**IERI TECHNICAL NOTES**

# Rescaling sampling weights and selecting mini-samples from large-scale assessment databases

**Eugenio J. Gonzalez**
*Educational Testing Service, Princeton, New Jersey, United States*

This edition of IERI Technical Notes addresses two different, yet related, issues that researchers often encounter when working with large-scale assessment (LSA) databases. The first issue discussed is that of rescaling sample weights so that they add to a predetermined number that is set according to design or computational needs. This procedure is necessary when we want to work with entire databases, such as those assembled by international large-scale assessments, and when we want groups of cases that exist in different proportions in the population, such as countries, to contribute in equal proportions to summary statistics. A case in point is calculating item parameters with data from across all participating countries so that each country contributes equally in the calculation of these estimates. Rescaling the sample weights is also useful when we want the sum of the sample weights to add to a number equal to, for example, the sample size or effective sample size, in cases where the statistical software used has no straightforward way to adjust for the use of sampling weights.

The second issue is that of selecting subsets of cases from the database while taking into account their original selection probabilities and sampling weights. The procedure described in this technical note involves selecting one or multiple samples from the database, such that the resulting subset of cases can be treated as a simple random sample from the population. As a result of this selection, each record has equal weight, eliminating the need to use sampling weights in the calculations because the resulting data are the equivalent of selecting a simple random sample. This type of selection is useful when, as with rescaling sample weights (issue one above), we want to select samples of equal size from within different groups of the population. Using a file built in this way leads to each group of the population contributing equally to summary statistics. This type of selection is also useful when we want to select multiple samples

*IERI Monograph Series: Issues and Methodologies in Large-Scale Assessments Volume 5*

from a database and to use these to study the variability of estimates, a process that is akin to using bootstrapping techniques, but involves selecting without replacement.

## SAMPLE SELECTION IN LSA

Sample selection in LSA usually involves two or more stages. During the first stage, schools are selected proportional to their size. During the second stage, students are chosen within the selected school. Depending on the sample design, classrooms might be selected within the school, and then including some or all of the students in the selected classroom. Students might also be selected directly across classes. Although these procedures vary somewhat, they all make it possible to calculate the probability of selection at each stage. Because the probabilities of selection at each stage are independent of one another, the overall probability of selection of any case can also be calculated as the product of the individual probabilities at the different stages. Further detail on how specific international LSA programs carry out their sample selection is beyond the scope of this paper, but can be found in the corresponding study technical reports (see, for example, Martin, Mullis, & Kennedy, 2007; Olson, Martin, & Mullis, 2008; Organisation for Economic Co-operation and Development [OECD], 2009). Kalton (1983), Kish (1968), and Ross (2005) provide additional useful information on sampling principles and methods.

## SAMPLING WEIGHTS IN LSA

A sampling weight is the inverse of the probability of selection of a unit; it can be loosely interpreted as the number of units in the population that are represented by the selected unit. In the case of LSA, these units are usually students, although they can also be teachers, schools, parents, and so on. We therefore need to use sampling weights when analyzing the data so that each student's contribution to the statistical estimates is proportional to the number of students represented in the population. Using sampling weights helps to adjust the proportional contribution of the elements that make up the total estimate. In most cases, the sampling weights incorporate one or more adjustments for non-participation or post-stratification.

Using sampling weights to calculate a mean is relatively simple. We sum the product of each observed value by its weight, and divide by the sum of the weights. For example, if we have the values 10, 12, and 4, with weights 40, 60, and 80, respectively, the weighted average of these values would be as follows: (10*40+12*60+4*80) / (40+60+80) = 8.0.

The magnitude of the weights, simply calculated as the inverse of the probability of selection, is proportional to a population total. But the key property of the sampling weights is that they capture the relative proportion of each sampled unit with respect to the overall population. As a consequence, the sampling weights can be transformed by multiplying them by a constant, a step that preserves this proportionality. It also simply alters the overall sum of the weights, but not the statistic calculated. If, in the case of the example above, we divide the weights by 10, we get the same result: (10*4+12*6+4*8) / (4+6+8) = 8.0.

A simple example helps illustrate this point. Let's assume we select a simple random sample of 10% of students from a city with 1,800 students. Each of these students is enrolled in one of three educational tracks: Track V, with 1,000 students; Track A, with 500 students; and Track G, with 300 students. We would expect that our sample of 10% will have approximately 100 students from Track V, 50 students from Track A, and 30 from Track G. Because we have selected 10% of students, or 1 out of every 10, the probability of selection of each student is 1/10, and the corresponding weight, or the inverse of this probability, would be 10/1. Therefore, we have in our sample 100 Track V students, each "weighing" or representing 10; 50 Track A students, each weighing or representing 10; and 30 Track G students, each weighing or representing 10. If we add the number that each selected student represents, we obtain 1,800, which is the same as the overall number of students in the population.

