given cycle) is the result of looking for 0-1 year-olds in 15 of the rotation groups of the LFS covering the reference period. Therefore, to get our cross-sectional sample of 0-5 year-olds at cycle 3, in all 45 rotation groups were probed. The question is: should the variance methodology hinge on all 45 rotation groups? How? We can decide to work as if we had just one sample responsible of getting us the 0-5 year-olds *provided* we see that sample as having been stratified by age group: 0-1 / 2-3 / 4-5. It then becomes clear what to make of all 45 groups: we need to only consider that for a given stratum we sampled from 15 rotation groups.

NLSCY being a longitudinal survey we need to have bootstrap replicates at every cycle of a cohort's life. What is used in the NLSCY is what some call coordinated bootstrap: a set of bootstrap replicates is chosen once when the cohort is born and these are used at every subsequent cycle; a new set of replicates is not drawn at each cycle. (This topic, among others, is discussed in Roberts *et al.* (2001)) To renew the resampling at each cycle would only introduce further instability in the variance estimates.

One last thing worth describing in details is the faith of out-of-scope units with regard to the bootstrap. In the NLSCY there are essentially two ways for a selected unit to be out-of-scope. First, a household selected from the LFS can be out-of-scope for the NLSCY if it has no eligible children to provide to the NLSCY. Since the NLSCY targets a rare population from LFS' standpoint, such out-of-scope units are very common. Second, should a child die or leave the country after entering the NLSCY's sample then he or she will become cross-sectionally out-of-scope, though longitudinally such children remain in-scope for the entire life of the cohort they're in. Either way, out-of-scope units are dealt with in NLSCY's weighting using the theory of domain estimation (see Särndal *et al.* (1992) section 14.7, especially item i) of sub-section 14.7.2 for further details). So, for example, while a household from the LFS having no eligible child can be seen as being out-of-scope for the NLSCY, it's actually treated in NLSCY's weighting as not belonging to the domain of children targeted by the NLSCY. One implication of this is that the LFS-weight assigned to these out-of-scope households is "lost" and not redistributed among the in-scope ones.

In the sequel we won't mention explicitly again out-of-scope units though what to do with them with regard to the bootstrap will be implicit from the exposé on domain estimation.

In a few words...

- Upon closer look, there's nothing truly specific to the NLSCY in this section: all of it is relevant to any given survey, longitudinal or not, LFS-based or not. The example of the NLSCY is used here merely to illustrate how some issues "come to life" in a given survey context; you may ignore some (or all!) of the specificities if you want – the paper is about the bootstrap after all, not the NLSCY!

- The longitudinal component of the NLSCY has brought the need for coordinated bootstrap which avoids introducing additional noise in variance estimation by using the same set of replicates through all cycles of a cohort.

- A survey which rests upon a sampling design with greatly unequal weights will be facing issues with stability of its estimates (point and variance); what would have required few units and/or few replicates will all of a sudden require much more.

- The treatment of out-of-scope units in the NLSCY is carried out using the theory of domain estimation.

5. INITIATING THE BOOTSTRAP

5.1 Multiplicities

Before computing bootstrap weights, we need to *create* the *B* replicates themselves. With the advent of SAS *proc surveyselect* this is a walk in the park; this is much appreciated by those who have attempted implementing in SAS a procedure of their own that performs with-replacement sampling. The abundant use of *proc surveyselect* in the SAS simulations accompanying this document should prove to be more than enough to get you well-acquainted with its syntax.

To carry out the bootstrap, we first need an observed sample; this is illustrated by the SAS code below which draws a sample of 100 units from a SRSWOR design:

```
proc surveyselect data=population method=srs seed=1 n=100
out=observed_sample;
run;
```

Note: If you're not familiar with the concept of a *seed*, let's simply say that specifying a value for it is a way to recover time after time, run after run, the *same* results. (If you're writing a document like this one and you haven't set a value for the *seed*, then you can kiss consistency goodbye because users won't be able to reproduce (and/or check) your results.)

We then need to draw with replacement some large number *B* of sub-samples from an observed sample, each of size *n*-1 (which is 99 in our example). This would typically be carried out using a SAS code similar to, choosing to draw *B*=1,000 replicates:

```
proc surveyselect data=observed_sample method=urs seed=2 n=99
rep=1000 out=multiplicities;
run;
```

As a result of performing with-replacement (*i.e.*, method=urs) with *proc surveyselect*, you'll get a file of multiplicities. It describes a given replicate by listing all units that were chosen *at least once* along with the number of times each unit was chosen (*i.e.*, the multiplicity). The sum of the multiplicities within a given replicate *and* stratum ought to give the number of units you've sampled with replacement in that stratum (*i.e.*, n_h-1). In the example above, summing within a given replicate the variable *NumberHits* found in the *proc surveyselect* output file *multiplicities* will yield 99.

The first step in carrying out the bootstrap for a real survey thus appears benign enough: we just need to repeatedly sample with-replacement n_h-1 units within each stratum from the observed sample. But taking the NLSCY as an example we'll show how things can get messy pretty quickly if one is not careful. As mentioned above already, in the NLSCY the PSU is the LFS rotation group indexed by the date it rotated in. Also, the number of PSUs in a LFS stratum that are visited for NLSCY's sampling purposes often exceed 12, while the exact number depend on cycle and province. But the important point here is this: not all these PSUs have eligible kids to provide to the NLSCY; many are actually empty in this regard *i.e.*, do not yield *any* 0-1 year-olds. Though these selected but empty PSUs have no bearing on the weighting of the NLSCY (since they can be ignored altogether as if NLSCY's population simply was a domain of the LFS), they have to be accounted for in the variance calculation. Indeed, we have to pay a price for "wasting" some of our sample size the way we do trying to find members of our target population within the LFS. So, out of 12 selected PSUs in a stratum, say only 5 turned up to be non-empty; we must then re-sample from all 12 PSUs not just the 5 non-empty ones. You may wonder at this point if it really *matters* whether we re-sample from the whole set of 12 PSUs (by taking 11) or are the 5 enough (and hence choose 4)? Free of NLSCY's jargon the question here is this: when the sample you use actually is a subset of a sample that was drawn from some design *i.e.*, a domain of the main survey, can you bootstrap it directly or do you rather need to bootstrap the main sample?

If you do wonder whether it matters or not, then you're actually asking what the distribution of the multiplicities within a replicate and stratum looks like (and does it depend or not on the sample we bootstrapped from). Other distribution-related questions are: How likely is a given unit not selected at all to form a replicate? If we sample with replacement 99 units out of a sample of 100, will we often encounter a unit that has been selected say more than 10 times? If you're curious about these issues (or you can never say no to an exercise in combinatorics!) then Computational Tip#6 is for you; otherwise, then you can skim through it but pay close attention to the paragraph that follows it where the lesson learned in the tip is restated.

Computational Tip#6

To help answer the questions that have lead us here, let P_{sample} stand for the probability that any given unit of the full sample gets to appear exactly r times in a bootstrap replicate of n-1 units selected from n units. Then,

$$P_{sample} = \binom{n-1}{r}\left(\frac{1}{n}\right)^{r}\left(1-\frac{1}{n}\right)^{n-1-r} \qquad (13)$$

(Indeed, out of the n-1 trials, there were r successes and each trial had a $1/n$ chance of resulting in a success *i.e.*, the selection of the unit.) In other words, the multiplicities are binomially distributed.

Similarly, let P_{domain} stand for the corresponding probability under bootstrap of $n_D - 1$ units selected from n_D ($<n$) units. (The example that lead us here used n=12 and $n_D = 5$, the domain D being the NLSCY.) This corresponds to bootstrapping only to the part of the sample that was found to pertain to the domain D. Then,

$$P_{domain} = \binom{n_D - 1}{r}\left(\frac{1}{n_D}\right)^r\left(1 - \frac{1}{n_D}\right)^{n_D - 1 - r} \qquad (14)$$

Here are a few interesting observations to be made about these two probabilities:

1) With the exception of a tiny domain size n_D (say < 10 units), both probabilities are (for all practical purposes) equal for the same given multiplicity r. In words, a given unit bootstrapped from either the full sample or the domain has equal chances of getting any given multiplicity!

2) For not-too-small n_D (and n), say, then one can show that both probabilities are (essentially) equal to $\exp(-1)/r!$. In other words, the distribution of multiplicities is well-approximated by the Poisson distribution with $\lambda = \frac{n-1}{n} \cong 1$. Actually, comparing (13) with n=20 to the Poisson yields:

r	P_{sample}	Poisson
0	0.3774	0.3679
1	0.3774	0.3679
2	0.1787	0.1839
3	0.0533	0.0613
4	0.0112	0.0153
5	0.0018	0.0031
6	0.0002	0.0005

They both say the same thing: don't count upon getting a multiplicity over 4 (let alone 10, as we considered in the discussion above that has lead us here) regardless of whether you re-sample 1,000 replicates from 20 or a trillion units.

But for variance purposes, this is not enough; it's not because re-sampling from 20 units or more yields the same (univariate) distribution of multiplicities that all is well. We need to make sure that p-variate vectors of multiplicities are also as likely to be observed and that's neither as simple nor as true as the univariate case. Yet, more importantly, by resampling within the domain we're guaranteed if a non-random sample size within replicates equal to n_D-1 whereas resampling done in the sample will lead to a random sample size within replicates for the domain D.

So, when facing the issue of re-sampling from a stratum with 12 PSUs, of which only 5 are non-empty, Computational Tip #6 tells us that we need to re-sample the 12 PSUs to have any chance of getting the variance estimation right. More generally, all PSUs that were consulted for the sampling of the NLSCY, regardless of whether they had something interesting or not for the survey, have to be accounted for in the variance. This means taking them into account to obtain the proper set multiplicities.

Computational Tip #7

While it's now clear that we need all sampled PSUs for re-sampling purposes, it's quite easy to forget all about the empty ones when actually merging the various survey files in the ongoing process of implementing the bootstrap. Indeed, in the timeline of a survey of NLSCY's scale, several months go by from the time a cohort is sampled to when its variance estimation is undertaken. Also, variance estimation is often carried out straight from estimation files (which need not include the empty PSUs). Add to all this staff movement and in practice things definitely *can* get missed. So, if one starts up all this from the survey weights file, then in all likelihood these PSUs will get missed. To prevent that from happening in NLSCY, a pseudo-household was created in the main internal file (called the Histoire file, for it keeps track of the trek of every kid and household into the NLSCY at any given time) that ensures a record for the corresponding empty PSU. This way we make sure that the PSUs we need to bootstrap are all accounted for and as soon as replication is done (*i.e.*, obtaining the set of multiplicities) we get rid of the pseudo-households.

5.2 Bootstrap weights

Once the file of multiplicities is obtained, at cycle 1 of a longitudinal survey like the NLSCY there's no other choice but to initiate the bootstrap using the design weights inherited from the LFS using (4). So, the preliminary (*i.e.*, before adjustment due to nonresponse) b^{th} bootstrap weight of unit k at cycle 1 is computed as:

$$w_{k,b}^{C1,initial} = \left(\frac{n_h}{n_h - 1}\right) \times mult_{k,b} \times w_k \qquad (15)$$

Then comes the nonresponse adjustment[7] to the b^{th} bootstrap weight of unit k by going through the Response Homogeneous Group (RHG) it was put in:

$$w_{k,b}^{C1,nr} = w_{k,b}^{C1,initial} \times \left(\frac{\sum\limits_{r+nr \in RHG} w_{j,b}^{C1,initial}}{\sum\limits_{r \in RHG} w_{j,b}^{C1,initial}} \right) \qquad (16)$$

It's when we reach cycle 2 (or any subsequent one) of a longitudinal survey that things can get messy pretty quickly if we don't make the proper turn at the crossroads. (In what follows we'll ground the discussion on cycle 2 vs. cycle 1 to simplify the exposé but the reasoning applies to a cycle i+1 that follows any cycle i.)

[7] Nonresponse and post-stratification in the context of the bootstrap will be discussed later. Nonresponse is only mentioned here because it's the factor in the NLSCY leading to cycle 2's set of units being a subset of cycle 1's.

To properly conduct the bootstrap at cycle 2, the bootstrap file constructed for cycle 1 has to contain the following information: identification, design weight, RHG affiliation, all B "basic" bootstrap weights, all B nonresponse adjustments, post-strata affiliation and all B post-stratification adjustments. Only with such a file at hand can we start up the bootstrap at cycle 2 where we left things up last time around *i.e.*, at cycle 1. If the cycles of a longitudinal survey were occurring closely in time, then this would be so natural to do that we'd have a hard time justifying why we're even talking about it! But in reality a lot of time elapses between successive cycles (in the case of the NLSCY two years go by in-between cycles) and maintaining such files with their huge number variables is an annoyance at best, one that many would be glad to go along without. So, what one often finds in practice is a bootstrap file, at a given cycle, containing only the B final bootstrap weights (along with identifiers, RHG and post-strata affiliations of course but *none* of the intermediate adjustments). This saves storage space. The proponents of that approach don't see the need to pursue the bootstrap cycle 2 from cycle 1's bootstrap weights: cycle 1 sample weights, as "start-up" weights, will do just fine. The rationale here is that since the sample based weights used to jump-start the bootstrap already encompass the nonresponse adjustment needed to go from cycles 1 to 2, then nothing is lost… right? Wrong.

The remainder of this section is to show why (and how) the above rationale is wrong: you can't use the survey (or sample, if you prefer) weights at a given cycle and think that the cumulative nonresponse adjustments they contain will suffice to capture the variance these adjustments have introduced so far in the estimator. There's no way around it: you need to pursue the bootstrap from the very beginning until the end, no matter how cumbersome that may be. (This does not literally mean that you need to process the bootstrap from the beginning of a longitudinal cohort for every cycle. Indeed, the file(s) about the latest cycle may very well do *provided* you took care to store in them all the required information to carry the bootstrap forward.)

The discussion which follows hinges on one principle: whatever way you choose to conduct your estimation (*i.e.*, obtaining sample weights), you must be sure that the methodology you've actually implemented will allow for the variance to be captured when replicated. Often, implementation shortcuts, like carrying forward estimation from the latest cycle, will work just for the sample weights: shortcut or not, the final weight is the same. This is a first-order moment criterion (*i.e.*, estimates to be obtained from those sample weights will be, on average, what they ought to be) but not a second-order one: estimated variance under shortcut or no-shortcut need not be equal. And this is what matters for variance estimation.

At cycle 2, ignoring for the moment incentives to post-stratify to focus solely on nonresponse, the <u>weight for the sample s</u> certainly can be computed in any of two ways:

$$w_k^{C2,s} = w_k \times (\text{C1's nonresponse adjustment}) \times (\text{C2's nonresponse adjustment}) \qquad (17)$$

i.e., start all the way from the LFS; or we could use the shortcut that we've already done some of this work for cycle 1's purpose:

$$w_k^{C2,s} = w_k^{C1,nr} \times \text{(C2's nonresponse adjustment)} \qquad (18)$$

where: $w_k^{C1,nr} = w_k \times$(C1's nonresponse adjustment).

It seems pointless to make the distinction between the two ways of computing a weight at cycle 2 *i.e.*, proceeding according to either (17) or (18); it actually is pointless *here* in the case of the sample, because we're dealing solely with point-estimation and *not* variance estimation. In other words, both approaches are equivalent as long as all we're interested in the end is getting the same point-estimate. But for the purpose of calculating bootstrap weights to capture adequately the variance introduced by the adjustment the two are not equivalent.

With many cycles under the survey's belt the advantage of the shortcut computation (18) over (17) becomes obvious. Now, remember we're told with the bootstrap we have to "replicate" what was done with the sample. Can we use here also the shortcut of starting our computations at cycle 2 from the sample-based weight obtained after cycle 1? (This is the trap we alluded to in the footnote 4 about item 4 of the implementation of the bootstrap.) The answer is a big "no"! Why? While it's true that cycle 1 sample-based weight does include the "first-order moment effect" of cycle 1's nonresponse through the adjustment it was exposed to, it doesn't capture what is required to evaluate the second-order moment of the adjustment. And there's no other way to capture that than to go all the way back to the LFS weight and have each replicate go successively through each wave of nonresponse. (Of course, if in cycle 1 bootstrap weights file all intermediate weights were kept one can start from that file but, as we've mentioned already, one often throws them away due to the resulting size of the file.)

Let's now see what the bootstrap weights would look like depending on whether they were obtained trying to replicate the long-form (17) or the shortcut (18).

After wave 1 of nonresponse we have:

$$w_{k,b}^{C1,nr} = w_k \times \frac{n_h}{n_h - 1} \times mult_{k,b} \times \left(\frac{\sum\limits_{r+nr \in RHG} w_j \times \frac{n_{h(j)}}{n_{h(j)} - 1} \times mult_{j,b}}{\sum\limits_{r \in wave1 \cap RHG} w_j \times \frac{n_{h(j)}}{n_{h(j)} - 1} \times mult_{j,b}} \right)$$

The notation $n_{h(j)}$ is used to make clear that the units from a given RHG need not come from the same stratum.

After wave 2 of nonresponse, starting "from the ground up" as we should, we get (17boot) which is the bootstrap counterpart of equation (17):

$$w_{k,b}^{C2,nr} = w_{k,b}^{C1,nr} \times \left(\frac{\sum\limits_{r \in wave1 \cap RHG} w_{j,b}^{C1,nr}}{\sum\limits_{r \in wave2 \cap RHG} w_{j,b}^{C1,nr}} \right) \qquad (17boot)$$

Observe that in (17boot) the starting point of the bootstrap weight for cycle 2 is the replicate-dependent bootstrap weight of cycle 1 (adjusted for C1's nonresponse): different replicates yield different starting points. This is the key to capturing the variance this adjustment has introduced in the overall picture. Let's see what we'd have instead if we initiated the bootstrap using C1's sample-based weight (adjusted for C1's nonresponse).

After wave 2, the shortcut-form for the bootstrap weights builds on the (sample-based) weight of the latest wave and leads to:

$$w_{k,b}^{C2,nr,shortcut} = w_{k}^{C1,nr} \times \frac{n_h}{n_h - 1} \times mult_{k,b} \times \left(\frac{\sum\limits_{r \in wave1 \cap RHG} w_{j}^{C1,nr} \times \frac{n_{h(j)}}{n_{h(j)} - 1} \times mult_{j,b}}{\sum\limits_{r \in wave2 \cap RHG} w_{j}^{C1,nr} \times \frac{n_{h(j)}}{n_{h(j)} - 1} \times mult_{j,b}} \right) \qquad (18boot)$$

Note: The notation used here for RHGs is admittedly clumsy; there's a dependency on the cycle that ought to be acknowledged somehow in the last few equations and instead "RHG" is used throughout here as a generic term. This was intentional. Indeed, since knowing *what* the RHGs really are is totally secondary to the issue here, we preferred to keep things "conceptually" simple by not cluttering up equations with additional notation.

Observe that contrarily to the long-form (17boot), the cycle-2-start-up weight $w_k^{C1,nr}$ in (18boot) is the same for all replicates. In other words, with this shortcut- form, whatever effect cycle 1's nonresponse may have had on the estimates is captured solely through the effect it had on the sample (*i.e.*, first-order or "average" effect), not on each of the replicates (*i.e.*, second-order). This is a crucial difference with (17boot).

Again, one may proceed this way because he/she feels that keeping all the (relevant) intermediate bootstrap information is just a waste of space and (processing) time. Failing to have that information, one tries to compensate by supplying instead sample-based information but this is far from being enough.

Here are two additional ways (other than the main one given above on first-/second-order-moments considerations) to see why the shortcut is inappropriate for variance calculations; just pick the one that you find more telling.

To further help compare the two algebraic expressions for the bootstrap weight we get after wave 2, we'll consider a particular case by assuming that wave 2 of nonresponse is actually a ghost: no nonresponse at cycle 2 has actually occurred. After two waves of collection we're left with those respondents of wave 1; therefore, as far as weighting is concerned wave 2 never has occurred. In this case we still have two choices on how to conduct our bootstrap for wave 2: start from the design weight (*i.e.*, from ground up) or from wave 1's nonresponse adjusted weight (*i.e.*, shortcut). If the two approaches were equivalent, we should be indifferent to which one we pick in this special case.

So, the equivalent of (17) in this special case (sc) is:

$$w_{k,b}^{C2,nr} = w_k \times \frac{n_h}{n_h - 1} \times mult_{k,b} \times \left(\frac{\sum\limits_{j \in r+nr \cap RHG} w_j \times \frac{n_{h(j)}}{n_{h(j)}-1} \times mult_{j,b}}{\sum\limits_{j \in r \cap RHG} w_j \times \frac{n_{h(j)}}{n_{h(j)}-1} \times mult_{j,b}} \right) \underbrace{\times 1}_{wave2} \tag{17sc}$$

And the equivalent here of (18) is then:

$$w_{k,b}^{C2,nr,shortcut} = w_{k,b}^{C1,nr} \times \frac{n_h}{n_h - 1} \times mult_{k,b} \times \underbrace{\left(\frac{\sum\limits_{j \in r+nr \cap RHG} {}_{C1}w_j^{sample,nr} \times \frac{n_{h(j)}}{n_{h(j)}-1} \times mult_{j,b}}{\sum\limits_{j \in r \cap RHG} {}_{C1}w_j^{sample,nr} \times \frac{n_{h(j)}}{n_{h(j)}-1} \times mult_{j,b}} \right)}_{wave2=1} \tag{18sc}$$

Let's rewrite $w_{k,b}^{C1,nr}$ in (18sc) to expose the design weight w_k as to make the two expressions (17sc) and (18sc) more directly comparable:

$$w_{k,b}^{C2,nr,shortcut} = w_k \times \left(\frac{\sum\limits_{j \in r+nr \cap RHG} w_j}{\sum\limits_{j \in r \cap RHG} w_j} \right) \times \frac{n_h}{n_h - 1} \times mult_{k,b} \tag{18sc'}$$

Comparing (17sc) and (18sc') one notices that the nonresponse adjustment is different: starting from the design weights the multiplicities (in 17sc) are factored in while they're left out in (18sc'). This means that in the shortcut the adjustment for nonresponse in a replicate is *independent* of the replicate. Instead of having a nonresponse adjustment that fluctuates from one replicate to another the shortcut uses "their average" instead.

Another way of reaching the same conclusion (which you may find more methodologically intuitive and *less* dependent upon equations) is not to see the set of bootstrap weights obtained through the shortcut as inadequate for our estimator but rather find the actual estimator for which this would be an appropriate way to compute its variance. It will then become apparent that the estimator we obtain this way is different from the one we're using, hence the error. Consider

$$\sum_s w_k y_k \frac{N_{RHG}}{N_{RHG}^s} \tag{19}$$

Here $N_{RHG} = \sum_U I(RHG)$ is the known count (taken from the frame) for the RHG and N^s_{RHG} is the corresponding count estimated using the *observed* sample *s*. A "hat" was *not* put onto N^s_{RHG} to emphasize that when calculating the variance of (19) we don't intend to re-compute sample after sample this component: we'll use in all calculations the one ratio we got using the sample we *observed*. N^s_{RHG} is not known prior to sampling but once *s* is obtained we'll regard this quantity from now on as known and we'd use it as such in variance computations. Compare that to the estimator:

$$\sum_s w_k y_k \frac{N_{RHG}}{\hat{N}_{RHG}} \tag{20}$$

The "hat" in $\hat{N}_{RHG}$ puts emphasis on the fact that we intend to re-compute this quantity sample after sample in order to compute its variance.

Now, while both estimators have approximately the same expected value with regard to the design, they don't have the same variance.

The morale here is this: the shortcut-form of the weighting is not the one to replicate and, more generally, while the bootstrap is about doing to each of the replicates what was done with the sample, one must make sure that whatever shortcut was first used with the sample *does* capture adequately the corresponding variance component *when* replicated.

In a few words…

- Bootstrap must be carefully investigated to ensure that the proper replication is made *i.e.*, variance is properly estimated through bootstrapping. This was illustrated by the issue of whether we can just bootstrap the domain of interest of a survey (*e.g.*, the NLSCY) or we actually need to bootstrap the entire sample (*e.g.*, the LFS).

- Shortcuts in creating weights must be carefully assessed. They are not appropriate for variance computations purposes *if* they don't capture the second-order moment of the intended adjustment to the weights. Indeed, shortcuts are introduced to simplify the implementation of the weighting methodology and for that purpose they are (usually) adequate. But this in itself does not mean that they are adequate substitutes from a variance perspective; this is the pitfall awaiting the methodologist when using shortcuts. In the NLSCY, we've come to affectionately describe the methodology of using shortcuts with the bootstrap as "shortstrap". So, the bottom line is: be very careful whenever you shortstrap!

