SUGI 28 Coders' Corner

Paper 75-28

Splitting a Large SAS® Data Set

Selvaratnam Sridharma, Census Bureau, Washington, D.C.

Abstract

Sometimes it is necessary to split a large data set into smaller, more manageable data sets. Here we show how to split a large data set into smaller sized data sets. The number of observations in each smaller sized data set will be equal to a given number except possibly for one smaller sized data set: this might have smaller number of observations than the given number

The %split1 Macro

For a given number **n**, the **%split1** macro, given below, will split a large dataset into smaller number of datasets with **n** number of observations except possibly for one smaller dataset: this one might have less than **n** number of observations. First we find the number of observations in the large data set . Then divide this number by the given number **n** to determine the number of smaller datasets needed. Then we use a simple data step to split the large dataset into smaller data sets.

```
%macro split1(num);

data _null_;
  if 0 then set orig nobs=count;
  call symput('numobs',put(count,8.));
run;

%let m=%sysevalf(&numobs/&num,ceil);

data %do J=1 %to &m; orig_&J %end; ;
  set orig;

%do I=1 %to &m;
```

The %split2 Macro

For a given number **n**, the **%split2** macro, given below, will split a large dataset into **n** smaller datasets. All the smaller data sets will have equal number of observations except possibly one smaller data set which might have smaller number of observations. First we find the number of observations in the large data set. Then divide this number by the given number **n** to determine the number of observations for the smaller data sets. Then we use a simple data step to split the large dataset into smaller data sets.

```
%macro split2(num);

data _null_;
  if 0 then set orig nobs=count;
  call symput('numobs',put(count,8.));
run;

%let n=%sysevalf(&numobs/&num,ceil);
data %do J=1 %to # orig_&J %end; ;
set orig;

%do I=1 %to #

if %eval(&n*(&i-1)) <_n_ <= %eval(&n*&I)</pre>
```

SUGI 28 Coders' Corner

```
then output orig_&I;
%end;
run;
%mend split2;
```

Examples

Example1

The following simple example shows how the %split1 macro works.

```
data orig;
  do i =1 to 84;
   output;
  end;
run;
%split1(18);
```

Here the dataset **orig** will be split into smaller datasets **orig_1**, **orig_2**, **orig_3**, **orig_4** and **orig_5**. The first four will have 18 observations each and the last one will have 12 observations.

Example2

```
data orig;
  do i =1 to 85;
   output;
  end;
run;
%split2(6);
```

Here the dataset orig will be split into smaller datasets orig_1, orig_2, orig_3, orig_4, orig_5, and orig_6. The first five will have 15 observations and the last one will have 10 observations.

Conclusion

The %split1 and %split2 macros are simple and very efficient.

Acknowledgements

I would like to thank Doug Hillmer for coming up with this problem.

Contact Information

Selvaratnam Sridharma U.S. Census Bureau 4700 Silver Hill Road, Stop 8400 Washington, D.C. 20233

301-763-5398