In the example above, all selected students, regardless of the track they are in, have the same probability of selection and therefore the same sample weight. But what if we notice that we have only 30 students from Track G, but have 100, or over three times more, from Track V? We could then decide to select 10% of the students from the population, but this time select an equal number of students from within each track. We then proceed to select 60 out of 1,000 students from Track V, 60 out of 500 students from Track A, and 60 out of 300 students from Track G. As is evident in Table 1, the probabilities of selection within each track have changed, and so have the weights that would be assigned to each student. However, the proportional contribution of each group to the overall, when using the weights, has remained constant. The sampling weights thus allow us to adjust the relative contribution of the sampled elements when we need to sample at different rates within different groups.

Table 1: Weights with unequal sampling probabilities, by group

| Description | *N* | Number of selections | Probability of selection (*N* of selections/*N*) | Sample weight (1/probability of selection) | Units represented (number of selections * sample weight) |
|---|---|---|---|---|---|
| Track V | 1,000 | 60 | 0.06 | 16.67 | 1,000 |
| Track A | 500 | 60 | 0.12 | 8.33 | 500 |
| Track G | 300 | 60 | 0.20 | 5.00 | 300 |
| Total | 1,800 | 180 | | | 1,800 |

Table 2 provides another example. In it, we present summary statistics for five educational systems that participated in the Trends in International Mathematics and Science Study (TIMSS) 2007, conducted by the International Association for the Evaluation of Educational Achievement (IEA). We calculated these numbers from the publicly available international database. Even though each of the represented countries is very different in population size, a similar number of students (between 4,117 and 5,726) were selected within each country to participate in the study.

Table 2: Sample sizes in selected countries in TIMSS 2007

| Country | Sample size | Average sample weight | Population estimate |
|---|---|---|---|
| Australia | 4,791 | 53.73 | 257,407 |
| Bahrain | 4,199 | 2.51 | 10,543 |
| Armenia | 5,726 | 9.52 | 54,502 |
| Bulgaria | 4,117 | 21.28 | 87,603 |
| Belgium (Flemish) | 4,970 | 14.21 | 70,637 |

## RESCALING THE WEIGHTS

As we indicated earlier, when we rescale weights, all we are doing is multiplying them by a constant. We do this so that the sum of the weights will be a number that is set according to design and computational needs. For example, some statistics use the sum of the weights as part of the calculations, such as with categorical data analysis statistics. In this case, we want the weights to add up to the sample size so that when we use this value in the calculations, it will be equivalent to the sample size. In other cases, we might want to control the proportional contribution of different groups that contribute to a statistic.

For example, let's say we want to compute an international average for the five countries in the data presented in Table 2 above. We could have each country contribute proportionally to however many students exist in the population, in which case Australia would contribute almost 25 times as much as Bahrain would. Alternatively, we could have each country contribute equally to a summary statistic. But doing this depends on how we define the average. If the average for the population is defined as the total of the scores for all students over the population, regardless of where they come from, then each country should contribute proportionally to however many students exist in the population, in which case Australia would contribute about 25 times as much as Bahrain would. However, if the average for the population of interest is defined as an arithmetic average of the mean scores for the countries, we would need to calculate an international average whereby each country contributes equally.

The equal contribution from each country can be achieved in two ways: by calculating the statistic for each country and then taking the average, or by rescaling the weights within each country to add to a constant and then using these rescaled weights in the calculation. In general, when we calculate so called "international" statistics, the

recommended approach is first to compute the individual country estimates, and then to take the simple average of these. This approach has two main advantages. First, by allowing us to easily calculate the standard error of the international average as the standard error of aggregated means from independent samples, it prevents us from making assumptions about how the replication procedure for calculating the standard error was conducted across the countries. The second, and perhaps most important, advantage is that we do not have the problem of unequal representation that arises when data are missing at different rates within countries for the same variable, or when groups within the population exist in different proportions.

Take, for example, the case where we want to compute the international percent of correct answers for several items, and we want each country to contribute equally. Because, in LSA, each item is taken by different numbers of students within each country, with each set of students being a representative sample of the student population within the country, we need to adjust the contribution of each group to the overall estimate. If we were to use the rescaled weights in this analysis, we would need to rescale them for each of the items to ensure that the pattern of missing data does not affect the desired equal contribution from each of the countries.

