Paper AA-02-2013

PROC SURVEYSELECT as a Tool for Drawing Random Samples

Taylor Lewis, University of Maryland, College Park, MD

ABSTRACT

This paper illustrates some of the many sampling algorithms built into PROC SURVEYSELECT, particularly those pertinent to complex surveys, such as systematic, probability proportional to size (PPS), stratified, and cluster sampling. The primary objectives of the paper are to provide background on why these techniques are used in practice and to demonstrate their application via syntax examples. Hence, this is not a how-to paper on designing a statistically efficient sample—there are entire textbooks devoted to that subject. One exception is that the paper will discuss a few recently incorporated sample allocation strategies—specifically, proportional, Neyman, and optimal allocation. The paper concludes with a few examples demonstrating how one can use PROC SURVEYSELECT to handle certain frequently-encountered sample design issues such as alternative sampling methods across strata and multi-stage cluster sampling.

INTRODUCTION

The purpose of this paper is to highlight some of the capabilities of PROC SURVEYSELECT in SAS® to facilitate the task of drawing a random sample. Although the various sampling approaches will be introduced with moderate amounts of background and discussion regarding why and how they are used in practice, the intent is to demonstrate the necessary PROC SURVEYSELECT syntax to carry out these techniques. For an accessible introduction into the issues and concepts of survey sampling, see Kalton (1983). The reader seeking a more in-depth treatment of the underlying theory involved in designing an efficient complex survey sample is referred to one of the many excellent texts on the subject, such as Kish (1965), Cochran (1977), Scheaffer, Mendenhall, and Ott (1996), Lohr (1999), and Valliant, Dever, and Kreuter (2013) to name a few.

This paper is structured into three main sections. The first touches on three fundamental random sampling techniques: (1) simple random sampling; (2) systematic sampling; and (3) probability proportional to size sampling. The second section discusses how the STRATA statement can be used to apply these techniques independently within two or more strata defined on the sampling frame. The third section illustrates how the CLUSTER statement can be used to select a random sample of groups of the underlying units of analysis.

To motivate exposition of the various sampling techniques, suppose a market research firm has been hired to evaluate the spending habits of N = 2,000 adults living in a small city. Two example statistics of interest are the average amount of money an adult spent during the previous year on over-the-counter (OTC) medications and the average amount spent on travel outside the city. Assume the data set FRAME is the sample frame containing one record for each of the N = 2000 unique adults in the population and consisting of the following variables:

- ADULTID – a numeric variable ranging from 1 to 2,000 that uniquely identifies each adult.
- BLOCKID – a numeric variable ranging from 1 to 100 that denotes the distinct block on which each adult lives. All C = 100 blocks in FRAME consist of exactly N_c = 20 adults.
- CITYSIDE – a character variable with two possible values—"East" or "West"—that distinguishes which side of a river dissecting the city a given block falls.
- INCOME – an aggregate measure of income for each adult during the most recent year, which we will assume has been obtained from the local taxing authority.

Over the course of the paper, a variety of progressively more complex sample designs will be demonstrated, each with the fixed sample size of n = 400. There is nothing particularly noteworthy about this number, but perhaps we can think of it as the maximum sample size permitted by the market research firm's data collection budget.

FUNDAMENTAL SAMPLING TECHNIQUES

SIMPLE RANDOM SAMPLING

The example syntax below shows how to conduct the most basic (and default) method available in PROC SURVEYSELECT, simple random sampling without replacement (SRSWOR). All of the options utilized appear in the PROC statement. The DATA= option points to the sample frame, while the OUT= statement names the output data set SAMPLE_SRSWOR that will house the resulting sample. The SAMPSIZE= option is used to declare a sample size of n = 400. Assigning a random number with the SEED= option ensures the exact sample will be selected if the PROC SURVEYSELECT syntax is resubmitted at a later time—assuming an equivalent input data set in the same sort order. Although this is technically

optional, it is generally good practice to do so. As the reader will observe, each PROC SURVEYSELECT step demonstrated in this paper specifies a unique seed.

```
proc surveyselect data=frame out=sample_SRSWOR sampsize=400 seed=40029;
run;
```

The real "output" from PROC SURVEYSELECT is the SAMPLE_SRSWOR data set consisting of 400 observations drawn randomly from FRAME, but a brief rundown of what occurred is reported in the listing. For example, the summary generated from the syntax submitted above appears below. For brevity purposes, this is the only occasion output such as this will be included in the paper.

```
                    The SURVEYSELECT Procedure

                Selection Method    Simple Random Sampling

                  Input Data Set                   FRAME
                  Random Number Seed               40029
                  Sample Size                        400
                  Selection Probability              0.2
                  Sampling Weight                      5
                  Output Data Set          SAMPLE_SRSWOR
```

There is a tacit METHOD=SRS option in the PROC statement in the example above. A variety of alternative randomized sampling schemes are available. For instance, to request simple random sampling with replacement (SRSWR), we can specify METHOD=URS. (URS stands for *unrestricted random sampling*.) The following example illustrates the syntax to conduct this method for the same sample size of 400.

```
proc surveyselect data=frame out=sample_SRSWR sampsize=400 seed=22207 method=URS outhits;
run;
```

Aside from specifying METHOD=URS and a new seed, the SRSWR syntax is very similar to that of SRSWOR. One exception is the OUTHITS option appearing in the PROC statement. Whenever a with-replacement design is specified by the user, the default output data set consists of one row for each unique record sampled and a numeric variable called NUMBERHITS indicating how many times the particular record was chosen. There may be occasions when this is preferable, but the OUTHITS option requests a separate record be output for each selection, forcing the number of rows in the output data set to match the sample size. The NUMBERHITS variable is still retained in the OUT= data set, however.

