Paper ST-05

PROC SURVEY… Says!: Selecting and Analyzing Stratified Samples

Darryl Putnam, CACI Inc., Elkridge, MD

ABSTRACT

Statisticians and analysts need to design stratified survey plans and analyze the results of those surveys. Gone are the days when the analyst can ignore survey design tools when drawing inferences from the surveys. By forgoing the SAS© survey analysis procedures, estimates of the mean and standard error will be incorrect. By combining DATA STEP processing with the SAS© survey analysis procedures of PROC SURVEYSELECT and PROC SURVEYMEANS, we can determine the sample size, allocate the sample size across strata, and then draw correct inferences. This paper will demonstrate how to use these survey design and analysis tools with a stratified sample of an inventory audit.

INTRODUCTION

Statistical sampling is used when an auditor wants to draw conclusions about a population without performing an examination of all the items (Hitzig). When performing a financial audit, the analyst is tempted to use the statistical methods that were learned in introductory statistics classes. One key but often overlooked assumption in classical statistics is that the data are derived from an infinite population (Cassel). This key assumption drives the properties of the data such as: independent observations and identically distributed errors. By using PROC MEANS or PROC UNIVARIATE, we can simply calculate the mean and standard error to draw our inferences. Unfortunately, this approach can lead us to incorrect conclusions.

Survey data on the other hand assumes that the underlying data are derived from a finite population and as a result the properties of independent observations and identically distributed errors do not apply. Data used in financial audits fall into the survey data arena and survey statistics are needed to avoid possible misstatements. SAS has many procedures that use survey statistics instead of classical statistics.
By using PROC SURVEYMEANS instead of PROC MEANS, we can get the correct standard errors and draw the correct inferences.

The decision on whether to stratify our survey data depends on the data itself. In stratified sampling, the population is divided into sub-populations called strata and in each of these sub-populations a simple random sample is taken. The main benefit of stratified sampling is to take a heterogeneous population and sub-divide it so the sub-divisions are relatively homogenous. If each stratum is more homogenous than the population at large, then the precise estimate of any stratum mean can be obtained from a small sample size. This is especially important when the distribution of the data is highly skewed. These estimates can then be combined into a precise estimate for the whole population.

EXAMPLE – USING SASHELP.SHOES

Since every installation of SAS comes with sample data in the SASHELP library, the data file SASHELP.SHOES will be used to demonstrate the stratification process and the statistical methods used to analyze the sample. Our hypothetical shoe company has sales and holds inventory all over the world and our goal is to audit the inventory value. The auditors require a precision/relative error of 5% and a confidence level of 95%.