6. DOMAIN ESTIMATION

In this short section we investigate how the theory of domain estimation is carried out with regard to the bootstrap; we'll try to dissipate some of the confusion there seems to exist about the mechanics of its implementation.

We've already tackled one important part of this in Computational Tip #6 where we saw that we need to bootstrap the sample not the domain, *even* if the domain is all we're ever interested in. While point-estimation will be the same either way (*i.e.*, shortcut or no shortcut), it remains that variance estimation requires, to some extent, the full sample. Indeed, one consequence of re-sampling n_D-1 units from a "sample" of n_D units in the sampled domain is that the sum of the multiplicities in any given bootstrap replicate will be exactly n_D-1. If we instead resample n-1 units from n, then the portion of a domain in any given replicate will be random in size. This is the way the bootstrap will penalize you for conducting the estimation on a domain rather than on a stratum.

Now, suppose we want to estimate a domain total

$$Y_D = \sum_{k \in D} y_k$$

then the estimator to use is:

$$\hat{Y}_D = \sum_{k \in s \cap D} w_k y_k$$

In practice, this estimate will *numerically* match with the estimate provided by the following:

$$\hat{Y}_D^{(1)} = \sum_{k \in s} w_k^* y_k$$

where: $w_k^* = \begin{cases} w_k \text{ if k} \in \text{D} \\ 0 \text{ if k} \notin \text{D} \end{cases}$.

Note that the estimate $\hat{Y}_D^{(1)}$ is constructed from the entire sample *but* taking care first to set the weight of units outside of D to zero. There's nothing sacred about setting the weights of units outside of D to zero *and* keeping intact the corresponding y-values; we could have actually chosen to do the opposite and set the y-values to zero instead:

$$\hat{Y}_D^{(2)} = \sum_{k \in s} w_k y_k^*$$

where: $y_k^* = \begin{cases} y_k \text{ if k} \in \text{D} \\ 0 \text{ if k} \notin \text{D} \end{cases}$.

While the end result is the same both ways (*i.e.*, you get the same numerical value either way), none is conceptually satisfying: they involve tampering with either the weights or the *y*-values. What we should compute instead is:

$$\hat{Y}_D = \sum_{k \in s \cap D} w_k y_k = \sum_{k \in s} w_k y_k I_k(D)$$

where: $I_k(D) = \begin{cases} 1 \text{ if } \mathrm{k} \in \mathrm{D} \\ 0 \text{ if } \mathrm{k} \notin \mathrm{D} \end{cases}$.

This form neutralizes the units outside of the domain with the help of an indicator variable and avoids tampering of any kind with either the weights or the *y*-values.

The corresponding b^{th} bootstrap estimate would then be constructed as:

$$\hat{Y}_D^{(b)} = \sum_{k \in b} w_k^{(b)} y_k I_k(D)$$

Again, all three ways to compute the estimate here are *numerically* equivalent for a *total*. But we believe the latter form is more satisfying conceptually since 1) we don't mess with either the weights or the *y*-values in order to "neutralize" the contribution to the estimate of units outside of the domain *D*; 2) it's more easily exportable to other estimators.

7. NONRESPONSE AND TWO-STAGE SAMPLING

In this section we investigate the bootstrap's capacity to capture adequately the total variance arising from either nonresponse or a second-stage of sampling. Let's first focus on nonresponse. Many people take the usual motto of bootstrap "do with the replicate all that was done for the sample" to mean that all that is needed to fully capture the component of variance due to nonresponse is to replicate the nonresponse adjustment. Popular belief as it is, it's not adequate. While there are situations where the bootstrap *appears* to capture well the nonresponse contribution to variance (*e.g.*, a negligible sampling fraction), it's deceptive: even in those favourable situations the bootstrap misses one component of the total variance. It just happens that, in those situations, the component that is missed is so small that it can legitimately be ignored.

The bootstrap's motto never was meant for nonresponse (or a further stage of sampling for that matter), but rather was intended for methodological processes like post-stratification with no "randomness of its own". While it's true that post-stratification involves randomness (*i.e.*, the sample size that will materialize itself within a given post-stratum is random), it's entirely determined as a process once the sample has been observed. In other words, conditionally on the observed sample, post-stratification no longer is a random process though nonresponse (or a second stage of sampling) still is random even when the sample is observed. Actually, the nonresponse model that you adopt is a description of the randomness which is introduced by the nonresponse and remains even after the sample has been observed.

7.1 Stochastically modelled nonresponse and second-stage sampling components to variance

The reader may find odd that this sub-section puts nonresponse and a second-stage of sampling into the same pot. Odd as it is there's no need to treat them differently when it comes to variance estimation. First, let's start by saying that two grand sampling schemes are the two-phase sampling and the two-stage sampling. To help tell one term from the other, it is customary to over-simplify and describe two-phase sampling as sampling in two steps the same type of units (say, individuals at both steps) while two-stage sampling involves different types of units from one step to the other (say, hospitals in the first step and doctors in the second step). A more substantial *methodological* distinction between the two would emphasize the fact that a two-phase sampling design is about gaining efficiency despite a poor-on-auxiliary-data frame to begin with. Indeed, ideally one could say that had the frame been rich in information, one phase would have sufficed. But since it's not the case, we gain from a preliminary phase where the frame gets enriched so that the second phase can then be "efficient". Two-stage sampling does not have the benefit of a frame comprising the units of interest, even one poor in content would have been welcome! In such a case we reach the units of interest by constructing intermediate frames from which we sample (*e.g.*, a frame of all hospitals and within selected hospitals a frame of all their doctors, assuming here we're after doctors).

How does nonresponse fit into that at all? Traditionally, to help capture the variance component due to nonresponse, the nonresponse mechanism given the observed sample is modelled *as if* it operated on the sample the way a (random) sampling mechanism would. In other words, sample first according to some design followed with an episode of nonresponse is treated for variance purposes as if two-phases of sampling had taken place: the first phase covers the (intentional) sampling by the methodologist and the second phase describes the (un-intentional) sampling forced upon the methodologist by nonresponse. (See Sections 9.8 and 15.6 of Särndal *et al.* (1992) for a detailed exposé of the two-phase model for nonresponse.) Bottom line: mathematically, nonresponse's impact on variance is gauged through a two-phase sampling design.

To summarize, two-phase sampling (of which a one-phase survey facing nonresponse is an important case) and two-stage sampling certainly are different from a *methodological* perspective. But the key aspect here is that *mathematically*, both designs are covered from one and the same inferential setting: the two-phase *framework*. The latter serves to track the variance of a process having two phases of random sampling. (The reader may also want to read Section 9.1 of Särndal *et al.* (1992).) Loosely speaking, the two-phase framework allows us to consider random processes like nonresponse where given the observables of the first phase, randomness is still at work to yield the second phase observables (*i.e.*, the observables of the second phase are not entirely determined now that phase one has occurred).

In the two-phase framework, you can decompose the total variance component into two; one of such decomposition known as reverse[8] two-phase is due to Fay (1991) and can be stated as (see equation (4) of Shao and Steel (1999)):

$$V_t(\hat{\theta}) = \underbrace{E_{PH2}V_{PH1}(\hat{\theta})}_{V_1^{rev}} + \underbrace{V_{PH2}E_{PH1}(\hat{\theta})}_{V_2^{rev}} \qquad (21)$$

Note: The inner expectation (respectively, variance) is an expectation (respectively, a variance) conditional on the observables of phase 2.

At first glance, this reverse two-phase may appear quite twisted from a methodological standpoint, not to mention nonsensical: how can phase two observables be available *before* we even got phase 1's observables? While not all nonresponse models will allow such a decomposition to take place, most will. What is merely required of the model for the reverse approach to (mathematically) work is that a unit probability of response is assumed to be independent of other units. In other words, as soon as a given unit is in one sample or another it is assumed that its *true* probability of response is the same

[8] Briefly put, this approach treats nonresponse *as if* the initial sampling was carried out on two sub-populations: one of respondents and one of nonrespondents. This is obviously bogus from a methodological perspective but then *mathematically* it makes no difference one way or the other. Consequently, one picks in a given context the one mathematical formulation which is easier to work from and as it turns out the reverse approach does wonders to account for the nonresponse component to variance.

(though in practice different samples this unit is in may yield different estimated probabilities of responding, but this is a separate issue).

Methodologically twisted as it looks, you must realize one thing: what only matters sometimes (and it's the case here) is the mathematics of an argument, not the way it would look if implemented. For any given methodology, its practical implementation and its underlying mathematics are two very separate things. In order for the mathematical development of a methodology to be valid, it need not correspond to the way one would actually go about implementing it in practice. So, in the end, if it's mathematically simpler for us to decompose a mathematical entity like variance in what is methodologically a reverse way, then let's by all means make progress in our variance calculations by going full reverse! ☺

The crucial fact to know about that variance decomposition (21) and the bootstrap is the following: the bootstrap can (essentially) only act as an unbiased estimator of the first variance component V_1. (We'll see why in Example 1 below.) So what about the whole of V_t? If the sampling fraction at phase 1 is negligible then as it turns out V_2 can be considered to be tiny compared to V_1, small enough to be ignored (or "unbiasedly" estimated by 0 if you prefer); this means that the bootstrap which estimates only V_1 is essentially unbiased for V_t as well. For the practitioner this means that the bootstrap performs well for a one-phase design survey dealing with nonresponse or a two-stage survey if the sampling at first phase/stage is small. If, on the other hand, the sampling fraction at phase 1 is not so small, then V_2 also becomes non-negligible. And to make matters worse, the smaller the sampling fraction at phase 2 the larger the component V_2 becomes.

For the user this spells trouble in two cases: 1) when sampling within strata involves large sampling fractions and nonresponse occurs; 2) the survey relies on a two-stage design with sampling at first stage involving a large sampling fraction of PSUs. When this happens, the component to the total variance missed by the bootstrap can no longer be neglected; there's currently nothing in the literature on the bootstrap that can be used to compensate: we still don't know how to bootstrap adequately in presence of nonresponse. Our current knowledge allows us to capture only part of the total variance, and often times this is enough.

A simple example of two-phase sampling can be investigated to help see what is going on. The SAS code behind Example 1 is provided in Program G. The reader fluent in French will find a thorough discussion along those lines in Haziza (2005).

Example 1

Consider drawing in phase one of a survey a sample of size n using SRSWOR followed at phase two by nonresponse occurring totally at random and yielding r respondents. (Without going into the details, the latter assumption on the response mechanism ensures that the reverse approach will work here.) In this case, one can show that in this context an unbiased variance estimator of the mean is the one we usually get under SRSWOR with sample size equal to r (see Exercise 9.14 of Särndal *et al.* (1992)). We can get the corresponding decomposition that suits this situation by adapting equations 15 and 17 from Shao and Steel (1999). Actually, they considered the case of SRSWOR with ratio imputation performed for nonrespondents; one recovers our particular situation by noticing successively that: 1) imputation by the mean is a special case of imputation by a ratio (set $x_i = 1$ for all i, in which case $\bar{x} = x_i$ and $s_x^2 = s_{dx} = 0$); 2) imputation by the mean within the sample amounts here to re-weighting[9]:

$$V_t(\hat{\theta}) = \underbrace{E_r V_D(\hat{\theta}|r)}_{V1} + \underbrace{V_r E_D(\hat{\theta}|r)}_{V2} = \underbrace{\left(1 - \frac{n}{N}\right)\frac{s_r^2}{r}}_{V1} + \underbrace{\left(\frac{n}{N} - \frac{r}{N}\right)\frac{s_r^2}{r}}_{V2} \tag{22}$$

Note: E_r(respectively, E_D) is the expectation with regard to the response mechanism (respectively, design) and the conditioning is done with respect to the observed set of respondents r (respectively, observed set of sampled values s) and similarly for the variances. (Note: The notation "r" used describes either the set of respondents or the size of that set; the very place where "r" enters (22) should make it clear which of the two it represents.)

In this specific example it's easier to see the connection between the bootstrap and the variance component V_1 we alluded to above. So, let's say we seek using our one-and-only observed sample s an estimate of

$$V_1(\hat{\theta}) = E_r V_D(\hat{\theta}|r) \tag{23}$$

Since in practice all we'll ever have is *one* estimate $\hat{V}_D(\hat{\theta}|r)$ of $V_D(\hat{\theta}|r)$, this implies that our best estimate of $E_r V_D(\hat{\theta}|r)$ is indeed then just $\hat{V}_D(\hat{\theta}|r)$. (When you have only one observed value x of a random variable X for which you seek an estimate of *E(X)*, then the best you can do is estimate it by x.) Now, the conditioning in $\hat{V}_D(\hat{\theta}|r)$ is there to remind us that we're working from a *given* set of respondents r; sure, we *could* be observing other sets under the response mechanism but fact is, in practice, we're given *that r*. This means that although the response indicator associated to each unit is a random variable, the conditioning that was done makes the observed response indicator a typical survey (*i.e.*, fixed or non-random) variable. And the bootstrap is meant to act as a design-based estimator of variance of a quantity made of fixed variables *i.e.*, an estimator of $V_D(\hat{\theta}|r)$.

Using the SAS program provided in Program G, the empirical results in Table 3 below were obtained which support (at least in the case of SRSWOR and

[9] If this statement comes in as a surprise, then take a look at Annex A.

uniformly occurring nonresponse) the trends depicted above in terms of sampling fractions at both phases. Note just how in Program G the nonresponse was dealt with in each replicate. Practitioners believe that this ought to be enough to capture the whole variance V_t in *any* setting but it's not whenever the sampling fraction at phase 1 is non-negligible *even* when the weights (9) are used. Had we not processed the nonresponse adequately in the replication program we would have observed that the ensuing "bootstrap variance estimate" doesn't even estimate V_1 well. So, *all* we accomplished in the program by feeding the replicates through the nonresponse methodology was to keep track of V_1: V_2 still evades us.

f	r	V_1	V_2	V_t	v_{bs}
10%	100%	0.3759	0	0.3759	0.3772
	80%	0.4663	0.0099	0.4762	0.4697
	50%	0.8030	0.0448	0.8478	0.8613
25%	100%	0.1219	0	0.1219	0.1213
	80%	0.1515	0.0101	0.1616	0.1521
	50%	0.2590	0.0443	0.3032	0.2662
50%	100%	0.0399	0	0.0399	0.0418
	80%	0.0490	0.0096	0.0586	0.0498
	50%	0.0788	0.0395	0.1183	0.0823
75%	100%	0.0131	0	0.0131	0.0136
	80%	0.0163	0.0098	0.0262	0.0164
	50%	0.0261	0.0388	0.0649	0.0272

Table 3: Variance components of V_t in equation (22) as functions of the sampling fraction at first (*f*) and second (*r*) phases

In Table 3: observe notably that:

- if there's no nonresponse, then the bootstrap captures well the overall variance V_t (which is only in fact V_1, V_2 being negligible here);
- for a given *f*, the smaller the *r*, the larger the second variance component V_2;
- the second variance component V_2 only becomes an issue (compared to V_1) when the sampling fraction at first phase *f* is not small.

The morale of Example 1 is clear: the bootstrap's motto does not capture the entire nonresponse component to variance. It is therefore not enough to bootstrap the sample and have the replicate go through the nonresponse adjustments and think you've captured all of the variance you were after.

But the motto *does* work for post-stratification and domain estimation because contrarily to nonresponse, the post-stratification and domain estimation we carry out are entirely determined once the sample is drawn. Let's consider in more details the case of post-stratification. More precisely, let's suppose the sampling design is SRSWOR and post-stratification has given rise to G post-strata. In such a situation, the advocated post-stratified estimator for the mean of a variable of interest Y is (see Särndal *et al.* (1992), on the SI design in Section 7.6):

$$\hat{\bar{y}} = \sum_{g=1}^{G} \frac{N_g}{N} \left(\frac{\sum_{s_g} y_k}{n_{s_g}} \right) \tag{24}$$

We're not going to decompose the total variance according to the reverse approach but rather using the direct two-phase approach this time; the latter is more natural in this case since there's really here just one phase of sampling. It spells out as:

$$V_{total}(\hat{\bar{y}}) = \underbrace{V_{SRSWOR} E_{POST-STR}(\hat{\bar{y}}|s)}_{V_1^{DIR}} + \underbrace{E_{SRSWOR} V_{POST-STR}(\hat{\bar{y}}|s)}_{V_2^{DIR}} \tag{25}$$

The second component V_2^{DIR} is 0 because once the sample is given no randomness remains (*i.e.*, given s the only quantity that was random to start with, n_{s_g}, is now a constant.). In other words, V_2^{DIR} is 0 because $V_{POST-STR}(\hat{\bar{y}}|s)$ is 0. For the same reasons, $E_{POST-STR}(\hat{\bar{y}}|s)$ simply is $\hat{\bar{y}}$, so that

$$V_{total}(\hat{\bar{y}}) = V_{SRSWOR}(\hat{\bar{y}})$$

A similar reasoning will show that with domain estimation, as with post-stratification, there's not truly two phases of randomness: the first-phase sampling "is responsible" for it all. As a result, if we're interested in an estimator[10] of a domain D mean like (see Särndal *et al.* (1992), Section 10.3)

$$\hat{\bar{y}}_D = \frac{\sum_{s_D} w_k y_k}{\hat{N}_D = \sum_{s_D} w_k}$$

then the bootstrap ought to capture all of its variance. (See Program H for the SAS code which tests this.)

Now that the applicability of the bootstrap's motto has been clarified, does it mean we never were meant to capture the whole variance V_t under nonresponse even when the design is SRSWOR? Actually we could provided we "bootstrapped the right design" or succeeded in conducting two bootstraps: one for each component of (22).

[10] If you find puzzling that the estimator proposed is this one regardless of whether the domain size N_D is known or not, then you may want to read Annex A.

Let's first see what is meant by "bootstrapping the right design" by considering the case of SRSWOR yielding a sample of size n followed by uniformly occurring nonresponse which yields a set of respondents of size r. This is the framework in Example 1 above. We saw that bootstrapping the SRSWOR design and then feeding the replicates through the nonresponse methodology devised on the sample was not the way to go. But we have already noticed in Example 1 that, in this case, the resulting set of respondents r could be treated as if it had been obtained *directly* from a SRSWOR design yielding a sample of size r. Having concluded *that* we can take advantage of it and bootstrap directly the set r of respondents. This would imply computing the weights according to

$$w_{k,b} = \frac{N}{r}\left(1-\sqrt{1-\frac{r}{N}}+\sqrt{1-\frac{r}{N}}mult_{r,k,b}\,\frac{r}{r-1}\right) \tag{26}$$

rather than starting with

$$w_{k,b} = \frac{N}{n}\left(1-\sqrt{1-\frac{n}{N}}+\sqrt{1-\frac{n}{N}}mult_{n,k,b}\,\frac{n}{n-1}\right) \tag{27}$$

and have the weights go through the nonresponse adjustments as we did in Example 1 above. The obvious problem with this solution is that a multi-phase framework usually doesn't conveniently "collapse" into a one-phase well-known design as it was the case here.

Another tentative solution is to have *two* bootstrap processes, one for each component of (22). To help see how this could work, let's re-write the second component of (22) as:

$$V_t(\hat{\theta}) = \underbrace{\left(1-\frac{n}{N}\right)\frac{s_r^2}{r}}_{V1} + \underbrace{\left(\frac{n}{N}-\frac{r}{N}\right)\frac{s_r^2}{r}}_{V2} = \underbrace{\left(1-\frac{n}{N}\right)\frac{s_r^2}{r}}_{V1} + \underbrace{\left(\frac{n}{N}\right)\left(1-\frac{r}{n}\right)\frac{s_r^2}{r}}_{V2}$$

So, the second component of the variance for the <u>mean</u> can be captured by a bootstrap procedure of its own because it corresponds to a SRSWOR that draws r units from the observed sample of n units (with the drawn sample playing the role of the population here) with the sampling fraction at phase 1 as an adjustment factor in front of it all. This is implemented in Program G2. As a consequence, for the <u>mean</u>, to get both variance components of (22) you need to run Programs G (for V_1) and G2 (for V_2) and add the results. The problem with this approach is that we're totally dependent on the mean for the decomposition of the variance (not to mention the dependence on the design SRSWOR). While (22) tells us how the two fit into the total variance there's no reason for that decomposition to hold for estimators other than the mean. Indeed, one can use Programs G and G2 for the median and show that the variance estimates obtained don't add up to that of the median obtained under SRSWOR of r units drawn from N.

7.2 Replicating the nonresponse adjustment

We've already encountered nonresponse as the factor in a longitudinal survey like the NLSCY which is responsible for the attrition in sample sizes observed as cycles of collection take place.

To capture the bulk of the (total) variance attributed to the nonresponse adjustment under a without-replacement design with small sampling fraction using the bootstrap, the best known practices call for the replication of the adjustment[11]; this should yield the best estimate possible of V_1 under the circumstances (though not, as argued above, the whole of V_t). Many surveys, NLSCY included, rest their nonresponse methodology on the concept of Response Homogeneous Group (RHG). A RHG is, by definition, a group of units which are deemed to have the same propensity to respond. A RHG can notably be obtained by crossing several variables or by grouping predicted response probabilities from a logistic regression model fit to the data at hand.

So, in concordance with the best known practices, we need to compute in each RHG the following adjustment for each replicate b:

$$\text{“Replicate-based” nonresponse adjustment} = \frac{\sum_{r\in b} w_k + \sum_{nr\in b} w_k}{\sum_{r\in b} w_k} \tag{28}$$

Computationally, this is quite involved; it requires evaluating two quantities for each replicate: the numerator of the adjustment for a given replicate along with the corresponding denominator. This is not that big of a deal in the current era of powerful computers but for surveys whose variance methodology based on the bootstrap was implemented a while ago this could very well have been an issue. So, it's possible that in order to reduce computations one computed the numerator using the sample instead of the b^{th} replicate:

$$\text{“Sample-assisted” nonresponse adjustment} = \frac{\sum_{r\in s} w_k + \sum_{nr\in s} w_k}{\sum_{r\in b} w_k} \cong \frac{\sum_{r\in s} w_k^{nr}}{\sum_{r\in b} w_k} \tag{29}$$

where: w_k^{nr} is the design weight adjusted for nonresponse occurring in the *sample*.

The latter is called "sample-assisted" because the numerator of the adjustment is now made *independent* of the replicates and is determined solely by the sample. This approximation is attractive on the grounds that the average of all adjustments for a given

[11] Actually, some argue that we should replicate the entire nonresponse methodology by revisiting the RHG construction every time (like re-estimating within each replicate the logistic model parameters). If truly needed, then this would be very time consuming since the construction of RHGs often can't always be easily automated as it requires lots of specific decision making and investigation. (See Faucher *et al.* (2003) for a discussion on this.)

RHG over all replicates should be indeed about (28). In other words, the first-order moment of (28) is (essentially) preserved by using the less computer-intensive (29). Also, it doesn't require messing around with the nonrespondents. But here's the catch: we're after variance estimation *and* to be adequately captured, variance requires that an adjustment preserves the *second*-order-moment, in addition to the first. So, as attractive as it is, the adjustment (29) doesn't preserve the second-order moment at all and for variance this is a big "no". In other words, there's variability in the adjustment (to be captured as variance) that comes from having a replicate-dependent numerator. (You should by now have a strong feeling of déjà vu: this shortcut of building the bootstrap nonresponse adjustment (partly) on the sample is a close-cousin of the shortcut described in Section 5 about initiating the bootstrap at a given cycle from the sample-based weight of the previous cycle.)

To help see just how (29) is inadequate in preserving the second-order moment of the adjustment (28), consider that we only have 5 replicates to work from and one RHG; the following adjustments were computed according to (28) and (29):

Replicate	(28) *i.e.*, correct	(29) *i.e.*, erroneous
1	110/92	100/92
2	105/91	100/91
3	100/90	100/90
4	95/89	100/89
5	90/88	100/88

Table 4: Numerical example illustrating the difference between adjustments (28) and (29).

Observe that while the average of the 5 adjustments (over the 5 replicates) is about the same for both adjustments (*i.e.*, first-order moment preservation), their *spread* about that common average is far from being the same (*i.e.*, (29) fails to capture the second-order moment of (28)).

