Paper AD190

Using PROC SURVEYSELECT: Random Sampling

Raissa Kouadjo Bordenave, Florida A&M University

ABSTRACT

This paper will examine some of the capabilities of PROC SURVEYSELECT in SAS to show the task of drawing a random sample. PROC SURVEYSELECT allows the user the flexibility to customize the design parameters and every SAS programmer needs to know how to design a statistically efficient sample. An overview of sampling designs, selection methods, and examples are presented in this paper. The examples were run on SAS 9.4.

A MEMBER OF THE FAMILY

PROC SURVEYSELECT is a powerful tool by SAS to assist users in designing and developing reliable and representative samples for study. This procedure is a member of the SURVEY family with a list that includes PROC SURVEYMEANS, SURVEYREG, SURVEYLOGISTIC, SURVEYSELECT, and others that support analysis of complex surveys. PROC SURVEYSELECT offers two main methods of sample selection: equal probability sampling and proportional probability sampling (PPS). Both of these methods are customizable depending on preferences and deconstructed throughout the paper.

It is imperative to understand the complexities of designing an effective sampling method to produce an appropriate sample. The following is a review of possible unfamiliar terms:

- Sample - a subset of the study population that is ideally free from bias, representative of the target population, and has an operative sample size that warrants statistical analyses
- Sampling frame - a list of all the members of the population
- Sampling Unit - Each member, individuals or groups, of the sampling frame
- Sampling fraction - the ratio of the sample size to the study population size
- Sample size - the number of observations or units in the sample

SAMPLE DESIGNS

The choice of sample design is based on the characteristics of the target population and the research question. The correct selection of the design aims to reduce biases and errors and increases the validity of the results. Sampling methods can be broken down into two main umbrellas: equal probability sampling, or random sampling, and non-equal probability sampling, also known as PPS (proportional probability sampling) in SAS. Equal probability sampling is preferred over non-equal probability sampling because there is an absence of the researcher's preference of the selection of the units and there is a greater chance of randomization. It allows

everyone in the sample an equal opportunity of being selected through an unbiased process. Non-equal probability sampling places the selection of the units at the discretion of the researcher. This process increases the chance of bias but is preferred when the study population has limited access and there is limited time, money, and labor. Some of the main categories of sample designs that fall under the two main umbrellas are listed in in Table 1. Table 1 displays a generic delineation of the two main categories when the sampling method used is used exclusively. However, in practice, these methods can be combined into a multi-stage sample design and the various subtypes can be used in both equal and non-equal probability sampling.

Equal Probability Sampling	Non-equal Probability Sampling
Simple Random	Convenience
Stratified	Sequential
Cluster	Quota
Systematic	Snowball

Table 1: Sample Design

OVERVIEW OF PROC SURVEYSELECT

The PROC SURVEYSELECT procedure consists of multiple options to construct a sampling method. The two main types of sampling designs discussed previously are supported by the procedure; equal probability sampling has the same name in SAS and non-equal probability sampling is now called proportional probability sampling. Figure 1 shows that both types can be modified to either be a one-stage method or a multiple stage methods. Table 2 presents the different options supported by PROC SURVEYSELECT.

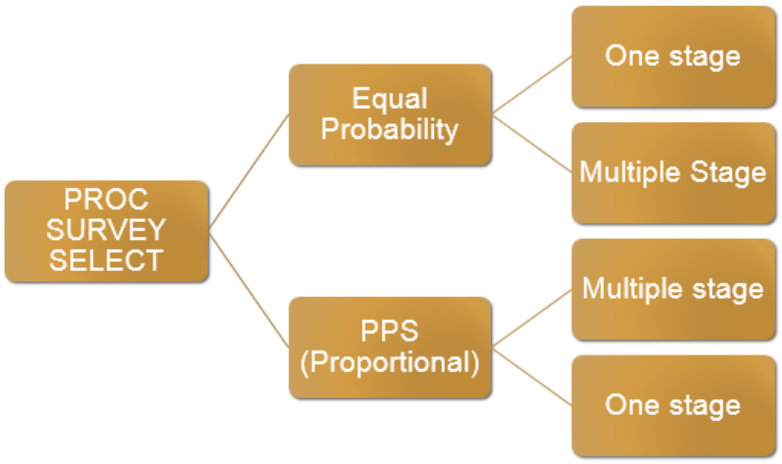

Figure 1: Overview of PROC SURVEYSELECT

Equal Probability Sampling	Proportional Probability Sampling (PPS)
Simple random sampling (without replacement)	Sampling without replacement
Unrestricted random sampling (with replacement)	Sampling with replacement
Systematic random sampling	Systematic
Sequential random sampling	Sequential
	Methods: Brewer's / Murthy's / Sampford's

Table 2: PROC SURVEYSELECT Methods

I will not be able to discuss fully all the available statements and options that PROC SURVEYSELECT has to offer. Some of the procedure statements and options are dependent upon the chosen sampling method. I will only focus on the options that pertains to my examples in the following section.