SYSTEMATIC RANDOM SAMPLING

Another widely used approach in applied survey research is *systematic sampling*. The basic idea is to select every k^{th} unit into the sample, where k is typically an integer. This method is particularly utile in scenarios where it would be exorbitantly difficult or altogether impossible to construct a sample frame. Consider a doctor's office that maintains each patient's information in a physical folder sorted alphabetically by surname. If the sampling unit is the patient, it would be much easier to sample, say, every 50^{th} folder than to enumerate all folders, draw a sample, and assemble the list of folders by retrieving them one at a time. Another example is a customer satisfaction survey for a grocery store. It is improbable an exhaustive list of patrons exists, so one rational method of random sampling would be to try to engage every 20^{th} customer leaving the store—of course, it would be smart to also randomly assign the days and time(s) of day during which these attempts are made.

Of course, the technique can also be used when a well-defined sample frame exists, as is the case with the hypothetical expenditure survey. The example below demonstrates the basic syntax for selecting a systematic sample of $n = 400$ adults. Specifying METHOD=SYS in the PROC statement initiates this selection technique. PROC SURVEYSELECT calculates the sampling interval k as N / n, where N is inferred from the number of observations in the input data set. If k is not explicitly an integer, a fractional interval is used such that the exact sample size requested is returned (see the documentation for more details). For ease of exposition, we intentionally allow for an integer interval of $k = 2000 / 400 = 5$.

In essence, PROC SURVEYSELECT begins by randomly choosing a starting point between the 1^{st} and k^{th} observation in the input data set. We might denote this observation r. The sample will consist of the r^{th} observation and the $(r + k)^{th}$, $(r + 2k)^{th}$, $(r + 3k)^{th}$, etc., on down through the end of the data set. For instance, suppose the first adult selected is the 4^{th}. The sample in the output data set SAMPLE_SYS would consist of this individual followed by the 9^{th}, 14^{th}, 19^{th}, …, and 1999^{th}.

```
proc surveyselect data=frame out=sample_SYS sampsize=400 seed=65401 method=SYS;
run;
```

The advantages and disadvantages of systematic sampling are laid out plainly in Chapter 8 of Cochran (1977). One salient disadvantage is that there is no "standard" variance formula adaptation(s) to be applied as there is with stratified or clustered sampling. Cochran illustrates scenarios where the sample could behave like a stratified *or* clustered sample. The former is much more welcomed than the latter. The real danger occurs when there is some periodicity in the sort order of the data set that coincides with the sampling interval. For example, suppose the first stage of sampling involved randomly selecting a set of days within a three-month period for some subsequent data collection process. Suppose further that the days in this span were enumerated sequentially (e.g., Monday, Tuesday, …, Sunday, Monday,…) and that a systematic sample of them was drawn using an interval of $k = 7$. There would be no variation in the sample with regard to the day of the week, which would arguably defeat the fundamental purpose of sampling these time units.

To guard against these unfortunate scenarios, many practitioners advocate sorting the input data set beforehand by one of more *control variables* on the sample frame. This increases the degree of representativeness and fosters a greater likelihood the sample will behave as if it were a stratified one (with respect to the control variables, at least). You can do the sorting manually in a PROC SORT step prior to the PROC SURVEYSELECT, but a somewhat more syntactically efficient alternative is to specify the control variable(s) in the CONTROL statement of PROC SURVEYSELECT. For example, suppose we wanted to use the INCOME variable on sample frame for this purpose; the syntax example below accomplishes this task.

```
proc surveyselect data=frame out=sample_SYS_c sampsize=400 seed=94424 method=SYS;
  control income;
run;
```

Note that the CONTROL statement causes the input data set to be overwritten with its sorted version. To prevent this default action, you can specify something like OUTSORT=data-set-name. (PROC SURVEYSELECT requires the sorted sample frame to be output somewhere.)

PROBABILITY PROPORTIONAL TO SIZE SAMPLING

Another popular sampling algorithm is *probability proportional to size sampling*, or PPS sampling for short. Among its many advantages is the ability to control the sample-to-sample variability of certain common estimators. For instance, if the sample frame contains an auxiliary variable that is strongly correlated with a key outcome variable to be measured in the survey, sampling with probability proportional to size of this variable can dramatically reduce the variability of the estimated total.

The idea is to select each unit on the frame with probability proportional to its share of the total with respect to some auxiliary variable's *measure of size* (MOS). A frequently used example by one of this author's professors is a survey of hospitals aimed at measured the total profit earned during some period. Substantial efficiencies for this estimated total could be achieved by sampling hospitals in proportion to their number of beds. In general terms, if we denote the i^{th} unit's measure of size MOS_i and the aggregate measure for all N units on the sample frame $\sum_{i=1}^{N} MOS_i$, the i^{th} unit's probability of selection would be $n \times \frac{MOS_i}{\sum_{i=1}^{N} MOS_i}$.