The consequence of using (29) instead of (28) can be pretty dramatic and is best seen from a post-stratification standpoint. Indeed, while a true post-stratification adjustment has as its numerator some known total (from an external-to-the-survey source), (29) uses an *internal* total. In other words, the RHGs were used *as if* they were post-strata. Contrarily to post-stratification, the internal-to-the-sample total is not known prior to sampling; but once the sample is drawn we can think of calibrating our respondents-only weights to it. (Why we'd want to do that is a totally different issue!) Post-stratifying, either to an external or an internal total, has one effect on variance: it brings it to zero for count estimates of domains which are a (direct) sum of post-strata. Incidentally, some people feel uneasy about reporting a zero variance under post-stratification. A zero variance is correct in the case of the usual post-stratification because we assume the known total to be true (*i.e.*, there'd be no bias introduced by aiming at *that* target because it's assumed to be *the* target). Consequently, our best estimate sample after sample ought to be that calibrated-to total, hence no variance and our claim that our post-stratified estimator is unbiased (*i.e.*, no overall sampling error). So, any uneasiness about zero variance here directly spells out as concerns about the known total to be true. But it is wrong for an internal total: what guarantees do we have that this sample-based total is true? None, whatsoever. By forcing all the replicates to yield the sample-based estimate of the RHG total (which can only coincide with the true total by mere luck) our estimator becomes *biased* though it has zero variance.

Indeed, suppose the domain of interest coincides with the RHG (our makeshift post-stratum). In such a case we have the following replicate-based estimate of the number of units in the domain (which, we re-emphasize, is the RHG here):

$$\cancel{\sum_{k\in D\cap resp\cap i} w_k} \times \frac{\sum_{j\in D\cap resp} w_j^{nr}}{\cancel{\sum_{j\in D\cap resp\cap i} w_j}} = \sum_{j\in D\cap resp} w_j^{nr} \qquad (30)$$

Equation (29) thus leads to a replicate-*independent* estimate which is equal, by construction, to the sample-based one. In the example above, this means that the 5 replicate-based estimates would be 110,105,100,95 and 90, if (28) was used, and 100,100,100,100 and 100, if (29) was instead used, hence a reported bootstrap variance of 0 in this latter case. If the bias introduced is not properly gauged, then the estimated variance as a stand alone estimate of the (total) sampling error becomes absurd.

This is the chief reason why the variance estimate in Table 1 at cycle 2 is so low; had it not been for the usual post-stratification that was put on top of that and thus brought in some noise, the estimated variance reported for cycle 2 would have been zero! Indeed, the domain variable *alone* happened to play a very special role in the methodology of the NLSCY: it is the one (categorical) variable behind the construction of RHGs in the Atlantic for cycle 2!

7.3 On the construction of RHGs

While the fashionable trend is to form RHGs by grouping predicted response probabilities arising from a logistic model fitted to the data, many surveys still rely on chi-square detection of distortions in distributions. The latter methodology may not sound familiar to you but the software that is often used to carry it out may ring a bell: *KnowledgeSeeker*. Loosely speaking, the chi-square-based methodology identifies variables whose distribution of values after nonresponse represents a distortion of what it was before nonresponse occurred. For example, say men and women in a given survey were of equal number before nonresponse only to have men dominate the set of respondents; this is an instance of a distortion that could be detected using *KnowledgeSeeker*.

The danger with such a methodology is to let the tool get out of hand by creating a huge number of RHGs. For instance, at cycle 1 of the NLSCY, there were well over 400 RHGs. Sorting first in ascending order all the ensuing nonresponse adjustments, one can then compute the average relative difference between two consecutive adjustments. For example, successive adjustments of 1.1 and 1.2 differ by 9.1%. The average relative difference computed for cycle 1's adjustments is 0.09%!! The huge number of RHGs, and incidentally the limited number of units they're bound to contain, is a contributing factor to unstable variance estimates across cycles. Indeed, too small a RHG will present highly variable nonresponse adjustments from one replicate to another depending on the nonresponse composition of each replicate. This further becomes an issue when considering that nonresponse adjustments in a longitudinal survey get compounded across cycles, and thus RHGs are to become smaller and smaller over successive cycles.

Whether it's through *KnowledgeSeeker*, logistic regression or what have you, nonresponse methodology is all about *modelling*. For instance, if for your cross-sectional survey you establish two RHGs based on gender, one for each, then your nonresponse model has one factor, gender. It's customary to hear practitioners say in such a situation that they've found gender to *explain* the occurring nonresponse. The issue with this terminology is that it's too emotionally charged. If *explain* is used as in "the set of explanatory (dependent) variables of the model", then there's no harm in it. But people get easily caught up in the terminology and lose sight of what the modelling circumstances (or model assumptions if you prefer) were. For instance, after explaining nonresponse through gender an estimate of age, say, was produced for which the true total is known as the variable lies on the survey frame, and the two don't match: "Isn't that embarrassing given that gender was meant to explain nonresponse and obviously it doesn't explain *everything*? There's still some explaining to do!" The way out is simple: nonresponse was modelled through the effect it had, in this case, on gender (and nothing else). Was it a bad choice of leaving all other variables out? If so then this was a bad modelling exercise. And if not, then it's a collateral damage: the model can only be expected (and you knew about it all along) to correct for what it explicitly accounted for. As far as modelling goes, you just don't get something for nothing.

To help keep the proper perspective on the modelling exercise, here's another point of view. Don't perceive nonresponse modelling as explaining nonresponse but rather as putting to light effects it has had on your ability to recover stuff that was known (and deemed important) prior to nonresponse. For example, before nonresponse say we were able to get count estimates of both men and women; let's suppose these summed to a ratio of roughly 2:1 in favour of women: we have an estimate of women twice as large as that of men. While these totals may not be the true values (which they'd be if we'd had post-stratified on gender), we may feel that the sample is the best information we've got and in particular that the rough ratio of 2:1 is to be trusted (short of putting our hands on the true totals). And then nonresponse comes and creates one big mess: the ratio in estimates using respondents only is now 1:1, *not* 2:1. *If* we feel strongly about our earlier estimates on gender (and more generally the distribution we obtained for gender in the population) and about the relevance gender has in our survey, then we'll want to rectify this "imbalance" introduced by nonresponse. We do that by proposing a model based on gender *i.e.*, a meaningful way to summarize the *effects* nonresponse has had on gender. Therefore, the model is simply seen as a mathematical way to acknowledge the shift in the distribution of gender we've noticed with the advent of nonresponse, and ultimately our intent to correct it.

This point of view sure is attractive: it stipulates clearly what we've attempted to do (and equally importantly what we didn't attempt to do), namely rectify the distortion nonresponse has introduced in our earlier sample-based estimates of gender. Nothing less but also nothing more. So if we're interested in age and find dubious the results obtained with the modelling then it's hardly any surprise: we've only (and openly) attempted to rectify the shift observed on gender. Should we have attempted to rectify age also? Sure, if we had any reason to care about it. To drive the point: suppose on the frame on children we have the benefit of knowing the eye color of the maternal great-grand-mother of every child and found that its respondents-only distribution differed from that obtained before… Why should we care?

7.4 On the estimation of "response probabilities": the multiplication of adjustments over cycles

Successive waves of nonresponse call for successive nonresponse models; how should they interact with one another? This is an issue that we'll investigate some in this sub-section. This is more a weighting issue than a bootstrap one. But then, if the weighting strategy is not appropriate, then replicating it for variance estimation purposes is certainly not adequate.

What is often seen is that nonresponse modelling at cycle i+1 is carried out ignoring the model from cycle i; in other words, all respondents of cycle i are pooled and modelling is done to explain how a (sub-)set of these have become the respondents of cycle i+1. This is the view that nonresponse modelling should bring you "one step back", back into known territory (*i.e.*, prior to latest wave of nonresponse).

To see what is going on, consider the following setting. By the time cycle 2, say, is over, the NLSCY has gone through two waves of nonresponse. So, starting from the design weights, we need to factor in two nonresponse adjustments in order for those remaining individuals at cycle 2 to represent adequately the cohort's target population. Since the occurring nonresponse is modelled as being the result of a random process, which is reminiscent of a multi-phase approach, the "response probabilities" are treated just as if they were true selection probabilities. The issue here has to do with the way by which we estimate, or obtain, these probabilities. Suppose that at cycle 1 gender was used to form two RHGs, *men* and *women*. The cycle 1 adjusted weights for nonresponse are thus devised to take this distribution imbalance about gender into account. Suppose now we're at cycle 2, with further attrition going on. What is found in NLSCY amounts to the following: *pool* all of cycle 2's respondents regardless of their gender (*i.e.*, cycle 1's RHG) and propose an RHG model which explains the cycle 1-to-cycle 2 attrition. Form then a weight to account for two cycles of nonresponse by simply multiplying the inverse of the two nonresponse estimated probabilities.

But when is such a pooling justified? Proponents of this approach rely (whether they realize it or not) on the (largely un-verifiable) assumption of independence-over-cycles of nonresponse factors. This issue is not explicitly addressed in the sampling books. For instance, in Survey methods and practices, section 7.1 on Weighting one reads:

"*Thus, for a two-phase sample where a unit's probability of selection is* π_1 *at the first phase and* π_2 *at the second phase, a sample unit's design weight is*

$$w_d = \frac{1}{\pi_1} \times \frac{1}{\pi_2} \text{"} \qquad (31)$$

There's an imprecision about this advice which is at the heart of this sub-issue: the second-phase selection probability will in general be *dependent* upon the first-phase result, hence the following notation reminiscent of that used in Särndal *et al.* (1992):

$$w_d = \frac{1}{\pi_1} \times \frac{1}{\pi_{2|1}} \qquad (32)$$

But even with appropriate notation, the issue nonetheless remains: how to account adequately *in practice* for the dependency of successive waves of nonresponse? A good modelling exercise will do that, but what would a good model here look like anyway?

Possibly the easiest way to implement such a model is to *nest* RHGs: a RHG at cycle 2 is constructed, if needed, within a given RHG of cycle 1, and so on. Using our example on gender above, this would call for nonresponse modelling at cycle 2 *by* gender (*i.e.*, the RHG of cycle 1). So, males could have a RHG at cycle 2 based on marital status (married vs. alone, say) while women's RHG at cycle 2 could be based on education level. But this raises a further issue of its own: the number of groups created in such a nested fashion may very well grow too quickly to handle as the survey goes through several cycles of nonresponse.

Here again, the calibration point-of-view can help see the good in nesting. At cycle 1, by forming RHGs based on gender, we essentially said that we wanted the initial distribution of gender restored so we proceeded with the adjustments on weights of the responding units to straighten it up. Equivalently said, we calibrated cycle 1's weights after nonresponse onto the counts obtained before nonresponse has occurred; those (internal) totals matter to us. Now comes a second wave of nonresponse which compromises yet again our ability to yield the original sample-based counts on gender. Since cycle 1's nonresponse weights were adjusted to preserve them, it seems natural to work from now on within gender to ensure consistency. Whatever way we'll split men into RHGs, the sum of all cycle 2 nonresponse adjusted weights for men will match the cycle 1's target men total. It thus appear natural from a calibration stand point that any adjustment you'd like to make in the present must not in any way compromise all the past calibration you've performed. And nesting is such a way (conceptually at least if not practical).

Now that you know about this imprecision, you may choose to ignore it. The consequence is certainly more a first-order preoccupation than a second-order one. If you intend to do so, you should conduct simulations to test under SRSWR what would be the impact.

In a few words...

- While two-phase sampling and two-stage sampling are methodologically different, they are mathematically the same for variance estimation purposes.

- The bootstrap captures only a portion of the total variance under a two-phase setting. When the sampling fraction at phase one is negligible, the portion not captured by the bootstrap is similarly negligible. As a consequence, the bootstrap "works" for a survey with nonresponse or one with a two-stage sampling design precisely as long as the sampling fraction at phase one is negligible. It is not currently known how to implement the bootstrap in order to capture the total variance under nonresponse.

- A methodology should be chosen with *some* consideration for variance estimation. This would avoid creating, for example, a hopelessly large number of RHGs which is detrimental to variance estimation (and hardly beneficial anyway for point-estimation).

8. POST-STRATIFICATION

Post-stratification is routinely carried out in many surveys as the last methodological step in the creation of releasable survey weights; the post-stratified weight is usually computed as:

$$w_k^{post} = w_k^{nr} \times \frac{T}{\sum\limits_{post-stratum} w_j^{nr}} \qquad (33)$$

where: T is the known total of the corresponding post-stratum.

One thing should strike you if you've adopted the calibration point of view to nonresponse: we're not nesting the post-strata and the RHGs. As a result, whatever "calibration" we performed through nonresponse is compromised by post-stratification carried out this way. (However, the nesting issue should be minimal here since the post-stratification factor should be close to 1.) The question naturally presents itself: can we forget all about RHGs and post-stratification and do everything under the Sun through calibration. The answer is yes, at least in principle. See Singh *et al.* (2005) for a view along such lines. The idea is the following: you want to replace the set of design weights by a set of final weights such that some functions are invariant (or kept fixed) under the weights used. For example, those functions could be about yielding counts (to match at all times known population counts) and/or totals (to match at all times earlier cycle released estimates for some variables of interest in a longitudinal survey). But not any set of weights satisfying the invariance condition on these functions would do. Indeed, we're not ready to throw out our design weights for just about anything: we want to maintain whatever feature the design has. We do that by imposing that a distance between the optimized weights and the design weights is minimal. This way we ensure that our design is kept well in mind in our efforts to have the weights yet satisfy conditions not verified by the design weights themselves. (This is incidentally the same reason Deville and Särndal (1992) built their GREG optimization around minimizing a distance.) The issue of calibration is not as much conceptual as computational. In order to have some benefit, extensive optimization has to take place and this is computer-intensive. An in-depth study of the possibilities and limitations of calibration to address at once nonresponse and post-stratification issues is the focus of research to come.

Returning to the usual post-stratification, what is a primary concern for variance estimation is the size of post-strata. Indeed, they must contain enough units as to make the replicates' (and not just the sample's) composition of the post-strata adequate. In other words, post-strata must not become empty (or even close) for some replicates.

In the NLSCY the post-stratification rests upon crossing age (of the kid at cycle 1), gender and province of residence at cycle 1; there are 240 such post-strata in the NLSCY. Table 5 below gives the post-strata composition for the original cohort of the NLSCY at cycle 6 in terms of number of kids. The shaded cells are those that become empty for at least one replicate. Table 6 is expressed in terms of number of PSUs and therefore there's double-counting: a same PSU may (and generally will) contribute to more than just one post-stratum, which is not the case for children (*i.e.*, Table 5).

Sex	Age	NFD	PEI	NS	NB	QC	ON	MB	SK	AB	BC
Female	0	44	15	43	30	140	187	64	66	57	53
	1	35	22	55	51	154	203	62	51	60	57
	2	26	16	30	30	83	121	43	46	56	42
	3	17	12	42	34	91	113	43	35	45	51
	4	23	13	35	31	81	131	33	33	47	51
	5	32	16	42	30	74	98	35	43	36	28
	6	18	17	21	33	69	96	23	30	40	40
	7	36	10	30	20	72	107	23	30	39	44
	8	30	8	32	22	68	105	38	44	42	31
	9	24	21	29	23	77	88	29	38	42	40
	10	36	18	30	25	79	97	28	36	38	37
	11	36	14	28	23	61	87	26	30	41	31
Male	0	37	20	49	44	142	188	58	52	70	64
	1	37	22	65	44	145	168	63	62	78	55
	2	23	10	35	32	112	146	41	48	54	46
	3	36	13	33	40	106	131	44	51	37	34
	4	25	14	41	28	85	105	23	41	43	43
	5	17	13	34	35	88	112	43	51	33	42
	6	24	18	30	29	64	103	39	26	40	46
	7	26	9	27	22	69	95	29	37	46	34
	8	27	8	39	24	68	116	36	24	45	36
	9	35	10	19	23	63	86	31	31	35	29
	10	26	11	25	21	58	84	20	26	49	33
	11	19	9	25	16	59	67	31	27	36	25

Table 5: Number of children within each post-stratum.

Sex	Age	NFD	PEI	NS	NB	QC	ON	MB	SK	AB	BC
Female	0	35	13	38	27	105	150	53	51	43	47
	1	28	20	42	37	119	167	46	37	52	45
	2	22	15	21	24	67	96	32	40	46	36
	3	15	8	30	22	74	84	32	27	36	40
	4	19	9	23	23	64	101	31	29	37	42
	5	26	14	34	21	62	76	26	30	27	24
	6	15	16	16	24	52	75	18	25	31	28
	7	27	10	20	14	51	79	19	25	27	31
	8	21	5	21	16	54	73	27	30	29	22
	9	17	12	20	15	55	71	23	25	27	29
	10	22	12	19	18	58	66	21	28	25	28
	11	27	10	20	15	46	70	14	23	28	22

Male	0	29	16	38	34	114	151	44	41	56	49
	1	26	18	53	33	110	134	52	48	61	44
	2	20	10	29	28	96	111	29	39	41	41
	3	24	10	26	28	82	108	38	36	30	28
	4	21	12	31	26	67	80	21	32	33	36
	5	14	10	24	28	70	89	38	40	27	35
	6	17	13	22	23	52	79	30	21	29	38
	7	20	7	22	14	50	69	21	32	36	29
	8	19	5	27	21	48	89	22	18	31	30
	9	26	9	15	14	46	62	23	23	25	21
	10	17	11	19	14	42	59	13	16	37	20
	11	14	8	20	10	41	47	22	20	23	21

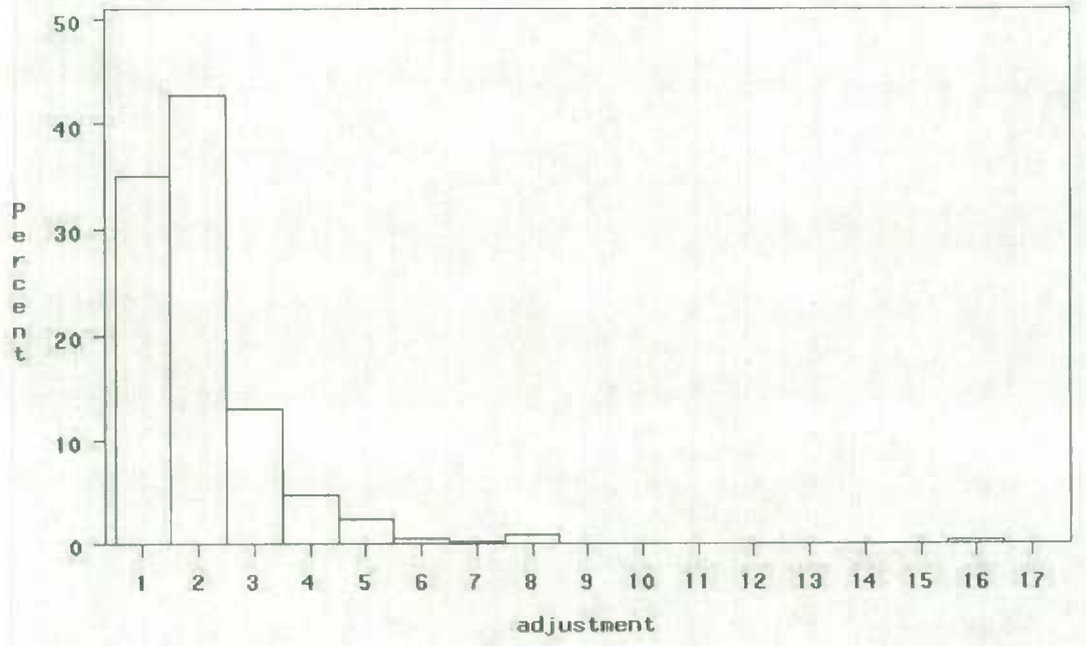
Table 6: Number of PSUs (*i.e.*, selection units) per post-stratum.

Currently, when a post-stratum becomes empty for a given replicate, the corresponding bootstrap weight is set to 0 for all implicated units. This patch doesn't address the real issue: post-strata are unstable the way in which they were designed. One solution is to collapse some of them… but according to which of the three variables? This is both a subject matter issue (for which sub-populations do we really need to have our weights set out to yield known counts?) and a methodological one (which variables can really be collapsed?). The methodological issue is not trivial; let's first look at the current post-stratification adjustments:

Sex	Age	NFD	PEI	NS	NB	QC	ON	MB	SK	AB	BC
Female	0	0.82	1.59	1.34	1.69	1.08	1.23	0.82	0.85	1.45	1.31
	1	1.09	1.09	0.95	0.83	0.99	1.01	1.19	1.33	1.46	1.25
	2	1.35	0.96	1.82	1.14	1.50	1.22	1.28	1.15	1.04	1.58
	3	1.68	1.44	0.97	1.07	1.20	1.53	1.32	1.32	1.38	1.27
	4	1.60	1.44	1.02	1.15	1.53	1.07	1.66	1.63	1.04	1.04
	5	0.91	0.98	1.02	1.05	1.23	1.20	1.06	1.04	1.50	1.63
	6	1.98	0.97	1.59	1.05	1.27	1.43	1.67	1.35	1.13	1.25
	7	0.74	1.38	1.04	1.66	1.16	1.10	1.76	1.26	1.08	0.93
	8	1.04	1.79	1.41	1.51	1.10	1.39	1.10	0.95	1.11	1.47
	9	1.29	0.57	1.29	1.47	1.39	1.42	1.15	1.04	1.19	1.15
	10	0.96	0.76	1.54	1.24	1.26	1.29	1.56	1.05	1.16	1.21
	11	1.01	1.14	1.23	1.45	1.36	1.46	1.24	1.49	1.15	1.61
Male	0	1.04	1.09	1.12	1.02	1.05	1.23	1.40	1.17	1.12	1.30
	1	1.18	1.09	0.98	1.05	1.05	1.76	1.19	1.03	1.13	1.72
	2	1.50	1.86	1.44	1.59	1.18	1.08	1.64	0.96	1.08	1.32
	3	0.96	1.39	1.38	0.92	1.67	1.09	0.96	0.96	1.68	1.85
	4	1.27	1.10	1.18	1.57	1.30	1.45	1.87	1.20	1.36	1.28
	5	2.14	1.02	1.20	0.96	1.29	1.41	0.99	0.84	1.56	1.08
	6	1.19	0.84	1.03	1.09	1.58	1.38	1.34	1.46	1.16	1.07
	7	1.25	1.49	1.53	1.36	1.22	1.14	1.45	1.25	1.22	1.42
	8	1.16	1.68	1.00	1.59	1.34	1.08	1.17	1.83	0.96	1.18
	9	1.01	1.91	2.17	1.52	2.19	1.35	1.35	1.47	1.24	1.43
	10	1.22	1.49	1.62	1.86	1.25	1.42	1.96	1.57	1.13	1.29
	11	2.38	1.92	1.57	2.13	1.50	1.89	1.32	1.72	1.87	1.69

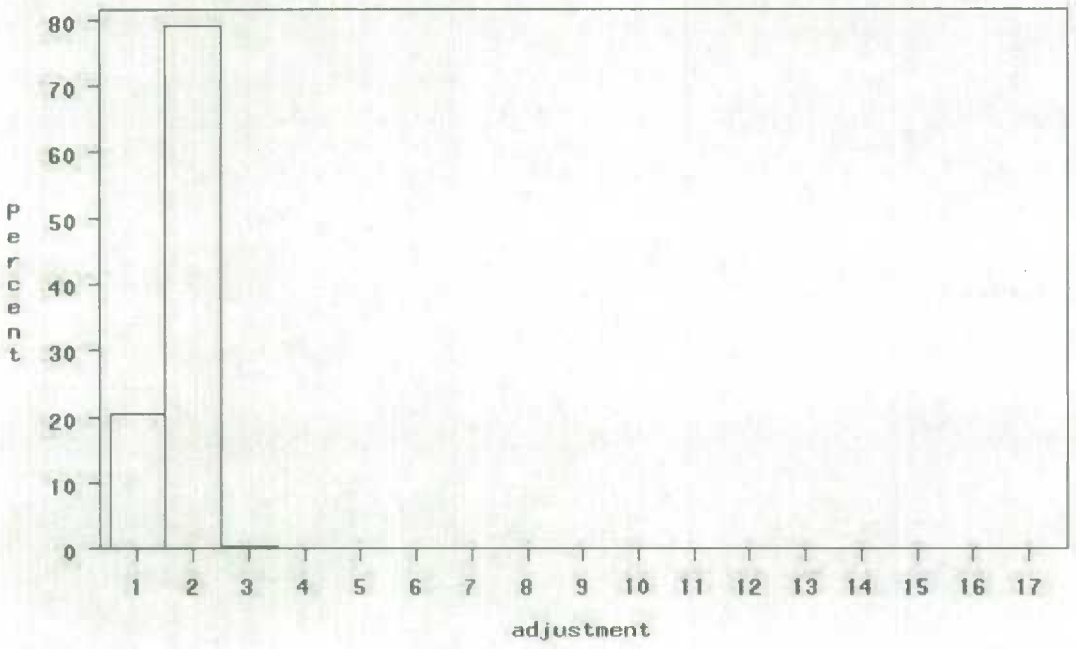
Table 7: Current post-stratification adjustments.

