Paper 262-28

Complex Sampling Designs Meet the Flaming Turkey of Glory

David L. Cassell, Design Pathways
AnnMaria Rousey, Spirit Lake Consulting

ABSTRACT

Sophisticated sampling designs can save money, provide more accurate answers, and answer questions that simple random sampling (or non-random data) cannot. Yet too many investigators are guilty of a multitude of sins. These include not collecting data optimally, not analyzing data according to the underlying features of the sampling designs or (horrors!) all of the above. Survey data are the lifeblood of marketing analyses, healthcare studies, environmental and sociological research, and many other areas. It is essential that data be collected and analyzed in the best possible ways. This talk will focus on key aspects of sampling theory, such as cluster sampling, weighted samples, stratified sampling, sampling errors, and non-response errors. This will be combined with a discussion of the SAS® PROCs which help one design surveys, and then analyze those surveys: PROC SURVEYSELECT, PROC SURVEYMEANS, and PROC SURVEYREG, along with the %SMSUB macro and some future V9 enhancements. Attendees will also learn the hazards of using PROC MEANS when PROC SURVEYMEANS was warranted, which, despite the title, do not include bursting into flame. Whether or not such punishment is warranted will also be discussed, with examples.

PROLOGUE

Once again, we have seen that fiction is no stranger than truth. The children's cartoon "!Mucha Lucha!" thoughtfully illustrated this recently. This show is set in an imaginary world in which everyone and everything is a masked Mexican wrestler. The main characters of the show are Ricochet, Buena Girl, and The Flea: three ordinary masked-wrestler schoolchildren who go to a masked-wrestler school replete with masked-wrestler teachers. A scene in this show accurately reflected the complexities of consulting on sampling designs. The three children prepare to perform their signature moves to combat a bad guy...

Buena Girl: I choose.. the Torch of Truth!
< she transforms into a living torch, complete with flame>
Ricochet: I choose.. Mirror Mirror!
<he transforms into a living mirror>
The Flea: [in an odd Peter Lorre-like voice] The Flea chooses.. the Flaming Turkey of Glory!
<he transforms into a plucked, headless, cooked turkey which is on fire>
<his teammates promptly turn and stare at him>
The Flea: What?!? The Flea LIKES the Flaming Turkey of Glory!

This is hardly different from the real world...

AnnMaria: I would choose a sampling design which will reveal accurate estimates for a minimal cost.
David: I would choose a sampling design which will reflect the complexities of the target population.
The Client: [in an odd Peter Lorre-like voice] The Client chooses.. stratified sampling!
<pause>
The Client: What?!? The Client LIKES the stratified sampling!

Note that the client did not actually transform into a flaming turkey. He only seemed to, in our imaginations. And only because this would be a situation where stratified sampling was inappropriate or sub-optimal. But this is one of many statements that have been made by close relatives of The Flea. For example, many of you have heard clients or end-users say [in an odd Peter Lorre-like voice] something like one of the following statements:

What?!? The Analyst LIKES using PROC MEANS!

What?!? The Client LIKES using data from his web survey!

What?!? The End-user LIKES assuming simple random sampling!

WHAT IS A PROBABILITY SAMPLE?

Too often, a client or end-user will have a data set ready for analysis, and then run into the annoying roadblock called The Statistician. (Of course, we all know that in reality the consultee only thinks his data set is ready, and the data will really require hundreds of person-hours to do all the quality assurance needed to make it usable. But that's another talk.) So first of all, we need to explain what constitutes a legit sample.. and what does not. In general, when a complete census is logistically or budgetarily unreasonable, the user will want a subset of the possible population values, so that realistic estimates can be made. But how do we decide what a 'subset' is? Or a 'population'? Or an 'estimate'? Or even 'realistic'? First, we must define terms.

A sample design should be more than just a collection of information that hopefully comes from the target population. It should be a functional process which embodies the entire process of sample definition, sample selection, sample collection, data preparation, data analysis, and data use.