Figure 1 is a pie chart portraying the concept of PPS sampling for a sample frame with $N = 8$ units of variable measures of size. We can think of PPS sampling as the event of throwing a dart onto the image, a dart that has an equal probability of landing anywhere inside the circle. Whichever pie slice the dart lands in is selected into the sample. We might expect the slice corresponding to Unit 8 being "hit" more frequently than the slice corresponding to Unit 1.

There is marked difference, both conceptually and formulaically, between without- and with-replacement PPS sampling. We can think of without-replacement sampling as follows. Suppose the first dart fell within the boundary of Unit 4. Prior to the second draw, we would remove that unit's pie slice and redistribute the remaining slices' boundaries such that their respective areas (i.e., probabilities) sum to 1. Subsequent selections would be handled comparably. On the other hand, with-replacement sampling would involve throwing multiple darts independently at the same pie chart. Of course, this could lead to a particular unit being selected more than once, which is probably not desirable.

The other complexity inherent in the without-replacement sampling paradigm is the need to account for the joint selection probabilities into measures of variability—for an example, see equation 3.30 of Valliant et al. (2013). PROC SURVEYSELECT will output these onto the sample data set if you specify the option JTPROBS in the PROC statement, but there is no easy way to account for them in the variance approximations computed by the suite of SURVEY procedures (e.g., PROC SURVEYMEANS). Because of these complexities, PPS sampling is a setting in which the with-replacement assumption is often adopted, even if sampling was actually implemented without replacement.

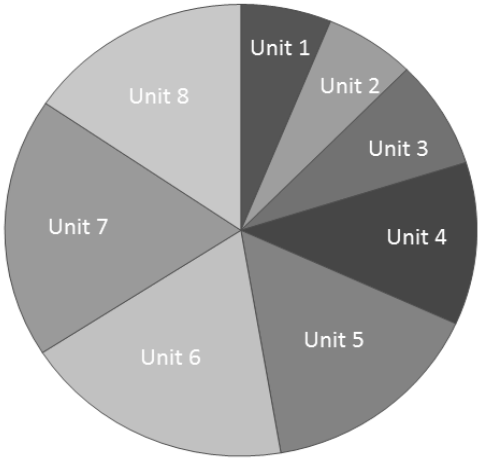

Figure 1. Illustration of Probability Proportional to Size (PPS) Sampling with a Sample Frame of Size *N* = 8

A sensible auxiliary variable for PPS sampling in the hypothetical expenditure survey is the INCOME variable on the sample frame. The reason is that it seems plausible certain expenditures are related to one's income. PPS sampling with respect to this auxiliary variable may improve precision relative to an equal-probability design.

Below is syntax to carry out this method for the same target sample size of *n* = 400 on the FRAME data set. Specifying METHOD=PPS invokes PROC SURVEYSELECT's PPS without-replacement (PPSWOR) sampling algorithm. The method requires the numeric measure of size variable be stated in the SIZE statement.

```
proc surveyselect data=frame out=sample_PPSWOR sampsize=400 seed=32124 method=PPS;
  size income;
run;
```

For complex sampling techniques, PROC SURVEYSELECT will calculate and store selection probabilities and corresponding sample weights (the inverses of these probabilities) in the output data set. Figure 2 below shows how this was done automatically in SAMPLE_PPSWOR. The default labels affixed mask the underlying variable names, which are SELECTIONPROB and SAMPLINGWEIGHT, respectively. We observe that the weights are larger for units with smaller values of INCOME and vice versa, as would be expected under the present design.

	cityside	income	Probability of Selection	Sampling Weight
1	West	10072.4	0.0549563534	18.196258268
2	East	11540.4	0.0629659566	15.881597845
3	East	12258.39	0.0668834054	14.951391804
4	East	12437.51	0.067860708	14.736067893
5	East	12439.57	0.0678719476	14.733627591
6	West	12625.65	0.0688872248	14.516479688
7	East	12634.77	0.0689369848	14.506001437
8	East	12925.01	0.0705205728	14.180259185
9	East	13054.54	0.0712273057	14.039559554
10	West	13358.14	0.0728837876	13.720472444
11	East	13613.08	0.0742747742	13.463521244
12	West	13733.98	0.0749344206	13.345002088
13	East	13767.92	0.0751196018	13.312104644
14	East	13894.9	0.0758124216	13.190450581
15	East	13910.25	0.0758961732	13.175894881
16	East	14028.56	0.0765416883	13.064775841
17	East	14157.06	0.0772428014	12.946190224
18	East	14331.01	0.0781918957	12.789049186
19	East	14405.83	0.0786001236	12.72262631
20	West	14669.7	0.0800398334	12.493779135

Figure 2. First 20 Observations of Output Data Set SAMPLE_PPSWOR from Probability Proportional to Size (PPS) Sampling Example

Be advised that the variables SELECTIONPROB and SAMPLINGWEIGHT are not automatically appended to the output data set for certain basic sample designs such as METHOD=SRS. This may be due to the fact that SRSWOR is a *self-weighting* design, in which certain statistics (e.g., an estimated mean) are equivalent whether or not the weights are used. Whatever the reason, specifying the STATS option in the PROC statement will force these variables onto the output data set.