For the full detail of all the capabilities of this powerful procedure, you can find it on the SAS website at: http://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#surveyselect_toc.htm

Other resources are also listed in the reference section.

BASIC SYNTAX

The following syntax shows the basic SAS code for the procedure. DATA= lets SAS know the name of the input dataset. OUT= lets SAS know the name of the output dataset. METHOD= dictates the type of sampling method. SAMPSIZE= is the sample size. SEED= is a random number entered by the user to regenerate the same sample for further analysis. It is best practice to use the SEED= statement.

```
Proc SurveySelect <options>
Data= <input dataset name>
Out= <output dataset name>
Method= <options>
Sampsize= <number>
Seed= <random number>; Run;
```

THREE EXAMPLES

The following examples are run on a Windows 7 operating system using SAS 9.4. The sample selection consists of units originating from the 2010 National Veterans Survey (NVS). My sampling frame is all 8,710 veterans captured in the survey. It is critical to create a large enough sample size that will yield adequate statistical power for inferential statistics, however because this paper does not delve into any statistics on the generated samples, I use a random sample number for each example. Each example also has a random number seed to ensure that the sample is repeatable. The selected sample will output into a SAS dataset that can then be used for various statistics. I include examples of the following sampling method: SRS, stratified random sampling, PPS systematic sampling. I have provided codes as well as the logs because it is important to always check the log for accuracy.

The following code brings the NVS excel file into the SAS environment and checks for accuracy. I have included explanations of the code as comments in the code. SAS successfully imported the excel file with the expected total of 8,710 observations and 229 variables. The Mylib.NSV2010 SAS dataset was created and modified to only include 9 variables in the Mylib.NSV dataset.

```
/* Importing the excel file  of the survey into the SAS environment */
Proc Import
Datafile= "C:\Users\Djrice\Desktop\Publicdata2010.xls" /*Location of
file */
Dbms = xls /* Type of file */
```

```
Out= Mylib.NSV2010 /* Output data name and location in SAS */
Replace; /* Replace previous file with name */
Getnames= yes; /* Use first row as variable name */
Sheet= "Variables A-N"; /* Sheet name */ Run;
```

```
NOTE: The import data set has 8710 observations and 229 variables.
NOTE: MYLIB.NSV2010 data set was successfully created.
NOTE: PROCEDURE IMPORT used (Total process time):
      real time           1.40 seconds
      cpu time            0.96 seconds
```

Log 1: Import Excel File

```
/* The following codes are housekeeping chores: I am keeping only 9 out
of the 229 variables. I am also checking the contents of the file and
how the characteristics of the variables */
TITLE "2010 Ntl Veterans Survey Data";
Data Mylib.NSV;
Set Mylib.NSV2010 (Keep= ACTEVER ACTLAST HLTH1 EVHZRD ARMY NAVY AIRF
MARINE CGUARD);
Proc Contents Data= Mylib.NSV; Run;
Proc Print Data= Mylib.NSV(Obs=5); Run;
```

```
NOTE: There were 8710 observations read from the data set MYLIB.NSV2010.
NOTE: The data set MYLIB.NSV has 8710 observations and 9 variables.
NOTE: DATA statement used (Total process time):
      real time           0.29 seconds
      cpu time            0.04 seconds

24   Proc Contents Data= Mylib.NSV; Run;
NOTE: Writing HTML Body file: sashtml.htm
NOTE: PROCEDURE CONTENTS used (Total process time):
      real time           1.23 seconds
      cpu time            0.29 seconds

25   Proc Print Data= Mylib.NSV(Obs=5); Run;
NOTE: There were 5 observations read from the data set MYLIB.NSV.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.20 seconds
      cpu time            0.04 seconds
```

Log 2: Data Cleaning

1. Simple Random Sampling (SRS)

Simple random sampling is the simplest sampling method and the default method for PROC SURVEYSELECT. In PROC SURVEYSELECT, it is without replacement. This method has a predetermined sample size that gives each unit in the sampling frame an equal chance of

being selected for the sample, thus there is no replacement. Each option of the syntax of SRS is defined in the code. A code can be written to calculate an appropriate sample size for statistical analysis, but I will use a random number of 1000.

```
/* Simple Random Sampling */
TITLE "2010 National Veterans Survey";
TITLE2 "Simple Random Sampling";
Proc SurveySelect
Data= Mylib.NSV /* Input dataset name */
Out= Mylib.SRS /* Output dataset name */
Method= SRS /* Selection of sampling method */
Sampsize= 500 /* Selection of sample size */
Seed= 13571; Run; /* Selection of random number to duplicate sample */
```

```
NOTE: The data set MYLIB.SRS has 500 observations and 9 variables.
NOTE: PROCEDURE SURVEYSELECT used (Total process time):
      real time           0.28 seconds
      cpu time            0.04 seconds
```