Cochran (1977) specified eleven steps in designing, performing and evaluating a survey. These steps, as he

listed them, are:

(1) specifying the objectives of the survey;
(2) determining the population to be sampled;
(3) deciding the data to be collected;
(4) determining the degree of precision to be required;
(5) determining the methods of measurement;
(6) developing the sampling frame;
(7) selecting of the sample;
(8) performing the pretest;
(9) organizing of the field work;
(10) summarizing and analyzing the data; and
(11) gathering information for future surveys.

A 'population' in this context will be the target group or groups to be studied. The population cannot be properly defined until the objectives of the study are fully defined, since changing the objectives can alter the requirements on the population to be studied. If the survey objectives are not specified, then determining which data must be collected is a hit-or-miss proposition. This is often addressed by attempting to gather every bit of available information, in the hope that whatever will be needed is in the final database. Doing a sample this way tends to have painful effects on budgets and data management. On the other hand, having people like this as clients is another way to ensure that SAS® will never go out of style.

Often, the goal of such a study is to estimate certain properties or characteristics of this population, which we will call population values. The true population values will be unknown, and may not even be fixed constants. We will (in principle) use the data from the sample to try to come up with 'sample estimates'. But the accuracy of the sample estimates is dependent on the size and scope of the survey design, as well as the kind of survey design used. The survey has to be designed carefully enough that we know exactly what the 'sampling units' are. The sampling units are the elements of the target population. But the traditional survey sample of people need not have a person as the sampling unit. Lots of surveys of people are really only looking at the household as the sampling unit. Such surveys then have to address such complexities as households with more than one family, families spread across more than one household, people who spend time in more than one household, people who have no household, etc. Until the sampling unit is fully defined and these sorts of questions are addressed, we are not really ready to build a sample, much less collect data.

Once these questions have been answered, then the 'sampling frame' can be constructed. A sampling frame is (typically) an enumeration of all the possible sampling units in the target population. This may be a list of people, a list of households, a list of hospitals, a list of patients, a list of lakes, a list of farms, a list of businesses, or any other list which gives us a list which can be used for sample selection. In cases where a complete enumeration is not possible, then the sampling frame has to be built to take this into account. Unfortunately, despite our best intentions (as George Bernard Shaw would say), we sometimes find ourselves on a road which has no reliable list of all sampling units, and no cost-effective way of obtaining one. Such a problem has been addressed in survey sampling theory, and the simplest approach to this problem is often cluster sampling. Under a cluster sampling design, we may end up taking blocks (i.e, clusters) of the population and building complete sampling frames for the clusters only. Cluster sampling is not the only option at this point, just the one that more people have heard of.

Note that only after the first six steps above are we ready to select the sample. This is the point at which PROC SURVEYSELECT is used. While many people think about experimental designs as a model for what we want to do in sampling designs, experimental designs actually have some major differences from sampling designs. A primary difference is that experimental designs are usually built under a structure that incorporates a finite sample from a theoretically infinite sequence of observations. Survey sampling classically has a finite population, and has necessary adjustments in error variances to compensate.

Step nine - organizing of the field work - has no statistical analysis and no statistical computations. But it is a crucial component of the sampling process. Response errors - errors which occur in the values collected - can vastly inflate the standard errors of the sample estimates, and can even introduce subtle but crucial biases in the point estimates. Different interviewers asking the same questions may get completely different answers.

Equally important are non-response errors, the cases where no values are obtained for some variable or variables. Those same interviewers may make the difference in getting people to answer a question. The design of the questionnaire may determine whether there is a reasonable chance that *any* interviewer can get the subject to answer a sensitive question. Different interviewers may get very different rates of refusal, so that one interviewer may have evaluations on nearly all of her units, while another interviewer may get less than half of his evaluations.

It is only at step ten that we are ready to use PROC SURVEYMEANS or PROC SURVEYREG to analyze the data. The unfortunate fact is that the client may envision the entire process of the survey as only drawing a sample and then analyzing it. Or in some cases, only performing the analysis on already-collected data.