The PPSWOR algorithm PROC SURVEYSELECT uses requires that no single unit's share of the SIZE statement variable exceed 1 / *n*. If this condition is not met, the procedure will terminate and send an error message to the log. There are several ways around this barrier. One is to use the MAXSIZE=*value* option in the PROC statement. This forces the maximum measure of size for any unit on the sample frame to be adjusted downwardly if it happens to exceed the value specified. So for the example above it could be assigned as $\frac{\sum_{i=1}^{N} MOS_i}{n}$.

Another way to avert termination is to use the CERTSIZE=*value* option. Sometimes a unit is deemed so large or so influential that is selected with certainty. This option can be used to declare such a threshold. Certainty sampling for a portion of the sampling units occurs frequently in area sampling. Doing so is legitimate, they are just assigned as weight of one (and if they happen to be a cluster of units, they are treated as distinct strata). An alternative format is to specify the threshold in proportion terms using syntax such as CERTSIZE=P=*proportion*.

Scanning the documentation, one will quickly observe there are quite a few variations of PPS sampling available, including a with-replacement algorithm (METHOD=PPS_WR), a systematic selection algorithm (METHOD=PPS_SYS), and a few additional methods for PPS sampling of clusters proposed by Brewer (1963), Murthy (1957), and Sampford (1967). It is beyond the scope of this paper to discuss these techniques. Consult the documentation if necessary.

STRATIFIED SAMPLING

The notion behind *stratified sampling* is to partition the sample frame into *H* mutually exclusive and exhaustive groups called *strata* and drawn an independent sample within each. The notation gets somewhat more complicated, since many of the terms require a subscript denoting the particular stratum at hand (e.g., $\sum_{h=1}^{H} N_h = N$), but the same fundamental techniques outlined in the previous section apply. There is no obligation to conduct the same sampling technique within all strata, although implementing different methods in SAS will require multiple runs to PROC SURVEYSELECT, each with different METHOD= options specified. A simple example of this will be demonstrated later.

SIMPLE RANDOM SAMPLING WITHIN STRATA

Suppose we sought to implement a stratified design in which n_1 = 100 adults were chosen from the east side of the city and n_2 = 400 adults from the west. The example syntax below carries out this sample design. The two strata are defined in FRAME by the two distinct values of the CITYSIDE variable, which appears in the STRATA statement. Note that the input data set must first be sorted by the stratification variable(s). The other requirement is to inform PROC SURVEYSELECT of the stratum-specific sample sizes, or n_h's. Although we could use the syntax SAMPSIZE=(100 300) in the PROC statement, the preferred alternative is to create a supplemental data set with a like-named and like-formatted stratification variable and the key variable _NSIZE_. We then point PROC SURVEYSELECT to it using the SAMPSIZE=*data-set-name* option. The parenthetical syntax defining sample sizes is probably sufficient with only a few strata, but seems prone to error when more are involved.

```
proc sort data=frame;
  by cityside;
run;

data sampsizes;
  length cityside $4;
  input cityside  _NSIZE_;
datalines;
East 100
West 300
;
run;

proc surveyselect data=frame out=sample_STR_SRS sampsize=sampsizes seed=89045;
  strata cityside;
run;
```

Recall there are $N_1 = N_2 = 1{,}000$ adults living on either side of the city. Therefore, the probabilities of selection are not identical for adults in either stratum. As with the PPSWOR example, the output data set SAMPLE_STR_SRS is appended with the SELECTIONPROB and SAMPLINGWEIGHT variables reflecting this inequity.

SAMPLE ALLOCATION STRATEGIES

The figures $n_1 = 100$ and $n_2 = 300$ in the example above were chosen arbitrarily to illustrate simple random sampling within strata for an overall sample size of $n = n_1 + n_2 = 400$. It turns out, however, an entire class of allocation strategies has been developed to help researchers decide how best to distribute a fixed sample size n over the H strata defined on the sample frame. These apply mainly to single-stage designs without clustering, which is somewhat limiting in applied survey research. Another downside is that they are univariate in nature. The preferred allocation with respect to one variable may not mesh with another. Nonetheless, several well-known methods are available within PROC SURVEYSELECT and were deemed worthy of illustration via a few simple examples.

PROPORTIONAL ALLOCATION

The first strategy we will consider is *proportional allocation*. The technique is intuitive: sample a given stratum in proportion to its share of the overall population size. Symbolically, this means we assign a sample size for the h^{th} stratum as $n_h = n \times \left(\frac{N_h}{N} \right)$. Since $\sum_{h=1}^{H} \frac{N_h}{N} = 1$, it follows that $\sum_{h=1}^{H} n_h = n$. Proportional allocation has the effect of making the sample a miniature replica of the population. Another simplifying aspect is that, since the sampling rates are more or less identical across all strata, or $f_h = \frac{n_h}{N_h} = \frac{n}{N} \approx f$, it results in a more or less self-weighting design. At first thought, one might wonder how this technique is superior to simple random sampling, but Lohr (1999, pp. 105 – 106) asserts that estimated variances under this stratified design are generally more precise than those under a simple random sample design of the same size.