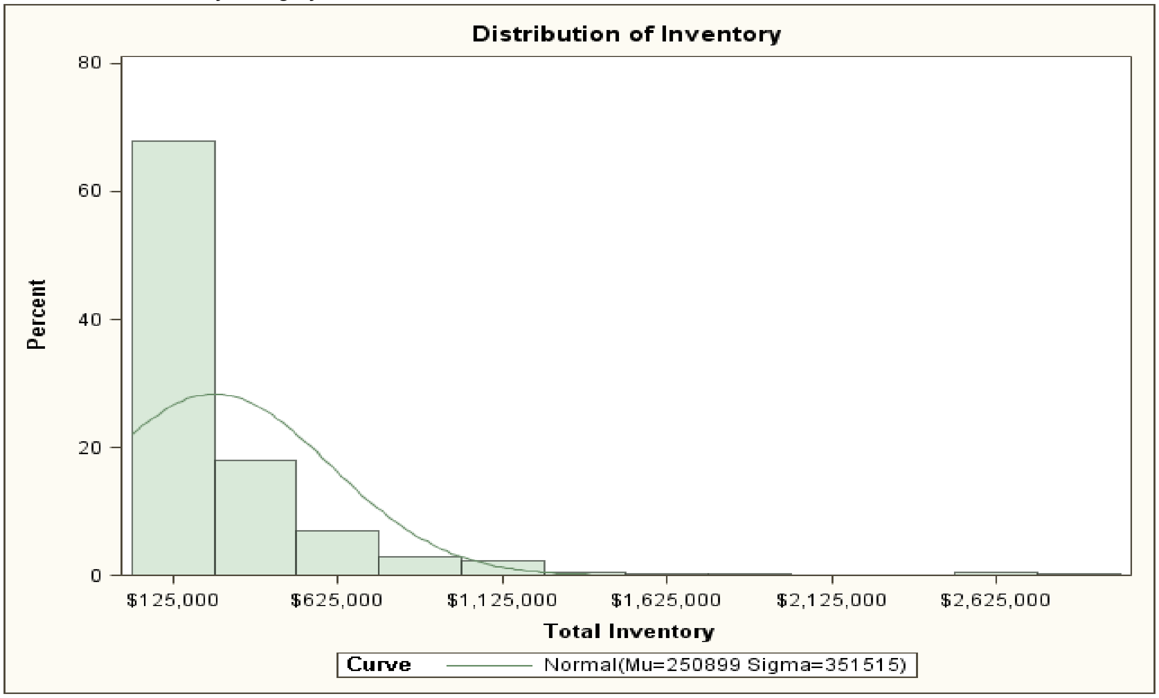
By using Cochran's seminal work "Sampling Techniques" as our guide we will:

1. Create the Strata.
2. Calculate Total Sample Size.
3. Allocate Total Sample Size Among the Stratums.
4. Calculate our confidence levels.
5. Analyze the results.

But first let us describe the data.

HISTOGRAM OF THE DATA

Running a simple PROC UNIVARIATE to get the distribution of the inventory valuation, we find that the value of the inventory is highly skewed.



A simple random sample will give each observation an equal chance of being picked. Performing an audit cost the same whether the item has a value of $1 or $100,000; the auditor still needs to verify the existence of the inventory item. The auditors will want to sample more from the higher value items than from the lower value items. We can accomplish this with two methods: we can sample using probability proportional to size which weights the universe by the value of inventory or we can stratify the universe by inventory value. For this paper we will stratify the data so that the first stratum has the highest value items and the last stratum has the lowest value items.

CREATE THE STRATA

The optimal way to construct the strata and how many stratums are created is beyond the scope of this paper. For this paper, four stratums were created based on the value of inventory. The high value inventories are in stratum 1 and the low value inventories are in stratum 4. In order for the reader to follow along, the code used to create the strata is below:

```
data universe;
  set sashelp.shoes;
  if        inventory>=940851 then stratum=1;
    else if inventory>=527226 then stratum=2;
    else if inventory>=275526 then stratum=3;
    else stratum=4;
run;
```

CALCULATE SAMPLE SIZE

One of the first steps in a survey design is to figure out what sample size to use. If the sample size is too big, the audit may be cost prohibitive. If the sample size is too small, the statistical power of the survey can suffer. Theory can tell us the minimum sample size to use to meet our criteria. As mentioned above, the auditors would accept a sample that produces a 5% precision within the typical 95% confidence level. Using our trusty formulas by Cochran, we can calculate the sample size by using the number of stratums, the confidence level, and the precision as inputs, along with the stratum means and variances. By using DATA STEP processing, we can get the total sample size that meets our criteria. In our example, the formulas computed a total sample size of 102. The code that produces all the sample size calculations is located at the end of this section.