WHAT IS NOT A PROBABILITY SAMPLE?

When the client comes forward and says that she has a terrific data set and it all comes from a really nice web survey her people constructed, the first thing you have to do is.. not laugh. It isn't clear to most people what constitutes a probability sample and what does not. So let's focus on a few common problems. In the theory of sample surveys, we assume that each unit in the population has a set value for each character Y, and that we got the correct value each time we collected data. We also assume that we know how 'likely' it is that a sampling unit was selected, and how likely it is that two different sampling units both ended up in our sample. These aspects are typically driven by our sampling design. When we cannot know the correct value for each collected Y, or we cannot determine the real probability of a sampling unit being included, then we are going to have serious problems in our analysis.

The advent of the Internet has made it possible for anyone to collect survey data, just by putting up a webpage. But does this constitute a survey design? Let's look at an example and see where things can go wrong. A few years ago, Major League Baseball added to the omnipresent confusion of fan voting for the All-Star game by adding on-line voting. In order for a fan to vote, he or she had to fill out a short form, then select the players of interest and submit the vote. On the last day of on-line voting, Nomar Garciaparra of the Boston Red Sox trailed Derek Jeter of the New York Yankees by roughly twenty thousand votes for American League shortstop. A Boston Red Sox fan who is a well-known Perl programmer and author decided to address that gap. He wrote a small Perl program which went to the webform, filled it out, and then voted for Nomar Garciaparra.. twenty-five thousand times.. in just a few hours. He didn't make any attempt to disguise his ballot-stuffing, so his votes were thrown out after the fact. But he could have forged his email address and masked his origins with ten or twenty more lines of Perl code. So, does this balloting yield a sample survey design?

Of course not. Even if we define some manner of proxy for the target population, say, all people on earth who might vote for their favorite baseball player at any position, we still have no way of determining how likely it is that a particular fan may vote. Or may vote as often as he can scare up a new email address on America Online. This type of data collection is prone to the problem of self-selection. Any time people are allowed to choose whether or not to participate in a survey, there is a risk of selection bias. Webpolls asking for a yes-no opinion produce a strong self-selection bias, since the people most likely to vote are those who carry the strongest opinions one way or the other. Webpolls showing intermediate results to potential voters may further encourage voting from those who feel that their opinion is not being represented, thus producing a 'leveling' effect. This effect does not yield accuracy, it only causes the percentage to move away from the printed value.

If two million people are watching a sports news show which asks a question and gives a URL for voting, and forty thousand people vote, then the response rate could be considered to be five percent. While five percent of a target population may be sufficient for a well-designed sample survey, a self-selected subset of five percent of an already-biased population is unlikely to yield an unbiased estimate of the real population value. And, even worse, there is no way to tell how big that bias on that sample estimate is.

SOME BASIC PROBABILITY SAMPLES

PROC SURVEYSELECT has the capability of generating sample designs that can accommodate a variety of common survey methodologies. The best known sample design is simple random sampling, either with or without replacement of the sampling units. This often assumes that all the probabilities of selection are equal. But sampling can be done with unequal probabilities of selection. When the sampling units are ordered sequentially, sampling can be performed in a systematic manner, taking every *k*th unit starting from a randomly chosen position. Sampling can be done after breaking the sample into 'strata', and it can be done when choosing to sample within 'clusters' of the population. Sampling can be done as a multi-stage process or a single-stage process. Replicates of the sample can be created, as well. PROC SURVEYSELECT can do all of these processes, and more. But these techniques will be an adequate basis for this paper.

As an aside, let us note that the efficiency of PROC SURVEYSELECT makes it less useful for resampling techniques such as bootstrapping and jackknifing. PROC SURVEYSELECT is smart enough to notice when a sample of size n is requested from a population of n units, and simply generates an exact copy of the original sampling units, with no randomization. Currently, the easiest way to generate a bootstrap sample from a data set is to use the macro wrapper programs %RAND_GEN and %RAND_ANL as described in the paper "A Randomization-test Wrapper for SAS® PROCs" from SUGI 27. The macro %RAND_GEN generates all the replicates for the resampling procedure in a single file, which may grow extremely large when working with a large number of replicates of a large data set. Caution should be used when the size of this bootstrap data set will be of the same order as the available disk space.