The example below demonstrates PROC SURVEYSELECT syntax to conduct proportional allocation. We specify $n = 400$ using the SAMPSIZE= option, and request proportional allocation to the $H = 2$ strata defined by the CITYSIDE variable by specifying ALLOC=PROP after the slash in the STRATA statement. By default, the procedure will calculate the appropriate sample sizes and execute the sample selection all at once, appending the pertinent stratum-specific summary variables to the data set named in the OUT= option of the PROC statement. This author's preference is to use the NOSAMPLE option to store only the summary information in the OUT= data set. This intermediate step allows one to examine the calculated sample sizes to ensure they are sensible before proceeding. As an example, there may be occasions when PROC SURVEYSELECT returns $n_h = 1$. This could present a dilemma during the analysis stage, because the default variance estimation method used by the SURVEY procedures requires the number of units sampled from each stratum to be greater than or equal to 2. (The option ALLOCMIN= after the slash in the STRATA statement, however, can be used to specify a minimum stratum sample size.) Assuming the sample sizes are deemed appropriate, the summary data set—or information gleaned from it—can be used in a subsequent PROC SURVEYSELECT step to actually draw the sample.

```
proc sort data=frame;
  by cityside;
run;

proc surveyselect data=frame out=sampsizes_prop sampsize=400;
  strata cityside / alloc=prop nosample;
run;
```

Figure 3 shows what the SAMPSIZES_PROP data set looks like. There is one row for each stratum and the following variables:

- CITYSIDE – the stratum identification variable
- TOTAL – the stratum population size N_h derived from the data set FRAME
- ALLOCPROP – target allocation proportion
- SAMPLESIZE – actual sample size allocated to the stratum n_h
- ACTUALPROP – actual allocation proportion, n_h / n

The reason both ALLOCPROP and ACTUALPROP appear is that the target stratum sample size calculated as $n^*(N_h / N)$, may not be an integer, yet the sample size clearly must. If necessary, PROC SURVEYSELECT uses a rounding routine to convert the target sample size to an integer, and these two columns differentiate the target from the actual sampling proportion for the given stratum. A similar argument applies to other allocation strategies available in PROC SURVEYSELECT, as we will see shortly.

	cityside	Total Number of Sampling Units	Allocation Proportion	Sample Size	Actual Proportion of Total Sample Size
1	East	1000	0.5	200	0.5
2	West	1000	0.5	200	0.5

Figure 3. View of Output Data Set SAMPSIZES_PROP from the Proportional Sample Allocation

This was perhaps not the most interesting example, since with N_1 = 1000 and N_2 = 1000, one may have quickly gathered the sample size would be split evenly across the two strata. Next, we consider somewhat more complex criteria for sample allocation.

NEYMAN ALLOCATION

Proportional allocation is a sound strategy when the element variances are commensurate across strata. When they exhibit substantial variation, *Neyman allocation* is a preferred alternative. To be truly optimal, the approach requires knowledge of the element variances for all population units in the stratum, the σ_h^2's, which we should immediately recognize is conceptually different from the estimated element variances from a given sample, the s_h^2's. Of course, in the absence of the true population parameters, the sample-based estimates from a prior survey or pilot study could serve as a resourceful substitute. These are not always available, however, in which case a compromise might be to substitute relative values (see discussion below).

Cochran (1977, p. 98) notes that the variance of a sample mean in a stratified random sample can be minimized for a fixed sample size *n* if stratum sample sizes are defined as

$$n_h = n \times \frac{N_h \sigma_h}{\sum_{h=1}^{H} N_h \sigma_h} \qquad (1)$$

Careful examination of (1) reveals how stratum sample sizes are dependent not only on the population size, but also variability. This allocation strategy can be conducted in PROC SURVEYSELECT by specifying ALLOC=NEYMAN after the slash in the STRATA statement.

Imagine the results of a comparable expenditure survey conducted several years ago were available and it was determined that adults living on the west side of the city exhibit far more variance with respect to travel expenditures than those living on the east. Specifically, the west side's variance was estimated at \$3,000, whereas the east side's was \$250. The example below requests Neyman allocation amongst the two strata for a total sample size of *n* = 400. Much of the syntax follows what was shown for proportional allocation, but there is an added requirement to inform PROC SURVEYSELECT of the stratum-specific element variances. (Note that you must specify the element variances, not the element standard deviations that appear in the equation above.) This is accomplished by pointing PROC SURVEYSELECT to a supplemental data set STRATUM_VARS using the VAR= option after the slash in the STRATA statement. The key variable SAS will be looking for is _VAR_.

```
proc sort data=frame;
  by cityside;
run;

* provide SURVEYSELECT a supplemental data set containing stratum-specific element vari-
ances;
data stratum_vars;
  length cityside $4;
  input cityside _VAR_;
datalines;
East 250
West 3000
;
run;

proc surveyselect data=frame out=sampsizes_neyman sampsize=400;
  strata cityside / alloc=neyman var=stratum_vars nosample;
run;
```

Figure 4 displays the summary data set SAMPSIZES_NEYMAN. Its structure parallels the SAMPSIZES_PROP data set from the previous section, except the allocation is now different. We can observe the target sample size is much larger for the west side (310) than the east side (90), a reflection of the former's much greater variability.

	cityside	Total Number of Sampling Units	Variance	Allocation Proportion	Sample Size	Actual Proportion of Total Sample Size
1	East	1000	250	0.2240092377	90	0.225
2	West	1000	3000	0.7759907623	310	0.775

Figure 4. View of Output Data Set SAMPSIZES_ NEYMAN from the Neyman Sample Allocation

It was mentioned earlier how relative variances could be substituted for actual variances. For example, resubmitting the code above with a supplemental data set consisting of _VAR_ = 1 for the "East" stratum and _VAR_ = 12 for the "West" stratum would yield the same allocation (since \$250 represents 1/12 of \$3,000). Developing acceptable relative element variances could prove easier than estimating the stratum-specific variances from an external source.

