Paper 3240-2015

Sampling Financial Records Using SurveySelect

Roger L. Goodwin, US Government Printing Office

ABSTRACT

This paper presents an application of the procedure SurveySelect ®. The objective is to draw a systematic random sample from financial data for review. Topics covered in this paper include a brief review of systematic sampling, variable definitions, serpentine sorting, and an interpretation of the output.

INTRODUCTION

[Sarndal, Swensson, and Wretman (2), pages 73-83] define the following notation for circular, systematic sampling. The integer *N* is the population size. The integer *n* such that $n \leq N$ is the sample size. The integer *r* such that $1 \leq r \leq N$ is called the random start. The probability of selecting any element in the sample is $\pi = \frac{n}{N}$.

The circular, systematic random sample selection procedure is as follow:

1. Select the random start *r* from 1 to *N.*

2. The selected sample for $j = 1, \ldots, n$ for periodic $a = \frac{N}{n}$ is

$$k = \begin{cases} r + (j-1)a, & If\ r + (j-1)a \leq N. \\ r + (j-1)a - N, & If\ r + (j-1)a > N. \end{cases}$$

In case the value of *a* is a real number, choose *a* such that it is the integer closest to $\frac{N}{n}$.

SAS Institute ® provides a canned procedure named SurveySelect to perform sample selection. We will present the parameters to the procedure for performing circular, systematic random sampling on financial records. Prior to running the SurveySelect procedure, the procedure needs data steps to read the data for sampling. Figure 1 shows the flow chart of the SAS ® code.

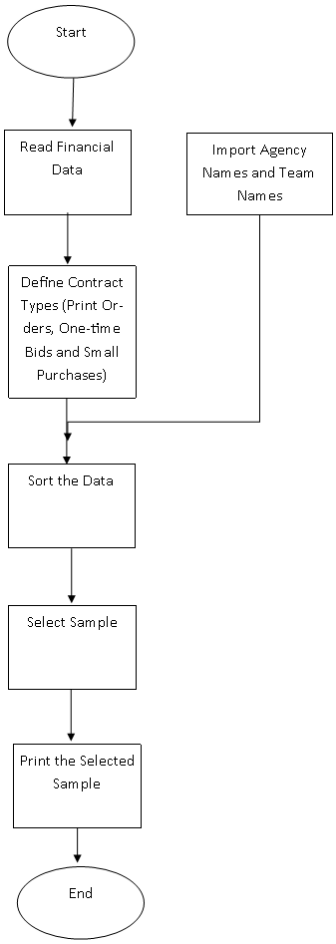


Figure 1. This figure shows the flowchart of the SAS code.

EXAMPLE OF SAMPLE SELECTION

Hypothetically, suppose we have a population size of $N = 65$. Suppose we wish to draw a circular, systematic random sample of size $n = 10$. We can deduce that $a = \frac{N}{n} = \frac{65}{10} = 6.5$. We choose the integer nearest to $\frac{N}{n}$. Rounding up, we obtain $a = 7$. Further suppose that the random start in the list is $r = 12$. Then, Table 1 summarizes the selection.

Notice that on $j = 9$ and $j = 10$, the algorithm looped back to the front of the list to choose elements $k=3$ and $k=10$, respectively.

Table 1: This table shows the sample selection algorithm performed on 65 elements. Column k shows the selected sample element.

Index j	Population Size N	Periodic a	Random Start r	Element Selected k
1	**65**	**7**	**12**	**12**
2	**65**	**7**	**12**	**19**
3	**65**	**7**	**12**	**26**
4	**65**	**7**	**12**	**33**
5	**65**	**7**	**12**	**40**
6	**65**	**7**	**12**	**47**
7	**65**	**7**	**12**	**54**
8	**65**	**7**	**12**	**61**
9	**65**	**7**	**12**	**3**
10	**65**	**7**	**12**	**10**

DATA STEPS

We define the following variables. We use these variables to define other variables and as input parameters to the SurveySelect procedure. These variables and fields reference the historical Voucher Payment and Processing System files (VOPPS) [US Government Printing Office (3)]. This system stored transactions on payments to contractors whom the US Government Printing Office (GPO) hired for procurement work for Federal agencies. Later, GPO modernized the system to Oracle and SAP Business Objects.