One cannot better illustrate the (adverse) effect of small post-strata on variance than to draw the histogram of the 1,000 post-stratification adjustments. Here are the non-zero bootstrap adjustments for the post-stratum of 8 year-old male kids in PEI:



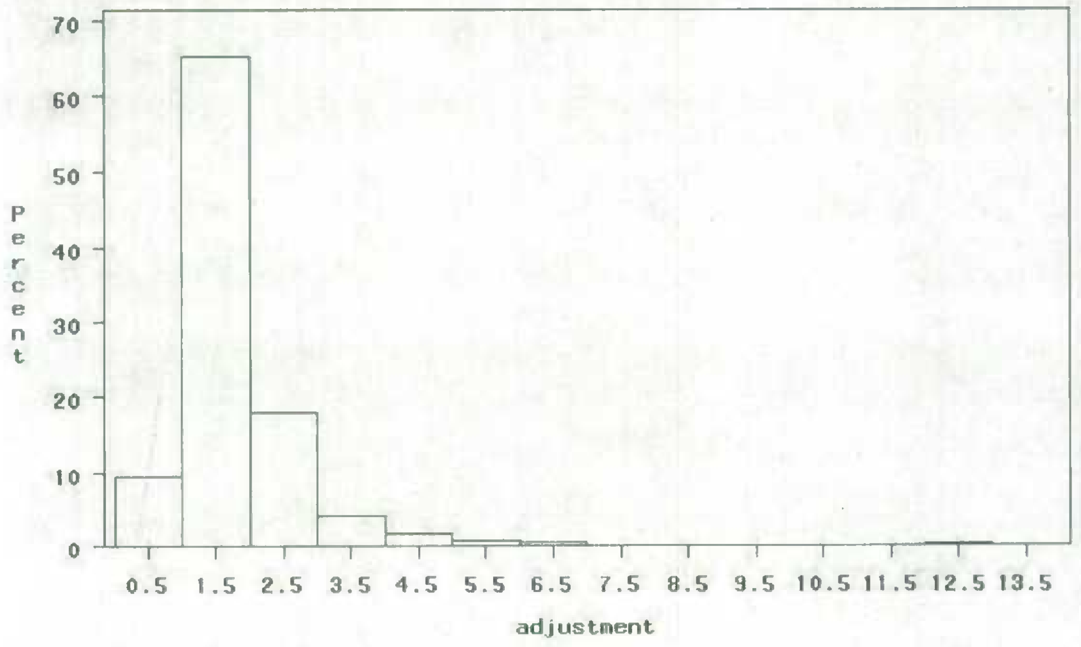
Graph 6: Distribution of post-stratification adjustments within replicates for the post-stratum of 8 year-old males in PEI

To give you a comparison point, here's the histogram (using the same scale) of bootstrap adjustments for a post-stratum with a similar (sample-based) adjustment but more densely populated: 3 year-old males in Québec:

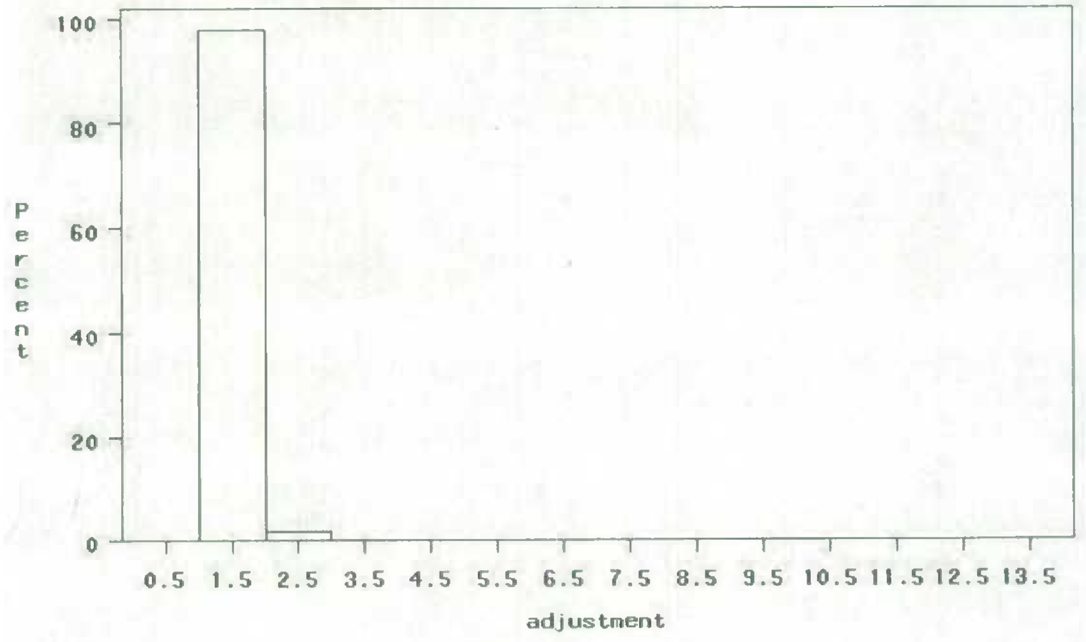


Graph 7: Distribution of post-stratification adjustments within replicates for the post-stratum of 3 year-old males in Québec

Another telling example is taken from 7 year-old males in PEI compared to 4 year-old males in Ontario (again, the latter was chosen because the sample-based adjustment was similar and is more densely populated):



Graph 8: Distribution of post-stratification adjustments within replicates for the post-stratum of 7 year-old males in PEI



Graph 9: Distribution of post-stratification adjustments within replicates for the post-stratum of 7 year-old males in Ontario

Should we consider in the smaller province to collapse according to gender, even if that proved to be the scenario chosen by subject-matter? The danger with any collapsing is to blend two trends. To illustrate, look at 5 year-olds in Newfoundland. We gather from the corresponding adjustments that female weights had to be reduced through post-stratification in order to match known counts on females while for the boys the opposite took place: we had to (considerably) inflate their weights to match known totals on boys. If we pool these two sub-populations into one, then at least one of the two trends (downward for the females and upward for the males) will be lost, if not both! What we'd like from a methodological standpoint is to recover (as closely as possible) sub-population totals while working at a more aggregate level. To illustrate, suppose post-strata involving 1 year-olds in PEI were collapsed to form just one, there'd then be no harm done since in both cases the trend was upward by 9% (*i.e.*, an adjustment of 1.09).

In a few words...

- As with nonresponse, post-stratification should be implemented with considerations for variance estimation. Tiny post-strata are detrimental to variance estimation and after-the-fact collapsing is at best an *ad hoc* approach that should be avoided with proper planning instead.

9. MISCELLANEOUS

In this section we cover other important but isolated issues relating to the bootstrap.

9.1 Stability of the variance estimates

As we pointed out already in Section 3, variance estimation using the bootstrap involves two separate random mechanisms: the original sampling and the re-sampling. A satisfactory variance estimator under bootstrap, $v_{boot}(\hat{\theta})$, of $V(\hat{\theta})$ should be unbiased:

$$E_D E_{boot}\left(v_{boot}(\hat{\theta})\middle|s\right) = V(\hat{\theta}) \tag{34}$$

If $v_{boot}(\hat{\theta})$ indeed satisfies (approximately) (34), then we'll customarily say that the bootstrap "works" for $\hat{\theta}$. In that sense, bootstrap works for many types of estimators which include as an important sub-class those estimators which arise from a smooth transformation of the mean, and even works for the median (and more generally for quantiles). But here's the catch: in practice we don't have the benefit of evaluating the double expectation in (34) and we must instead rely on *approximations*. The question of interest then becomes: are approximations obtained in practice "all equally reliable"? The answer is no. There are inferential frameworks (*i.e.*, given estimator and sampling design) where $v_{boot}(\hat{\theta})$ is very stable as in the case of the estimator of the mean and SRS. Indeed, provided sufficient sample size and replicates are used, the estimate can almost be confounded with the unknown value itself. And there are other settings where the uncertainty around $v_{boot}(\hat{\theta})$ cannot be as readily dismissed; this is the case of the estimator of the median under SRS. The measure we'll use to gauge stability is (an approximation under Monte Carlo of) the variance of the variance estimate; we'll show that under SRS the bootstrap variance estimator of the mean is much more reliable than its counterpart for the median. It is important to stress that the instability is not due to the re-sampling (or the bootstrap if you prefer) but rather to the nature of (design) variance estimation in presence of the median. We'll have more to say on this in Section 9.2.

Consider the following (general) question: if for some reason the relation $V_1 < V_2$ should hold for two (exact) variances V_1, V_2 then would we necessarily have $\hat{V}_1 < \hat{V}_2$ for any two of their estimates? To put this in the context of the NLSCY, let $V_1 = V_{C1}$, the variance of an estimator at cycle 1 for a static characteristic and $V_2 = V_{C2}$ the corresponding variance at cycle 2 with attrition due to nonresponse being the sole factor explaining how one gets to cycle 2 from cycle 1. Consequently, we should have $V_{C1} < V_{C2}$ since, all other things being equal, the effective sample size at cycle 2 is smaller than that of cycle 1. The relationship between the true variances will hold for any two variance estimates, that is estimates obtained from possibly *different* samples, *provided* that the variance estimation of V_1 and V_2 is a reliable enough process. This simply means that we require that the $\hat{V}_1$'s and the $\hat{V}_2$'s are tightly spread about V_1 and V_2, respectively (and thus lay sufficiently far away from V_2 and V_1, respectively), in order to preserve the relation between *any* two variance estimates. Our concern here, with Table 1 well in mind, is: when will we really see a variance estimate of a cycle exceed that of the previous cycle (as it should for a static variable) as a result of occurring nonresponse?

To answer questions of this kind, we must keep in mind that there are always two forces at work which can make the variance estimates unreliable: the sampling design and the bootstrap (hence the two expectations in (34)). For one, the number of replicates used to conduct the bootstrap will have an effect on how good the computed variance estimate will be *as far as replication is concerned*. This is what people typically find in practice when they notice that in a *given* situation not much gain in stability in the variance estimates is achieved by using 1,000 replicates instead of 500, say. More precisely, the variability in the variance estimates due to the bootstrap comes from which set of B replicates (among all possible such sets) is actually used: different *sets* of B replicates will yield different bootstrap variance estimates. This is the sampling framework underlying the use of the bootstrap.

Observe that with the bootstrap, one set of B replicates will yield B estimates; these bootstrap estimates can in turn be likened to B estimates taken directly from the sampling distribution of the estimator. This means we can see the bootstrap as a sampling mechanism operating from the (sampling) distribution of the estimator. In other words, the replication acts *as if* the B bootstrap estimates actually were (up to a translation factor) B observed survey estimates: the distribution of the bootstrap estimates attempts to recreate the sampling distribution of the estimator.

Reiterating a point made earlier, if one had access in any given situation to a closed-form expression for the variance estimate, one provided by the sampling theory, then one could see the bootstrap method used in practice as an attempt to yield a variance estimate as close as possible to that coveted closed-form estimate. The closed-form conceptually corresponds to having all possible replicates. *But* even a variance estimate obtained from sampling theory in the form of a closed algebraic expression would vary from sample to sample under hypothetical repeated sampling from the population. This is simply the (design) variance of the variance estimator itself. This component of the (overall) variance of the variance estimates is the one we would like most to guard ourselves against, but one for which the bootstrap has no influence on whatsoever. Indeed, recall that the bootstrap can only attempt to provide us with an estimate "close" to the theory-supported variance estimate if such an estimate was actually available: the greater the number of replicates used, the closer the bootstrap estimate gets to its closed-form counterpart. But once the bootstrap (reasonably) achieves that, there's nothing else it can do about the sample-to-sample variability which is *still* present in the variance estimates themselves. This main component to the variability in the variance estimates thus depends on specific features of the sampling design, like the sample size used.

In practice one can usually guard oneself against the variability arising from the bootstrap by choosing a large enough number of replicates, as large as available time and resources permit. But while this keeps one source of variability in the variance estimates under control, it achieves nothing with regard to the other component which then can easily be forgotten. Unfortunately, the component of variance in the variance estimates arising from the number of replicates used is usually *not* the dominant one. Furthermore, control over the minor component (*i.e.*, arising from the bootstrap) must not be mistaken as control over both components at once.

A more visual way of saying the same thing is to represent the situation of estimating the variance using the bootstrap in the form of a grid which describes all possible variance estimates that can be obtained under the two random mechanisms at work.

	Set of *B* replicates used				
Observed sample	**1**	**2**	**3**	**4**	**and so on…**
1	$\hat{V}_{11}$	$\hat{V}_{12}$	$\hat{V}_{13}$	$\hat{V}_{14}$	…
2	$\hat{V}_{21}$	$\hat{V}_{22}$	$\hat{V}_{23}$	$\hat{V}_{24}$	…
3	$\hat{V}_{31}$	$\hat{V}_{32}$	$\hat{V}_{33}$	$\hat{V}_{43}$	…
4	$\hat{V}_{41}$	$\hat{V}_{42}$	$\hat{V}_{43}$	$\hat{V}_{44}$	…
and so on…	…	…	…	…	…

Table 8: A two-dimensional representation of all possible variance estimates arising from bootstrapping samples.

In Table 8 there's one line for each of the different observed samples one could have to work from (all obtained from the same sampling design of course) and there's one column for each *set* of *B* replicates one could possibly draw. (Observe that it's *not* the number of replicates *B* that changes from one set to another but rather which units of the observed sample get to make the *B* replicates.) The main entries of the Table 8 represent the various variance estimates one would obtain under these two sampling mechanisms.

In practice, that is bootstrapping only one sample, one obtains only one entry of the two-dimensional Table 8 above; let's say it's the entry (1,1): first line and first column. Indeed, one usually has only *one* observed sample from which to conduct the bootstrap using only *one* set of *B* replicates. So, in practice we want to know just how $\hat{V}_{11}$, as an estimate itself, is variable. But there are two ways of answering this. Indeed, we can ask what other variance estimates we would have obtained had the set of *B* replicates been different than the one actually used to compute $\hat{V}_{11}$. These variances make the entries on the same *line* as $\hat{V}_{11}$. Or we can ask what other variances we would have obtained had the observed sample been different than the one used to compute $\hat{V}_{11}$. These variances are found in the same *column* as $\hat{V}_{11}$. (For the intellectual gymnastic of changing samples but keeping the same set of replicates, we need to think of the bootstrap as selecting rankings in a sample rather than labels. So a given replicate did not result in the choice of unit labelled 2, say, but rather the second unit in the (random) ordering by which units of the sample are listed.) The column-to-column variability in the variance estimates is due to the bootstrapping while the line-to-line variability is due to the sampling design.

We can therefore liken the computation of bootstrap variance estimates on a very large number of replicates to an attempt to provide ourselves with a variance estimate close to the estimate we would get if a closed-form expression provided by the sampling theory was available to us.

Closed-form variance expressions were referred to quite a few times above as the sampling theory counterparts to the computed bootstrap variance estimates. It may help to see this parallel if we put the closed-form variance estimation framework in a grid-form like the one above. For that purpose, consider that the design is SRSWR and we're interested in estimating the variance of estimates of the mean. In such a simple setting we have a closed-form expression for the variance estimates, it is:

$$\hat{V}(\hat{\mu}) = \frac{S_s^2}{n} \tag{35}$$

The grid in this case can be seen as the two-dimensional Table 8 above "collapsed" to only one column: one set made of all possible replicates, since if we took all possible replicates we'd fall back on the usual variance estimate.

Observed sample	Variance estimate
1	$\hat{V}_1 = S_{s=1}^2 / n$
2	$\hat{V}_2 = S_{s=2}^2 / n$
3	$\hat{V}_3 = S_{s=3}^2 / n$
4	$\hat{V}_4 = S_{s=4}^2 / n$
and so on…	…

Table 9: This is the two-dimensional Table 8 collapsed to just one dimension when all replicates can be used

To help gauge the two components to variance estimation just mentioned, we'll repeatedly use the following result from classical statistics. But since our context is not classical statistics but rather survey sampling, we need to keep in mind that this result is at best heuristic in our context. Actually, the main reason a result like this is considered here is to ground simulations. Indeed, running simulations can be hazardous, especially in uncharted territory, due to computational errors, misunderstood assumptions, *etc.* It considerably helps (and secure later findings) if one starts with simulations in simple frameworks where outcomes can somewhat be predicted. Once a simulation is validated this way, we can have some confidence extending it to more complex frameworks where essentially it will sail on its own. With hindsight, you'll find that each simulation that was used in this paper never was solo. For example, if it was about the performance of a little-tested estimator like the median, then it also featured a better-known estimator like the mean to help "cross-validate it".

The result is the following - see Dudewicz and Mishra (1988) p.325 (Efron and Tibshirani (1998) get essentially to the same result, but by different means):

Result 1

Let $X_1, X_2, ..., X_M$ be M i.i.d. variables with mean μ and variance σ^2. The sample-based variance estimate $\hat{S}^2 = \frac{\sum_{i=1}^{M}(X_i - \bar{X})^2}{M}$ of σ^2 has a variance which is given by

$$V(\hat{S}^2) \cong \frac{(\alpha_4 - 1)\sigma^4}{M} \qquad (36)$$

where α_4 is the kurtosis of the distribution of the X_i's.

Note: Since there's no consensus on the definition of the kurtosis, it's important to make precise which one we use. For our purposes the kurtosis of the distribution of probabilities associated to the random variable X with mean μ and variance σ^2 is defined as:

$$\alpha_4 = \frac{E(X - \mu^4)}{\sigma^4} \qquad (37)$$

In some books and, most importantly for the user, in SAS the kurtosis used is the kurtosis defined by (37) *minus* 3. This is done so normal distributions get a kurtosis value of 0 (instead of the unremarkable value of "3" they get under our definition).

Let's now consider the two cases of stability separately and use Result 1 to evaluate how important each component is.

If the sample's composition changes but the set of replicates used remains the same, then we're assessing the stability of the variance estimates with regard to the sampling from the population. (By considering the same set of replicates for each drawn sample, we're doing as if that set provided us with an error-free estimate as far as bootstrap goes.) Therefore, for a given set of B replicates, the stability of the bootstrap variance estimates depend on how the characteristic of interest is distributed in the population. In this case, we can rewrite (36) in terms of CV as:

$$CV_B(\hat{S}^2) = \sqrt{\frac{(\alpha_4^{pop} - 1)}{n}} \qquad (38)$$

where α_4^{pop} is the kurtosis of the distribution of the characteristic of interest.

Simulation D

In Program D, one hundred samples of size 100 (500 and 1,000) were drawn from a population of 10,000 under SRSWOR; there are two characteristics of interest: one normally distributed (with a kurtosis of 3) and the other exponentially distributed (with a kurtosis of 8.75). Each sample was bootstrapped using the *same* set of 1,000 replicates. We're interested in the stability under the sampling design of variance estimates for the estimator of the mean. Tables 10 to 12 below give the CV of various variance estimators for sample sizes of 100, 500 and 1,000.

Estimator	Source	Normal	Exponential
Mean	Closed form	15.2	25.7
	Heuristic	14.1	27.8
	Bootstrap	15.6	26.5
Median	Bootstrap	47.0	48.4

Table 10: Comparison of CVs obtained under a sample size of 100.

We can see from the Table 10 that all results concur in the case of the mean; variance estimates for the exponential variable vary more from one sample to another than those for the normal because of its greater kurtosis. In the case of the median we only have one source, the bootstrap. According to it, the variance estimates for the median vary much more from one sample to another than those for the mean for the same variable of interest.

Estimator	Source	Normal	Exponential
Mean	Closed form	5.5	11.3
	Heuristic	6.3	12.4
	Bootstrap	6.6	12.1
Median	Bootstrap	29.9	36.1

Table 11: Comparison of CVs obtained under a sample size of 500.

Estimator	Source	Normal	Exponential
Mean	Closed form	4.3	7.8
	Heuristic	4.5	8.8
	Bootstrap	6.4	9.7
Median	Bootstrap	25.5	30.2

Table 12: Comparison of CVs obtained under a sample size of 1,000.

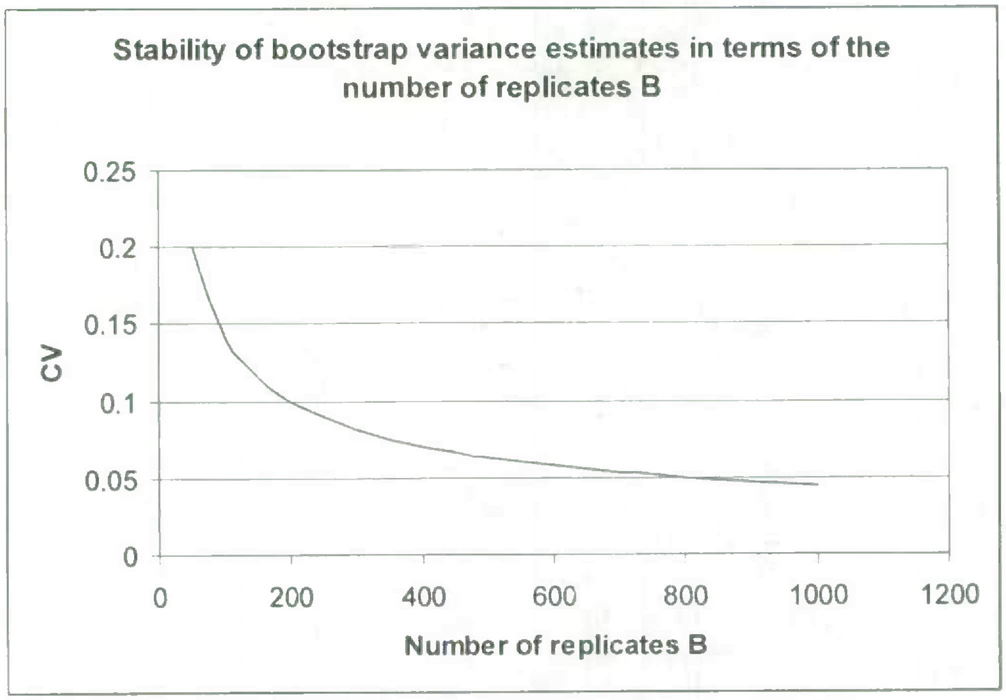
The lesson here is this: for a given sample size, variance estimates for the median are much less stable than corresponding variance estimates for the mean. So, while the bootstrap "works" for both the mean and the median in the sense that the expectation of the bootstrap variance estimator (essentially) is the true corresponding variance, in practice variance estimates for the two are not equally reliable.

On the other hand, if we work from the same observed sample, then the stability of bootstrap variance estimates is now dependent on the re-sampling. By fixing the observed sample, we can then liken the bootstrapping to sampling directly from the sampling distribution of the estimator. The precision of the estimate of its variance is then solely expressed in terms of the number of replicates used, say B: hence:

$$CV\left(\hat{S}^2\right) = \sqrt{\frac{\left(\alpha_4^{estimator} - 1\right)}{B}} \tag{39}$$

The kurtosis now entering (36) is that of the sampling distribution of the estimator; for all practical purposes we can assume this distribution is normal and hence get the following CV:

$$CV_{p,n_D}\left(\hat{V}_{bootstrap}\right) = \sqrt{\frac{\left(\alpha_4^{estimator} - 1\right)}{B}} = \sqrt{\frac{2}{B}} \tag{40}$$



Graph 10: Stability of the variance estimates as a function of the number of replicates used in the bootstrap.

Graph 10 depicts something which is already well-known of users about the bootstrap through simulations: the largest gains in stability are provided by the first 250 replicates. In Simulation E we illustrate the heuristic result regarding the stability in variance estimates with respect to the bootstrapping.