OPTIMAL ALLOCATION

The third and final allocation strategy we will demonstrate is *optimal allocation*. This approach expands upon Neyman allocation by allowing for stratum-specific unit costs of data collection to be incorporated. In fact, Neyman allocation can be perceived as the special case of optimal allocation in which the per-unit data collection costs are equal across all strata. Optimal allocation is the result of minimizing the expected variance for the overall mean of a stratified random sample when the following simple data collection cost model applies (see p. 107 of Lohr, 1999):

$$C = C_0 + \sum_{h=1}^{H} c_h n_h \qquad (2)$$

In (2), the total cost C is assumed to be the sum of certain sunk costs C_0 (e.g., overhead associated with the survey effort) and n_h stratum-specific unit costs c_h. Under this model and known stratum-specific element variances, the optimal allocation is

$$n_h = n \times \frac{N_h \sigma_h / \sqrt{c_h}}{\sum_{h=1}^{H} N_h \sigma_h / \sqrt{c_h}} \qquad (3)$$

Relative to equation (1), the only new terms are those pertaining to costs. Since they appear in the denominator, we can reason that, all else equal, higher costs drive down the sample size allocated to a particular stratum.

To see how this might influence the allocation, let us return to the hypothetical expenditure survey. Suppose it was estimated that the cost of contacting and surveying adults living on the west side of the city was \$100 per completed interview, whereas adults on the east side could be surveyed for a cost of \$25. The example below shows the syntax necessary to perform optimal allocation. On top of stratum-specific element variances, we must also inform PROC SURVEYSELECT of the unit costs. This is accomplished by populating a key variable named _COST_ in the supplemental data set STRATUM_VARS_COSTS. We point to this data set using the COST=*data-set-name* option after the ALLOC=OPTIMAL option in the STRATA statement. Note that the VAR option (without an equals sign) informs PROC SURVEYSELECT that the variable _VAR_ also resides on the data set containing _COST_.

```
proc sort data=frame;
  by cityside;
run;

 * provide SURVEYSELECT a supplemental data set containing stratum-specific element vari-
ances and unit costs;
data stratum_vars_costs;
  length cityside $4;
  input cityside _VAR_ _COST_;
datalines;
East 250  25
West 3000 100
;
run;

proc surveyselect data=frame out=sampsizes_optimal sampsize=400;
  strata cityside / alloc=optimal cost=stratum_vars_costs var nosample;
```

```
run;
```

Figure 5 shows how the summary data set SAMPSIZES_OPTIMAL reflects a different mix of sample sizes relative to Neyman allocation. Accounting for the cost structure has boosted the sample size for adults in the east side of the city at the expense of the sample size for adults in the west. This is an intuitive result considering the cost of data collection is four times as expensive for the west side adults.

	cityside	Total Number of Sampling Units	Variance	Cost	Allocation Proportion	Sample Size	Actual Proportion of Total Sample Size
1	East	1000	250	25	0.3660254038	146	0.365
2	West	1000	3000	100	0.6339745962	254	0.635

Figure 5. View of Output Data Set SAMPSIZES_ OPTIMAL from the Optimal Sample Allocation

VARIABLE METHODS OF SAMPLING ACROSS STRATA

We conclude the section on stratified sampling with a quick example illustrating how one might operationalize different sampling methods for different strata. There is no way to do this with one pass of PROC SURVEYSELECT, since the METHOD= option in the PROC statement applies to all strata identified on the input data set, but because stratification involves independent sampling across strata boundaries, the sample frame can be partitioned as needed.

For sake of an example, suppose the research firm was particularly concerned about the representativeness of the sample of adults drawn from the west side of the city and it was argued the best way to ensure this was systematically selecting a sample of 200 from a list presorted on income. Acknowledging the downsides of systematic sampling and the hunch that variation was far less amongst adults living on the east side, the firm arrived at a compromise to select a simple random sample of 200 adults from this other stratum.

The syntax to conduct this sample design is demonstrated below. There is nothing new about the PROC SURVEYSELECT syntax given, only that WHERE= data set options are used to subset the sample frame on the fly for the two stratum-specific runs. The STATS option in the second run forces the SAMPLINGWEIGHT variable to be appended to the output data set, which does not occur by default with the implied METHOD=SRS. Since all cases meeting the WHERE= condition emanate from a single stratum, there is no need for a STRATA statement. The DATA step at the end concatenates the two stratum-specific samples into one data set.

```
proc surveyselect data=frame (where=(cityside='West')) out=sample_STR_part1 sampsize=200
seed=63215 method=SYS;
  control income;
run;

proc surveyselect data=frame (where=(cityside='East')) out=sample_STR_part2 sampsize=200
seed=72005 stats;
run;

data sample_STR_all;
  set sample_STR_part1
      sample_STR_part2;
run;
```

CLUSTER SAMPLING

In applied survey research, the primary sampling units (PSUs) can sometimes be clusters of the ultimate units of analysis. This may occur naturally or purposively to facilitate the data collection process. Examples demonstrated thus far in this paper have assumed the adult is the PSU. Another choice for PSUs are the C = 100 city blocks identifiable on the data set FRAME by the variable BLOCKID. This might be operationally more convenient if the expenditure survey were to be conducted in-person by interviewers. Examples presented in this section adhere to this paradigm.