Log 3: Simple Random Sampling

2010 National Veterans Survey
Simple Random Sampling

The SURVEYSELECT Procedure

Selection Method	Simple Random Sampling

Input Data Set	NSV
Random Number Seed	13571
Sample Size	500
Selection Probability	0.057405
Sampling Weight	17.42
Output Data Set	SRS

Output 1: Simple Random Sampling

Output 1 is the results of the SRS code. SAS displays the selection method as well as additional information about the produced sample. The input dataset, random number seed, sample size, and output dataset were all options I defined in the code. The selection probability is product of the ratio between the sample size and the sampling frame: 500/8710 = 0.057405. The

sampling weight of 8.71 is the inverse of the selection probability. We now have a representative sample size of 500 for the veterans in the NVS sampling frame.

2. Stratified Random Sampling = Simple Random Sampling by Strata

Stratified random sampling occurs when the sampling frame is categorized into strata, or groups, and the SRS method is applied to each strata for a sample. It is best practice to first sort the dataset then apply the PROC SURVEYSELECT procedure. To accomplish this, the dataset will first be modified to include the strata of interest then the SRS method will be applied.

The following SAS code will select samples stratified by a newly created variable, branch; it is a discrete variable with the main branches of the military. The new variable branch, from the combination of the 5 military branch variables in the original dataset, is created with the SELECT and WHEN statement. The original dataset had a variable for each of the branch that was not well organized. I dropped the rest of the variables in the dataset to increase the processing time. A format is applied to the branch variable. The dataset is then sorted and finally samples are selected from the different military branches. The STRATA= statement is used to tell SAS which variable to use for the groups and implicitly dictates the sampling method. There is no METHOD= option.

```
/* Stratified Random Sampling */

/* I. Combining 5 variables into 1 new variable, Branch */
Data Branches (Drop= ACTEVER ACTLAST HLTH1 EVHZRD ARMY NAVY AIRF MARINE
CGUARD);
Set Mylib.NSV;
Branch=.;
Label Branch = " Veterans' Military Branch";
Select;
When (ARMY=1) Branch=2;
When (NAVY=1) Branch=3;
When (AIRF=1) Branch=4;
When (MARINE=1) Branch=5;
When (CGUARD=1) Branch=6;
Otherwise Branch=7;
End; Run;
TITLE "2010 National Veterans Survey";
TITLE2 "The New Branch Variable";
Proc Freq Data= Branches;
```

```
Tables Branch; Run;
```

```
NOTE: There were 8710 observations read from the data set MYLIB.NSV.
NOTE: The data set WORK.BRANCHES has 8710 observations and 1 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds

NOTE: There were 8710 observations read from the data set WORK.BRANCHES.
NOTE: PROCEDURE FREQ used (Total process time):
      real time           0.09 seconds
      cpu time            0.01 seconds
```

Log 4: Creation of the Branches Dataset

2010 National Veterans Survey
The New Branch Variable

The FREQ Procedure

Veterans' Military Branch				
Branch	Frequency	Percent	Cumulative Frequency	Cumulative Percent
2	4177	47.96	4177	47.96
3	1919	22.03	6096	69.99
4	1605	18.43	7701	88.42
5	734	8.43	8435	96.84
6	108	1.24	8543	98.08
7	167	1.92	8710	100.00

Output 2: Proc Freq of Branches dataset

```
/* II. Formatting the Branch variable */
Proc Format;
Value Branch
2="Army"
3="Navy"
4="Airforce"
5="Marine"
6="Coast Guard"
7="Other"; Run;
Data Branches;
Set Branches;
Format Branch Branch. ; Run;
TITLE "2010 National Veterans Survey";
TITLE2 "Formatted Branch Variable";
Proc Freq Data= Branches; Tables Branch; Run;
```

```
NOTE: Format BRANCH has been output.

NOTE: PROCEDURE FORMAT used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds

NOTE: There were 8710 observations read from the data set WORK.BRANCHES.
NOTE: The data set WORK.BRANCHES has 8710 observations and 1 variables.
NOTE: DATA statement used (Total process time):
      real time           0.01 seconds
      cpu time            0.00 seconds

NOTE: There were 8710 observations read from the data set WORK.BRANCHES.
NOTE: PROCEDURE FREQ used (Total process time):
      real time           0.09 seconds
      cpu time            0.01 seconds
```