SIMPLE RANDOM SAMPLING IN A SURVEY

Prior to PROC SURVEYSELECT, the best options for drawing a simple random sample from a sampling frame were SAS® code in a DATA step or two. Some excellent examples of macros to do this have been presented on the SAS-L mailing list by Dale McLerran. But, as of version 8, we have survey sampling tools in SAS®.

To draw a simple random sample without replacement (i.e., the probability of getting the same unit twice in the sample is zero), you can now invoke PROC SURVEYSELECT. A recurring question in SAS-L is the issue of drawing a simple random sample of a set size or sampling rate. To draw a simple random sample of size 200 out of a sampling frame called MyFrame, we can write:

```
proc surveyselect
     data=MyFrame
     out=MySample
     method=SRS
     seed=12345678
     sampsize=200
     stats;
run;
```

Note that we create an output data set called MySample, which will have all the design data that we need later on.. when we are finally forced to analyze the resulting collected data. Here, because we used the option STATS, the data set will include a SelectionProb variable and a SamplingWeight variable. SelectionProb will hold the likelihood of selecting that datum from the sampling frame. SamplingWeight will hold the (initial, unadjusted) weight that would be used as the sample weight when analyzing the data in PROC SURVEYMEANS or PROC SURVEYREG. The output data

set will automatically include these variables if unequal probability sampling is performed, or if stratified sampling is done. The output data set can also contain any variables included using an ID statement.

We recommend that you always specify your own random seed, so that your sample can be replicated.. as will be necessary as soon as you forget to use a random seed. We also recommend that you not use the seed 12345678 over and over again, unless you want to have to stand up at some point and loudly state [in an odd Peter Lorre-like voice]:

What?!? The Programmers LIKES 12345678!

To instead draw a sample of fifteen percent from the same frame, we can specify a different option:

```
proc surveyselect
     data=MyFrame
     out=MySample
     method=SRS
     seed=12345678
     samprate=15
     stats;
run;
```

This sampling design could also be specified as:

```
proc surveyselect
     data=MyFrame
     out=MySample
     method=SRS
     seed=12345678
     samprate=.15
     stats;
run;
```

Note the two different values for SAMPRATE which yield the same results. The PROC SURVEYSELECT translates sampling rates above one into a sampling fraction by assuming the number to be a percentage. However, if you write

```
    samprate=1
```

SAS® will assume you want a 100% sample, rather than a 1% sample. For a 1% sample, you must write:

```
    samprate=.01
```

If, instead of simple random sampling with equal probabilities, you need a simple random sample where some sampling units should be sampled with a higher probability than others, you can perform sampling with unequal probability of selection, simply by specifying a PPS method and having a size variable in the MyFrame data set. The size of a unit should be proportional to the probability of selection. So, if you want each woman listed in your sampling frame to have twice the likelihood of being selected as each man, then you would want to create a sampling size variable in the MyFrame data set and use that variable in the SIZE statement of PROC SURVEYSELECT:

```
data Frame2 / view=Frame2;
  multiplier = 1 + (sex='F');
run;

proc surveyselect
     data=Frame2
     out=MySample
     method=PPS
     seed=12345678
     samprate=.15
     /* STATS is no longer needed */ ;
  size multiplier;
run;
```

PROC SURVEYSELECT will generate the sampling weights that are appropriate for the design. These are the weights that will be needed later on, after the data have been collected. At that point, sample weights may have to be adjusted due to problems with the collection of the sample data. The adjusted sample weights will then be used when running PROC SURVEYMEANS and PROC SURVEYREG.