Simulation E

In Program E, a population of 10,000 units was created with 2 variables of interest: one is normally distributed and the other exponentially distributed. A sample of 100 units is drawn under SRSWOR to play the role of an observed sample in practice. The observed sample is bootstrapped to yield 1,000 replicates. The bootstrapping is done 100 times. The kurtosis is assumed to be 3 (*i.e.*, a normal sampling distribution for the estimator). (The program also expresses the stability of the variance estimates for the mean with regard to the bootstrap in terms of root mean-squared errors - RMSE).

Seed (sample)	Mean normal	Mean exp.	Median normal	Median exp.
1	4.9	4.7	3.3	3.3
2	4.1	4.4	5.3	4.7
3	4.6	4.6	5.5	4.8
4	4.4	4.9	5.1	4.2
5	4.9	4.4	7.4	4.0

Table 13: Stability of variance estimates under bootstrap for the mean and median in terms of CV.

The context in Simulation E allowed us to ignore the finite population correction factor in the bootstrap. A question that arises is whether the bootstrap in the context of a non-negligible sampling fraction is as stable as in the negligible case. It turns out that whether or not you work with a negligible sampling fraction the same number of replicates will buy you the same stability in the bootstrap variance estimates as Simulation E2, below, shows.

Simulation E2

Program E2 is a variant of Program E that allows for larger sampling fractions to be processed in a timely fashion. The population made of 1,000 units was created with the same two characteristics as in Simulation E. Sample sizes of 100, 250 and 500 were considered allowing testing the stability of the bootstrap over larger and larger sampling fractions. The observed sample was bootstrap 100 times, each bootstrap run using 1,000 replicates.

The stability of the variance estimates is expressed in terms of CVs.

Sampling fraction	Mean normal	Mean exp
10%	4.8	4.2
25%	4.7	4.4
50%	4.6	4.3

As you can see the CV doesn't change significantly with the sampling fraction used in drawing the observed sample.

Now that we've gone to great lengths describing both random processes behind bootstrap variance estimation in practice, we're ready to give a more compact description of it all.

The idea here is to decompose the total variance *i.e.*, the variance due to all random processes at work confounded, into its components using conditional variance/expectation. (This trick was used above when we investigated the two-phase framework.) We get:

$$V\left(v_b(\hat{\theta})\right)=E_{design}V_{boot}\left(v_b(\hat{\theta})|s\right)+V_{design}E_{boot}\left(v_b(\hat{\theta})|s\right)$$

The component $E_{design}V_{boot}\left(v_b(\hat{\theta})|s\right)$ evaluates the contribution to the overall variance of the fact that bootstrap involves a sampling process of its own. Indeed, given a sample s $V_{boot}\left(v_b(\hat{\theta})|s\right)$ evaluates the variability we'd observe in the bootstrap variance estimate $v_b\left(\hat{\theta}\right)$ as we go through different sets of replicates (each set containing the same given number B of replicates). This is the column-wise variability of Table 8. Once that variance has been calculated, we must "average it out" over all possible samples to relieve the dependency of the calculation performed on the specific sample that was used in the conditioning. As we saw above the component $E_{design}V_{boot}\left(v_b(\hat{\theta})|s\right)$ is usually not the major one and actually it goes to 0 as the number of replicates increases to ultimately reach the total number of replicates the re-sampling gave rise to.

The component $V_{design}E_{boot}\left(v_b(\hat{\theta})|s\right)$ evaluates the contribution to the overall variance of the design variance. Indeed, given a sample s $E_{boot}\left(v_b(\hat{\theta})|s\right)$ first "stabilizes" the bootstrap variance estimates with regard to re-sampling. By averaging out with respect to the bootstrap process, we take the replicate-effect out in a way, thus revealing the sampling design aspect *before* even thinking of assessing its role in the overall picture by computing $V_{design}(\bullet)$. As we saw above, this is of the two the major component to the overall variance (i.e., variance estimates usually vary more from one observed sample to another than from one set of B replicates to another).

9.2 Bootstrap and the median

The median is often cited as an example of a non-smooth estimator for which the bootstrap works. But what is a smooth estimator anyway? An estimator is smooth if it can be expressed as a function *g* of an estimated total whose derivative is itself a continuous function. What's the big fuss about smooth functions then? A big part of the answer comes from classical statistics where the following can be shown to hold.

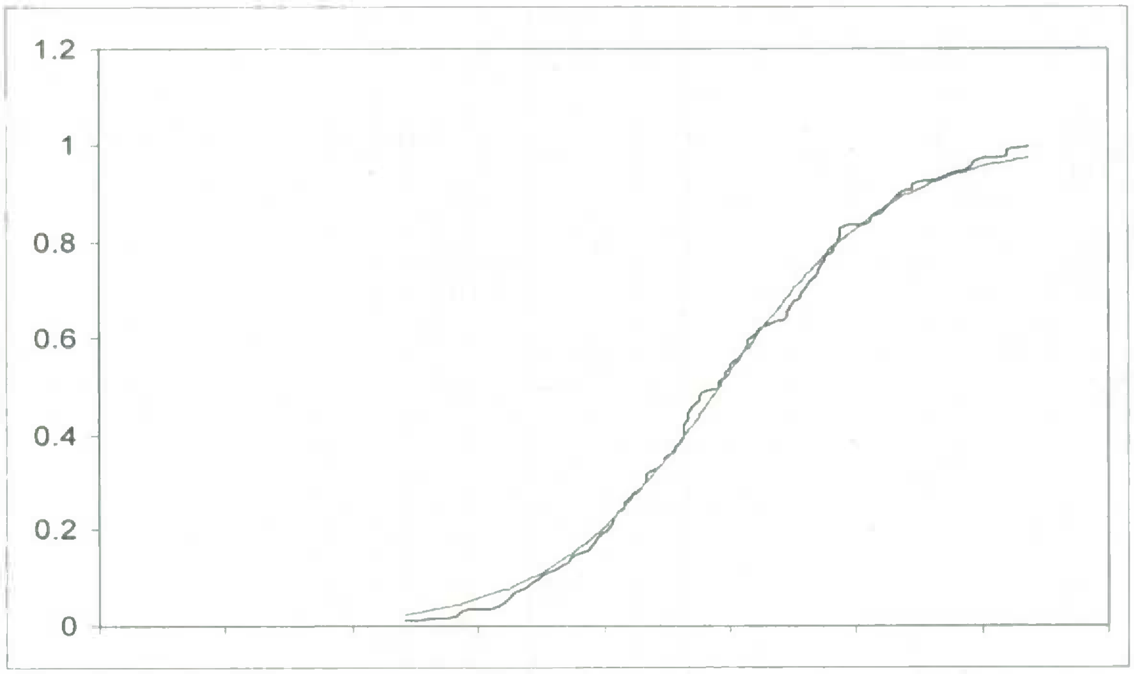
Result 2 (Theorem 6.3.18, Dudewicz and Mishra (1988), p.327).

> Suppose that $\sqrt{n}(X_n - \theta) \xrightarrow{d} N(0, \sigma^2)$ and that *g*(x) is a function for which the derivative *g*'(x) exists and is continuous in some neighbourhood. Then
>
> $$\sqrt{n}(g(X_n) - g(\theta)) \xrightarrow{d} N(0, [g'(\theta)]^2 \sigma^2) \qquad (41)$$

Into words: the transformation of a "normal-inclined" random variable yields also a normal-inclined *provided* the transformation is a smooth one.

We've already noticed that the variance estimate under SRSWOR of the median doesn't enjoy the same level of stability as that of the mean. This is an important issue to keep in mind because in practice we only have the benefit of one variance estimate. And things can potentially get worse for other quantiles. Indeed, for most distributions the practitioner will encounter (*i.e.*, unimodal ones), the median will reside in a "densely" populated neighbourhood of the distribution. This would not be true though for a distribution with two humps, one looking like a camel's back: the median would then be located in a scarcely populated area. Quantiles other than the median tend to occupy scarcely populated area of the cumulative function; think of the 95% quantile for instance. Furthermore, Woodruff's method hints that in order for a variance estimation method of a quantile to be efficient, the local shape of the underlying cumulative function must be "stable".

To illustrate, take a normally distributed variable of interest *Y*. Its (discrete) cumulative distribution is given by the broken (or blue or dark) line while the continuous (or pink or light-coloured) line is the cumulative we'd observe if *Y* were continuous (*i.e.*, infinite population). See how the two lines are locally, here and there, quite apart; these discrepancies will create instability in the inferential process for the quantiles.



Graph 11: The cumulative function of a normally distributed variable on a discrete support.

Computational Tip #8

There's an issue with the way SAS computes a median from weighted data that the user needs to know. SAS first computes the 50th point on the weighted data and this is done the expected way. If it falls between two successive values in the dataset then SAS chooses as the median the larger of these two values. In other words, the weighted median of a variable of interest is always an observed value. Consequently, the further apart successive values (or the units' weight) in the "middle" of the dataset are, the more discrete is the set of "possible" medians. For example, under re-sampling, a given replicate will come to a median value that is close to the one obtained based on the sample. So, if in that area values are distant from one another (or weights are large), then the set of distinct medians generated from all replicate will be small: the set is very discrete (as opposed to being "continuous"). As a result, the histogram of the bootstrap replicates will look little like the (continuous) distribution of medians obtained under a Monte Carlo simulation.

In Program F you'll find the SAS code which estimates the median in the case of weighted data using interpolation.

Simulation F

As usual, a population of 10,000 was generated and the variable of interest is normally distributed. We're interested in estimating its median using a SRSWOR sample of size 100 and estimating the variance of the estimator using the bootstrap.

The Monte Carlo approximation of the (exact) variance of the median is calculated to be (look in the Output window for the summary statistics yielded by the *proc univariate*):

$$V_{MC}(median) = 6.0203$$

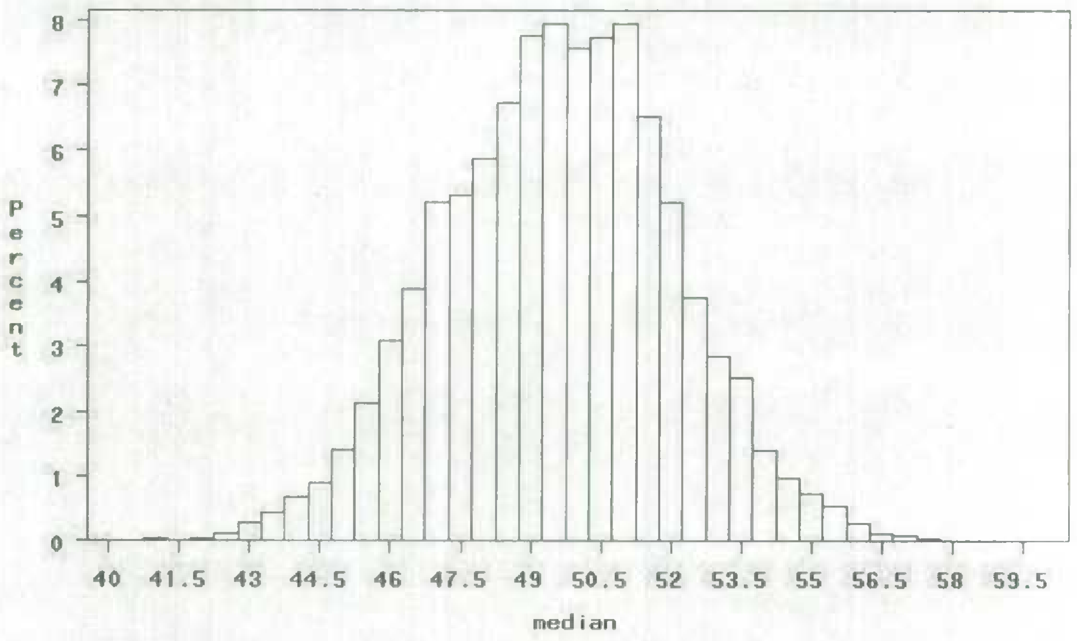
We can check that result with the following classical result (see Dudewicz and Mishra (1988) p. 374) which states that for a normal variable the following approximation stands:

$$V(median) \cong \frac{\pi}{2} V(mean)$$

We find:

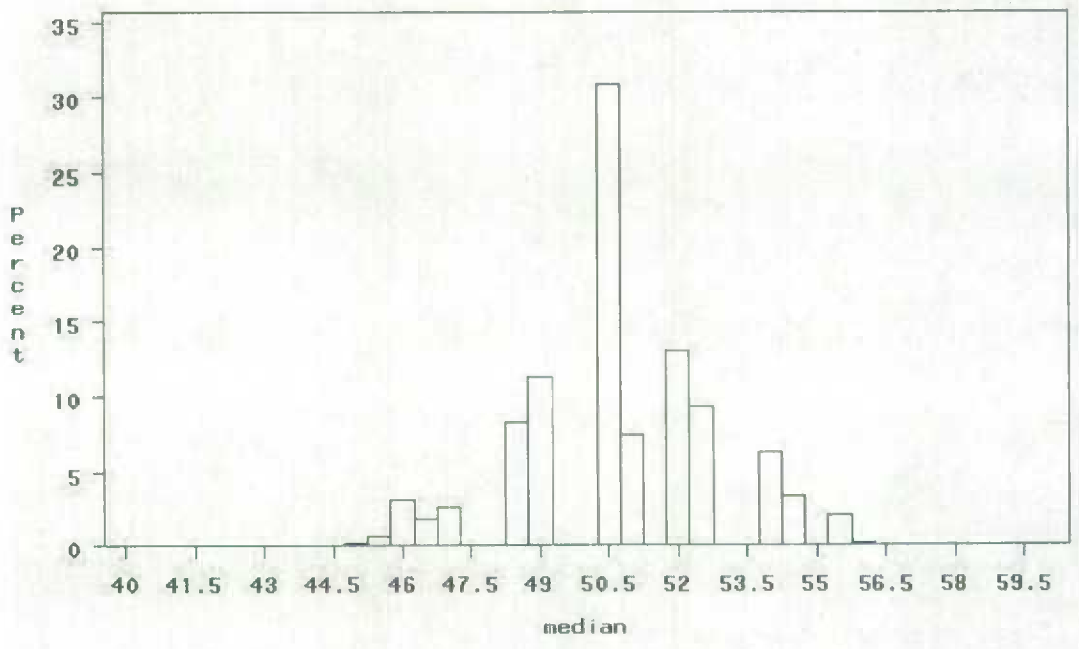
$$V_{Classical}(median) = 6.0369$$

The Monte Carlo approximation of the sampling distribution of the median estimator is provided in Graph 12.



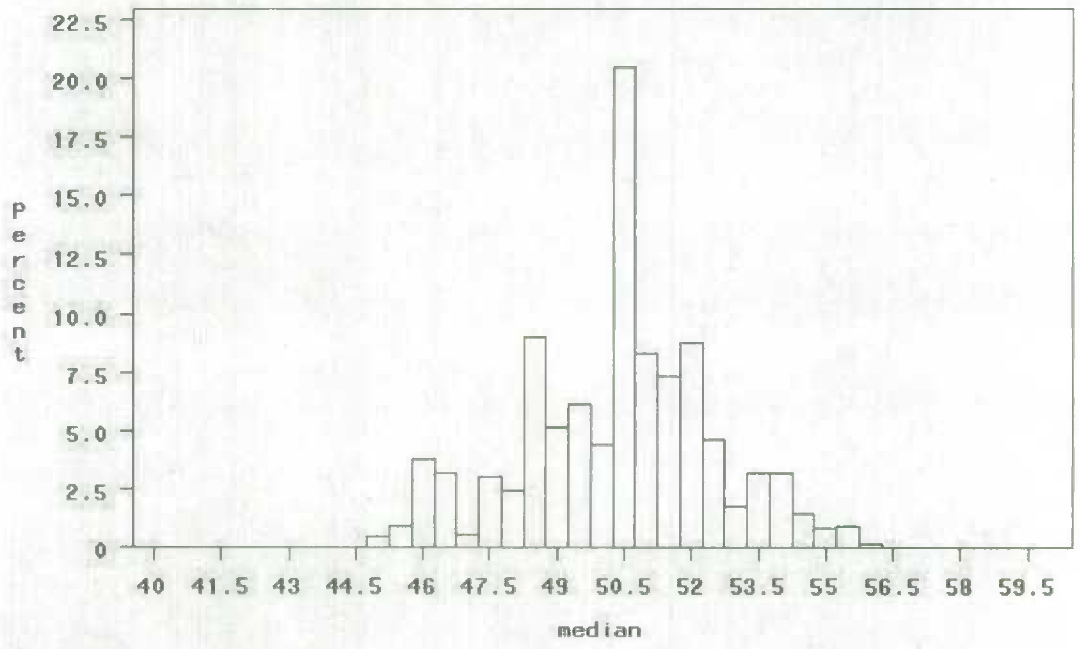
Graph 12: Sampling distribution of the median estimator as depicted through a Monte Carlo simulation.

The bootstrap approximation of the sampling distribution of the median estimator relying on SAS procedures to compute the (bootstrap-)weighted medians is:



Graph 13: Sampling distribution of the median estimator as depicted by the bootstrap using SAS *proc summary*.

The distribution is very discrete. Using now the linear interpolation, the same 1,000 replicates yield a more continuous approximation of the distribution:



Graph 14: Sampling distribution of the median estimator as depicted by the bootstrap using interpolation.

But either way the computed bootstrap variance is

$$v_{BS}(median) = 9.68$$

Using a different sample (*i.e.*, seed value of 4 instead of a seed value of 1 in the corresponding *proc surveyselect*) yields:

$$v_{BS}(median) = 4.61$$

In other words, with a sample size of 100 under SRSWOR the variance estimates for the median are all over the place. But this ought not to be a surprise since in Table 10 we already had established that the CV of the variance estimator for the median in this was about 47%.

Translating that into a 95% confidence interval about the (approximate) exact variance, we get: $(6.0203 \pm 2 \times 0.47 \times 6.0203) = (0.36, 11.68)$. Nothing to write home about, especially for a sample size of 100!

We reiterate at this point that it's not the bootstrap that necessarily has problems in the case of the median. Indeed, a parameter which strongly-depends on the entire distribution of the estimator will prove difficult to estimate whenever sample sizes are not at least of moderate size. The median is one such example and it is expected that other percentiles of the distribution are in fact behaving worse; take for instance the 90th percentile which is in a scarcely populated area of the distribution and probably less robust to changes in the depicted distribution (obtained as you depict the distribution bootstrapping one sample or another) than the median. An alternative to the bootstrap for a survey that solely needs it to take care of the median is Woodruff's method which is described in Annex *B*. Actually, whenever possible the median should be avoided altogether in favor of the better-behaved mean. Yes in the case of strongly skewed distribution there's hardly any escaping the median. But then, in practice distributions may appear more skewed than they really are because the survey data contains severe outliers that have gone undetected and thus unaddressed. This is likely to appear, for instance, in a survey where data collection allows income to be captured hourly, daily, weekly, monthly or annually. Errors like a misplaced decimal in a reported figure or a correct figure but reported over the wrong period will cause outlying values.

9.3 The mean bootstrap

To address small domains (or deal with confidentiality issues), a variant of the bootstrap called the *mean bootstrap* has recently emerged ("mean" in the sense of "averaging", not "nasty", though… ☺). The idea behind the mean bootstrap is the following. As we've seen the usual bootstrap only allows integer multiplicities; there are therefore only two types of outcome for a unit: either it enters a given replicate, *i.e.*, its multiplicity is an integer >0, or it doesn't enter the replicate *i.e.*, its multiplicity is 0. For small domains, for instance, each time the latter outcome happens the domain gets deprived of a critical representation in the replicate. The idea is to offer to the unit more "choices" as to its multiplicity; under mean bootstrap the multiplicity is allowed to be a (positive) rational number. So, instead of being either chosen or not, the multiplicity under mean bootstrap can be 1/2, 1/3, 1/4, *etc*. To implement the mean bootstrap resulting in B replicates being created, one actually constitutes under the usual bootstrap $M*B$ so-called auxiliary replicates. M is some not-too-large a number, say 10 to fix ideas. One forms each of the final B replicate by averaging the multiplicities over the corresponding M auxiliary replicates. One can readily see the appeal of the mean bootstrap: in order for the final multiplicity (*i.e.*, the averaged one) to be 0 for a given unit, one needs to find that the given unit was chosen in *none* of the M auxiliary replicates, which is a quite rare event.

The problem with the mean bootstrap at this point in time is that instead of providing us with a solution closer to the nurturing ground of the traditional bootstrap we're making a step further into the unknown. And we currently have our hands pretty full with unknowns concerning the bootstrap in relation to survey sampling as it is! Actually, as these words were written, it appears that whatever gains there'd be in using the mean bootstrap are out-weighted by its (many) shortcomings. There's talk that the mean bootstrap leads to moderate to severe bias in many inferential contexts. The interested reader should seek further information on the mean bootstrap before tackling it.

9.4 Unequal weights

NLSCY, as other surveys feeding from the LFS, has to deal with a stratified multi-stage design involving, as we've seen above, largely unequal weights.

Nonresponse occurring over successive cycles *may* have other impacts on the variance than those already described. Indeed, a stratified design of (homogeneous) PSUs, as is the case of stratified multi-stage designs, combines two poles in terms of efficiency: stratification tends to reduce the variance while homogeneous PSUs work to increase it (all other things being equal of course). With nonresponse, the PSU may become over time less homogeneous. Therefore the stratification may benefit from this and contribute to a non-increasing variance over cycles (*i.e.*, as stratification moves from an originally inefficient methodology to a… less inefficient one!). This is merely a hypothesis to be kept in mind, and tested if needed, rather than a fact at this point and time.

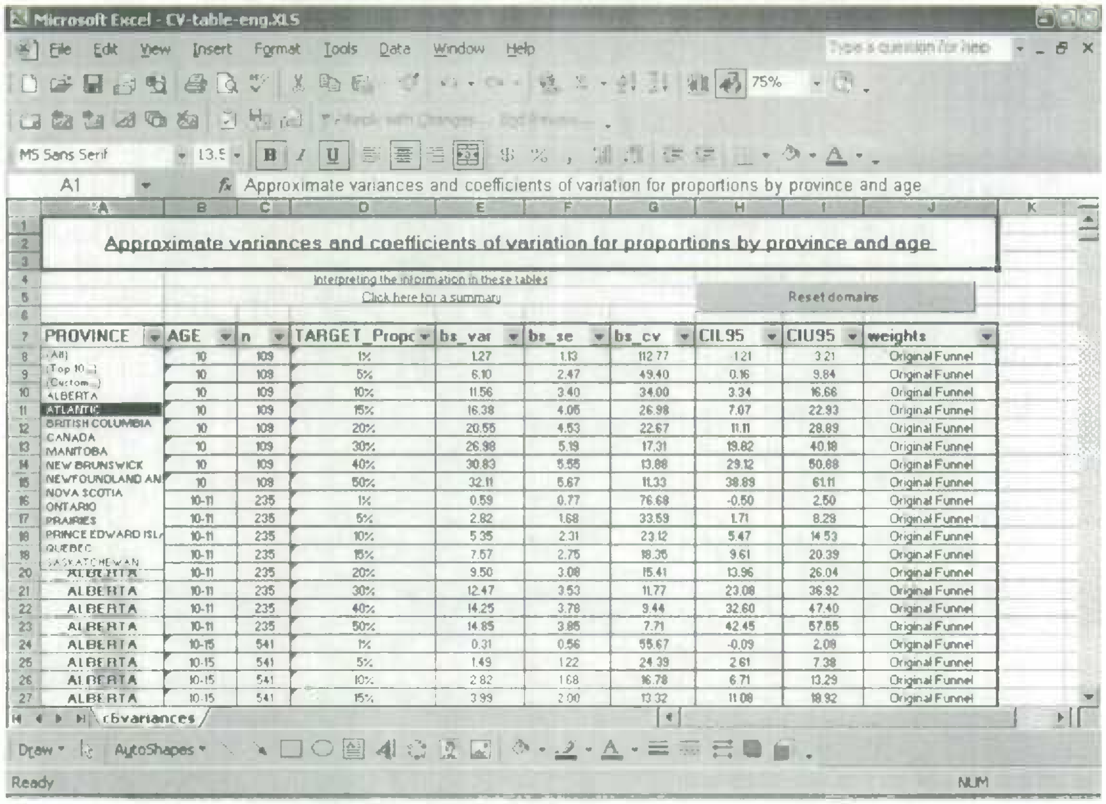
Also, unequal weights contribute to making the variance unstable, especially in the (common) situation where all analyses are carried out with one-size-fits-all set of weights (both sample-based and bootstrap). Indeed, two highly correlated variables of interest may have significantly different sample-based estimates, but also bootstrap estimates, because the units for which they don't concur have the larger weights. Furthermore, a given variable, static over time, may see its estimated variance not increase over time simply because the tails of the bootstrap distribution of estimates exist thanks to contributing units having the larger/smaller weights. As those units are lost to nonresponse, the one-size-fits-all methodology for nonresponse may not compensate adequately for this one loss (though it's expected to work well "on average" *i.e.*, for all variables) and therefore the tails may simply vanish (or appear reduced in importance as cycles go by). As a consequence, what used to be an extreme (bootstrap) estimate at a given cycle may altogether disappear and never reappear in subsequent cycles. With the ensuing distribution getting tighter about the (unaffected-by-the-losses) mean the estimated variance does not increase as it's expected to.

