

Utilizing Hash Tables to Obtain Matched Post-Hoc Control Populations

Elayne Reiss, University of Central Florida
Jeffrey Reiss, Independent Statistical Consultant

Contents



- Introduction to the problem
- Hash table overview
- Why use hash tables?
- Creating matching post-hoc control populations
- Conclusion

Introduction



- Statisticians in the social sciences are often asked to conduct post-hoc studies
- Studies from outside parties: where's the control group?

Introduction



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

- Depending on the nature of the dataset, finding similar samples after the fact may not be easy
- SAS (version 9) can help resolve this problem, through...

Hash functions



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

HASH TABLE OVERVIEW

Hash Table Overview

SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008



Hash tables provide a quick and simple solution for finding key observations within a larger dataset



Components of a Hash Table – Example 1



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

```
data sample;  
if _n_ = 1 then do;  
    declare hash h(dataset:  
        "sampleset");  
    h.defineKey('sku');  
    h.defineDone();  
end;  
set productlisting;  
if h.find() = 0 then output;  
run;
```

- Hash objects require declaration
 - Only run once
- *Declare hash h*
 - Defines *h* as a hash object using the dataset “sampleset”
- *h.definekey()*
 - Specifies variables for matching
- *h.definedone()*
 - Ends the definition of *h*

Components of a Hash Table – Example 1



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

```
data sample;  
if _n_ = 1 then do;  
    declare hash h(dataset:  
        "sampleset");  
    h.defineKey('sku');  
    h.defineDone();  
end;  
set productlisting;  
if h.find() = 0 then output;  
run;
```

- Set declaration must occur after the hash declaration
- *h.find()* searches through “productlisting” for all observations that match the key variable
 - 0 represents “true” in any hash function
- All matches are output to the “sample” dataset

Components of a Hash Table –

Example 1



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

The end result is the dataset “sample” with all entries that match the values of the ‘sku’ variable in the “sampleset” dataset

Components of a Hash Table – Example 2



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

```
data sample;  
format sale 5.2;  
if _n_ = 1 then do;  
    declare hash h(dataset:  
        "sampleset");  
    h.defineKey('sku');  
    h.defineData('sale');  
    h.defineDone();  
end;  
set productlisting;  
if h.find() = 0 then output;  
run;
```

- Additional variables must be formatted prior to the hash definition
- *h.defineData()* allows for variables in the hash table to be included but not compared

Components of a Hash Table – Example 2



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

- Example 2 produces the same output as Example 1, but also includes the ‘sale’ variable from the “sampleset” dataset for all observations in the resulting “sample” dataset
- Example 2: Useful for merging datasets

WHY USE HASH TABLES?

Sorting

SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008



Hash tables can merge without pre-sorted data.

Merge/By Method

```
proc sort data = sampleset;
    by sku;
run;
proc sort data = productlisting;
    by sku;
run;
data sample;
    merge productlisting sampleset;
    by sku;
    if sku ^= '';
run;
```

Hash Table Method

```
data sample;
format sale 5.2;
if _n_ = 1 then do;
    declare hash h(dataset:
        "sampleset");
    h.defineKey('sku');
    h.defineData('sale');
    h.defineDone();
end;
set productlisting;
if h.find() = 0 then output;
run;
```

Adding and Removing



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

- A master database can have items added and removed via hash tables
 - The master database is used as a hash table
 - A dataset of desired changes is used to find items to add or remove
 - The master database is replaced with the refreshed version when complete
- An example of this process can be found in the paper

Caveats



- Larger hash tables may present issues with computer memory usage
 - The hash table's speed is counterbalanced by the amount of memory required
 - Slower processes, such as merge/by statements, use less memory



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

CREATING MATCHING POST- HOC CONTROL POPULATIONS

The Issue



- Question: have at-risk students who participated in an alternative high school program performed better academically and behaviorally than if they remained in their zoned school?
- Irresolvable Issue: you can't turn back time and have students repeat the process over again
- Resolvable Issue: finding students with similar traits to serve as a matching control group

Ordinal Range Datasets



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

- Ordinal range variables must be prepared before utilizing hash tables (i.e. yearly subject GPA)
 - Introductory approach: expand each observation to include all possible combinations in a discrete range
 - This method is not considered for ranges with many possible combinations (this issue will be covered momentarily)

Sample Observation



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

VIEWTABLE: Work.Cclc0405

	FIRSTNAME	LASTNAME	markla1	markmath1
33	NIKITA		2.50	1.50
34	NIKITA		2.50	2.00
35	NIKITA		2.50	2.50
36	NIKITA		3.00	1.50
37	NIKITA	Starting Point -->	3.00	2.00
38	NIKITA		3.00	2.50
39	NIKITA		3.50	1.50
40	NIKITA		3.50	2.00
41	NIKITA		3.50	2.50

Running the Hash Table



SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008

- This scenario requires a multivariate key and **no extra variables**
 - The *h.definekey()* function will contain all variables serving as keys for matching
 - Post-process sorting may be required

Continuous Range Datasets



- Preparation is necessary for continuous variables (i.e. overall GPA)
 - Variables are needed to define both the high and low points of the range (i.e. +/- .25)
 - The hash table will bring these variables in as non-key variables

Continuous Range Sample Code

SouthEast SAS®
Users Group 2008
St. Pete Beach, Florida
October 19-22, 2008



Prep Code

```
data quest; /*prep for continuous*/
  set source.quest
    (rename=(hszone=school));
  gpalow = pr_gpa-.25;
  gpahi = pr_gpa + .25;
run;
```

Hash Code

```
data source.questhash2;
  format gpalow gpahi 6.3;
  if _n_ = 1 then do;
    declare hash h(dataset:
      "quest");
    h.definekey('school',
      'pr_grade', 'grade', 'ethnic',
      'frl', 'py_rmast',
      'py_mmast');
    h.definedata('gpalow', 'gpahi');
    h.definedone();
  end;
  set source.control ;
  if h.find()=0 then do;
    if gpalow <= pr_gpa <= gpahi
      then do;
        output;
      end;
  end;
run;
```

Taking a Final Sample



- The matching process will bring more observations to the control set than are necessary
- Use PROC SURVEYSELECT with options for SRS and a stratification variable
- Resulting dataset will contain the same proportion of 9th and 10th grade students as found in the actual alternative school testing population

Conclusions



- Don't fear the hash table!
- Hash tables serve as an efficient tool to match a key dataset to a large number of observations without a series of sorts, merges, and loops
- Creative coding can allow both discrete and continuous variables, even with ranges, to serve as keys for merging

Questions/Comments?



Contact Information

Elayne Reiss
University of Central Florida
12424 Research Pkwy
Orlando FL 32826
ereiss@mail.ucf.edu

Jeffrey Reiss
Independent Statistical Consultant
jeff10@bellsouth.net

This presentation will be posted online at <http://www.uaps.ucf.edu>.