One common misconception found among analysts is their belief that rescaling the weights needs to be done only once. For example, the IEA TIMSS and Progress in International Reading Literacy Study (PIRLS) databases include a "senate weight" that adds up to 500 within a country, and the OECD Programme for International Student Assessment (PISA) databases contain a "weight factor" that can be used to transform the weights so that the sum within each country equals 1,000. Using these weights or factors will make the sum of the weights overall for each country equal to a constant. But the sum of the weights for subgroups within each country will vary. So, in the IEA databases, for example, the sum of the senate weights for boys within each country is proportional to the number of boys in the country in comparison to the number of girls and the amount of missing data for the variable for that country.

Rescaling the weights is as simple as multiplying the original sampling weights by a constant that will yield a desirable result based on a design or computational need. This occurs when we rescale weights in order to add to a constant across different groups, as with the senate weights in the IEA databases, or when we transform them to add to a specific number within each country (as is the case for the "house"[1] weights in the IEA databases).

The mechanics of rescaling the weights are as follows:

1. Select the constant to which the sum of the weights will be rescaled (in our example, "K").
2. Compute the sum of the sampling weights within each of the groups within which the rescaling will be done.
3. Multiply the sample weights by the result of dividing the constant selected in Step 1 above by the sum of the weights in Step 2.

1 House weights included in the IEA databases are transformed within each country to add to the overall sample of the country.

## RESCALING THE WEIGHTS USING SPSS

As part of this technical note, we provide the SPSS[2] code that can be used to rescale the weights to a specified constant. The code is available online at www.ierinstitute.org/IERI_TechNote1.zip. It consists of two parts—a macro and a macro call—which we have placed in two separate files. The macro can be called from within any SPSS command syntax and executes a specified set of commands. Figure 1 shows the macro "RescaleWGTS" that we use in the example below.

To execute the macro, we need to specify a set of parameters in the call of the macro. The call of the macro is an SPSS command syntax that contains the necessary parameters for the macro to run. The macro can be executed from within any SPSS syntax window. Therefore, when using the macro RescaleWGTS, we need to specify these parameters:

- INFILE: The name of the file that has the original sampling weights. This file is not overwritten by the program.
- OUTFILE: The name of the file where the new weights will be saved. This file will preserve all the variables in the original file. It will also have a variable with the rescaled weights.
- DIR: The name of the directory where the INFILE and OUTFILE are located.
- CVAR: This lists the classification variables used to group the data. The sum of the weights will add to a constant within each unique combination of the variables defined by these classification variables.
- WGTS: The name of the variable in the original file that has the sampling weight. Here we can specify one or more weights in case we want to rescale several weights simultaneously, as would be the case when rescaling replicate weights.
- NEWWGTS: The name of the variable in the new file that will contain the rescaled sampling weights. There needs to be as many entries for this parameter as were made for the WGTS parameter. Not having these will have unexpected results.
- K: The constant that we want the weights to add up to.

Figure 2 shows an example of how the macro is called. In this example, we are doing the following:

1. Working with files located in "C:\IERI_TechNote1."
2. Reading the data from the file "RescaleFrom.sav."
3. Rescaling the weight variables called "TOTWGT."

These steps lead to:

4. The rescaled weights being saved to variable "R_TOTWGT."
5. The weights adding up to 1,000 for each IDCNTRY by ITSEX combination.
6. The resulting file being saved to "RescaleTo.sav."

2 The code is also available in SAS from www.ierinstitute.org/IERI_TechNote1.zip

We need, at this point, to offer several clarifications and recommendations:

1. Although not strictly necessary, saving the resulting file with a different name from that of the original file is good practice. Not doing this could have unexpected results.
2. The weight variable specified in the WGTS parameter must exist in the original file. One or more weights can be specified.
3. The variable name for the rescaled weights (NEWWGTS) should be different from the original weight variable. Not doing this will overwrite the original weights.
4. The weights should be rescaled to a constant that is "reasonable." While defining reasonable might be difficult, recognizing unreasonable is not. LSA databases tend to provide rescaled weights that add up to a number close to, yet not necessarily the same as, the effective sample size for a country.
5. Checking the results and verifying that the outcome is the desired one is very important. As part of quality control, the macro computes the sum of the weights within each of the groups and presents the results. This is shown in Figure 3, where we notice that the sum of the new variable, called "r_totwgt", equals 1,000 within each IDCNTRY by ITSEX combination, even in the case where ITSEX is coded as a user-defined missing value.