SYSTEMATIC AND SEQUENTIAL SAMPLING

Once upon a time, before computers, systematic sampling was an extremely useful tool. By ordering one's frame sequentially and selecting a starting point, an entire sample could be selected without generating any more random numbers. Unfortunately, a single systematic sample actually has no unbiased variance estimate. However, the use of more than one systematic sample from the same frame will yield an unbiased variance estimate. The authors do not recommend the use of standard systematic sampling with only a single replicate taken. Sequential sampling processes are available from PROC SURVEYSELECT using the CONTROL statement. Chromy (1979) devised a way of walking through the sequentially-ordered sampling units so that there is an unbiased variance estimator. Systematic and sequential sampling can, in theory, provide sampling variances smaller than that of simple random sampling when the frame has been sorted so that the correlation between pairs of selected units in a sample is sufficiently small. So here is a survey design using a sequential sample:

```
proc surveyselect
     data=MyFrame
     out=MySample
     method=PPS_SEQ
     seed=12345678
     samprate=.15;
run;
```

STRATIFIED SAMPLING

When the target population can be divided into non-overlapping subpopulations, stratified sampling may come to mind. It may not be the technique which should come to mind, but it usually does. One of the authors has frequent problems with end-users who have heard of stratified sampling and want it, whether it is a reasonable choice or not.

There are several reasons why stratified sampling should be considered. Cochran (1977) listed the classic four cases: when known precision (i.e., known sample size) is desired for

specific non-overlapping subdivisions of the population; administrative convenience, such as sampling when field offices will be conducting the survey for a designated portion of the population; studies in which sampling problems may differ in different parts of the population, such as a survey which samples people living in care facilities and also people living at home; and finally, studies in which the stratification may produce a gain in the precision of the variance estimates.

Unfortunately, stratified sampling is not always appropriate. When known precision is desired, sampling proportional to size can provide sample sizes close to the precisely-desired stratum sizes. When gains in precision are desired, the stratification must be very precise: a misclassification rate of as little as twenty percent on the stratum definitions can invalidate any gain in precision from using stratified sampling over simple random sampling. Logistic reasons are the best reasons for performing stratified sampling.

Here is an example of stratified sampling where there are four regions which will be used as strata, and the sample sizes for the strata are specified so that the total sample of 200 is split as desired among the strata. The STRATA statement works analogously to the BY statement in many PROCs. It can take more than one stratification variable, and it even has DESCENDING and NOTSORTED keywords like the BY statement. It can even use an index built on the stratum variables in PROC DATASETS, just as the BY statement can. As with the BY statement, NOTSORTED does not mean that the data are unsorted: it means that the data are arranged into the strata, but the strata are not necessarily in ascending or descending order. Here the data are arranged by region but the regions have been appended in what may not be a sorted order. Note that you cannot select proportional to size or using a CONTROL statement unless the strata are sorted or indexed in ascending order.

```
proc surveyselect
     data=MyFrame
     out=MySample
     method=SRS
     seed=12345678
     sampsize=(24 48 56 72);
  strata region notsorted;
run;
```

Notice that SAMPSIZE is now used to provide an array of sample sizes, one per stratum. The order of the strata is now crucial, or else the wrong sample sizes may be applied to the strata. The SURVEYSELECT procedure provides a way of addressing this problem, as well as the problem of providing stratum sizes when there are a large number of strata. The SAMPSIZE option can also take the name of an auxiliary data set. This data set need only have the stratum names and stratum sizes. The stratum names in the auxiliary data set must be exactly as provided in the MyFrame data set, with precisely the same length and type. The stratum sizes should be put into a variable named _NSIZE_ . The values for _NSIZE_ must be positive integers, and if sampling is done without replacement, the stratum sizes must be at most the number of sample units in the stratum. Furthermore, the order of the strata in the auxiliary data set must match the order of the strata in the frame data set. If an auxiliary data set MyStratSizes is used to build the stratum sizes, then the procedure would be invoked like this:

```
proc surveyselect
     data=MyFrame
     out=MySample
     method=SRS
     seed=12345678
     sampsize=MyStratSizes;
  strata region notsorted;
run;
```

This will work properly as long as the strata in MyFrame and MyStratSizes are arranged in the same order, even if that order is not a sorted order.

