1. Introduction

Because of the CPU time and disk I/O required to convert a very large raw data file into a SAS data set, it is sometimes impractical - or impossible - to do so on a limited or highly utilized system. PROC RANDOM and PROC RANSTRAT provide simple mechanisms for choosing either a simple random sample (RANDOM) or a stratified random sample using one stratification variable (RANSTRAT) of unwieldy data sets. The raw data samples can be stored in temporary data sets, or they could be stored permanently on either tape or mass storage. The sample raw data files can then be read into a SAS program using the usual INFILE and INPUT statements. All processing after the DATA step is customary.

Please note that PROC RANDOM and PROC RANSTRAT are designed only for use with raw data sets, not on SAS data sets.

2. Input Data Set Specifications

The user must provide the necessary information to access the data set to be sampled using JCL (MVS) or the appropriate FILEDEF (CMS) statements. The number of lines per observation must be provided in the procedures with default to one line per observation if not specified. Variable numbers of lines per observation are not provided for. The record length is determined by the procedures from the JCL, system, or data set labels.

3. Generating Pseudo-Random Numbers

Both PROC RANDOM and PROC RANSTRAT rely on the generation of pseudo-random numbers generated by the multiplicative-congruential method which is defined by:

\[ x_i = (ax_{i-1}) \mod M \]

for \( i = 1, 2, 3, ... \)

where \( x_1, a, \) and \( M \) are integers and

\[ 0 \leq x_i \leq M. \]

For the purposes of the programs

\[ M = 2^{31} - 1. \]

For the multiplicative congruential method with modulus \( M \) and multiplier \( a \), the period is the smallest integer \( V \) for which

\[ a^V = 1 \mod M. \]

The value \( x_0 \) is the initial value in the generation program, and it is referred to as the 'seed' of the generation process. By defining \( x_0 \), it is possible to reproduce the same pseudo-random numbers, and therefore, the same random sample if desired.

The integers produced are in the interval \((0,M)\). They are transformed by \( x_i / M \) into \((0,1)\) over which they approximate a \( U(0,1) \) process. To transform the \( U(0,1) \) deviates to discrete uniform \((0,N)\) where \( N \) is the number of observations in the original data set, multiply by \( N \), add 1 to the result, and truncate the result to an integer.

The pseudo-random deviates are the observation numbers of those observations to be kept in PROC RANDOM. In PROC RANSTRAT, the deviates serve as pointers to the observation numbers of those observations to be kept for the stratified random sample.

4. PROC RANDOM

The statement to control PROC RANDOM is:

PROC RANDOM options;

The following options are available:

INDD=ddname
specifies the ddname of the file to be sampled. It must be a valid ddname. You must include this option.

OUTDD=ddname
specifies the ddname of the output file to which the sample is sent. It must be a valid ddname. You must include this option.

SEED=integer
specifies the initial value used in the generation of the discrete uniform random deviates. It must be an integer value in the range 1 to 2,147,483,647. The default is 11.

PERCENT=number
specifies the percentage of the input file to be kept as a sample. This percentage must be in the range 1 to 100 with no more than two (2) decimal places. The default is 10.

NUMBER=integer
specifies the number of input records that make up a single observation. You use this option only when the input file has more than one record per observation. The integer must be positive. The default is 1.
Printed output consists of the number of input records read, the number of observations read, and the number of output records written to the new sample data set. Also, the final seed for the generation of deviates is printed to allow a continuation of the sample in a later run if necessary.

A warning is printed if you do not specify OUTDD or SEED. A note is printed if you do not specify PERCENT or NUMBER. In all cases where you omit optional parameter values, the default value is printed.