Figure 1: SPSS macro to rescale sampling weights

```
SET Length = None Width = 255
SET format f8.2.

* Rescales the weights within each <cvar> grouping and makes them add to <k>.
* The variable "newwgt" contains the rescaled weight.
* The resulting file is saved to <outfile> and has all the records from the original file.
* All variables in the original file are preserved.

define RescaleWgts
        (dir     = !charend('/') /
         infile  = !charend('/') /
         outfile = !charend('/') /
         cvar    = !charend('/') /
         wgts    = !charend('/') /
         newwgts = !charend('/') /
         k       = !charend('/') ).

set mprint = on.

* Count number of weights to rescale.
!let !nw = !null
!do !w !in(!wgts)
!let !nw = !concat(!nw,"w")
!doend
!let !nwgts = !length(!nw)

get file = !quote(!concat(!dir,"\",!infile,".sav")).
weight off.
sort cases by !cvar.

save outfile = !quote(!concat(!dir,"\",tmp0)).
```

Figure 1: SPSS macro to rescale sampling weights (contd.)

```
aggregate outfile = !quote(!concat(!dir,"\",tmp1))
  / break = !cvar
  / !do !w !in(!wgts) !concat("t",!w) !doend  = sum(!wgts).

match files
 / file = !quote(!concat(!dir,"\",tmp0))
 / table = !quote(!concat(!dir,"\",tmp1))
  / by !cvar.

!let !tmpnew = !newwgts.
!let !tmpold = !wgts.
!let !tmptot = !null.

!do !w !in(!wgts)
!let !tmptot =  !concat(!tmptot," t",!w).
!doend

!do !w = 1 !to !nwgts
compute !head(!tmpnew) = !head(!tmpold) * (!k / !head(!tmptot)).
!let !tmpnew = !tail(!tmpnew).
!let !tmpold = !tail(!tmpold).
!let !tmptot = !tail(!tmptot).
!doend.

execute.

mean tables = !newwgts !do !cv !in(!cvar) by !cv !doend
 / cells = sum min max count
 / missing = include.

save outfile = !quote(!concat(!dir,"\",!outfile,".sav"))
  / drop = !do !w !in(!wgts) !concat("t",!w) !doend.

new file.

erase file=!quote(!concat(!dir,"\",tmp0)).
erase file=!quote(!concat(!dir,"\",tmp1)).

!enddefine.
```

Figure 2: SPSS syntax to rescale the sampling weights

```
* Sample call of the macro to rescale the weights.

include file = "C:\IERI_TechNote1\RescaleWGTS.spm".

RescaleWGTS dir     = C:\IERI_TechNote1
        / infile  = RescaleFrom.sav
        / outfile = RescaleTo.sav
        / cvar    = idcntry itsex
        / wgts    = totwgt
        / newwgts = r_totwgt
        / k       = 1000.
```

Figure 3: SPSS output from rescaling the weights

| *COUNTRY ID* | *SEX OF STUDENTS* | Sum | Minimum | Maximum | *N* |
|---|---|---|---|---|---|
| Australia | GIRL | 1,000.00 | 0.01 | 1.54 | 2,443 |
| | BOY | 1,000.00 | 0.01 | 1.61 | 2,348 |
| | Total | 2,000.00 | 0.01 | 1.61 | 4,791 |
| Bahrain | GIRL | 1,000.00 | 0.19 | 0.71 | 2,025 |
| | BOY | 1,000.00 | 0.19 | 0.77 | 2,174 |
| | Total | 2,000.00 | 0.19 | 0.77 | 4,199 |
| Armenia | GIRL | 1,000.00 | 0.13 | 1.45 | 3,003 |
| | BOY | 1,000.00 | 0.15 | 1.62 | 2,696 |
| | OMITTED | 1,000.00 | 23.30 | 160.76 | 27 |
| | Total | 3,000.00 | 0.13 | 160.76 | 5,726 |
| Bulgaria | GIRL | 1,000.00 | 0.12 | 2.31 | 2,015 |
| | BOY | 1,000.00 | 0.11 | 2.15 | 2,102 |
| | Total | 2,000.00 | 0.11 | 2.31 | 4,117 |
| Belgium (Flemish) | GIRL | 1,000.00 | 0.22 | 1.03 | 2,620 |
| | BOY | 1,000.00 | 0.25 | 1.14 | 2,350 |
| | Total | 2,000.00 | 0.22 | 1.14 | 4,970 |
| Total | GIRL | 5,000.00 | 0.01 | 2.31 | 12,106 |
| | BOY | 5,000.00 | 0.01 | 2.15 | 11,670 |
| | OMITTED | 1,000.00 | 23.30 | 160.76 | 27 |
| | Total | 11,000.00 | 0.01 | 160.76 | 23,803 |

## SELECTING MINI-SAMPLES FROM LSA

Despite recent increases in computing speed and capacity, those of us using LSA databases might still want to work with a smaller subset of the entire database, or select multiple subsets of the larger database. There are many reasons for taking these actions. For example, we might want to conduct preliminary or exploratory analysis with the data without the burden of having to use the entire dataset; or we might want to create multiple samples to validate results obtained using a particular statistical procedure, such as validating a factor analysis solution. In some instances, newer software systems are not very efficient at handling large databases, and for some procedures the facility of using sampling weights might yet not be available. In these cases, the use of mini-samples or subsets from the entire database can be useful.