Once we calculated the total sample size, we need to allocate it across the strata. One of the best methods in allocating the sample size across the stratums is the Neyman allocation. The Neyman allocation has been proven optimal in a wide number of circumstances (Cochran). We can do that via PROC SURVEYSELECT. Many think that PROC SURVEYSELECT can only be used for generating samples, but it can also be used to generate stratum level sample sizes.

```
** the universe needs to sorted in stratum order for input;
** into PROC SURVEYSELECT;
proc sort data=universe;
  by stratum descending inventory;
run;

** Run descriptive statistics on the stratums;
** Output population descriptive statistics as a SAS data set;
** This will used as an input into the sample size algorithm;
proc summary data=universe;
  class stratum;
  var inventory;
  output out=pop_strata_stats1
    sum=pop_sum n=pop_n mean=pop_mean std=pop_std var=pop_var
       min=pop_min max=pop_max;
run;

proc surveyselect data=universe1 n=102 out=samplesize_ps ;
  strata stratum / alloc=neyman
                   var=pop_strata_stats1
                      (where=(_type_ ne 0) rename=(pop_var=_var_))
                   nosample;
run;
```

Below are the options of interest:

- n=total sample size.
- strata=stratum is the name of our strata.
- out=name of data set containing the sample sizes for each stratum.
- alloc=neyman allocates the total sample size by using the Neymen allocation.
- var=data set containing the variance for each stratum. The name of the variance variable must be _VAR_.
- nosample tells SAS not to take a sample.

All is good, right? After an examination of the log we get this note.

```
NOTE: The sample size allocation does not strictly follow Neyman proportions
because a stratum sample size cannot exceed the stratum total for without-
replacement selection.
```

In at least one stratum, the sample size was bigger than the population size. PROC SURVERSELECT makes adjustments to the stratum level sample size to keep the total sample size of 102. However, there are some unintended consequences in ignoring this note; the allocation is no longer optimal. The precision of the estimate will be higher than expected and it is up the analyst to determine the impact of this fact. Fortunately for us, Cochran has a formula to revise the optimal sample size. In our quest for statistical perfection we will now use the DATA STEP to get the revised optimum sample size. The code at the end of this section creates a SAS data set which contains the revised optimum sample for each stratum based on our precision and confidence level. In order to maintain our precision, the total sample size was increased from 102 to 112.

From our analysis of the sample size algorithm, we can see that the Neyman allocation method wanted a sample size of 27 for stratum 1, but stratum 1 only had a population of 17. This was the cause of the note in our log when we attempted to use PROC SURVEYSELECT to allocate the sample size across the stratums. The below chart summarizes the difference in sample sizes for the different methods used. PROC SURVEYSELECT proportionally spreads the extra wanted 10 items in stratum 1 across the remaining stratums. The revised optimal formula recalculates the total sample size with regard to the stratum population sizes and variances. See Cochran “Sampling Techniques” p.104 for a thorough discussion on this topic.

Summary of sample size revisions

Stratum	Population Size	Original Neyman Allocation	PROC SURVEYSELECT	Revised Neyman Allocation
1	17	27	17	17
2	36	11	12	14
3	65	12	13	15
4	277	52	60	66
Total	**395**	**102**	**102**	**112**