CLUSTER SAMPLING AND MULTISTAGE SAMPLING

There are two main reasons for using cluster sampling. The first is logistical: sometimes it is not possible to build a complete, reliable sampling frame for the entire population. In such cases, clusters of the population can be constructed, and sampling frames can be built for those clusters. The second reason is budgetary: it is often far more cost-effective to sample all students within a school, all patients in a clinic, or all discharges from a hospital than to randomly sample a few here and a few there from thousands of institutions. For example, the NIS selects all discharges with a set of sampled hospitals.

A multistage sample might be a sample in which you first draw a first-stage sample as if you were performing a cluster sample, and then you create a second-stage sample as subsets of the clusters. The elements composing the frame from which the first sample is drawn are called the primary sampling units. Then a second-stage sample is built by taking a subset of elements from each of the primary sampling units.

While cluster sampling can make your sample more cost-effective, it can also cause a serious loss in precision of your estimates as compared to a simple random sample. A standard technique to deal with this problem in cluster sampling is to make sure that your sample within each cluster is as heterogeneous as possible for the variables of interest.

Let us suppose that we want a cluster sample of hospitals (using the hospital ID variable HospID), and that we will want to stratify based on the variable REGION, due to the complications of taking samples in different regions using different administrative staff and different field personnel. The stratum sizes are already in the auxiliary data set MyStratSizes. Here the primary sampling units are the hospitals, not the patients. We will take a sample of hospitals, and then survey all patients in each selected hospital. Since the hospitals are our primary sampling units, we will only need to use the design information used in selecting the hospitals in order to use the SURVEYREG and SURVEYMEANS procedures. The approximations used in these procedures are based on the design used for the primary sampling units.

```
proc surveyselect
     data=MyFrame
     out=MySample
     method=SRS
```

```
     seed=12345678
     sampsize=MyStratSizes;
  strata region;
  id HospID;
run;
```

This stratified sample of hospitals now yields a stratified, clustered sample when we evaluate all patients in each selected hospital.

If we then choose to take a sample of the patients in each hospital, rather than surveying all the patients, we would have a two-stage sample. We can achieve this by building a new frame comprised of all those patients in the selected hospitals, and selecting a sample from those patients.

Suppose we have a second data set MyPatients, composed of all patients in all of our hospitals. We need to subset that to the patients in the selected hospitals. Then we need to select a specific number of patients in each hospital. We can do that in our second stage of sampling by treating the hospitals as if they were strata, selecting patients within each stratum, and providing the appropriate stratum sizes in an auxiliary data set, which we will call MyPatientSizes. If MyPatients is already sorted in the same order as MySample, then we can accomplish the second stage of sampling like this:

```
data MyFrame2;
  merge
    MyPatients
    MySample (keep=HospID in=inSample);
  by HospID;
  if inSample;
run;

proc surveyselect
     data=MyFrame2
     out=MySample2
     method=SRS
     seed=87654321
     sampsize=MyPatientSizes;
  strata HospID;
run;
```

Now the DATA step above may not be the optimal approach. Depending on data set sizes and available disk space, it might be preferable to create the data set MyFrame2 as a data step view. Or, if the two files are not sorted or indexed on HospID, the programmer might prefer to use PROC SQL to do this merge. If the data sets are extremely large, or must be read off tapes, then it might be preferable to read all the HospID values into an array, then read in the MyPatients data set and search the array to find matching patient records to be output in the MyFrame2 data set.

Since the above process can be repeated, there is no reason why this sample cannot be performed as a multi-stage sample, with more than two stages. Perhaps sampling needs to be done by hospital, and then within each hospital we need to do cluster sampling by care facility, with sub-sampling of patients done in each care facility. We could use the same techniques to perform a three-stage sample. We build a facility frame based on the selected hospitals, and select the facilities to be sampled. Then we build a patient frame based on the selected facilities within hospitals, and sample those patients within care facilities in the same way that we sampled the facilities within hospitals.