The procedure described in this technical note amounts to selecting one or more samples from the database, where the resulting subset of cases is statistically equivalent to a simple random sample from the population. As a result of this selection, each record has equal weights, thus eliminating the need to use sampling weights in the calculations. This selection is useful when, as in the previous issue, we want to select samples of equal size from within different groups of the population.

Using a file built in this way effectively makes each group of the population contribute equally to summary statistics. This type of selection is also useful when we want to select multiple samples from a database and then use these to study the variability of estimates, a procedure akin to using bootstrapping techniques.

The results obtained from using these mini-samples are not expected to match exactly those results from the entire dataset, nor are they a substitute. Because LSA data are collected using complex sampling procedures, the process of selecting a subset of records from the database that is still representative of the population is not straightforward.

When selecting a subset, we need to take two matters into account: the probability of selecting the sampled units, and the desired composition of the resulting subset. We can take the first matter into account by sampling the units with probability proportional to their sampling weight. This results in a self-weighted sample in which all the units have equal weight. The sampling weight is the inverse of the probability of selection, and the sampling weights resulting from each stage are multiplicative, meaning the final sampling weight is the product of the weights at the different stages of selection.

This outcome can be easily shown. If we select the units from within the larger sample with probability proportional to their sampling weight, their probability of selection from the subset is then equal to:

$$Probability\ of\ Selection\ into\ subset = \left(\frac{Weight}{\Sigma\ Weights} * Number\ of\ Selections\right)$$

The sampling weight for this stage is the inverse of the probability of selection, or:

$$Weight\ for\ Selection\ into\ subset = \left(\frac{\Sigma\ Weights}{Weight * Number\ of\ Selections}\right)$$

Therefore, the resulting sampling weights, once the records have been selected for inclusion into the mini-sample, equal:

$$Final\ Weight\ for\ Selection\ into\ subset = Weight * \left(\frac{\Sigma\ Weights}{Weight * Number\ of\ Selections}\right)$$

After the redundant terms have been eliminated, the final weight for selection into the subset equals:

$$Final\ Weight\ for\ Selection\ into\ subset = Weight * \left(\frac{\Sigma\ Weights}{Number\ of\ Selections}\right)$$

As we can see, the final sampling weight for the units selected into the subset of cases simply equals the overall sum of the sampling weights divided by the number of selections.

A few points of clarification are necessary at this point:

1. The sum of the weights refers to the sum of the weights for the cases within the stratum or population group from where the selection is made.
2. The number of selections refers to the number of selections made from within the stratum from where the specific case is selected.
3. The weights of the units selected from within each stratum will be equal for each of the selected units.

Addressing the composition of the resulting subset requires a little more explanation. For example, assume we want to select a given number of cases from across the entire database, say a subset of 10,000 students. The selection results in a subset that will include different numbers of students from within each country, depending mostly on the overall size of the population for the country. Alternatively, we might be interested in selecting equal numbers of students from within each country or group (as defined by a set of grouping variables) while preserving the information provided by the sampling weights. Such is the case, for example, when we want to select a subset of students so that each country is represented by equal numbers of cases. Of course, we could address the equal representation of the countries in our analysis by transforming the weights to add up to a constant within each country and then using these transformed weights. But this approach will still leave us with all the cases in the data file and the computing burden unchanged, and it will not provide us with different samples to use in the validation of results.

Selecting representative subsets of students from within each stratum could be the technique to use when, for example, we want to validate data-processing procedures without using the entire database, or when we want to compare the results from two comparable samples of the population to assess the stability of a particular statistical estimate. LSA programs used this procedure during the 1990s in order to study the stability of item parameters at a time when computing capabilities made it prohibitive to use the entire database for calibration purposes.

The purpose of selecting a mini-sample is to have a more manageable set of cases that can be used in preliminary analyses, for didactic purposes, and as other examples. For example, the National Assessment of Educational Progress (NAEP) primer used mini-samples to describe and give examples of the use of NAEP data (Beaton & Gonzalez, 1995). Because individuals wanting to use the complete NAEP database have to have a license to do so, using mini-samples enables individuals without the license to run many graphs and tables. Mini-samples are also used to describe and provide examples of various techniques for estimating population parameters using the NAEP data (e.g., the average NAEP scores or percentages of students exceeding NAEP achievement levels in various demographic groups).

