Class Exercise 12

This exercise is based upon Chapter 16 of the SAS Advanced Certification Prep Guide. We will review hash objects to demonstrate a wider variety of applications than we covered during the lesson plan.

Begin by revisiting the 3-dimensional lookup table in Chapter 16. I am concerned that the example we studied might lead you to believe that there is something inherently more difficult about multi-dimensional lookups. I would like you to create the data set inputdata again, and set up the appropriate informats. Then we can simplify the declaration of the array, and the assignment of input values to the array:

data one;

set inputdata;

array answer {1:2,1:3,1:5} \_temporary\_

(4 4 5 5 5

 3 4 4 5 5

 2 3 4 4 5

 4 5 5 5 5

 3 4 5 5 5

 2 3 4 5 5 );

adjusted=answer(input(sex,gender.), input(age,age.),score);

proc print; run;

Comment on the changes in code—does this now seem similar to how we handled two-dimensional arrays?

In our investigation of hash objects, we really did not explore some alternate uses. Let’s take a closer look at a couple features. I started with a saved version of the E Coli data set as a permanent SAS data set named ecoli in my main course directory:

libname stat541 "/home/grego1/STAT 541/";

I’d like to use a hash object to assign different thresholds for E Coli to different watersheds. The stations for the watersheds include prefixes that identify the watershed (e.g., B=Broad River, E=Edisto River, CSTL=Santee Cooper Lakes). It seemed to me that I could create a table of unique stations in PROC SQL, save it, then build a hash object (Note—PROC SORT NODUPKEY could do this as well). First I created the table and saved it in WORK.stations (be sure to inspect the table to get a sense of the number of available stations):

proc sql;

create table stations as select

distinct station from stat541.ecoli;

quit;

So Station will be my hash key, and I can use PROC SQL to create a threshold for different values of the key. The Edisto River (E- stations) is the longest free-running river in the state, so I’ll assign a tougher standard to it. I arbitrarily assigned a looser standard to the Broad River (B- stations), though technically a looser standard would be assigned to rivers with less potential for recreation (e.g., non-navigable rivers). Look at the data set ecolihash that is now ready to be converted to a hash object. As you can see, its structure is pretty simple.

proc sql;

create table ecolihash as select station label="Station",

case

 when station contains "B" then 400

 when station contains "E" then 100

 else 200

end as ecolistandard label="E Coli Standard"

from stations;

quit;

Note that I didn’t set up my logic carefully enough—some stations may contain “B” or “E”, but would not actually belong to the Edisto or Broad River watersheds. Graduate students should provide code that corrects this error

I’ll create a hash object **ecolistan** from **ecolihash**, and then use it to assign thresholds to all the records in **stat541.ecoli**. I’ll also create a variable **ecoliviolation** that prints a message indicating E Coli status for each record. All of this will be saved in the data set **violations**. The code here is similar to what we did in class—just make sure you have all the data objects straight in your head as you inspect this code.

data violations;

length ecoliviolation $13;

if \_N\_=1 then do;

if 0 then set ecolihash;

declare hash ecolistan(dataset: "ecolihash");

ecolistan.definekey("Station");

ecolistan.definedata("Ecolistandard");

ecolistan.definedone();

end;

set stat541.ecoli;

keep station station\_description collection\_date ecoli ecoliviolation ecolistandard;

ecolistan.find(key:station);

if ecoli>ecolistandard then ecoliviolation="Exceedance";

else ecoliviolation="In Compliance";

run;

Notice that I kept only the variables related to E Coli. Finally, I want to print some of the records to check how things turned out. Comment.

proc sort data=violations;

by collection\_date; run;

title "E Coli Data for 2009";

proc print data=violations (obs=20) label;

label ecolistandard="E Coli Standard" ecoliviolation="E Coli Status" station\_description="Station Name" collection\_date="Date";

format collection\_date mmddyy8.;

run;