Log 5: Formatting the Branch Variable

Output 3: Formatted Branch Variable

2010 National Veterans Survey
Formatted Branch Variable

The FREQ Procedure

Veterans' Military Branch				
Branch	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Army	4177	47.96	4177	47.96
Navy	1919	22.03	6096	69.99
Airforce	1605	18.43	7701	88.42
Marine	734	8.43	8435	96.84
Coast Guard	108	1.24	8543	98.08
Other	167	1.92	8710	100.00

```
/* III. Sorting the dataset*/
Proc Sort
Data= Branches;
By Branch; Run;
TITLE "2010 National Veterans Survey";
TITLE2 "Sorted Branch Variable";
```

```
Proc Print Data= Branches (Obs=5); Run;
```

```
NOTE: There were 8710 observations read from the data set WORK.BRANCHES.
NOTE: The data set WORK.BRANCHES has 8710 observations and 1 variables.
NOTE: PROCEDURE SORT used (Total process time):
      real time          0.15 seconds
      cpu time           0.01 seconds

NOTE: There were 5 observations read from the data set WORK.BRANCHES.
NOTE: PROCEDURE PRINT used (Total process time):
      real time          0.06 seconds
      cpu time           0.01 seconds
```

```
/* IV. Stratified sampling on the dataset */
TITLE "2010 National Veterans Survey";
TITLE2 "Stratified Branch Variable";
Proc SurveySelect
Data= Branches
Out= Mylib.Bybranch
Sampsize= 50
Seed= 23462;
Strata Branch; Run; /* Selection of samples by the variable Branch */
Proc Print Data= Mylib.Bybranch (Obs= 5); Run;
```

```
NOTE: The data set MYLIB.BYBRANCH has 300 observations and 3 variables.
NOTE: PROCEDURE SURVEYSELECT used (Total process time):
      real time          0.15 seconds
      cpu time           0.04 seconds

NOTE: There were 5 observations read from the data set MYLIB.BYBRANCH.
NOTE: PROCEDURE PRINT used (Total process time):
      real time          0.06 seconds
      cpu time           0.01 seconds
```

2010 National Veterans Survey
Stratified Branch Variable

The SURVEYSELECT Procedure

Selection Method	Simple Random Sampling
Strata Variable	Branch

Input Data Set	BRANCHES
Random Number Seed	23462
Stratum Sample Size	50
Number of Strata	6
Total Sample Size	300
Output Data Set	BYBRANCH

2010 National Veterans Survey
Stratified Branch Variable

Obs	Branch	SelectionProb	SamplingWeight
1	Army	0.011970	83.54
2	Army	0.011970	83.54
3	Army	0.011970	83.54
4	Army	0.011970	83.54
5	Army	0.011970	83.54

3. PPS SYSTEMATIC SAMPLING

PPS systematic sampling is a non-equal probability method that selects units at a regular interval point throughout the sampling frame. The method is selected by the PPS_SYS option in METHOD=. The ID statement specifies which variables to include in the output data. Due to the non-equal probability method, the SIZE statement stipulates which numeric variable to add weight to when selecting the units in the sample. It also excludes any missing variables from the sample. The hierarchy of the observations in the SIZE variable defines the non-equal selection of the units.

```
/* PPS_Systematic Sampling */
TITLE "2010 National Veterans Survey";
TITLE2 "PPS Systematic Sampling";
Proc SurveySelect
Data= Mylib.NSV
Out= Mylib.PPSsys
Method= PPS_SYS
Sampsize= 400
Seed= 38749;
ID EVHZRD;
Size EVHZRD; Run;
Proc Print Data= Mylib.PPSsys (Obs= 5); Run;
```

```
NOTE: 76 sampling units were omitted due to missing or nonpositive size measures.
NOTE: The data set MYLIB.PPSSYS has 400 observations and 4 variables.
NOTE: PROCEDURE SURVEYSELECT used (Total process time):
      real time           0.09 seconds
      cpu time            0.01 seconds


 NOTE: There were 5 observations read from the data set MYLIB.PPSSYS.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.09 seconds
      cpu time            0.00 seconds
```

Log 6: PPS Systematic Sampling

2010 National Veterans Survey
PPS Systematic Sampling

The SURVEYSELECT Procedure

Selection Method	PPS Systematic
Size Measure	EVHZRD

Input Data Set	NSV
Random Number Seed	38749
Sample Size	400
Output Data Set	PPSSYS

2010 National Veterans Survey
PPS Systematic Sampling

Obs	EVHZRD	NumberHits	ExpectedHits	SamplingWeight
1	8	1	0.094826	10.5456
2	2	1	0.023707	42.1825
3	3	1	0.035560	28.1217
4	2	1	0.023707	42.1825
5	4	1	0.047413	21.0913

Output 4: PPS Systematic

CONCLUSION

This paper only touches the surface of the vast power of the PROC SURVEYSELECT procedure. It is crucial to define the purpose of the study and to understand sampling methods before using the procedure.

CONTACT INFORMATION

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

Name: Raissa Kouadjo Bordenave
Enterprise: Florida A&M University
E-mail: raissa1.kouadjo@famu.edu