What?!? The Analyst LIKES using PROC MEANS!

A key point to remember is that the standard SAS® statistical procedures assume that you are working with a simple random sample from an infinite population. Instead of a simple random sample (where every sampling unit has an equal chance of being selected), consider a sample where some units have a greater chance of selection. Samples proportional to size, stratified samples, and cluster samples can all lead to this situation.

If the data are not randomly sampled, then using PROC MEANS or PROC REG can mean that the standard errors are wrong. It may also (if the weights are not taken into account) mean that the point estimates form PROC MEANS are wrong too. And even if the weights are used in PROC REG, the point estimates for the regression coefficients cannot be trusted. But the joy of PROC SURVEYMEANS and PROC SURVEYREG is that you do not have to understand the intricacies of the sampling design as long as you wrote down that you have a stratified cluster sample with known weights, and you wrote down the names of the variables for stratification, clustering, and weighting. Even better, you do not have to look under the hood and study how the Taylor series expansion method is used to obtain the sampling errors for the sample estimators you want. All you need to know is what design was used to create the primary sampling units. This is due to the way in which the Taylor series expansion theory is used to estimate the sampling errors of the estimators. So, if you have a complex, multi-stage sampling design as in our hospital-facility-patient example, you still only need the information used in building the first stage of the design in order to use the SURVEYREG and SURVEYMEANS procedures.

USING PROC SURVEYMEANS

So, assuming you did write down the information above, you are ready to analyze the data properly. You know that the data are in a file called YourSample, with stratum variable REGION, cluster variable HospID, and weighting variable Adj_Weight. You want mean estimates and their standard errors for the population parameter LOS:

```
proc surveymeans data=Your sample;
  strata REGION;
  cluster Hispid;
  weight Adj_Weight;
  var LOS;
run;
```

If you wanted to perform this same analysis by gender, you might be tempted to sort by gender and then use the BY statement. Don't do that! At least, if you try this, SAS® will give you a note that tells you to use the DOMAIN statement. The SAS® log will contain the note:

```
NOTE: The BY statement provides
completely separate analyses of the BY
groups. It does not provide a
statistically valid subpopulation or
domain analysis, where the total number
```

```
of units in the subpopulation is not
known with certainty. If you want a
domain analysis, you should use the
DOMAIN statement.
```

Domain analysis, or sub-population analysis as it is sometimes called, will provide an appropriate analysis, by adding only one line to the above code:

```
proc surveymeans data=Your sample;
  domain GENDER;       /* new line */
  strata REGION;
  cluster Hispid;
  weight Adj_Weight;
  var LOS;
run;
```

The DOMAIN statement is available as of SAS® version 8.2, but if you have an earlier version of SAS® you can still make do. The %SMSUB macro is available on the SAS® website.. if you know how to search it. But even if you are running version 8.2 or version 9.x, you still have to watch out for large numbers of missing values in your data. The SURVEYMEANS procedure is based on the assumption that missing data are MCAR - Missing Completely At Random - so that the pattern of missing values should not affect the estimates. SAS®, unlike other statistical packages designed to analyze complex survey data, treats these as records to be deleted before the analysis, rather than as another subgroup of respondents, which both authors agree is the appropriate way to deal with them. Grouping the data by the occasionally-missing variable (we'll call it GENDER for our example) and using the %SMSUB macro will give you the results you would expect to get from other survey analysis packages. Once you use %INCLUDE to include the %SMSUB macro in your code, you can call it like this:

```
filename incl_lib "path to smsub macro";
%include incl_lib(smsub);

%smsub( data=Your sample,
        statistics=mean stderr,
        weight=Adj_Weight,
        strata=REGION,
        cluster=Hispid,
        var=LOS,
        table=GENDER );
```

Unfortunately, the %SMSUB macro creates a large number of temporary data sets, some of which are large. After using the %SMSUB macro, you will want to clean up your workspace to prevent %SMSUB from devouring your hard drive:

```
proc datasets library = work;
  delete _: ;
run;
```