It is important to note that national and international results computed using these mini-samples are expected to be close—but not identical to—published results in the reports. The reason, of course, is that these mini-samples are a random subsample from the full database and therefore subject to sampling fluctuations. National or

international estimates should not be made with these data; nor should they be published as official estimates of the LSA database. Results from these mini-samples should be used for exploratory training purposes only.

The mechanics of selecting a mini-sample are as follows:

1. Specify the number of cases to select within each group (in our example, NSEL).
2. Calculate the sum of the weights within each group (in our example, TMPSWGT, for total measure of size), and find the number of times NSEL fits within TMPSWGT. This calculation will result in the selection interval of our systematic random sampling selection (in our example, INTERVAL).
3. Pick a uniform random number between 0 and 1 (in our example, RNDSTART); this will determine the case to select within each INTERVAL.
4. Compute the boundaries of the measure of size for each case in the data file (in our example, TMPBMOS is the lower boundary for begin-measure-of-size, and TMPCMOS is the upper boundary for cumulative-measure-of-size).
5. The selection then proceeds as follows:
   a. The first element to select is that which contains [RNDSTART * INTERVAL] between TMPBMOS and TMPCMOS.
   b. The next element to select is that which contains [RNDSTART * INTERVAL + (TMPSelNum-1)*INTERVAL].
   c. Depending on the number of selections, sample size, measure of size, and overall weight of the individual case, the same case could be selected more than once.
6. After completing the selection, we save only those records that were selected in Step 5 above. Should a record be selected more than once, we would want to write it as many times as it was selected. We would achieve this with the XSAVE command available in SPSS and would then add to this file a variable that indicates the selection sequence of the case (in our example, SELVEC).

## SELECTING A MINI-SAMPLE USING SPSS

At this point, we want to provide the SPSS code for selecting a mini-sample while taking into account the sampling weights. The code consists of two parts that we have placed in two separate files: a macro and a call to the macro. The code is available online at (www.ierinstitute.org/IERI_TechNote1.zip).

The macro can be called from within any SPSS command syntax and executes a specified set of commands. Figure 4 on page 131 shows the macro "SamplePPS", which we use in our example. To execute the macro, we need to specify a set of parameters in the call of the macro. The call of the macro is an SPSS command syntax that contains the necessary parameters for the macro to run. The macro can be executed from within any SPSS syntax window.

When using the macro "SamplePPS" to select a mini-sample, we need to specify the following parameters:

- INFILE: The name of the file that has all the records. This file is not overwritten by the program.
- OUTFILE: The name of the file where the selected records will be saved. This file will preserve all the variables in the original file, but only those records that were selected into the mini-sample.
- DIR: The name of the directory where the INFILE and OUTFILE are located.
- CVAR: The listing of the classification variables that will be used to group the data. Equal numbers of selections will be made from within each unique combination of these grouping variables.
- WGT: The name of the variable in the original file that has the sampling weight.
- NEWWGT: The name of the variable that has the new sampling weight for the selected case. This is simply the sum of the sampling weights within the group divided by the number of selections. For most applications of the selection of cases, this weight will be discarded, but it is saved in the file as a measure of quality control.
- NSEL: The number of selections to make within each group.
- NSAMPLES: The number of mini-samples to select. Each of these samples is expected to be different from the others.
- SEED: The number used as the seed for the generation of random numbers. A different value for SEED will yield a different selection of cases into the mini-file. Only the same seed used on a file with the same number of records, sorted in the exact same order, will yield the exact same selection of cases. Any other combination will result in a randomly different selection of cases.
- IDVAR: Identification variables in the file that are used to sort the cases in the file prior to selection. If we want to ensure cases are selected from across different groups in the population, we need to specify the variables that identify these groups here. Although this parameter is optional, we highly recommend it. It works as an implicit stratification variable for the selection of cases to ensure cases are selected from a diversity of records.

Figure 5 shows an example of how the macro can be called, with some sample parameters. In this example, as evident in Figure 5, we are doing the following:

1. Working with files located in "C:\IERI_TechNote1."
2. Reading the data that are located in "SamplePPSFrom.sav."
3. Making 500 selections from within each IDCNTRY by ITSEX combination.

These steps lead to these outcomes:

4. The selection is made using the variable TOTWGT as the measure of size.
5. The cases are sorted, for the PPS selection, by the variables IDSCHOOL and IDSTUD.