9.5 CV Extraction Module

In a survey like the NLSCY, support to users of the data is part of the mandate of its methodologists. One of these tasks is to guide users on how to integrate bootstrap weights into their analysis. The CV Extraction Module (CVEM) was born from the need from users to obtain approximations to variance for an estimate of a proportion in a timely fashion; this is one of the many uses made of bootstrap weights. In the past, users have relied on CV look-up tables which provided some variance approximations by making various simplifying assumptions about the actual sampling design. An approximation to the bootstrap variance, rather than the full-blown calculation of it, may become handy to users who are in exploration mode for their analysis; it allows delineating which analysis can statistically be supported from those jeopardized by sample sizes too small to carry them out.

A CVEM was introduced in the NLSCY at cycle 4, and it may very well have been a first at that time for a survey at Statistics Canada. It takes the form in the NLSCY of an Excel spreadsheet with various pre-defined domains of interest as the rows and domain descriptor and various statistics about them as columns. In other words, the CVEM has in store a variance approximation that was calculated for a wide range of domains and proportions of interest. The user gets to probe the database of variance approximations by using a tool which extracts the relevant information for the request made. In the simplest of all cases, all pre-computed estimates are stored in an Excel table and the user probes its content using Excel autofilters.

The snapshot below is taken from NLSCY's CVEM for cycle 6. An autofilter used on each column creates a pull-down menu which contains the different values (and combinations of them) admitted by the column; in this example the user chose "Atlantic" from the column "PROVINCE". Once that choice is finalized by left-clicking on it, the table screens out automatically all domains whose geographic component is not "Atlantic". One can refine further the selection by using, if needed, the autofilters on the other columns.



Microsoft Excel - CV-table-eng.XLS

A1 — Approximate variances and coefficients of variation for proportions by province and age

Approximate variances and coefficients of variation for proportions by province and age

Interpreting the information in these tables

Click here for a summary

Reset domains

(All)
(Top 10...)
(Custom...)
ALBERTA
ATLANTIC
BRITISH COLUMBIA
CANADA
MANITOBA
NEW BRUNSWICK
NEWFOUNDLAND AN
NOVA SCOTIA
ONTARIO
PRAIRIES
PRINCE EDWARD ISL
QUEBEC
SASKATCHEWAN

PROVINCE	AGE	n	TARGET_Propc	bs_var	bs_se	bs_cv	CIL95	CIU95	weights
	10	109	1%	1.27	1.13	112.77	-1.21	3.21	Original Funnel
	10	109	5%	6.10	2.47	49.40	0.16	9.84	Original Funnel
	10	109	10%	11.56	3.40	34.00	3.34	16.66	Original Funnel
	10	109	15%	16.38	4.05	26.98	7.07	22.93	Original Funnel
	10	109	20%	20.55	4.53	22.67	11.11	28.89	Original Funnel
	10	109	30%	26.98	5.19	17.31	19.82	40.18	Original Funnel
	10	109	40%	30.83	5.55	13.88	29.12	50.88	Original Funnel
	10	109	50%	32.11	5.67	11.33	38.89	61.11	Original Funnel
	10-11	235	1%	0.59	0.77	76.68	-0.50	2.50	Original Funnel
	10-11	235	5%	2.82	1.68	33.59	1.71	8.29	Original Funnel
	10-11	235	10%	5.35	2.31	23.12	5.47	14.53	Original Funnel
	10-11	235	15%	7.57	2.75	18.36	9.61	20.39	Original Funnel
ALBERTA	10-11	235	20%	9.50	3.08	15.41	13.96	26.04	Original Funnel
ALBERTA	10-11	235	30%	12.47	3.53	11.77	23.08	36.92	Original Funnel
ALBERTA	10-11	235	40%	14.25	3.78	9.44	32.60	47.40	Original Funnel
ALBERTA	10-11	235	50%	14.85	3.85	7.71	42.45	57.55	Original Funnel
ALBERTA	10-15	541	1%	0.31	0.56	55.67	-0.09	2.08	Original Funnel
ALBERTA	10-15	541	5%	1.49	1.22	24.39	2.61	7.38	Original Funnel
ALBERTA	10-15	541	10%	2.82	1.68	16.78	6.71	13.29	Original Funnel
ALBERTA	10-15	541	15%	3.99	2.00	13.32	11.08	18.92	Original Funnel

c6variances

Ready — NUM

The Longitudinal Survey of Immigrants to Canada provided its users with an even more user-friendly (and fancier) tool which is an extension of NLSCY's CVEM; their version is now what people have in mind when they speak of a "CVEM". It uses a Visual Basic for Applications (VBA) supported interface which extracts from several Excel databases the information corresponding to what was captured in the fields used by the user to define his/her request. The VBA-based CVEM was introduced mainly because the LSIC had many more domains to cover and these outgrew the storage capacities of a single Excel spreadsheet; several now had to be used and the concept of auto-filters, at the heart of NLSCY's CVEM, was extended to a multi-sheet environment with the help of VBA.

The first generation of CVEM in the NLSCY produced an approximate variance for a domain of interest for a characteristic assumed to be held by p% of people by randomly generating first such a characteristic for the sample and computing its variance estimate using the 1,000 bootstrap weights. To avoid having this variance estimate depend on this one specific realization, 100 such characteristics were independently generated and the ensuing variance estimates averaged out to form the CVEM variance approximation. In other words, the approximation to the variance for p in a domain $D, AV_{CVEM}^{(D)}(p)$, computed by the CVEM is:

$$AV_{CVEM}^{(D)}(p) = \frac{1}{100}\sum_{k=1}^{100}\left[\frac{1}{1000}\sum_{b=1}^{1000}(\hat{\theta}_{b,p,k}^{(D)} - \hat{\theta}_{s,p,k}^{(D)})^2\right]$$

where:

$\hat{\theta}^{(D)}_{s,p,k} = \frac{\sum_{j \in D \cap s} w_j I_{k,j,p}}{\sum_{j \in D \cap s} w_j}$: sample-based estimate for p in D using the k^{th} generated characteristic

and

$\hat{\theta}^{(D)}_{b,p,k} = \frac{\sum_{j \in D \cap b} w_j^{(bs)} I_{k,j,p}}{\sum_{j \in D \cap b} w_j^{(bs)}}$: b^{th} bootstrap estimate for p in D using the k^{th} generated characteristic

In the case of the NLSCY, all the domains considered turn out to be direct sums of post-strata. As a consequence, the denominator of the estimate $\hat{\theta}^{(D)}_{b,p,k}$ is not random with respect to the bootstrap. Indeed, for all replicates, it's equal to the sum of the post-strata totals which make the domain *D*.

Before we go any further, a word of caution has to be said about the CVEM. By randomly generating in such a way characteristics of interest in the sample, we construct variance approximations under the assumption that the characteristic of interest and the design weight are independent. This is a <u>very strong</u> assumption to make, one that users should keep well in mind when exploiting the CVEM. Indeed, a real dichotomous characteristic seldom can be assumed to be independent of the weights; in the case of the NLSCY, any dichotomous characteristic related in one way or another with geography is bound to be correlated with the weights. The approximate variance computed above can only be accurate if the design is self-weighted as with SRSWOR for example; any departure from that scenario will compromise the reliability of the approximation to the variance calculated this way.

The main disadvantage with this first generation is the incredible amount of computer resources it takes to run. For example, to build in a timely fashion the CVEM inside the tight production window of the NLSCY at cycle 5, several computers were used (through SAS connect) to get the job done.

By the time cycle 6 of the NLSCY had arrived a simplification in the computation of the approximate variances was made which opened the way to a significant reduction in computing time in building the CVEM. To reveal it, let's first permute the summations signs in $AV^{(D)}_{CVEM}(p)$:

$$AV^{(D)}_{CVEM}(p) = \frac{1}{1000} \sum_{b=1}^{1000} \left[\frac{1}{100} \sum_{k=1}^{100} (\hat{\theta}^{(D)}_{b,p,k} - \hat{\theta}^{(D)}_{s,p,k})^2 \right]$$

In the bracket now is the average of the square term over the randomly generated characteristics. But this is simply taking the expectation with respect to a Bernoulli random process.

$$\frac{1}{100}\sum_{k=1}^{100}(\hat{\theta}_{b,p,k}^{(D)}-\hat{\theta}_{s,p,k}^{(D)})^2 \cong E_{BERN}(\hat{\theta}_{b,p}^{(D)}-\hat{\theta}_{s,p}^{(D)})^2$$

where $\hat{\theta}_{b,p}^{(D)}$ and $\hat{\theta}_{s,p}^{(D)}$ are functions of the random Bernoulli variable I_p whose 100 realizations have yielded us $\left\{\hat{\theta}_{b,p,k}^{(D)}\right\}_{k=1}^{100}$ and $\left\{\hat{\theta}_{s,p,k}^{(D)}\right\}_{k=1}^{100}$, respectively. In the case of the NLSCY things are even simpler because all domains considered are direct sum of post-strata. This means that the denominator of a proportion is a known total and thus does not vary from one replicate to another, hence there is no need to repeatedly calculate it.

We can work out the expectation under the Bernoulli process to find:

$$\begin{aligned}
E_{BERN}(\hat{\theta}_{b,p}^{(D)}-\hat{\theta}_{s,p}^{(D)})^2 &= \frac{1}{T_D^2}E_{BERN}\left(\sum_{k=1}^{D}\left(w_{b,k}-w_k\right)I_k\right)^2 \\
&= \frac{1}{T_D^2}V_{BERN}\left(\sum_{k=1}^{D}\left(w_{b,k}-w_k\right)I_k\right) \quad \left(\text{since } \sum_{k=1}^{D}\left(w_{b,k}-w_k\right)=0\right) \\
&= \frac{\sum_{k=1}^{D}\left(w_{b,k}-w_k\right)^2}{T_D^2}\,p(1-p) \quad (\text{since the } I_k\text{'s are independent})
\end{aligned}$$

where : $T_D=\sum_{k=1}^{D}w_{b,k}=\sum_{k=1}^{D}w_k$.

The approximate variance then becomes:

$$AV_{CVEM}^{(D)}(p)=\frac{1}{1{,}000}\sum_{b=1}^{1{,}000}\left[\frac{p(1-p)\sum_{k=1}^{D}(w_k^{bs}-w_k^{s})^2}{T_D^2}\right]$$

Computationally, this is much less-intensive than the previous form since each term is computed once, not one hundred times; at cycle 6 it took just a few minutes of running time on a single computer to gather the variance estimates required by the CVEM.

Computational time can be reduced even more if one exploits the following link:

$$AV_{CVEM}^{(D)}(p) = \frac{1}{1{,}000}\sum_{b=1}^{1{,}000}\left[\frac{p(1-p)\sum_{k=1}^{D}(w_k^{bs}-w_k^{s})^2}{T_D^2}\right] \cong E_{BIN}\left(\frac{p(1-p)\sum_{k=1}^{D}(w_k^{bs}-w_k^{s})^2}{T_D^2}\right)$$

Indeed, the average taken over the replicates used is an attempt to evaluate the expectation with respect to the Binomial process responsible for the multiplicities imbedded in the bootstrap weights (see Computational Tip #6). Unfortunately, in order to carry out the algebra involved in computing explicitly this expectation one has to make some simplifying assumptions which, in the end, we don't feel are worth the effort (and whatever little further gain in computing time there is to make).

9.6 Diagnostics on variance

Non-increasing variance estimates over cycles about a static-over-time characteristic in a longitudinal survey certainly is perplexing – but what increase would actually be sensible? It's tempting to assume that the design effect (DEFF) should remain roughly constant over cycles; it sure would be convenient! But this is not a valid assumption in general. If it were true, it would mean that the loss of sample size over cycles due to nonresponse (and other contributing factors) affects the variance of a complex survey the same way it does a survey relying on a SRSWOR design. This is a very strong assumption that has no reason to hold in general.

As we saw earlier, knowing the reliability of a variance estimate is useful information to take into account in practice. For instance, we've established that in the simplest of all cases, variance estimates of the median were much less reliable than those for the mean. Unstable variance estimates may also explain (to some extent) the bizarre non-increasing variance estimates. But then, how can we build a case about unstable variance estimates when in practice we only have the benefit of one observed sample? (Surely we have the benefit of taking several sets of bootstrap replicates but as we argued before the error due to re-sampling is not the one at fault here, at least not in magnitude.)

We propose a way to help assess in any given situation just how stable the bootstrap variance estimates are; it comes from the framework of the Coefficient of Variation Extraction Module (CVEM) described above. Recall that originally the CVEM involved generating 100 Bernoulli variables with probability of a success p and obtaining for each of these a bootstrap variance estimate for a domain of interest D. The variances were averaged out and reported in the CVEM as an approximation to the bootstrap variance estimate we'd get for a (survey) variable of interest of estimated proportion p. The idea here would be to take the variance of these variances and see if it's large or not. And 100 generated variables is probably overkill; 5 or 10 could very well be enough to give you a good idea of the instability in the variance estimates that you have to deal with.

If you want to avoid simulating Bernoulli variables altogether, then you'll have to compute:

$$V_{BERN}\left(\frac{1}{1000}\sum_{b=1}^{1000}(\hat{\theta}_{b,p,k}^{(D)}-\hat{\theta}_{s,p,k}^{(D)})^2\right)$$

which is at best messy; this computation may not be worth its while compared to the effort of approximating it using a few simulated Bernoulli variables.

In a few words...

- Variance estimates are not as stable with regard to the sampling design as one would think (and wish!).

- Variance estimation for the median is noticeably trickier than that for the mean, for example.

- The CVEM depends on a very strong assumption, one that should not be taken lightly.

- Possible diagnostics on variance are hard to come by; yet, with the advent of computers (and thus simulations) surely more options will become available.

A few final words...

While many different issues were covered here in depth, in the end a *single* unifying lesson stands out: we need to opt for methodologies that strike a proper balance between performance and what's required implementing them. For instance, we notice that often the weighting methodology used is (almost) completely driven by bias considerations at the expense of a sound and easy-to-implement variance methodology. A good example of this is the issue we covered in Section 8 about the post-stratification methodology used in the NLSCY. There we found that the post-strata introduced were too numerous and led, as cycles unfolded, to empty post-strata within some bootstrap replicates. While we can devise after-the-fact *ad hoc* solutions to deal with empty post-strata in the variance estimation phase of the NLSCY, it remains that it would be preferable to have a weighting methodology which avoids creating them in the first place.

In retrospect, the paper provided you with a panoramic view of the Rao-Wu rescaled bootstrap, describing it both from a theoretical and a practical standpoint, the latter focusing on pitfalls to avoid and providing computer-based tips to keep in mind while implementing the bootstrap. We hope this paper helped bridge the gap that exists between the theory and the practice of the bootstrap by exposing and discussing at length the issues that arose in the NLSCY implementing it.

References

Ardilly, P. 2000. Les techniques d'enquête. Éditions Technip. Paris.

Deville. J.-C. and C.-E. Särndal. *Calibration estimators in survey sampling.* Journal of the American Statistical Association, 87, pp.376-382.

Dudewicz, E. J. and S. N. Mishra. 1988. Modern Mathematical Statistics. Wiley New-York.

Efron, B. 1979. *Bootstrap methods: An another look at the jackknife.* Annals of Statistics, 7, pp.11-26.

Efron, B. 1982. The Jackknife, the Bootstrap and Other Resampling Plans. Philadelphia: Society for Industrial and Applied Mathematics (SIAM).

Efron, B. and R. J. Tibshirani. An Introduction to the Bootstrap. Monographs on Statistics and Applied Probability 57. Chapman and Hall. 1993.

Faucher, D., Langlet, É and É. Lesage. 2003 An application of the bootstrap variance estimation method to the participation and activity limitation survey. Proceedings of the SSC. pp. 105-110.

Fay, B.E. 1991. *A Design-Based Perspective on Missing Data Variance.* Proceedings of the 1991 Annual Research Committee, U.S. Bureau of the Census, pp. 429-440.

Girard, C. *An extension of Woodruff's method for calculating sampling variability for the median.* Proceedings of the JSM. 2005.

Grenier, D., C. Girard, A. Lévesque and M. Simard. Méthode de calcul de CV approximatifs. Document interne. Statistique Canada. 2006

Gross, S. 1980. *Median estimation in sample surveys.* Proceedings of the Section on Survey Research Methods, American Statistical Association, 181-184.

Hansen, M.H., N.H. Hurwitz and W.G.Madow. 1953. Sample survey methods and theory. Volume II. Wiley New-York.

Haziza, D. *Inférence en présence d'imputation simple dans les enquêtes : un survol.* Journal de la Société Française de Statistique, tome 146, no.4 2005.

Laflamme, Guy. *Le calcul de variance dans l'Enquête longitudinale nationale sur les enfants et les jeunes.* Division des méthodes d'enquêtes sociales, Statistique Canada, 17 avril 2002.

Lohr, S. 1999. Sampling: design and analysis. Duxbury Press.

Mach, L., J. Dumais, and L. Robidou. *A Study of the Properties of a Bootstrap Variance Estimator Under Sampling Without Replacement*. Federal Committee on Statistical Methodology (FCSM) 2005 Research Conference. (to appear).

Preston, J. and J.O. Chipperfield. Using a Generalized Estimation Methodology for ABS Business Surveys. (Presented to ABS Methodology Advisory Committee in 2002 and available through www.abs.gov.au).

Rao, J.N.K. and C.F.J. Wu. Re-sampling Inference With Complex Survey Data. JASA, March 1988, Vol. 83, No. 401.

Rao, J.N.K., C.F.J. Wu and K. Yue. *Some recent work on re-sampling methods for complex surveys*. Survey Methodology **18.** 1992. pp. 209-217.

Roberts, G., M. Kovacevic, H. Mantel and O. Phillips. 2001. *Cross-sectional inference based on longitudinal surveys: some experiences with Statistics Canada surveys*. Federal Committee on Statistical Methodology (FCSM).

Rust, K. and JNK Rao. *Variance estimation for complex surveys using replication techniques.* Statistical Methods in Medical Research 1996; **5**; pp.283-310.

Särndal, C-E., B. Swensson and J. Wretman. 1992. Model Assisted Survey Sampling. Springer. N-Y.

Singh, A.C., S. Wu and R. Boyer. *Longitudinal survey nonresponse adjustment by weight calibration for estimation of gross flows.* Proceedings of the section on survey research methods of the ASA. 1995. pp. 396-401.

Sitter, R.R. *Comparing three bootstrap methods for survey data.* The Canadian Journal of Statistics, Vol. 20, No. 2, 1992. p. 135-154.

Statistics Canada. 2003. Survey methods and practices. Statistics Canada

Annex A

Imputation by the mean in a group seen as an imputation class and its relation to re-weighting with the same group seen as a RHG

$$\hat{Y} = \sum_{k\in r} y_k w_k + \sum_{k\in n-r} y_k^{imp} w_k \tag{A1}$$

$$= \sum_{k\in r} y_k w_k + \sum_{k\in n-r} \left(\frac{\sum_{i\in r} y_i w_i}{\sum_{i\in r} w_i} \right) w_k$$

$$= \sum_{k\in r} y_k w_k \left(1 + \frac{\sum_{i\in n-r} w_i}{\sum_{i\in r} w_i} \right)$$

$$= \sum_{k\in r} y_k w_k \left(\frac{\sum_{i\in n-r} w_i + \sum_{i\in r} w_i}{\sum_{i\in r} w_i} \right)$$

About the estimator for the mean for a domain

There are two competing estimators here in our case of estimating a domain mean with domain size N_D known (see Särndal *et al.* (1992), particularly section 10.3:

$$\hat{\bar{y}}_1 = \frac{\sum_{k\in s_D} y_k w_k}{N_D} \tag{A2}$$

$$\hat{\bar{y}}_2 = \frac{\sum_{k\in s_D} y_k w_k}{\hat{N}_D = \sum_{k\in s_D} w_k} \tag{A3}$$

It's quite perplexing to even think about using $\hat{\bar{y}}_2$ since it apparently ignores the fact that N_D is *known* and relies instead on its estimate $\hat{N}_D$. But we have to go over that because in fact $\hat{\bar{y}}_2$ is a better choice than $\hat{\bar{y}}_1$! That would indeed most perplexing if it were true that $\hat{\bar{y}}_2$ ignored altogether N_D but appearances are deceptive, for:

$$\hat{\bar{y}}_2 = \frac{\sum_{k\in s_D} y_k w_k}{\hat{N}_D} = \frac{N_D}{\hat{N}_D} \times \frac{\sum_{k\in s_D} y_k w_k}{N_D} = \frac{N_D}{\hat{N}_D} \times \hat{\bar{y}}_1 \tag{A4}$$

Now, given a sample s, two things can make the estimate $\hat{\bar{y}}_1(s)$ unduly large (resp. small): 1) s contains more than its share of units with large Y values *i.e.*, a "bad" sample was observed; 2) s is over-represented (resp. under-represented) by units from the domain: there are more (resp. less) units from the domain than we would expect on average. While there's nothing that can be done about 1), 2) could have been taken into account in time to release a more proper estimate. Indeed, all that was needed is some way to quantify if we had over-/under-representation, and this is what the ratio $\frac{N_D}{\hat{N}_D}$ in (A4) is all about. When we have over-representation, for example, the ratio is smaller than 1 and thus serves to deflate the estimate $\hat{\bar{y}}_1$ which got fooled into yielding a large estimate just because it fed "blindly" from more units than it should have had to work from.

Annex B

An extension of Woodruff's method for calculating sampling variability for the median

Claude Girard
Senior methodologist, Statistics Canada

ABSTRACT

Woodruff (1952) introduced a method to construct confidence intervals for the median having approximately targeted level. The method, as used in the context of survey sampling which is the focus of this paper, is illustrated in standard books like Särndal *et al.* (1992) and Lohr (1999).

Also known, but not well-documented, is the σ-extension: the possibility to obtain from Woodruff's method a point estimate of sampling variability for the median estimate. In the case of simple designs, basic estimators like those for totals and means have variance estimators that can be easily computed. This is important for surveys for which replication methods like the bootstrap are counter-indicated due to, for example, high sampling fractions. The appeal of Woodruff's σ-extension to survey practitioners comes from extending the range of estimators for which variance estimates can be computed to include the median (and to some extent other quantiles). Indeed, Woodruff's σ-extension produces variance estimates for the median using nothing but what's already available to compute variance estimates for the basic estimators.

Unfortunately, the σ-extension yields sampling variability estimates that are dependent upon the level used to initially construct the confidence interval from which the sampling variability estimates are derived. Furthermore, these estimates are rather unstable. These problems may very well deter practitioners from using the σ-extension. Both the dependence on the confidence level and the instability arise from the very way Woodruff's method is usually implemented. We propose a modification to the implementation of Woodruff's method that gives rise to a σ-extension free of the dependence on confidence levels and yields more stable sampling variability estimates. The approach is possibly novel from a practitioner point of view, while from a theoretical perspective it uses ideas already presented in the literature but in a way not easily seen to be of immediate use to the practicing statistician.

KEYWORDS: Woodruff, Median, Survey sampling

Woodruff's method in survey sampling

Let's briefly describe the usual implementation of Woodruff's method; see Lohr (1999) and Särndal *et al*. (1992) for further details.

Let Z be a characteristic for which an estimate of the median is sought for some domain of interest D based on a sample s of units. Here are the 5 steps used to obtain a confidence interval of approximate level $1-\alpha$ using Woodruff's method.

1) Form a graph of the empirical cumulative distribution function (CDF), based on the sample, with Z values on the x-axis and probabilities ranging from 0 to 1 on the y-axis. The empirical CDF of a given value z of Z represents the (weighted) proportion of units in the population having a value of Z smaller than z.