The DATA STEP code used to generate the sample size is below:

```
%let nstrata=4;
%let precision=.05;
%let confidence_level=.95;

** Sample Size Algorithm **;
** The formulas are based from
"Sampling Techniques, William Cochran 3rd Edition";
** -1) Calculate variance of mean estimator "V";
** -2) Calculate total sample size;
** -3) Using Neyman allocation, allocate total sample among strata;
** -4) Test that the sample size allocation follows Neyman proportions because
       a stratum sample size cannot exceed the stratum total for without-
       replacement selection.;
** -5) Repeat 2-4 until all stratums pass the test;
data SampleSize ;

  ** Create vector arrays to hold the stratum level and total statistics;
  array pop_n      (0:&nstrata)    pop_n0-pop_n&nstrata;
  array pop_mean   (0:&nstrata)    pop_mean0-pop_mean&nstrata;
  array pop_std    (0:&nstrata)    pop_std0-pop_std&nstrata;
  array pop_var    (0:&nstrata)    pop_var0-pop_var&nstrata;
  array w          (0:&nstrata)    w0-w&nstrata;
  array samp_n     (0:&nstrata)    samp_n0-samp_n&nstrata;
  array osamp_n    (0:&nstrata)    osamp_n0-osamp_n&nstrata;
  ** coding aid for testing stratum sample size le population stratum size;
  array alloc_test (0:&nstrata)    alloc_test0-alloc_test&nstrata;

  * initialize values;
  i=0;                                * generic iterator;
  L=&nstrata;                         * number of strata;
  precision=&precision;               * precision value;
  confidence_level=&confidence_level; * confidence_level of estimate;

  * load population statistics into a single record;
  * consisting of multiple arrays. The output of the summary procedure;
  * has the total values on the _type_=0 record, which will be array;
  * item 0 as the totals and array items 1-n as the stratum statistics;
  do until (eof);
    set pop_strata_stats1 (keep=pop_n pop_mean pop_std pop_var
                           rename=(pop_n=n pop_mean=mean pop_std=std
                                   pop_var=var)) end=eof;
    pop_n[i]=n;
    pop_mean[i]=mean;
    pop_std[i]=std;
    pop_var[i]=var;
    alloc_test[i]=1;
```

```
    i+1;
  end;

  * Variance to minimize used by the optimal allocation method for determining
    the stratum sample size;
  * Desired variance is the margin of error, d=mean value*precision;
  * V=(d/z)^2;
  * In section 4.4 the margin of error calculation is discussed;
  * See section 5.9 for details, Cochran;
  desired_variance=(pop_mean[0] * precision / quantile(
                  'normal',confidence_level+(1-confidence_level)/2) )**2;

  ** Neyman Allocation with revision if stratum sample size is greater than the
     Stratum population size;
  ** Formula 5.27 to calculate initial total sample size;
  ** Section 5.8 formulas 5.41,5.42,5.42;
  allocation_pass=0;
  i=0;
  do until (allocation_pass=L or i=100);
    * initialize variables;
    sum_ws=0;  * sum of weighted std(standard deviation);
    sum_wv=0;  * sum of weighted var(variance);
    sum_ns=0;  * sum of for total stratum variances;
    ** calculate total sample size;
    do h=1 to L;
      w[h]=pop_n[h]/pop_n[0];
      sum_ws=sum_ws + (alloc_test[h] * w[h]* pop_std[h]);
      sum_wv=sum_wv + (alloc_test[h] * w[h]* pop_std[h]**2);
      sum_ns=sum_ns + (alloc_test[h] * pop_n[h] * pop_std[h]);
    end;
    sum_ws2=sum_ws**2;
    sum_wv2n=sum_wv/pop_n[0];

    total_samplesize=ceil(sum_ws2 / (desired_variance+sum_wv2n));
    if i=0 then do;
      call symputx('total_samplesize',total_samplesize);
      do h=1 to L;
        if i=0 then osamp_n[h]=ceil((total_samplesize * pop_n[h] * pop_std[h]) /
                              sum_ns);
      end;
    end;

    do h=1 to L;
      if alloc_test[h]=1 and samp_n[h] ne pop_n[0] then
         samp_n[h]=ceil((total_samplesize * pop_n[h] * pop_std[h])/ sum_ns);
      alloc_test[h]=(samp_n[h] le pop_n[h]);
      if alloc_test[h] eq 0 then samp_n[h]=pop_n[h];
    end;

    allocation_pass=sum(of alloc_test1-alloc_test&nstrata);
    final_samplesize=sum(of samp_n1-samp_n&nstrata);
    i=i+1;     * max out at 100 iteration to stop infinite looping;
  end;

  put 'Number of Iterations: ' i;
  do h=1 to l;
    put 'POP SIZE: ' pop_n[h] 'ORIGINAL SAMPLE SIZE: ' osamp_n[h]
        'REVISED SAMPLE SIZE: ' samp_n[h];
  end;
run;
```

TAKE A SIMPLE RANDOM SAMPLE WITHIN EACH STRATUM

The next step is to create the actual sample. By using PROC SURVEYSELECT, we can create a stratified simple random sample. But first we need to create a data set that has stratum and sample size. PROC SURVEYSELECT needs the name of the sample size variable to be _NSIZE_ and the name of the STRATA variable needs to be the same as it is the universe data set.