5. PROC RANDOM Example

The following program takes a five (5) percent random sample of the South Carolina publicly accessible vital statistics birth/infant death cohort for the years 1975 through 1980. The output gives the MEAN, STANDARD DEVIATION, and MINIMUM of both the sample and the population data for comparison. Note: This example assumes that you are running under MVS. For CMS users, replace the JCL statements with the appropriate commands and filedef statements.

```
// RANDOM JOB...
// EXEC SAS...
// INDD DD DSN=COHORT.YR75,DISP=SHR
// DD DSN=COHORT.YR76,DISP=SHR
// DD DSN=COHORT.YR77,DISP=SHR
// DD DSN=COHORT.YR78,DISP=SHR
// DD DSN=COHORT.YR79,DISP=SHR
// DD DSN=COHORT.YR80,DISP=SHR
// OUTDD DD DSN=&&TEMP,UNIT=SYSDA,
// SPACE=(TRK,(200,10),
// DCB=(LRECL=198,BLKSIZE=12870),
// DISP=(NEW,PASS)
// SYSIN DD *
// OPTIONS NOCENTER;

PROC RANDOM INDD=INDD
   OUTDD=OUTDD
   SEED=2144033
   PERCENT='5;

DATA SAMPLE;
   INPUT COUNTY 5-6 MAGE 43-44
   F_AGE 72-73 WEIGHT 115-118;
   PROC MEANS N MEAN STD MIN MAX DEC=3;
   TITLE 'STATISTICS FOR A 5% SAMPLE';
   VAR M_AGE F_AGE WEIGHT;

DATA WHOLE;
   INPUT COUNTY 5-6 MAGE 43-44
   F_AGE 72-73 WEIGHT 115-118;
   PROC MEANS N MEAN STD MIN MAX DEC=3;
   TITLE 'STATISTICS FOR POPULATION';
   VAR M_AGE F_AGE WEIGHT;

The output follows:
```

```

STATISTICS FOR 5% SAMPLE

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>N</th>
<th>MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_AGE</td>
<td>14973</td>
<td>23.987</td>
</tr>
<tr>
<td>F_AGE</td>
<td>13342</td>
<td>35.796</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>14973</td>
<td>3263.055</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VARIABLE STANDARD DEVIATION VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_AGE</td>
</tr>
<tr>
<td>F_AGE</td>
</tr>
<tr>
<td>WEIGHT</td>
</tr>
</tbody>
</table>

STATISTICS FOR POPULATION

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>N</th>
<th>MEAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_AGE</td>
<td>299455</td>
<td>24.006</td>
</tr>
<tr>
<td>F_AGE</td>
<td>266722</td>
<td>35.535</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>299454</td>
<td>3268.283</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VARIABLE STANDARD DEVIATION VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>M_AGE</td>
</tr>
<tr>
<td>F_AGE</td>
</tr>
<tr>
<td>WEIGHT</td>
</tr>
</tbody>
</table>
```

6. PROC RANSTRAT

The main difference between PROC RANDOM and PROC RANSTRAT is the specification of a stratification variable in RANSTRAT and the ability to specify the number of observations or a particular percentage sample depending on the value of the stratification variable.

The statement to control PROC RANSTRAT is:

```
PROC RANSTRAT options;
```

The following options are available:

- **INDD=ddname**
  Specifies the ddname of the file to be sampled. It must be a valid ddname. You must include this option.

- **OUTDD=ddname**
  Specifies the ddname of the output file to which the sample is sent. It must be a valid ddname. The default is OUTDD.

- **SEED=integer**
  Specifies the beginning value used in the generation of the discrete uniform random deviates. It must be an integer value in the range 1 to 2,147,483,647. The default is 1.

- **STRATA='start1-end1,start2-end2,...'**
  Specifies the columns in which the values that are to be concatenated to form the stratification variable are located. Up to 20 ranges may be specified. For example, 'STRATA='17-24,39-41'". A single column is specified as one number. e.g. STRATA='5'. The columns indicated are read as character values.
VALUE count='size,value'

count - specifies the number of the strata. It is indicated as 1, 2, ...

type - specifies either the number of observations from the indicated strata to be in the sample, or the percent of the observations in the strata to be in the sample. It followed with a '%', that percentage of the strata is sampled. Otherwise, that number of records indicated is sampled from the strata.

value - specifies the strata values to be sampled by the indicated size specification. Value may be a single string of characters, a range of values, or a list of values. Because the stratification values are characters, the length of the strings must match the length of the stratification variable exactly. The syntax for a single string is

VALUE count='size,value'

For a range, the syntax is

VALUE count='size,(value1-value2)'

For a list of values, the syntax is

VALUE count='size,value1,value2,....'