6. The resulting file is saved to "SamplePPSTo.sav."
7. Ten samples in total are selected.

We again, at this point, need to offer several clarifications and recommendations, some of which are similar to the ones presented earlier:

- Although saving the resulting file with a different name from that of the original file is not strictly necessary, it does prevent us from overwriting the original file.
- The weight variable must exist in the original file.
- Depending on how many selections we choose to make as well as on sample size and magnitude of the sampling weights, it is possible for the same case to be selected more than once into the mini-sample file. In our example, the variable SELVEC in the resulting file contains the number of times a case was selected into the file. Cases selected more than once will be repeated in the output file as many times as necessary.
  - The variable SELSEQ in the output file has a sequential selection number for each selected case. This number will start at 1 and continue sequentially until reaching a maximum of the number of selections per subgroup multiplied by the number of possible unique groups that can be formed with the grouping variables.
  - The resulting file does not contain the weight variable from the original file. Because this weight is no longer useful, we drop it from the file. However, we then include in the file a new weight equal for all cases within each grouping, with the name specified by using the parameter NEWWGT.
  - An important part of the process involves checking the results and verifying that the outcome is the desired one. As part of quality control, the macro computes the sum of the weights and number of cases within each of the groups, and presents the results. This is shown in Figure 6, which presents summary statistics for the first sample selection. Here we can see that the number of cases equals 500 within each IDCNTRY by ITSEX combination. The values for "maximum" indicate the maximum number of times a single case was selected to be in the sample. In the figure shown, a girl in Bulgaria was selected twice, and individuals with omitted values in the ITSEX variable were selected multiple times because there were very few cases in this group.
  - Last, but not least, we cannot emphasize enough that national and international results computed using these mini-samples are expected to be close—but not identical to—published results in the reports or when the complete dataset is used. The reason, of course, is that these mini-samples are a random subsample from the full database and therefore subject to sampling fluctuations. National or international estimates should not be made with these data, and they should not be published as official estimates of the LSA database. Results from these mini-samples should be used for exploratory training purposes only.

Because the resulting mini-samples are equivalent to a simple random sample, there is no need to use complex methods for estimating sampling variance to obtain variances of the estimates.

Figure 4: SPSS macro to select mini-samples

```
SET Length = None Width = 255.
SET format f8.2.

* Selects <nsel> cases PPS from <infile> within each <cvar> grouping using <wgt> as the measure of size.
* It makes <nsamples> selections and saves each to a different file numbered sequentially.
* The PPS selection is done within each group using systematic SRS with the records sorted by <idvar>.
* The variable "selvec" contains the number of times a record is selected (could be > 1).
* The resulting <outfile> has <nsel> records.
* The variable "selseq" contains the selection sequence.
* All variables in the original file are preserved.
* The <seed> is used to initialize the random number generator.
* Different values for <seed> will result in different case selection.

define samplepps
        (dir     = !charend('/') /
         infile  = !charend('/') /
         outfile = !charend('/') /
         idvar   = !charend('/') /
         cvar    = !charend('/') /
         wgt     = !charend('/') /
         newwgt  = !charend('/') /
         nsel    = !charend('/') /
         nsamples= !charend('/') /
         seed    = !charend('/') ).

set seed = !seed.
set mprint = on.

get file = !quote(!concat(!dir,"\",!infile,".sav")).
weight off.
sort cases by !cvar !idvar.

save outfile = !quote(!concat(!dir,"\",tmp0)).

aggregate outfile = *
  / break = !cvar
  / tmpswgt = sum(!wgt).

!do !s = 1 !to !nsamples
compute !concat(rstart,!s) = uniform(1).
!doend

save outfile = !quote(!concat(!dir,"\",tmp1)).

match files
 / file  = !quote(!concat(!dir,"\",tmp0))
 / table = !quote(!concat(!dir,"\",tmp1))
  / by !cvar
   / first = tmpfirst
   / last  = tmplast.

save outfile = !quote(!concat(!dir,"\",tmp2)).

execute.
```