SINGLE-STAGE CLUSTER SAMPLING

Recall that precisely N_c = 20 adults live on each block. Alternatively, a sample size of 400 could be realized by sampling all adults living in a sample of c = 20 blocks. The CLUSTER statement in PROC SURVEYSELECT can be used for samples of this nature in which PSUs are comprised of multiple observations on the input data set.

The example below demonstrates syntax to select a sample of 20 blocks from the FRAME data set. Note that in combination with a CLUSTER statement, the value specified in the SAMPSIZE= option refers to the number of clusters to be chosen, not observations. Hence, the output data set SAMPLE_CLUSTER still consists of 400 observations: n_c = 20 adults from each of c = 20 sampled blocks.

```
proc surveyselect data=frame out=sample_cluster sampsize=20 seed=82908;
  cluster blockID;
run;
```

PPS sampling is frequently used in cluster sampling, and can be implemented in various ways using PROC SURVEYSELECT. To illustrate, the next example welds concepts of prior examples by conducting a PPSWOR sample of *c* = 20 blocks using the FRAME variable INCOME as the measure of size. When a SIZE statement appears alongside a CLUSTER statement, PROC SURVEYSELECT computes a PSU-level aggregate measure of size equaling the sum of the SIZE variable for all units in the PSU. This aggregate measure is then utilized in whichever PPS algorithm is specified in the METHOD= option of the PROC statement. It is also appended to the output data set along with the familiar SELECTIONPROB and SAMPLINGWEIGHT variables.

```
proc surveyselect data=frame out=sample_PPS_cluster sampsize=20 seed=49721 method=PPS;
  cluster blockID;
  size income;
run;
```

PPS sampling is sometimes executed with the measure of size being the number of underlying units in the PSU. This is particularly common in the context of area sampling, where geographical units might be selected in proportion to the population of individuals therein. If the sample frame is oriented such that there is one observation for each underlying unit in the respective PSUs—that is, there is a complete enumeration of the ultimate sampling units with the PSU—there is an alternative form of PROC SURVEYSELECT syntax that can be used. Specifically, one would eliminate the SIZE statement and use the PPS option after the slash in the CLUSTER statement. Again, this will result in a PPS sample of clusters where the measure of size is the number of underlying observations in the input data. In our hypothetical survey with equally-sized clusters, the method would more or less reduce to simple random sampling of clusters, so no syntax example is given.

STRATIFIED CLUSTER SAMPLING

Clusters can also be stratified prior to sample selection. If done wisely, this could counteract some of the precision loss anticipated from sampling blocks in lieu of adults directly. Given the expected greater variability of expenditures for adults living on the west side, suppose the market research firm opted to oversample blocks in that stratum at a rate of 3 to 1 relative to the east side stratum. The syntax shown below selects a simple random sample according to these specifications. Recall that in cluster sampling the supplemental data set should house PSU sample sizes for the particular stratum, so the ultimate number of sampled adults output to SAMPLE_STR_CLUSTER is still 400.

```
proc sort data=frame;
  by cityside;
run;

data sampsizes_clusters;
  length cityside $4;
  input cityside  _NSIZE_;
datalines;
East 5
West 15
;
run;

proc surveyselect data=frame out=sample_str_cluster sampsize=sampsizes_clusters
seed=77472;
  strata cityside;
  cluster blockID;
run;
```

MULTI-STAGE CLUSTER SAMPLING

Cluster sampling in practice rarely concludes after a single step. As an example, an education survey's sample design might begin by selecting schools in the first stage (i.e., schools are the PSUs), and classrooms in the second stage (i.e., classrooms are the SSUs). That is, within each sampled school, a sample of classrooms is drawn. This is an example of a *multi-stage cluster sample*. These types of designs are perfectly legitimate, but we must keep track of the associated selection probabilities at each step because the ultimate weight assigned should be inversely proportional to the product of them. With a little care, a sequence of PROC SURVEYSELECT runs can be employed to carry out these types of designs. A simple example involving the hypothetical expenditure survey data set is given next to demonstrate some of the basic ideas.

Suppose that instead of surveying all 20 adults living in each block of the stratified sample, the market research firm decided to double the number of sampled blocks within each stratum, but survey only one-half of the adults therein. Even with the

same number of adults interviewed, this design would probably improve precision since the number of PSUs has increased; however, if cluster sampling was chosen to limit on-site interviewer expenses, this alternative design it is likely associated with an increased data collection cost.

The example below illustrates syntax to select this two-stage sample. As with any stratified design, we must first sort the sample frame by the stratification variable(s). A new supplemental data set SAMPSIZES_CLUSTERS2 informs PROC SURVEYSELECT of the PSU sample sizes. The first step is to select the sample of blocks. This is output to a data set SAMPLE_PRELIM. The variable renaming is done to distinguish the sampling probabilities and weights between this and the second stage, which uses comparable suffixing nomenclature.