Reshape the output from the previous DATA STEP into a format that PROC SURVEYSELECT can use.

```
data StratumSampleSize(keep=stratum _NSIZE_ _TOTAL_);
  set SampleSize;

  array samp_n (0:&nstrata) samp_n0-samp_n&nstrata;
  _TOTAL_=sum(of samp_n1-samp_n&nstrata);

  do stratum=1 to &nstrata;
    _NSIZE_=samp_n[stratum];
    output;
  end;
run;
```

Use PROC SURVEYSELECT to get the stratified random sample.

```
ods output summary=sampling_summary method=sampling_method;
proc surveyselect data=universe1 sampsize=StratumSampleSize  method=srs
  out=sample_data seed=8;
  strata stratum ;
run;
```

Below are the options of interest:

- sampsize=data set holding the sample size for each stratum.
- method=srs is for simple random sample.
- out=output data set containing the sample.
- strata=stratum is the name of our strata.
- seed=the random seed.

ANALYZING THE RESULTS

Let us create a data set with the audit results to demonstrate the differences between PROC MEANS and PROC SURVEYSELECT. The below code assumes that the inventory audit found a 40% loss of the inventory value in the Chicago subsidiary. Even though these results may be extreme, they will make a point. It would make sense that our statistical test should point out that the differences are significant.

```
data validated_data;
  set sample_data;

  validated_inventory=inventory;
  if Subsidiary='Chicago' then validated_inventory=inventory*0.6;

  diff=validated_inventory - inventory;
run;
```

In an audit, the auditors are looking for differences between the database and validated amounts that are significantly different from zero. If the results are outside of acceptable error, then there may be a problem.

Below is a typical PROC MEANS code on the sample results, which retrieves the mean, standard error of the mean, and the confidence intervals at the 95% level (which is the default setting).

```
title 'PROC MEANS';
proc means data=validated_data mean std stderr clm;
  var inventory validated_inventory diff;
run;
```

Since we took a weighted sample via stratification, we can add those weights produced by PROC SURVEYSELECT into PROC MEANS by using the WEIGHT statement.

```
title 'PROC MEANS with weights';
proc means data=validated_data mean std stderr clm;
  var inventory validated_inventory diff;
  weight SamplingWeight;
run;
```

Even though using PROC MEANS with the WEIGHT statement seems promising, the weights in PROC MEANS are not survey weights and may lead us astray (Cassel). We must use PROC SURVEYMEANS to use the correct sampling weights and add the finite population correction via the TOTAL= option.

```
title 'PROC SURVEYMEANS with Weights';
proc surveymeans data=validated_data mean std stderr clm total=395;
  var inventory validated_inventory diff;
  weight SamplingWeight;
run;
```

Since our sample design is stratified so we need to include the STRATA statement to get the correct results.

```
title 'PROC SURVEYMEANS with Strata and Weights';
proc surveymeans data=validated_data mean std stderr clm
total=StratumSampleSize;
  strata stratum;
  var inventory validated_inventory diff;
  weight SamplingWeight;
run;
```

Below are the options of interest:

- total=StratumSampleSize, this data set contains the variables STRATUM, _TOTAL_(the total sample size), _NSIZE_(stratum sample sizes). This is used to calculate the finite population correction. Notice the difference between total=365 (in PROC SURVEYMEANS with Weights) and total=StratumSampleSize from above.