- VOPPS jacket number and print order number --- Both of these variables together identify a procured job. The procured job can be printed work, CD and DVD reproduction, promotional items such as magnets, and so on.
- Fiscal Year --- The Fiscal Year can refer to three different years: 1) the Fiscal Year of the jacket number, 2) the Fiscal Year the contractor delivered the product, and 3) the Fiscal Year GPO paid the contractor.
- Award amount --- This is the amount awarded to the contractor. It does not include contract modifications (stored in different fields on the VOPPS file). Total award amount must be calculated.

A VOPPS file does not contain the Customer Services Team (or team number) that facilitated the procurement. The information comes from Customer Services. We use existing variables to define the three contract types:

- A print order --- The print order field does not equal to '00000'.
- A one-time bid --- The print order field does equal to '00000' and the award price is greater than or equal to $100,000.

- A small purchase --- The print order field does equal to '00000' and the award price is less than $100,000.

We used the phrase *print order* in two different ways. We used *print order* once as a variable and once as the size of a contract. In other data steps, we select a particular team and the team's particular programs for the given Fiscal Year 2009.

SYNTAX OF THE SURVEYSELECT PROCEDURE

This section discusses some of the syntax of the SurveySelect procedure in SAS. According to the SAS manual [SAS Institute (1)], the SurveySelect procedure provides a variety of methods for selecting probability-based random samples.

PROC SURVEYSELECT options;
STRATA variables </ options>;
SAMPLINGUNIT | CLUSTER variables </ options>;
CONTROL variables;
SIZE variable;
ID variables;

SURVEYSELECT STATEMENT

The options on the SurveySelect line allow the programmer to set most of the parameters to the SurveySelect procedure.

- Declare the name of the input file using the DATA option.
- Declare the sampling method using the METHOD option.
- Declare the sample size using the N option.
- Declare the output file name using the OUT option.

SERPENTINE SORTING

The SAS software defaults to serpentine sort for systematic sampling. Serpentine sorting applies to data ordered by more than one variable. This application uses two variables in the serpentine sort. A serpentine sort orders the first named variable in ascending order. For each of the first variable values, the order of the second variable is in descending order. For this application, the serpentine sort ordered the print order numbers in descending order for each VOPPS jacket number. The serpentine sort ordered the VOPPS jacket number in ascending order. To accomplish this sorting SAS provides the CONTROL statement with the SurveySelect procedure. The CONTROL statement names variables for sorting on the input data set before sample selection [SAS Institute (1)]. The programmer can name multiple variables on the CONTROL line. The CONTROL statement does not have any options.

Tip: The SORT ® procedure immediately before the SurveySelect procedure may be unnecessary. SAS sorts the data in serpentine order because of the CONTROL statement.

RECOVERING A PRIOR DRAWN SAMPLE

It is possible to recover the exact sample drawn as long as the input file and the sort arrangement have not changed. The SurveySelect statement has a SEED option. The SAS manual [SAS Institute, (1)] states that if you need to reproduce the same sample in a subsequent execution of Proc SurveySelect, you can. Specify the same seed value using the SEED= option, along with the same sample selection parameters. The Proc SurveySelect will reproduce the sample.

SAMPLING FINANCIAL RECORDS

The purpose of this code is to draw a sample of contracts from each team for review. In the system of record, to retrieve a commercial contract, we only need the jacket number and the print order number. We ran one SurveySelect procedure (three in all) for each of the contract sizes: 1) print orders, 2) one-time bids, and 3) small purchases. Some teams only have multi-year contracts. Therefore, obtaining the same sample size for each contract size for each team was not an objective. We use the following SAS code to draw a sample of financial records. Figure 2 shows the summary information from SurveySelect.