It is possible to choose a percent sample of all other stratification values not indicated explicitly, using the REST option. The syntax is

VALUE count='size,(REST)'

NUMBER=integer

specifies the number of input records that make up a single observation. You use this option only when the input file has more than one record per observation. The integer must be positive. The default is 1.

Printed output consists of the number of records indicated, the number of observations read, and the number of output records written to the new sample data set. Also, the final seed for the generation of deviates is printed to allow a continuation of the sample in a later run if necessary.

A warning is printed if you do not specify OUTDD or SEED. A note is printed if you do not specify PERCENT or NUMBER. In all cases where you omit optional parameter values, the default value is printed.

7. PROC RANSTRAT Example

The following program requests a 100 percent sample of the stratification values between '740' and '7499' and a 1 percent sample of the observations with any other stratification variable value. The data set sampled is the South Carolina publicly accessible vital statistics live birth/infant death cohort for the years 1975 through 1980. The output gives the MEAN, STANDARD DEVIATION, and MINIMUM of both the stratified sample and the stratified population for comparison.

//RANSTRAT JOB ...
// EXEC SAS
// INDD DD DSN=COHORT.YR75,DISP=SHR
// INDD DD DSN=COHORT.YR76,DISP=SHR
// INDD DD DSN=COHORT.YR77,DISP=SHR
// INDD DD DSN=COHORT.YR78,DISP=SHR
// INDD DD DSN=COHORT.YR79,DISP=SHR
// INDD DD DSN=COHORT.YR80,DISP=SHR
//OUTDD DD DSN=&STMP,UNIT=SYSDA,
// SPACE=(TRK,(200,10)),
// DCB=(LRECL=198, BLKSIZE=12870),
// DISP=(NEW,PASS)
//SYSIN DD *
OPTIONS NOCENTER;
PROC RANSTRAT
INDD=INDD
OUTDD=OUTDD
SEED=1747787
STRATA='125-128'
VALUE1='100%,(740 -7499)' VALUE2='1%,(REST)';
DATA RANSTRAT;
INFILE OUTDD;
INPUT M AGE 43-44 F AGE 72-73
WEIGHT 115-118 CONGMAL $ 125-128;
IF '740 '(sCONGMAL<='7499'
THEN STRATA=1;
ELSE STRATA=2;
PROC SORT;
BY STRATA;
PROC MEANS N MEAN STD MIN MAXDEC=3;
TITLE 'STATISTICS FROM RANSTRAT';
VAR M AGE F AGE WEIGHT;
BY STRATA;
DATA WHOLE;
INFILE INDD;
INPUT M AGE 43-44 F AGE 72-73
WEIGHT 115-118 CONGMAL $ 125-128;
IF '740 '(sCONGMAL<='7499'
THEN STRATA=1;
ELSE STRATA=2;
PROC SORT;
BY STRATA;
PROC MEANS N MEAN STD MIN MAXDEC=3;
TITLE 'STATISTICS FROM POPULATION';
VAR M AGE F AGE WEIGHT;
BY STRATA;
The output follows:
**Statistics from Ranstrat**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Strata</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>M AGE</td>
<td>672</td>
</tr>
<tr>
<td>F AGE</td>
<td>613</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>672</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mean</th>
<th>24.528</th>
<th>24.047</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>33.692</td>
<td>36.021</td>
</tr>
<tr>
<td></td>
<td>3105.811</td>
<td>3274.436</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Standard Deviation</th>
<th>Minimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>M AGE</td>
<td>5.405</td>
</tr>
<tr>
<td>F AGE</td>
<td>20.759</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>1032.574</td>
</tr>
</tbody>
</table>

**Statistics for Population**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Strata</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>M AGE</td>
<td>672</td>
</tr>
<tr>
<td>F AGE</td>
<td>613</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>672</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mean</th>
<th>24.528</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>33.692</td>
</tr>
<tr>
<td></td>
<td>3105.811</td>
</tr>
</tbody>
</table>

**Notes:**
- *SAS is the registered trademark of SAS Institute Inc., Cary, NC, USA.*