2) Using s, form the estimator $\hat{F}(z)$ yielding the (weighted) proportion of units having a value of Z smaller or equal to the sample-based estimate of the median $m\hat{e}d$. This is simply the weighted mean

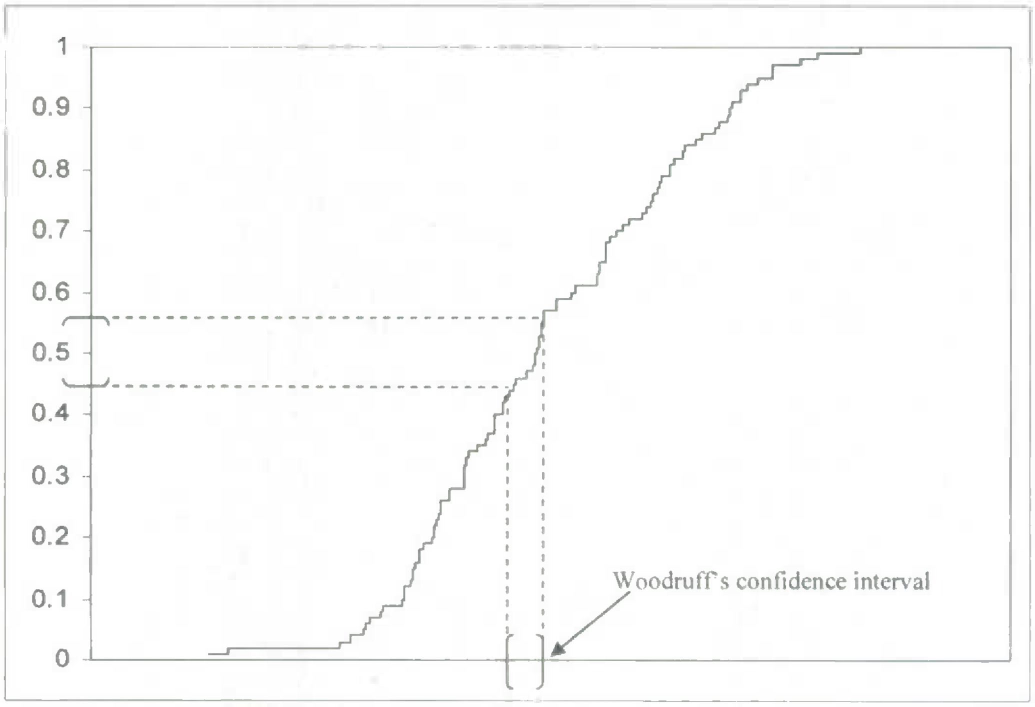
$$\hat{F}(z) = \frac{\sum w_k I_k(m\hat{e}d)}{\sum w_k}$$

where $I_k(m\hat{e}d)=1$ if $Z \leq m\hat{e}d$. (Strictly speaking, we should build the estimator around the true median *med* but since it's unknown we must rely instead on its estimate $m\hat{e}d$ and carry on!)

3) Obtain a variance estimate for the estimator used in 2). This variance computation should be possible under the given sampling design since it's for a basic estimator.

4) Form the confidence interval with the desired level using the variance obtained in 3). It can be represented on the y-axis on the graph of the CDF as in graph 1 below. Woodruff's method is about using the empirical CDF as a projector-line to project this (computable) interval onto the x-axis.

5) Find where the endpoints of the confidence interval intersect with the empirical CDF and use the corresponding x-values to define the endpoints of the confidence interval on the x-axis.



Graph 1: An empirical CDF with characteristic of interest Z on the x-axis and probabilities on the y-axis, illustrating Woodruff's method as projecting the computable y-axis confidence interval onto the x-axis to obtain the sought-after x-axis confidence interval

Observe that sample after sample we won't find the same weighted percentage of sampled *y*'s that are below the given *y*-value *mêd* but rather some array of percentages around 0.5 as estimates of it. The idea behind Woodruff's method is this: while there's no intrinsic interest in measuring the spread in these estimates, it turns out that it reflects our incapacity sample after sample to agree on one common estimate of the median. It's one thing to say that these two uncertainties are manifestations of the same thing but *how* do we get the scaling right between the two? The ratio is certainly not 1-1, so what is it? This is where the CDF comes into play; it's slope at the median (or in a smoothed-out neighbourhood of it) tells us exactly what the conversion rate ought to be.

σ-extension approach

The whole point to Woodruff's method is to *transform* using the CDF an inference we know how to make but which has no intrinsic interest of its own (*i.e.*, the confidence interval on the y-axis about 0.5) into the inference we're after *i.e.*, one on the *x*-axis. In other words, you transform what you know about a suitable weighted mean in terms of sampling variability into a statement about the estimated median through the CDF.

In some survey situations, a client may prefer to deal with point estimates of sampling variability rather than confidence intervals. For instance, a client may be interested in knowing the coefficient of variation (CV) associated with an estimate. It is known that Woodruff's method can be used to produce estimates of sampling variability: this is what we call Woodruff's σ-extension. The current implementation of the σ-extension uses the assumption of symmetry of Woodruff's confidence interval.

Assume for the moment that Woodruff's confidence interval $(L_{1-\alpha/2}, U_{1-\alpha/2})$ obtained from step 5 above on the x-axis is symmetric with midpoint *mid*:

$$(L_{1-\alpha/2}, U_{1-\alpha/2}) = (mid \pm z_{1-\alpha/2}\sigma_x) \tag{B1}$$

Then simple algebra yields the following point estimate of the sampling variability after matching the corresponding intervals endpoints:

$$\sigma_{x,1-\alpha/2} = \frac{U_{1-\alpha/2} - L_{1-\alpha/2}}{2z_{1-\alpha/2}} \tag{B2}$$

An additional subscript for σ_x indicating the confidence level was added to make explicit the dependency of σ_x on the confidence level when obtained that way. In other words, not only does the current σ-extension rests upon a dubious assumption of the symmetry of the Woodruff confidence interval, but also the results are confidence-level dependent. This would not *a priori* be a problem were the numerical estimates obtained using different confidence levels the same. But in practice they differ from one another, sometimes quite significantly. Indeed, the symmetry assumption is rarely met in practice because the empirical CDF is not a smooth projector-line (like an ideal CDF would be). Thus, we get confidence level dependent sampling variability estimates for the median since neighborhing y-values don't get projected in some consistent (read linearly) way onto the x-axis because of the stepwise CDF.

Now, suppose that instead of assuming symmetry we start from the assumption that the empirical CDF is actually linear in some neighborhood of the median (a neighborhood just large enough to contain the area used to project the y-axis confidence intervals onto the x-axis). In other words, assume that the line

$$y = mx + b \tag{B3}$$

provides a good fit to the CDF in some neighborhood of the median –see graph 2 below. Then we can show that the standard error on the y-axis, σ_y (the one we're able to compute because it's related to nothing else than some weighted mean) is related to the standard error on the x-axis, σ_x (the one we're after) through the slope in the following way:

$$\sigma_x = \sigma_y / m \tag{B4}$$

Indeed, since the confidence interval on the y-axis $(0.5-1.96\sigma_y, 0.5+1.96\sigma_y)$ is sent to $\left(\frac{0.5-1.96\sigma_y-b}{m}, \frac{0.5+1.96\sigma_y-b}{m}\right)$ the symmetric y-axis interval gets transformed into a symmetric interval on the x-axis. Assuming then that the latter was obtained as $(mid-1.96\sigma_x, mid+1.96\sigma_x)$ one gets (B4) by simply matching both intervals endpoints and solving for σ_x.

In words: if the CDF was indeed locally linear about the median, then both standard errors of interest to us would be related to one another simply by the slope of the CDF in the neighborhood just considered.

With relation (B4) in mind, let's compute the slope of the projector-line which is implicit to the symmetry assumption. Observe that the symmetry assumption comes down to using a linear projector-line determined by where the endpoints of the y-axis confidence interval *used* meet the CDF:

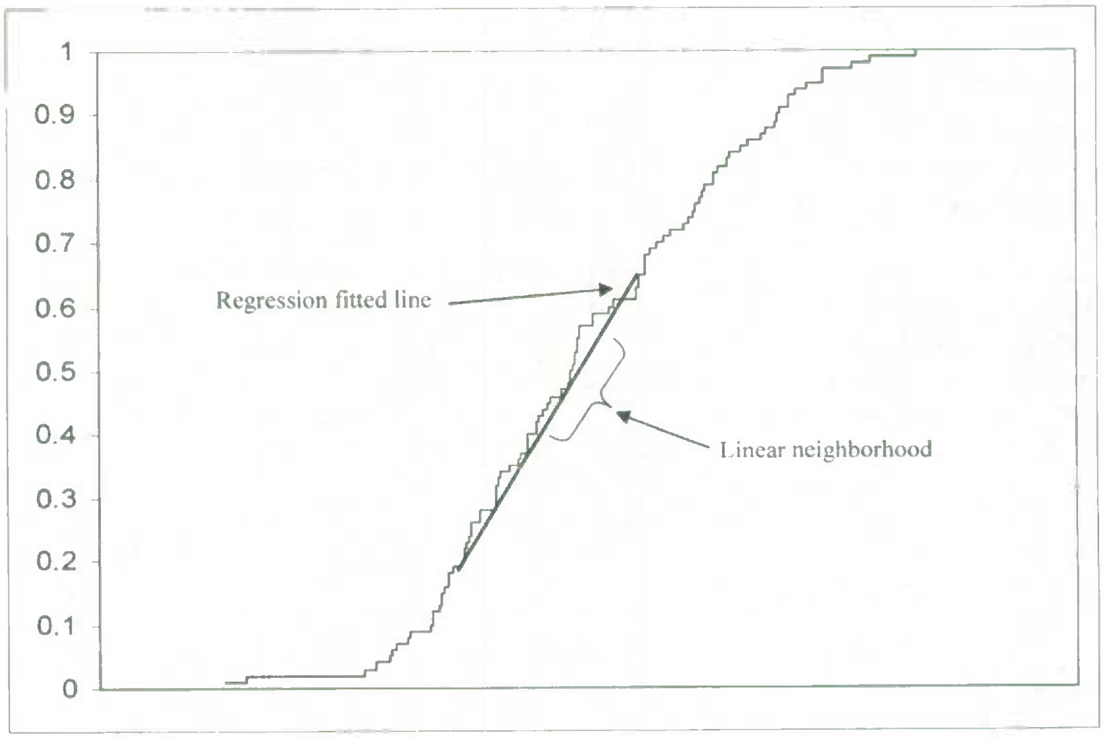
$(L_{1-\alpha/2}, 0.5-z_{1-\alpha/2}\sigma_y)$ and $(U_{1-\alpha/2}, 0.5+z_{1-\alpha/2}\sigma_y)$

This line can be seen to have slope

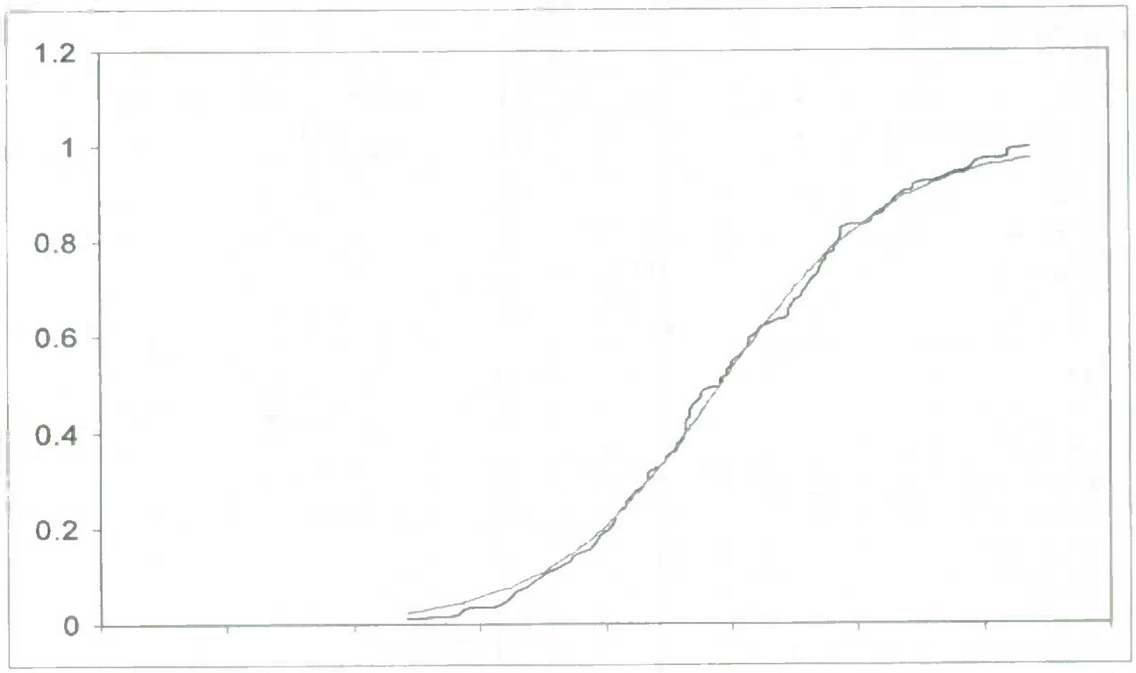
$$slope = m_{1-\alpha/2} = \frac{2\times z_{1-\alpha/2}\times\sigma_y}{U_{1-\alpha/2}-L_{1-\alpha/2}} \qquad \text{(B5)}$$

Now, feeding this into the slope-relation (B4) precisely yields estimate (B2). Thus, the assumption of symmetry boils down to assuming the CDF is linear in a neighborhood of the median and the fit made to the CDF corresponds to the line determined by just two points in the area: where the endpoints of the y-axis confidence interval meet the CDF. (In Graph 1 this is where the horizontal broken lines meet with the stepwise CDF.) The slope-relation (B4) suggests a new approach in practice: first fit a regression line to the data in some neighborhood of the median and use the ensuing slope estimate to obtain a sampling variability estimate on the x-axis through the slope-relation.

The idea is thus to first find a best projector-line by smoothing out the CDF. This can be easily done by "locally" fitting a regression line to the data or "globally" fitting a logistic type of curve to the entire CDF – see graphs 2 and 3. In either case the smoothing will rid the sampling variability estimates for the median of their initial dependence on the confidence level used to construct the confidence interval from which they're derived from.



Graph 2: Illustration of a linear neighborhood about the median and local regression fit of a projector-line.



Graph 3: Illustration of a global fit to a step CDF using a logistic-type curve.

Preliminary findings

The σ-extension was investigated for variables having a unimodal distribution using a (local) linear fit to the CDF in a neighborhood of the median using regression. Multi-mode distributions are among the counterindicated distributions since they're likely to display nonlinear neighborhoods about the median; this will happen, for instance, when the median is in between two modes.

For self-weighted surveys, the 99% endpoints yield better sampling variability estimates than both the lower 90% and 95% levels; it actually matches quite well with the regression-based estimate. This is because for steps of equal "depth" the further apart the endpoints are from one another, the more stable the slope of the line they determine will be. The most important gains in efficiency using the regression-based estimate were obtained in scenarios where the weights varied among sampled units. And depending on the weights of the units forming the neighborhood, the regression estimate of the slope can be much more reliable than the one obtained from fitting a line through the endpoints of the projected confidence interval as with the symmetry assumption. The range of probabilities used to define the linear neighborhood was, in most cases, from 0.4 to 0.6. At any rate, the practitioner should consult the graph of the step CDF which arises in his/her specific situation to help assess the proper range of probabilities to use as a neighborhood. Furthermore, the usual regression diagnostic tools like a measure of fit will help evaluate the reasonableness of the linear neighborhood assumption the practitioner is about to make.

Concluding remarks

Woodruff's approach is often considered for use because of the simplicity of its implementation. But differences between the ideal (or smooth) case and the one actually faced in practice have, in the past, deterred some practitioners from using Woodruff's approach. The approach we described in this paper is essentially about re-establishing most of the winning (*i.e.*, ideal) conditions by first smoothing out the data *before* proceeding with Woodruff's approach per se.

Those using the σ-extension by resorting to the symmetry assumption would gain by considering fitting first a regression line to a neighborhood of the median. As was seen, considerations about symmetry actually come down to assuming *some* neighborhood is linear, with an estimate of the slope built on just two points; these are the end points of the confidence interval used and delineate the neighborhood considered. It only seems natural to consider a global fit instead, obtained by fitting first a regression line through the cloud in a neighborhood of the median.

A similar gain could be obtained by smoothing the entire CDF first. For instance, many CDFs could be very well fitted by a logitistic type of a curve. In turn, the adjusted model's algebraic expression would indicate what slope value should be used to feed the slope relation.

REFERENCES

Lohr, S. (1999). Sampling: Design and Analysis. Duxbury Press, California.

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

Woodruff, R.S. (1952). Confidence intervals for medians and other position measures. Journal of the American Statistical Association 47: 635-646.

SAS Companion

This "annex" contains the SAS code that was used in the Branch Working Paper titled "*How to avoid getting all tied up bootstrapping a survey: A walk-through featuring the National Longitudinal Survey of Children and Youth*"

Program A

```
/*

Program created by; Claude Girard .

Last update; February 1st 2007

*/


data pop;
do i=1 to 10000;
y=abs(50+20*rannor(1));
output;
end;
run;

*The variables name is remisniscent of the notation used in
Sarndal et al. (1992);
proc sql;
select distinct mean(y) as true_mean,
                    (1-100/10000)*var(y)/100 as var_theor,
                    var(y) as s2u
from pop;
quit;

proc surveyselect data=pop method=srs n=100 rep=10000
seed=1
out=Monte_Carlo;
run;

data Monte_Carlo;
set Monte_Carlo;
designweight=10000/100;
run;

proc summary data=Monte_Carlo nway;
class replicate;
var y;
weight designweight;
output out=MC_estimates(drop=_TYPE_ _FREQ_) mean=MC_mean;
run;
```

```
*This odd-looking SQL statement actually is a neat and
painless
way to put something (here the population's mean) into a
macro
variable for later use (under its name &truemean).;
proc sql noprint;
select distinct mean(y)
into :truemean
from pop;
quit;

*"Later" actually came pretty quickly! ;
data MC_estimates;
set MC_estimates;
diff_thetahat_theta=(MC_mean-&truemean.);
label diff_thetahat_theta='error i.e., theta_hat minus
theta';
run;

*The midpoints statement is NOT necessary.  Actually, those
values
were found after running the preliminary univariate without
the midpoints option.  This is the way to make sure
different
histograms are on the same scale. You do not need to
specify
the variable for which a histogram is needed in the var
statement.
Just note that with a missing var statement proc univariate
will
provide you with statistics about everything your input
file
contains in terms of variables.;

proc univariate data=MC_estimates;
var diff_thetahat_theta;
histogram diff_thetahat_theta / midpoints=-7.75 to 7.75 by
0.25;
run;

*The Monte Carlo variance estimate computed from the SQL is
disclosed in the output window since no table (i.e. output
dataset) was requested.  Observe that this variance
estimate
was already found in the output of the proc univariate
above.;
proc sql;
```

```
select distinct var(MC_mean) as var_MC
from MC_estimates;
quit;

*Carrying out the bootstrap now.;

proc surveyselect data=pop method=srs n=100 seed=5
out=observed_sample;
run;

data observed_sample;
set observed_sample;
designweight=10000/100;
run;

proc sql;
select distinct (1-100/10000)*var(y)/100 as
var_theor_estimated,
var(y) as s2s
from observed_sample;
quit;

proc sql noprint;
select distinct sum(designweight*y)/sum(designweight)
into :estimatedmean
from observed_sample;
quit;

proc surveyselect data=observed_sample method=urs n=99
rep=10000
     seed=2 out=bootstrap_replicates;
run;

data bootstrap_replicates;
set bootstrap_replicates;
bootstrapweight=designweight*(100/99)*numberhits;
run;

proc summary data=bootstrap_replicates nway;
class replicate;
var y;
weight bootstrapweight;
output out=bootstrap_estimates(drop=_type_ _freq_)
mean=bs_mean;
run;

data bootstrap_estimates;
```

```
set bootstrap_estimates;
diff_thetabs_thetahat=bs_mean - &estimatedmean.;
label diff_thetabs_thetahat='bootstrap error i.e.,
theta_hat_bootstrap minus theta_hat';
run;

proc univariate data=bootstrap_estimates;
histogram diff_thetabs_thetahat / midpoints=-7.75 to 7.75
by 0.25;
run;

proc sql;
select distinct var(bs_mean) as var_bs_shortcut,
mean(diff_thetabs_thetahat**2) as var_bs
from bootstrap_estimates;
quit;
```

Program B

```
/*

Program created by; Claude Girard

Last update; February 1st 2007

*/

*Program B describes the rule of thumb mentioned in
Computational Tip #3.;

%let capn=10000;
%let smalln=100;

data pop;
do i=1 to &capn.;
y=50+20*rannor(1);
output;
end;
run;

*The result of the calculation of the SQLs below are given
in the
output window (because a "create table as" statement was
not used).;
proc sql;
select distinct (1-&smalln./&capn.)*var(y)/&smalln. as
var_theor,
var(y) as s2u
from pop;
quit;

proc surveyselect data=pop method=srs n=100 seed=1
out=observed_sample;
run;

data observed_sample;
set observed_sample;
designweight=&capn./&smalln.;
run;

proc sql;
select distinct (1-&smalln./&capn.)*var(y)/&smalln. as
var_theor_estimated,
var(y) as s2s
from observed_sample;
quit;

*We now bootstrap the observed sample by taking more and
more
replicates with each macro call.;
%macro repete(seed,nrepl);

%let nrep=%sysevalf(&smalln.-1);

proc surveyselect data=observed_sample method=urs n=&nrep.
rep=&nrepl.
     seed=&seed. out=bootstrap_replicates;
run;

data bootstrap_replicates;
set bootstrap_replicates;
bootstrapweight=designweight*(&smalln./&nrep.)*numberhits;
run;

proc summary data=bootstrap_replicates nway;
class replicate;
var y;
weight bootstrapweight;
output out=bootstrap_estimates(drop=_type_ _freq_)
mean=bs_mean;
run;

proc sql;
create table temp as
select distinct &nrepl. as replicates,var(bs_mean) as
variance
from bootstrap_estimates;
quit;

/*The next piece of code simply puts back all files
arising*/
/*from the various calls into a single one.*/

%if &nrepl.=50 %then %do;
data variances;
set temp;
run;
%end;
%else %do;
```

```
data variances;
set variances temp;
run;

%end;

%mend;
%repete(10,50);
%repete(10,100);
%repete(10,250);
%repete(10,500);
%repete(10,750);
%repete(10,1000);
%repete(10,1250);
%repete(10,1500);
%repete(10,1750);
%repete(10,2000);
%repete(10,2250);
%repete(10,2500);
%repete(10,2750);
%repete(10,3000);

/*The content of the one file variances can be either
plotted*/
/*using graph-n-go from SAS (Solutions->Reporting->Graph-n-
go)*/
/*or export the file to Excel and form the graph there. */
```

Program C

```
/*

Program created by; Claude Girard

Last update; February 1st 2007

*/

/*This program is associated with Simulation C*/

data pop;
do id=1 to 10000;
y=abs(50+20*rannor(1));
output;
end;
run;

%macro repete(samplesize);
/*Computing exact variance for SSRWR and SRSWOR - the
results*/
/*are displayed in the output window.*/


proc sql;
select distinct (1-&samplesize./10000)*var(y)/&samplesize.
     as var_srswor_&samplesize.,
     var(y)/&samplesize. as var_srswr_&samplesize.
from pop;
quit;

proc surveyselect data=pop method=srs n=&samplesize.
     seed=5 out=observed_sample;
run;

data observed_sample;
set observed_sample;
designweight=10000/&samplesize.;
run;

/*Now computing estimated variance for SSRWR and SRSWOR -
the results*/
/*are displayed in the output window.*/

proc sql;
select distinct (1-&samplesize./10000)*var(y)/&samplesize.
```

```
     as var_srswor_estimated_&samplesize.,
     var(y)/&samplesize. as
var_srswr_estimated_&samplesize.
from observed_sample;
quit;

/*Setting up the sample size required for the bootstrap.*/
%let nminus1=%sysevalf(&samplesize-1);

proc surveyselect data=observed_sample method=urs
n=&nminus1 rep=1000
     seed=4 out=bootstrap_replicates outall;
run;

/*Proceeding with the bootstrap with no finite population
correction*/
/*factor (nofpc).*/
data bootstrap_weights;
set bootstrap_replicates;
bootstrapweight_nofpc=designweight*(&samplesize./&nminus1.)
*numberhits;
f=&samplesize./10000;
bootstrapweight_fpc=
     designweight*(1-sqrt(1-f)+sqrt(1-
f)*(&samplesize./&nminus1.)*numberhits);
run;

proc summary data=bootstrap_weights nway;
class replicate;
var y;
weight bootstrapweight_nofpc;
output out=bootstrap_estimates_nofpc(drop=_type_ _freq_)
mean=bs_mean;
run;

proc sql;
select distinct var(bs_mean) as var_bs_nofpc_&samplesize.
from bootstrap_estimates_nofpc;
quit;

proc summary data=bootstrap_weights nway;
class replicate;
var y;
weight bootstrapweight_fpc;
output out=bootstrap_estimates_fpc(drop=_type_ _freq_)
mean=bs_mean;
run;
```

```
proc sql;
select distinct var(bs_mean) as var_bs_fpc_&samplesize.
from bootstrap_estimates_fpc;
quit;

%mend;
%repete(100);
%repete(500);
%repete(1000);
%repete(2000);
%repete(3000);
%repete(4000);
%repete(5000);
```

