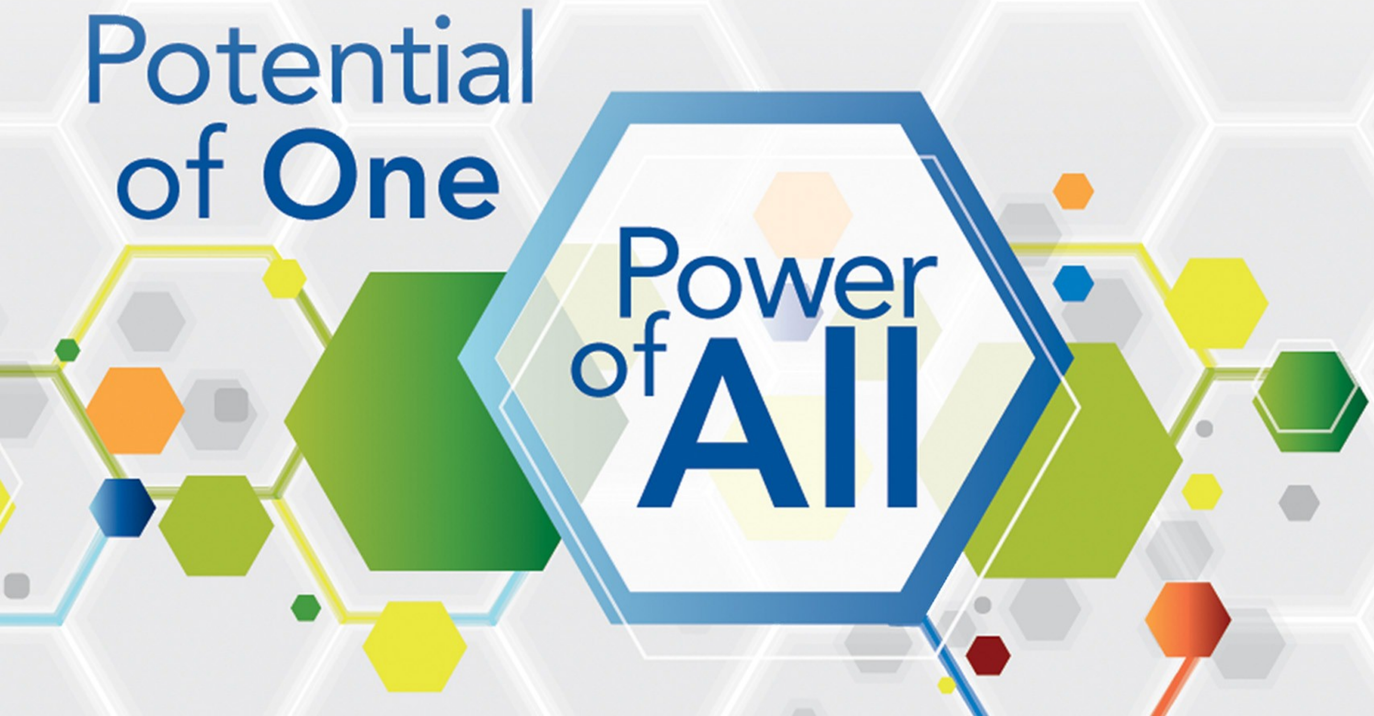
SAS GLOBAL FORUM

Extracting the needles from the haystack

Cutting big data down to size



Extracting the Needles from the Haystack

Sara Boltman
Butterfly Projects

Abstract

When you want to know the details about a small subset of a much larger data set, it can take a long time to pick out the records you need. This paper will show you how to create a user defined SAS® format to pull only the observations you want out of a big data source. Even when selecting a million records out of datasets that can have over 100 million records, this method is much quicker than either a PROC SQL join or a SAS merge.

Defining the problem

Let's say you're working for an insurance company that want to know which of their customers have used their online quote application in the last month. You may have a particular subset of high value customers that you wish to make an effort to retain, so knowing whether they have
requested a quotation this month is important to your business. First you need to identify a unique match key that is in both tables – for example for motor vehicle insurance you could use the vehicle registration number. In this example for simplicity I will assume that the numeric customer account number is present in both my little dataset and the big dataset. Initially you might sort both datasets by the account number and then merge them, like this:

```
data matches;
 merge little (in=a)
 bigdata (in=b);
  by account_ID;
   if a and b;
run;
```

If the larger dataset is not already sorted by account_ID this could take a very long time, so you might choose to use SQL instead. This also has the advantage that if the key variable is not named consistently across the datasets that doesn't matter – so if account_ID is actually called accnum in the larger table, you could use the following code:

```
proc sql;
 create table matches
            as select a.account_ID,
             b.*
 from little as a
 inner join bigdata as b on a.account_ID=b.accnum;
quit;
```

Using a format instead

SAS formats are usually used to apply a label to a range of values. The classic example is grading exam scores:

```
proc format;
 value grade 90 - 100 = 'A'
             80 - 89 = 'B'
             70 - 79 = 'C'
             60 - 69 = 'D'
             50 - 59 = 'E'
             0 - 49 = 'F' ;
```

The values on the left hand side are called the 'range' and typically have a 'start' and 'end' value. The values on the right hand side are called the 'labels'. The FORMAT procedure uses a binary search using the lookup table that makes very efficient use of computer resources.

But for this example we only need one label, ''Y'. As SAS reads in each row in the big table it's either one of the accounts we are interested in or it's not. If you have a very small number of account numbers you are trying to extract, you could define a stored format called $pick. using a value statement, where each of the account_IDs is the 'start' of the range. There is no need to specify an 'end' to any of the ranges.

```
proc format;
 value $pick 'VRN331630' ='Y'
 'VRN331769' ='Y'
 'VRN331801' ='Y';
```

You can then see if any of these three accounts were in the big data set using a simple WHERE statement.

```
data matches;
 set bigdata (where=
             (put(account_ID, $pick.) ='Y')
                                      );
run;
```

Building formats from data

The real power in this technique is of course if you already have the list of account_IDs you are interested in as a SAS dataset. First, you need to make sure you have a unique list of the accounts you want to extract, sorted by the account number.

```
Proc sort data=little nodupkey; by account_ID;
run;
```

Next you can use the CNTLIN function to create a SAS format that is effectively a lookup list.

```
data little_format (keep = fmtname type start label);
      retain type 'C';
 set little;
       start   =  account_ID;
       fmtname = 'pick'; label = "Y"; output;
 run;

Proc format cntlin=little_format; run;
```

This can be used to create the 'matches' dataset as before, or as one step in a longer process that could then become fully data driven.

Conclusions

The SAS format procedure is much more than a labelling tool, it is a powerful technique that will save you time in selecting the records of interest.

References

- SAS Formats – Simply Powerful. Howard Hagemann, South Central SAS User's Group Conference October 2007.
- Finding a Duplicate in a Haystack. Brett J. Peterson, SUGI 31

Your comments and questions are always welcome. Contact the author at Butterfly Projects Sara@butterflydev.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies

SAS® GLOBAL FORUM

Washington, D.C.

March 23–26, 2014