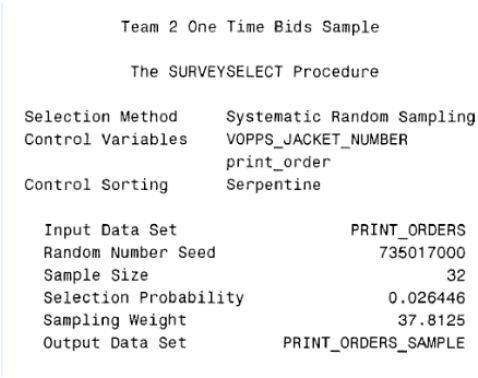
```
/* select the samples */
proc sort data = print_orders; by vopps_jacket_number print_order; run;
data print_orders;
set print_orders;
by vopps_jacket_number print_order;
if first.print_order;
run;

Proc surveyselect data = print_orders method = sys n = 32 out=print_orders_sample;
Control vopps_jacket_number print_order;
Run;
```

INTERPRETING THE OUTPUT

Figure 2 shows the summary information for a single run of SurveySelect. The first five lines repeat the parameters given to the procedure. We declared the selection method as systematic random sampling using the METHOD = SYS option. We declared the control variables VOPPS_JACKET_NUMBER and PRINT_ORDER using the CONTROL statement. SAS defaults the control sorting to serpentine. Finally, we defined the input data set using the option DATA = PRINT_ORDERS.

SAS generated the random number seed and printed it. As discussed in Section 4.3, this number may be useful later. The SAS user sets the sample size *n*. We set *n = 32* because this is the magic number most textbook tables define infinity.

The selection probability of $\pi = 0.026446$ is a calculation. As stated in Section 1, it is the sample size *n* divided by the population size *N*. We have $\pi = \frac{n}{N} = \frac{32}{1210} = 0.026446$. The sampling weight is equal to $\frac{1}{\pi} = \frac{1}{0.026446} = 37.8125$. Lastly, SAS prints the name of the output data set.



Team 2 One Time Bids Sample

The SURVEYSELECT Procedure

Selection Method	Systematic Random Sampling
Control Variables	VOPPS_JACKET_NUMBER print_order
Control Sorting	Serpentine

Input Data Set	PRINT_ORDERS
Random Number Seed	735017000
Sample Size	32
Selection Probability	0.026446
Sampling Weight	37.8125
Output Data Set	PRINT_ORDERS_SAMPLE

Figure 2: This figure shows summary information from SurveySelect.

The SurveySelect procedure generated the high-level summary statistics and the samples. To add additional variables, we used the procedure PRINT ® for the final samples.

The selected samples list the following variables.

- VOPPS Jacket number and print order number
- VOPPS Program number
- Payment date

For the small purchases and one-time bids, the program number column was blank. Additionally, for small purchases and one-time bids, the print order number column contains zeros. This is consistent with the definitions in Section 3.

REFERENCES

1. SAS Institute, *SAS/STAT(R) 9.3 User's Guide, SAS Stat*, Carey, North Carolina, July 2011.
2. C. E. Sarndal, B. Swensson, and J. Wretman, *Model Assisted Survey Sampling,* Springer-Verlang, New York, Inc., New York, pp 73-83, 1992.
3. US Government Printing Office, *Printing Procurement Regulation,* May 1999, page VI-6.
4. US Government Printing Office, *Printing Procurement Regulation, Chapter XIII. Contract Administration And Compliance,* April 2014.

ACKNOWLEDGMENTS

The author thanks Paul Giannini for showing the relationships between the fields of the historical files and defining the contract size.

CONTACT INFORMATION

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

Affiliate:

US Government Printing Office
Roger L Goodwin
Email: rgoodwin AT gpo DOT gov
732 North Capitol Street NW
Room C-628
Washington DC 20401

Roger L Goodwin
Email: rogergoodwin AT dcsug DOT org
1989 Leetown Road
Summit Point, WV 25446

ABOUT THE AUTHOR

Roger Goodwin has 15 years' experience with several government agencies. Two of the agencies are statistical in nature; the third agency is both production and commercial in nature. Roger Goodwin completed statistical assignments on many computer platforms, which include PCs, Vax VMS OS, Unix OS, and IBM mainframes. He usually performs his statistical analyses in SAS and uses Excel for simpler calculations. He developed reports for cost, progress, and billing reports using SAS and SAP Business Objects.

Roger Goodwin holds a BS in Computer Science and an MS in Applied Statistics from Old Dominion University. He completed a certificate in Software Engineering Processes from Learning Tree. He completed the Project Management Professional certification from PMI. He authored several papers in IEEE conferences and two online journals that summarize his experiences in government. He authored papers in the North East SAS User’s Group that describes some of the SAS code that he wrote.