Program D

```
/*

Program created by; Claude Girard

Last update; February 1st 2007

*/

%let capn=10000;
%let smalln=100;

/*rep specifies the number of observed samples that will be
bootstrapped.*/

%let rep=100;

data pop;
do i=1 to &capn.;
     income_nor=abs(50+20*rannor(1));
     income_exp=500*ranexp(111);
output;
end;
run;
*Note: The need to take the absolute value of income_nor
will be
explained when time comes below.;

*The histogram option of univariate is handy -  you may
even require
a best-fit curve to be added.;
proc univariate data=pop;
var income_nor income_exp;
histogram income_nor income_exp;
run;

*Drawing &rep samples of size &smalln under SRSWOR.;
proc surveyselect data=pop method=srs n=&smalln. seed=1
rep=&rep. noprint
     out=observed_samples;
run;

*For each observed sample i.e. each repetition,
we want the closed-form variance calculation.;
proc summary data=observed_samples nway;
```

```
class replicate;
var income_nor income_exp;
output out=s2u(drop=_type_ _freq_) var=s2nor s2exp;
run;

data closed_form;
set s2u;
closed_form_nor=(1-&smalln./&capn.)*s2nor/&smalln.;
closed_form_exp=(1-&smalln./&capn.)*s2exp/&smalln.;
run;

*CV of variance estimates obtained using the closed-form
formula.;
proc sql;
create table stability_closed_form as
select distinct
100*sqrt(var(closed_form_nor))/mean(closed_form_nor) as
cv_norm,
100*sqrt(var(closed_form_exp))/mean(closed_form_exp) as
cv_exp
from closed_form;
quit;

*To make sure we bootstrap each observed sample the same
way,
we'll re-sample labels from 1 to &smalln.  So, in a given
observed sample, we're not going to refer to a unit through
its population id but rather from its label.;

data labels;
do label=1 to &smalln.;
output;
end;
run;

*The bootstrap calls for sampling from one unit less than
the number
of units sampled.;
%let nrep=%sysevalf(&smalln.-1);

*Bootstrapping the sample per se: urs refers to SRSWR.;
proc surveyselect data=labels method=urs n=&nrep. seed=10
rep=1000 noprint
     out=bootstrap_sample;
run;

proc sort data=bootstrap_sample;
```

```
by label;
run;

*So far, we had our bootstrap replicates as lines. It will
be more
convenient for what follows to have the bootstrap
information coded
as columns.  Instead of using a messy proc transpose we
achieve that
through the following.;
data bootstrap_sample;
set bootstrap_sample;
by label;
array replicat{1000} replicat1-replicat1000 (1000*0);
if first.label then do i=1 to 1000;
     replicat{i}=0;
end;
replicat(replicate)=NumberHits;
if last.label;
drop replicate i numberhits;
run;
*Now the file has one entry for each label from 1 to 100
and the 1000
bootstrap replicates appear as columns;

*Computing from the multiplicities replicat1-replicat1000
the
ensuing bootstrap weights.  This comes from formula (9).;
data bootstrap_sample;
set bootstrap_sample;
array replicat{1000} replicat1-replicat1000;
array bsw{1000} bsw1-bsw1000;
do i=1 to 1000;
     bsw(i)=(&capn./&smalln.)*(1-sqrt(1-&smalln./&capn.)+
     sqrt(1-
&smalln./&capn.)*replicat(i)*(&smalln./&nrep.));
end;
keep label bsw1-bsw1000;
run;

*Associating to each selected unit of all observed sample
one label from
1 to 100.;
data observed_samples;
set observed_samples;
label=_N_-&smalln.*(replicate-1);
run;
```

```
proc sort data=observed_samples;
by label;
run;

data observed_samples_bootstrapped;
merge observed_samples bootstrap_sample;
by label;
run;

*There are 1000 bootstrap-weighted means to compute, one
for each
series of weights bsw. The problem is that SAS allows only
one
variable to be specified in a weight statement.
The following trick is also used in BootVar.  It consists
of computing
separately the 1000 numerators and denominators using two
proc
summaries with the bootstrap weights and variable of
interest interverted
with regard to the var and weight statements.

Warning: for the weight statement to work properly in such
an inversion,
the variable of interest has to be non-zero - this is why
the absolute
value was taken above ensuring that the normal variable was
positive.;
proc summary data=observed_samples_bootstrapped nway;
class replicate;
var bsw1-bsw1000;
weight income_nor;
output out=numerator_nor
sum=num1-num1000;
run;

proc summary data=observed_samples_bootstrapped nway;
class replicate;
var bsw1-bsw1000;
output out=denominator(drop=_type_ _freq_)
sum=denom1-denom1000;
run;

*Putting everything back together to compute the means per
se.;
data means_nor;
```

```
merge numerator_nor denominator;
by replicate;
array moy{1000} moy1-moy1000;
array num{1000} num1-num1000;
array denom{1000} denom1-denom1000;
do i=1 to 1000;
moy(i)=num(i)/denom(i);
end;
keep replicate moy1-moy1000;
run;

*Computing the (shortcut form) of the bootstrap variance
for
each observed sample.;
data bs_variances_nor;
set means_nor;
var_bs_nor=var(of moy1-moy1000);
keep replicate var_bs_nor;
run;

*Doing things all over with the exponentially distributed
variable now.;
proc summary data=observed_samples_bootstrapped nway;
class replicate;
var bsw1-bsw1000;
weight income_exp;
output out=numerator_exp(drop=_type_ _freq_)
sum=num1-num1000;
run;

data means_exp;
merge numerator_exp denominator;
by replicate;
array moy{1000} moy1-moy1000;
array num{1000} num1-num1000;
array denom{1000} denom1-denom1000;
do i=1 to 1000;
moy(i)=num(i)/denom(i);
end;
keep replicate moy1-moy1000;
run;

data bs_variances_exp;
set means_exp;
var_bs_exp=var(of moy1-moy1000);
keep replicate var_bs_exp;
run;
```

```
data bs_variances;
merge bs_variances_nor bs_variances_exp;
by replicate;
run;

proc sql;
create table stability_bs as
select distinct 100*sqrt(var(var_bs_nor))/mean(var_bs_nor)
as cv_nor,
100*sqrt(var(var_bs_exp))/mean(var_bs_exp) as cv_exp
from bs_variances;
quit;

%macro medianes(n);

%do i=1 %to &n.;

proc summary data=observed_samples_bootstrapped
(keep=replicate income_nor bsw&i.) nway;
class replicate;
var income_nor;
weight bsw&i.;
output out=med_nor_&i.(drop=_type_ _freq_) mean=mean&i.
median=mediane&i.;
run;

proc summary data=observed_samples_bootstrapped
(keep=replicate income_exp bsw&i.) nway;
class replicate;
var income_exp;
weight bsw&i.;
output out=med_exp_&i.(drop=_type_ _freq_) mean=mean&i.
median=mediane&i.;
run;

%end;
%mend;
%medianes(1000);

*The macro generated a whole bunch of output datasets that
we need
to have back as one file.  One painless way to do so is to
use
a "background" SAS file. It's actually a SQL-view, called
vcolumn
```

```
located in the library SASHELP.  It contais a myriad of
info
on datasets that one can exploit. (Another view from the
same library
which is useful from time to time is vtable.)  The code
below puts
into a list the names of all datasets of the WORK library
which
have MEM_NOR_ as their first 8 characters - these are
precisely the
files I want to put together.;

proc sql noprint;
select distinct MEMNAME
into :varlist separated by ' '
from sashelp.vcolumn
where LIBNAME="WORK" and substr(MEMNAME,1,8)="MED_NOR_";
quit;

*Had all the datatsets' name been simply concatenated as
MEM_NOR_1MEM_NOR_2etc. this would be of little use.  What
is
useful is to have a blank inserted in between each name to
get
in &varlist the following MEM_NOR_1 MEM_NOR_2 MEM_NOR_3 and
so on.
This is the role of "separated by" in the SQL above.;
data bs_med_nor;
merge &varlist.;
by replicate;
var_bs_med_nor=var(of mediane1-mediane1000);
var_bs_mean_nor=var(of mean1-mean1000);
keep replicate var_bs_med_nor var_bs_mean_nor;
run;

*Getting rid now of all those datasets since they're
useless now.;
proc datasets library=work nolist;
delete &varlist.;
quit;

proc sql noprint;
select distinct MEMNAME
into :varlist separated by ' '
from sashelp.vcolumn
where LIBNAME="WORK" and substr(MEMNAME,1,8)="MED_EXP_";
quit;
```

```
data bs_med_exp;
merge &varlist.;
by replicate;
var_bs_med_exp=var(of mediane1-mediane1000);
var_bs_mean_exp=var(of mean1-mean1000);
keep replicate var_bs_med_exp var_bs_mean_exp;
run;

proc datasets library=work;
delete &varlist.;
quit;

data bs_medianes;
merge bs_med_nor bs_med_exp;
by replicate;
run;

proc sql;
create table stability_medianes as
select distinct
100*sqrt(var(var_bs_med_nor))/mean(var_bs_med_nor) as
cv_med_nor,
100*sqrt(var(var_bs_mean_nor))/mean(var_bs_mean_nor) as
cv_mean_nor,
100*sqrt(var(var_bs_med_exp))/mean(var_bs_med_exp) as
cv_med_exp,
100*sqrt(var(var_bs_mean_exp))/mean(var_bs_mean_exp) as
cv_mean_exp
from bs_medianes;
quit;
```

Program E

```
/*

Program created by; Claude Girard

Last update; February 1st 2007

*/

%let capn=10000;
%let smalln=100;
%let seed=4;

data pop;
do i=1 to &capn.;
     income_nor=abs(50+20*rannor(1));
     income_exp=500*ranexp(111);
output;
end;
run;

proc univariate data=pop;
histogram income_nor income_exp;
run;

*One sample is drawn under SRSWOR and it will be
bootstrapped several
times below.;
proc surveyselect data=pop method=srs n=&smalln.
seed=&seed. noprint
     out=observed_sample;
run;

proc sql;
select distinct (1-
&smalln./&capn.)*var(income_nor)/&smalln.
,(1-&smalln./&capn.)*var(income_exp)/&smalln.

into :varnor,:varexp
from observed_sample;
quit;

%let nrep=%sysevalf(&smalln.-1);
```

```
proc surveyselect data=observed_sample method=urs n=&nrep.
seed=10 rep=100000 noprint
     out=bootstrap_sample;
run;

data bootstrap_sample;
set bootstrap_sample;
repetition=mod(replicate,100);
if repetition=0 then repetition=100;
bsw=(&capn./&smalln.)*(&smalln./&nrep.)*numberhits;
run;

proc summary data=bootstrap_sample nway;
class repetition replicate;
var income_nor income_exp;
weight bsw;
output out=estimates_bs(drop=_type_ _freq_) mean=moy_nor
moy_exp
     median=med_nor med_exp;
run;

proc univariate data=estimates_bs;
var moy_nor moy_exp med_nor med_exp;
histogram moy_nor moy_exp med_nor med_exp;
run;

proc summary data=estimates_bs nway;
class repetition;
var moy_nor moy_exp med_nor med_exp;
output out=variances_bs(drop=_type_ _freq_) var=;
run;

proc sql;
create table stability_bs as
select distinct 100*sqrt(var(moy_nor))/mean(moy_nor) as
cv_mean_nor,
100*sqrt(var(moy_exp))/mean(moy_exp) as cv_mean_exp,
100*sqrt(var(med_nor))/mean(med_nor) as cv_med_nor,
100*sqrt(var(med_exp))/mean(med_exp) as cv_med_exp
from variances_bs;
quit;

data variances_bs;
set variances_bs;
diff_sq_mean_nor=(&varnor.-moy_nor)**2;
diff_sq_mean_exp=(&varexp.-moy_exp)**2;
run;
```

```
proc sql;
select distinct 100*sqrt(mean(diff_sq_mean_nor))/&varnor.
as rmse_nor,
100*sqrt(mean(diff_sq_mean_exp))/&varexp. as rmse_exp,
100*(mean(moy_nor)-&varnor.)/&varnor. as rb_nor,
100*(mean(moy_exp)-&varexp.)/&varexp. as rb_exp
from variances_bs;
quit;
```

Program E2

```
/*

Program created by; Claude Girard

Last update; February 1st 2007

*/

%let capn=1000;
%let smalln=100;

/*The seed option controls the set of observed samples
that*/
/*are being drawn.*/
%let seed=4;

data pop;
do id=1 to &capn.;
     income_nor=abs(50+20*rannor(1));
     income_exp=500*ranexp(111);
output;
end;
run;

proc univariate data=pop;
histogram income_nor income_exp;
run;

*One sample is drawn under SRSWOR and it will be
bootstrapped several
times below.;
proc surveyselect data=pop method=srs n=&smalln.
seed=&seed. noprint
     out=observed_sample;
run;

%let nrep=%sysevalf(&smalln.-1);

proc surveyselect data=observed_sample method=urs n=&nrep.
seed=10 rep=100000 noprint
     out=bootstrap outall;
run;

/*The bootstrap weights are for negligible fpc*/
data bootstrap_sample;
```

```
set bootstrap;
repetition=mod(replicate,100);
if repetition=0 then repetition=100;
designweight=&capn./&smalln.;
bsw=designweight*
(1-sqrt(1-&smalln./&capn.)+sqrt(1-
&smalln./&capn.)*(&smalln./&nrep.)*numberhits);
run;

proc summary data=bootstrap_sample nway;
class repetition replicate;
var income_nor income_exp;
weight bsw;
output out=estimates_bs(drop=_type_ _freq_) mean=moy_nor
moy_exp;
run;

proc univariate data=estimates_bs;
var moy_nor moy_exp;
histogram moy_nor moy_exp;
run;

proc summary data=estimates_bs nway;
class repetition;
var moy_nor moy_exp;
output out=variances_bs(drop=_type_ _freq_) var=;
run;

proc sql;
create table stability_bs as
select distinct 100*sqrt(var(moy_nor))/mean(moy_nor) as
cv_mean_nor,
100*sqrt(var(moy_exp))/mean(moy_exp) as cv_mean_exp
from variances_bs;
quit;
```

Program F

```
/*

Program created by; Claude Girard

Last update; February 1st 2007

*/

%let capn=10000;
%let smalln=100;

data pop;
do i=1 to &capn.;
     income_nor=abs(50+20*rannor(1));
output;
end;
run;

proc surveyselect data=pop method=srs n=&smalln. rep=10000
seed=4 out=mc;
run;

proc summary data=mc nway;
class replicate;
var income_nor;
output out=mc_medians(drop=_type_ _freq_) median=mediane;
run;

proc sql;
select distinct var(mediane) as MC_median
from mc_medians;
quit;

/*This is a Monte Carlo approximation of what the sampling
distribution */
/*looks like.*/
proc univariate data=mc_medians;
var median;
histogram median;
run;

/*Computing an approximation of the exact variance based
on*/
/*asymptotics drawn from classical statistics.*/
```

```
proc sql;
select distinct (constant('PI')/2)*(1-
&smalln./&capn.)*var(income_nor)/&smalln. as var_median
from pop;
quit;

proc surveyselect data=pop method=srs n=&smalln. seed=4
noprint
     out=sample;
run;

%let nrep=%sysevalf(&smalln.-1);

proc surveyselect data=sample method=urs n=&nrep. seed=10
rep=10000 noprint
     out=bootstrap outall;
run;

data bootstrap;
set bootstrap;
poids=(&capn./&smalln.)*(1-sqrt(1-&smalln./&capn.)+
     sqrt(1-&smalln./&capn.)*numberhits*(&smalln./&nrep.));
run;

/*Computing the weighted medians the SAS way i.e. reporting
as estimate the one observation in the dataset whose
cumulated
weight first exceeds 50% of the total sum of weights. */
proc summary data=bootstrap nway;
class replicate;
var income_nor;
weight poids;
output out=SAS_medians(drop=_type_ _freq_) median=mediane;
run;

proc sql;
select distinct var(mediane) as SAS_median
from SAS_medians;
quit;

/*This results in a very discrete histogram*/
proc univariate data=SAS_medians;
var mediane;
histogram mediane;
run;

/*Preparing the ground for the interpolation*/
```

```
proc sql;
create table bootstrap2 as
select *,poids/sum(poids) as poids_normalise
from bootstrap
group by replicate
order by replicate,income_nor;
quit;

/*For some obscure reason, SQL has a hard time creating
exact */
/*fractions through its calculations.  So even if the 50th
*/
/*percentile is in the file, the cumulative weight
poids_cumul*/
/*with show up as something like 0.5000000001 or
0.49999999999*/
/*instead of 0.5, hence the criterion involving the 0.00001
below.*/
data bootstrap2;
set bootstrap2;
by replicate income_nor;
if first.replicate then poids_cumul=poids_normalise;
else poids_cumul+poids_normalise;
lag_income=max(lag(income_nor),0);
lag_poids=max(lag(poids),0);
lag_cumul=max(lag(poids_cumul),0);
median_tentative=(poids*income_nor+lag_poids*lag_income)/(p
oids+lag_poids);
if abs(poids_cumul-0.5)<0.00001 then mediane=income_nor;
else if lag_cumul<0.5 and poids_cumul>0.5 then
mediane=median_tentative;
if mediane ne .;
run;

proc sql;
select distinct var(mediane) as interpol_median
from bootstrap2;
quit;

proc univariate data=bootstrap2;
var mediane;
histogram mediane;
run;
```

Program G

```
/*

Program created by; Claude Girard

Last update; February 1st 2007

*/

%let capn=10000;
%let smalln=2500;
%let r=.5;

********************;

%let nombre=%sysevalf(&r.*&capn.);
data pop;
do id=1 to &capn.;
y=50+20*rannor(1);
if id<=&nombre then resp=1;
else resp=0;
output;
end;
run;

proc surveyselect data=pop method=srs n=&smalln. seed=1
out=observed_sample;
run;

%let rep=%sysevalf(&smalln.-1);

proc surveyselect data=observed_sample method=urs n=&rep.
rep=1000
     seed=2 out=bootstrap_replicates outall;
run;

data bootstrap_weights;
set bootstrap_replicates;
designweight=&capn./&smalln.;
bootstrapweight=designweight*
(1-sqrt(1-&smalln./&capn.)+sqrt(1-
&smalln./&capn.)*(&smalln./&rep.)*numberhits);
run;

proc sql;
```

```
create table bootstrap_nr as
select
*,bootstrapweight*sum(bootstrapweight)/sum(bootstrapweight*
(resp=1))
as bootstrapweightnr
from bootstrap_weights
group by replicate
having resp=1;
quit;

proc summary data=bootstrap_nr nway;
class replicate;
var y;
weight bootstrapweightnr;
output out=bs_estimates(drop=_type_ _freq_)
mean=bs_estimate;
run;

proc sql;
select
     (1-&smalln./&capn.)*var(y)/count(*) as v1,
     (&smalln./&capn.-count(*)/&capn.)*var(y)/count(*) as
v2,
     (1-count(*)/&capn.)*var(y)/count(*) as vt,
     count(*) as r, mean(y) as parameter_estimate
from observed_sample
where resp=1;
quit;

proc sql;
select distinct mean(bs_estimate) as bs_mean,
     var(bs_estimate) as v_bs
from bs_estimates;
quit;
```

Program G2

```
/*

Program created by; Claude Girard

Last update; February 1st 2007

*/

%let capn=10000;
%let smalln=5000;
%let r=.5;

********************;

%let nombre=%sysevalf(&r.*&capn.);
data pop;
do id=1 to &capn.;
y=50+20*rannor(1);
if id<=&nombre then resp=1;
else resp=0;
output;
end;
run;

proc surveyselect data=pop method=srs n=&smalln. seed=1
out=observed_sample;
run;

data respondents;
set observed_sample;
where resp=1;
run;

proc sql;
select distinct count(*),count(*)-1
into :r,:rep
from respondents;
quit;
*bootstrap des repondants;

proc surveyselect data=respondents method=urs n=&rep.
rep=1000
     seed=2 out=bootstrap_replicates(drop=y resp);
run;
```

```
data copies;
set respondents;
do replicate=1 to 1000;
output;
end;
run;

proc sort data=copies;
by replicate id;
run;

data bootstrap;
merge copies bootstrap_replicates;
by replicate id;
if numberhits=. then numberhits=0;
run;

data bootstrap;
set bootstrap;
designweight=&smalln./&r.;
bootstrapweight=designweight*
(1-sqrt((&smalln./&capn.)*(1-
&r./&smalln.))+sqrt((&smalln./&capn.)*(1-
&r./&smalln.))*(&r./&rep.)*numberhits);
run;

proc summary data=bootstrap nway;
class replicate;
var y;
weight bootstrapweight;
output out=bs_estimates(drop=_type_ _freq_)
mean=bs_estimate;
run;

proc sql;
select distinct mean(bs_estimate) as bs_mean,
     var(bs_estimate) as v_bs
from bs_estimates;
quit;
```

Program H

```
/*

Program created by; Claude Girard

Last update; February 1st 2007

*/
%let smalln=2500;

data pop;
do i=1 to 10000;
y=abs(50+20*rannor(1));
if i<=5000 then do;
     post_stratum='M';
     total=6000;
end;
else do;
     post_stratum='W';
     total=4000;
end;
output;
end;
run;

proc surveyselect data=pop method=srs n=&smalln. seed=4
out=observed_sample;
run;

proc sql;
create table var_components as
select distinct (total/10000)*mean(y) as est,
(total/10000)**2*(1/count(*)-1/total)*var(y) as component
from observed_sample
group by post_stratum;
quit;

proc sql;
select distinct sum(est) as estimate,
sum(component) as tot_var_estimated
from var_components;
quit;

%let nrep=%sysevalf(&smalln.-1);
```

```
proc surveyselect data=observed_sample method=urs n=&nrep.
rep=1000 seed=2
out=mult outall;
run;

data bs;
set mult;
bsweight=10000/&smalln.*(1-sqrt(1-&smalln./10000)+
sqrt(1-&smalln./10000)*numberhits*(&smalln./&nrep.));
run;

proc sql;
create table bs_pstr as
select *,bsweight*(total/sum(bsweight)) as bsweightpstr
from bs
group by replicate,post_stratum;
quit;

proc summary data=bs_pstr nway;
class replicate;
var y;
weight bsweightpstr;
output out=bs_estimates(drop=_type_ _freq_)
mean=bs_estimate;
run;

proc sql;
select distinct mean(bs_estimate) as avg_bs
,var(bs_estimate) as var_bs
from bs_estimates;
quit;
```

Program I

```
/*

Program created by; Claude Girard

Last update; February 1st 2007

*/

libname server "c:\";

data server.mult;
array mult{1000} mult1-mult1000;

do j=1 to 1000;
mult(j)=ranpoi(1,1);
end;
do i=1 to 10000;
output;
end;
run;

data server.bootstrap;
array bsw{1000} bsw1-bsw1000;

do j=1 to 1000;
bsw(j)=1000*ranuni(1)*constant('PI');
end;
do i=1 to 10000;
output;
end;
run;

data server.mult_comp(compress=BINARY);
set server.mult;
length mult1-mult1000 3.;
run;

data server.bootstrap_comp(compress=BINARY);
set server.bootstrap;
length bsw1-bsw1000 4.;
run;
```