Figure 4: SPSS macro to select mini-samples (contd.)

```
save outfile = !quote(!concat(!dir,"\",tmp2)).

execute.

!do !s = 1 !to !nsamples

get file = !quote(!concat(!dir,"\",tmp2)).

compute interval = tmpswgt / !nsel.
compute selvec = 0.

do if (tmpfirst=1).
compute tmpbmos = 0.
compute tmpcmos = !wgt.
else.
compute tmpbmos = tmpcmos.
compute tmpcmos = tmpbmos + !wgt.
end if.
leave tmpcmos.
execute.

do if (tmpfirst = 1).
compute tmpselnum = !concat(rstart,!s) * interval.
end if.

do if (tmpselnum ge tmpbmos and tmpselnum lt tmpcmos).
compute selvec = 1+trunc((tmpcmos-tmpselnum)/interval).
compute tmpselnum = tmpselnum + selvec * interval.
end if.

leave tmpselnum.
execute.

select if SelVec > 0.
loop #i = 1 to SelVec.
xsave outfile = !quote(!concat(!dir,"\",!outfile,!s,".sav")).
end loop.
execute.

get file = !quote(!concat(!dir,"\",!outfile,!s,".sav")).

compute selseq = selseq + 1.
leave selseq.

mean tables = selvec !do !cv !in(!cvar) by !cv !doend
 / cells = sum min max count
 / missing = include.
compute !newwgt = tmpswgt / !nsel.

save outfile = !quote(!concat(!dir,"\",!outfile,!s,".sav"))
  / drop = tmpfirst tmplast tmpselnum tmpcmos tmpbmos tmpswgt !wgt.

new file.

!doend

erase file=!quote(!concat(!dir,"\",tmp0)).
erase file=!quote(!concat(!dir,"\",tmp1)).
erase file=!quote(!concat(!dir,"\",tmp2)).

!enddefine.
```

Figure 5: SPSS syntax to select mini-samples

```
include file = "C:\IERI_TechNote1\SamplePPS.spm".

samplepps dir     = C:\IERI_TechNote1
      / infile  = SamplePPSFrom
      / outfile = SamplePPSTo
      / idvar   = idschool idstud
      / cvar    = idcntry itsex
      / wgt     = totwgt
      / newwgt  = n_totwgt
      / nsel    = 500
      / nsamples= 10
      / seed    = 72864.
```

Figure 6: SPSS output from selecting mini-samples

| *COUNTRY ID* | *SEX OF STUDENTS* | Sum | Minimum | Maximum | *N* |
|---|---|---|---|---|---|
| Australia | GIRL | 500.00 | 1.00 | 1.00 | 500 |
| | BOY | 500.00 | 1.00 | 1.00 | 500 |
| | Total | 1,000.00 | 1.00 | 1.00 | 1,000 |
| Bahrain | GIRL | 500.00 | 1.00 | 1.00 | 500 |
| | BOY | 500.00 | 1.00 | 1.00 | 500 |
| | Total | 1,000.00 | 1.00 | 1.00 | 1,000 |
| Armenia | GIRL | 500.00 | 1.00 | 1.00 | 500 |
| | BOY | 500.00 | 1.00 | 1.00 | 500 |
| | OMITTED | 13,548.00 | 12.00 | 80.00 | 500 |
| | Total | 14,548.00 | 1.00 | 80.00 | 1,500 |
| Bulgaria | GIRL | 502.00 | 1.00 | 2.00 | 500 |
| | BOY | 500.00 | 1.00 | 1.00 | 500 |
| | Total | 1,002.00 | 1.00 | 2.00 | 1,000 |
| Belgium (Flemish) | GIRL | 500.00 | 1.00 | 1.00 | 500 |
| | BOY | 500.00 | 1.00 | 1.00 | 500 |
| | Total | 1,000.00 | 1.00 | 1.00 | 1,000 |
| Total | GIRL | 2,502.00 | 1.00 | 2.00 | 2,500 |
| | BOY | 2,500.00 | 1.00 | 1.00 | 2,500 |
| | OMITTED | 13,548.00 | 12.00 | 80.00 | 500 |
| | Total | 18,550.00 | 1.00 | 80.00 | 5,500 |

One additional point that we need to address when selecting mini-samples is that of the optimal sample size. There is no single answer to what this should be. In general, the samples selected should be large enough to reach desired effect sizes yet small enough to achieve the computational efficiencies sought. There should also be sufficient variability across the selected samples to obtain an optimal measure of variability of the estimates. In addition, we would recommend avoiding samples where the same records are selected multiple times in high frequencies.

## References

Beaton, A., & Gonzalez, E. (1995). *The NAEP primer*. College Hill, MA: Center for the Study of Testing, Evaluation, and Educational Policy, Boston College.

Kalton, G. (1983). *Introduction to survey sampling* (SAGE university paper series, No. 35). Newbury Park, CA: SAGE Publications.

Kish, L. (1968). *Survey sampling*. New York, NY: Wiley.

Martin, M., Mullis, I., & Kennedy, A. (2007). *PIRLS 2006 technical report*. Chestnut Hill, MA: Boston College.

Olson, J., Martin, M., & Mullis, I. (2008). *TIMSS 2007 technical report*. Chestnut Hill, MA: Boston College.

Organisation for Economic Co-operation and Development (OECD). (2009). *PISA 2006 technical report*. Paris, France: Author.

Ross, K. (2005). *Sample design for educational survey research*. Paris, France: UNESCO.