  From the SAS documentation:

  - If your sample design is stratified with different sampling rates or population totals in different strata, use the RATE=***SAS-data-set*** or TOTAL=***SAS-data-set*** option to name a SAS data set that contains the stratum sampling rates or totals. This data set is called a ***secondary data set***, as opposed to the ***primary data set*** that you specify with the DATA= option.
  - The secondary data set must contain all the stratification variables listed in the STRATA statement and all the variables in the BY statement. If there are formats associated with the STRATA variables and the BY variables, then the formats must be consistent in the primary and the secondary data sets. If you specify the TOTAL=***SAS-data-set*** option, the secondary data set must have a variable named *_TOTAL_* that contains the stratum population totals. Or if you specify the RATE=***SAS-data-set*** option, the secondary data set must have a variable named *_RATE_* that contains the stratum sampling rates. If the secondary data set contains more than one observation for any one stratum, then the procedure uses the first value of *_TOTAL_* or *_RATE_* for that stratum and ignores the rest.

- Strata=stratum is the name of our strata.

Our solution comes with PROC SURVEYMEANS which is designed for survey analysis. PROC SURVEYMEANS is quite similar to PROC MEANS with nearly identical syntax. The main difference is the STRATA statement and the TOTAL= option. Since we used a stratified design, we must use the STRATA statement. The TOTAL = option is used to calculate the finite population correction (fpc) and adjusts the estimates with it. The fpc can be ignored with small sampling fractions, but since we are sampling about 1/3 of the data the fpc was included. If the fpc is not used then the standard error is overestimated (Cochran). As the sampling fraction grows we will know more of our population, and hence will need to estimate less of it (Cassel).

An analysis of the results shows the WEIGHT statement in both PROC MEANS and PROC SURVEYSELECT produce the same mean, but the standard errors are off which affects the confidence levels. In an audit, we would want to test if the difference is significantly different from zero. Notice that by using PROC MEANS with WEIGHTS, the confidence intervals have a zero within its range, while PROC SURVEYSELECT with STRATA and WEIGHTS confidence intervals do not include zero. By using PROC SURVEYSELECT with STRATA and WEIGHTS, we can conclude that a possible issue is occurring with inventory management.

Statistics for DIFF

Procedures	**Mean**	**Std Error**	**Lower 95% CL for Mean**	**Upper 95% CL for Mean**
PROC MEANS	-$10,476	$5,519	-$21,411	$459
PROC MEANS with WEIGHTS	-$6,512	$4,102	-$14,640	$1,616
PROC SURVEYSELECT with WEIGHTS	-$6,512	$2,850	-$12,159	-$865
PROC SURVEYSELECT with STRATA and WEIGHTS	-$6,512	$3,057	-$12,573	-$451

CONCLUSION

The statistical programmer has an arsenal of PROCs to choose from in SAS/STAT. Knowing when to use survey statistical methodology will lead to more accurate and defendable results. In some cases, using the wrong PROC can lead to misleading results. In our example, by using PROC MEANS we would have come up with the wrong conclusion, but by using PROC SURVEYMEANS we can get the correct inference. With SAS, we can have complete control over our statistical survey plan, from calculating sample sizes, producing samples, and reporting on the results.

REFERENCES

David L. Cassell, 2006," Sample Survey Data and the Procs You OUGHT To Be Using". Design Pathways, Corvallis, OR.

Cochran, W.G. 1977. "Sampling Techniques", 3rd Edition, Wiley.

Statistical Sampling Revisited, Neal B Hitzig, "The CPA Journal".

SAS OnlineDoc® 9.22, BASE SAS and SAS/STAT.

Diana Suhr, "Selecting a Stratified Sample with PROC SURVEYSELECT ", University of Northern Colorado.

CONTACT INFORMATION

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

Darryl Putnam
CACI, Inc.
6835 Deerpath Road
Elkridge, MD 21075
Work Phone: 410-762-6535
E-mail: dputnam@caci.com