Random Sample Selection

Imelda C. Go, Richland County School District One, Columbia, SC

ABSTRACT

SAS® functions and procedures can be used in a variety of ways to select random samples. The paper is a sampler of SAS code used to select different types of random samples. The types of random sampling methods exemplified include simple (without replacement), with replacement, systematic, stratified, and proportional stratification.

INTRODUCTION

One of the sampling strategies used in this paper is explained by the following example. Given a well-shuffled deck of 52 playing cards, the goal is to randomly select one card. Selecting any card in the well-shuffled deck of cards yields a randomly selected card. For example, if the rule is to select the 34th card in the deck, then the 34th card (whatever it might be) is the randomly selected card. Whether the 1st, 2nd, …, or 52nd card is selected; the resulting card is the randomly selected card. The selection process is random because the well-shuffled cards are in random order.

What if the goal is to randomly select two cards? Given a well-shuffled deck of cards, selecting any pair of cards from the deck yields a random sample of two cards. So if the rule is to select the first two cards in a well-shuffled deck of cards, the first two cards are the randomly selected cards.

In this paper, the records being considered for random selection are *shuffled* by using the RANUNI function to assign a random number to each record. The RANUNI function generates a random number from a continuous uniform distribution (the interval (0, 1)). (Some statistical discussion is omitted here.) Each random number generated has the same chance of being the largest, 2nd largest, …, or smallest among the random numbers generated. For simplicity, assume that none of the random numbers are duplicated.

By assigning a random number to each record, the records can then be sorted in increasing or decreasing order of the random numbers. For this paper, all sorting will be done in increasing order. *Shuffling* or rearranging a data set's records in this manner is analogous to shuffling the deck of cards in the above discussion.

NOTE ON SEEDS

The value generated by the RANUNI function depends on a seed. The seed should be a nonnegative integer from 1 to 2,147,483,646 in order to replicate the results of the RANUNI function. That is, given the same seed, the function produces the same result. If no seed, zero, or negative integers are specified as the seed, the computer clock sets the seed and results are not replicable.

NOTE ON SELECTION BIAS

Good random samples are representative of the population from which they are drawn from. Therefore, it is important to avoid selection bias during the sampling process. One source of selection bias is the order in which the records occur in a data set. For example, the goal is to select 50 students from a data set. The first 50 students in a data set of 10,000 may not always be the best sample because the first 50 students might all come from the same school (if the data from one school are in consecutive records and are followed by the data from another school).

EXAMPLES

Simple random sampling (without replacement)

Goal: Randomly select 10 numbers from numbers 1 to 100 (1, 2, …, 100) without replacement.
Process: Let data set `hundred` contain the 100 numbers. For the purpose of providing an example, the variable `seed` is also on the data set `hundred`. Assign a RANUNI function-generated random number (`shuffling`) to each record. Sort the records by increasing `shuffling` order. Select the first 10 records to be the sample.

```
data hundred;
  set hundred;
  shuffling=ranuni(seed);

proc sort data=hundred;
  by shuffling;

data sample;
  set hundred (obs=10);
```

Random sampling with replacement

Goal: Randomly select 2 numbers from numbers 1 to 100 (1, 2, …, 100) with replacement.
Process: Let data set `hundred` contain the 100 numbers. Assign two RANUNI function-generated random numbers (`shuffling1` and `shuffling2`) to each record. Sort the records in increasing `shuffling1` order. Select the first record to be the first selection in the sample. Sort the original records in increasing `shuffling2` order. Select the first record to be the second selection in the sample.

```
data hundred;
  set hundred;
  shuffling1=ranuni(seed1);
  shuffling2=ranuni(seed2);

proc sort data=hundred;
  by shuffling1;

data firstselection;
  set hundred (obs=1);

proc sort data=hundred;
  by shuffling2;

data secondselection;
  set hundred (obs=1);
```

Systematic random sampling

Goal: Randomly select 10 students from a data set of 200 records.
Process: 200÷10=20. Select every 20th record in the data set. To prevent selection bias given the order of the records on the data set, the records may be sorted by name prior to random selection.

```
proc sort data=students;
  by name;

data sample; set students;
  if mod(_n_,20)=0;
```

The MOD function returns the remainder that results from dividing the first argument (_n_) by the second argument (20). The _n_ automatic variable represents the number of times the DATA step has iterated. If each iteration involves only one record, _n_ can be used to indicate the ordinal position of the record within the data set. For example, _n_=1 for the 1st record in a data set. In general, _n_=i for the ith record in a data set.

Stratified random sampling

Goal: Randomly select 10 students from each age and sex group.
Process: Assign a RANUNI function-generated random number (shuffling) to each record. Sort the records by shuffling within each age and sex combination. Select the first 10 records from each age and sex combination to be part of the sample.

```
data students;
  set students;
  shuffling=ranuni(seed);

proc sort data=students;
  by age sex shuffling;

data sample;
  retain counter;
  set students;
  by age sex;
  if first.sex
    then counter=1;
    else counter=counter+1;
  if counter<=10;
```

Proportional stratification

Goal: Randomly select half of the males and half of the females in a data set of student records.
Process: To prevent selection bias given the order of the records on the data set, the records may be sorted by sex and by name prior to random selection. Select every other student (i.e., the first of every two) in each category of sex.

```
proc sort data=students;
  by sex name;

data malesample;
  set students;
  if sex='m' and mod(_n_,2)=1;

data femalesample;
  set students;
  if sex='f' and mod(_n_,2)=1;
```

Mod(_n_,2)=0 whenever _n_ is an even number and mod(_n_,2)=1 whenever _n_ is an odd number.

Goal: Randomly select 1/3 of the males and 2/3 of the females in a data set of student records.
Process: To prevent selection bias given the order of the records on the data set, the records may be sorted by sex and by name prior to random selection. For the males, one of every trio is selected. For the females, two of every trio are selected.

```
proc sort data=students;
  by sex name;

data malesample;
  set students;
  if sex='m' and mod(_n_,3)=1;

data femalesample;
  set students;
  if sex='f' and mod(_n_,3) in (1,2);
```

Mod(_n_,3)=0 whenever _n_ Is perfectly divisible by 3 (i.e., _n_ is 3, 6, 9, …); mod(_n_,3)=1 when _n_ is 1, 4, 7, …; and mod(_n_,3)=2 when _n_ is 2, 5, 8, … .

CONCLUSION

SAS can be used to facilitate the random sample selection process. It is important to note sources of selection bias in order to yield a sample that is representative of the population from which it was drawn from.

REFERENCES

SAS Institute Inc., *SAS® Language Reference, Version 8*, Cary, NC: SAS Institute Inc., 1999. 1256 pp.

SAS Institute Inc., *SAS OnLineDoc®, Version 8*, Cary, NC: SAS Institute Inc., 1999.

TRADEMARK NOTICE

Imelda C. Go
Office of Research and Evaluation
Richland County School District One
1616 Richland St.
Columbia, SC 29201

Tel.: (803) 733-6079
Fax: (803) 929-3873
icgo@juno.com