USING PROC SURVEYREG

Just as the SURVEYMEANS procedure provides the error-variance calculations that are needed for complex survey designs, so the SURVEYREG procedure does regression analyses using Taylor expansion theory to estimate the proper variance-covariance matrix for the regression coefficients. And, joy of joys, you still do not need to learn how Taylor expansion theory is used in survey sampling theory. It uses a structure analogous to PROC REG, only adding STRATA and CLUSTER statements. So a simple example of modeling the hospital expenses based on patient income, given the same sampling design as before, would look like this:

```
proc surveyreg data=Your sample;
  strata REGION;
  cluster Hispid;
  weight Adj_Weight;
  model expense = income;
run;
```

The SURVEYREG procedure also has the CLASS, CONTRAST, and ESTIMATE statements of PROC REG. This allows the programmer to use techniques that have been learned through years of working with the REG procedure.

The SURVEYREG procedure uses ODS directly, rather than using an OUTPUT statement as is often seen in older procedures. The parameter estimates for the above model can be output to a data set using the following code. Note that this does not require the usual ODS statements before and after the procedure. All the necessary ODS structure has been internalized.

```
proc surveyreg data=Your sample;
  strata REGION;
  cluster Hispid;
  weight Adj_Weight;
  model expense = income;
  ods output ParameterEstimates=MyParams;
run;
```

More output data sets can be obtained using the same statements. Here, we also obtain the fit statistics and the inverse of the **X'X** matrix.

```
proc surveyreg data=Your sample;
  strata REGION;
  cluster Hispid;
  weight Adj_Weight;
  model expense = income / inv;
  ods output
       ParameterEstimates = MyParams
       FitStatistics = FitForLife
       InvXPX = Inverted;
run;
```

The INV option in the MODEL statement is needed to obtain the inverse of the **X'X** matrix. All such requirements are listed in the Online documentation, under "ODS Table Names".

Logistic regression for sample surveys is not yet available in a procedure, but PROC SURVEYLOGISTIC will be available in SAS® version 9.1 .

EPILOGUE

There are many ways to properly draw a probability sample. The SURVEYSELECT procedure gives you a large number of ways to design sample surveys, so that you can cope with the complexities of real-life samples. When simple random sampling (with or without replacement) is appropriate, it can generate your sample. And, when clustering is needed, or stratification, or sample weights, or multi-stage sampling, the

SURVEYSELECT procedure can do that too.

Unfortunately, there are far more ways to collect a sample which is not a probability sample. And there are often poorly-designed survey samples which may not meet the objectives of the study due to poor precision or bad accuracy. Once the data have been collected, perhaps not even Buena Girl could salvage your study.

Analysis of a well-designed probability sample is usually straightforward with the proper tools. The SURVEYMEANS and SURVEYREG procedures will meet those needs.

Analysis of a badly-design sample will most likely yield numbers which can never be fully evaluated or statistically adjusted. Unfortunately, bad analysis of a well-designed sample survey can have equally ugly consequences. Remember the signature move of The Flea. No one wants to appear before a client or a boss while resembling a flaming turkey.

REFERENCES

Cassell, D.L., 2002. "A Randomization-test Wrapper for SAS® PROCs". SUGI 27 Conference Proceedings, SAS Institute, Inc.

Chromy, J.R., 1979. "Sequential Sample Selection Methods". Proceedings of the American Statistical Association, Survey Research Methods Section, pp. 401-406.

Cochran, W.G., 1977. Sampling Techniques. Third edition, New York: John Wiley & Sons, Inc.

ACKNOWLEDGMENTS

CONTACT INFORMATION

The authors welcome questions and comments. All serious gaffes are no doubt due to the lead author, who can be reached at:

David L. Cassell
Design Pathways
3115 NW Norwood Pl.
Corvallis, OR 97330
Cassell.David@epamail.epa.gov

The second author can be reached at:

AnnMaria Rousey
annmaria@spiritlakeconsulting.com