The data set SAMPLE_PRELIM consists of all unique adults within the sampled blocks. In the second sampling step, we randomly select one-half of them. Note that we no longer specify a CLUSTER statement in the second SURVEYSELECT run—if we did, we would be sampling whole clusters from the set of clusters already sampled. Instead, we add the BLOCKID variable to the STRATA statement. No separate PROC SORT is needed at present because SAMPLE_PRELIM is already sorted by CITYSIDE and BLOCKID from the first PROC SURVEYSELECT step. Note that an alternative method for defining the sample size is to specify a sampling rate using the RATE=*value* option. Since the (sub)sampling rate of adults is equivalent for all sampled blocks, providing one number is sufficient in this example. A little more work would be needed if the rates or sample sizes in the second stage were to vary, but a supplemental data set could be created and supplied via the SAMPRATE=*data-set-name* option or the SAMPSIZE=*data-set-name* option.

The DATA step after the second PROC SURVEYSELECT run computes the overall selection probabilities and sampling weights for the adults who were eventually chosen and output to the data set SAMPLE_MULTISTAGE. For adults in the east stratum, the weights turn out to be 10; for adults in the west stratum, the weights are 3.33. These make sense, as the ultimate sample size for the east is 100 and the west 300, samples that need to be weighted up to represent the respective stratum population counts of 1,000. It is always good practice to verify the weights sum to sensible totals. Though not shown, the output from PROC MEANS confirms they were calculated correctly.

```
proc sort data=frame;
  by cityside;
run;

data sampsizes_clusters2;
  length cityside $4;
  input cityside  _NSIZE_;
datalines;
East 10
West 30
;
run;

* Step 1) selection of PSUs (blocks);
proc surveyselect data=frame seed=78410 sampsize=sampsizes_clusters2
  out=sample_prelim (rename=(SelectionProb=SelectProb1 SamplingWeight=SamplingWeight1));
  strata cityside;
  cluster blockID;
run;

* Step 2) selection of SSUs (adults within blocks);
proc surveyselect data=sample_prelim seed=64107 rate=.5
  out=sample_multistage (rename=(SelectionProb=SelectProb2 Sam-
plingWeight=SamplingWeight2));
  strata cityside blockID;
run;

* assign variables SelectionProb and SamplingWeight to be the product of the corresponding
  variables output from the two SURVEYSELECT runs above;
data sample_multistage;
  set sample_multistage;
SelectionProb=SelectProb1*SelectProb2;
SamplingWeight=SamplingWeight1*SamplingWeight2;
run;

* verify the weights sum to known totals from the sample frame;
proc means data=sample_multistage nway n sum;
  class cityside;
```

```
    var SamplingWeight;
  run;
```

CONCLUSION

The goal of this paper was to introduce some of the common sample selection techniques employed in applied survey research and illustrate rudimentary PROC SURVEYSELECT syntax examples implementing them. In addition to discussing a few fundamental sampling approaches, we explored applications of stratified and cluster sampling. Although no specific examples were shown, it should be acknowledged that proper inferences from these samples requires the complex survey features (weights, stratification, and clustering) be accounted for. One built-in way to do this is to utilize one of the SAS/STAT® procedures prefixed SURVEY. For instance, PROC SURVEYMEANS is the cousin procedure of PROC MEANS that can be used when the input data set is derived from a complex survey sample design. See the documentation and/or proceedings papers such as Lewis (2010) for more details.

REFERENCES

Brewer, K. (1963). "A Model of Systematic Sampling with Unequal Probabilities," *Australian Journal of Statistics*, **5**, pp. 93–105.

Cochran, W. (1977). *Sampling Techniques*. *Third Edition*. New York: Wiley.

Kalton, G. (1983). *Introduction to Survey Sampling*. Sage University Paper series on Quantitative Applications in the Social Sciences, 07-035. Newbury Park, CA: Sage.

Kish, L. (1965). *Survey Sampling*. New York, NY: Wiley.

Lewis, T. (2010). "Principles of Proper Inferences from Complex Survey Data," *Paper presented at the SAS Global Forum*. Seattle, WA, April 11 – 14. Available on-line at: http://support.sas.com/resources/papers/proceedings10/266-2010.pdf.

Lohr, S. (1999). *Sampling: Design and Analysis*. Pacific Grove, CA: Duxbury Press.

Murthy, M. (1957). "Ordered and Unordered Estimators in Sampling without Replacement," *Sankhya*, **18**, pp. 379–390.

Sampford, M. (1967). "On Sampling without Replacement with Unequal Probabilities of Selection," *Biometrika*, **54**, pp. 499–513.

Scheaffer, R., Mendenhall, W., and Ott, R. (1996). *Elementary Survey Sampling*. *Fifth Edition*. Belmont, CA: Duxbury Press.

Valliant, R., Dever, J., and Kreuter, F. (2013). *Practical Tools for Designing and Weighting Survey Samples*. New York, NY: Springer.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Taylor Lewis
Joint Program in Survey Methodology (JPSM)
1218 LeFrak Hall
University of Maryland
College Park, MD 20742
Email: tlewis@survey.umd.edu