Class Exercise 4

This exercise is based upon Chapter 4 of the SAS Certification Prep Guide. In our class notes, we concentrated on basic applications of set concatenation commands and keywords. In this exercise, we will study concatenating larger data sets, and combining Chapter 4 features with features learned from earlier chapters.

Begin by importing the Bacterial data set from Excel into WORK.wq (note that you may need to rename any variable names that were truncated during the import). The following commands disaggregate WORK.wq by type of bacteria (E. Coli, Fecal Coliform, or Enterococci) and save each table separately (with the **CREATE TABLE** command).

proc sql;

create table ecoli as

select station, collection\_date, EColi from wq;

quit;

proc sql;

create table fcoli as

select station, collection\_date, fecalcoli from wq;

quit;

proc sql;

create table entero as

select station, collection\_date, enterococci from wq;

quit;

Confirm that the three data sets were saved correctly. Because of the way in which we disaggregated the data, every combination of station and date for each data set should also appear in every other data set. So the following command comparing, e.g., E Coli to Fecal Coliform, should not find records in E Coli that do not appear in Fecal Coliform. Confirm that this is true:

proc sql;

select station,collection\_date from ecoli

except select station,collection\_date from fcoli;

quit;

Next, we can use the **UNION** command to recombine the data into a “long and narrow” format.

proc sql outobs=30;

select station, collection\_date, 'EColi', Ecoli from ecoli

union select station, collection\_date,'Fecal Coliform',fecalcoli from fcoli

union select station, collection\_date,'Enterococci',enterococci from entero; quit;

The output needs improvement; we can provided a name for the column containing the type of bacteria, and also provide better labels. After running the code below, look at **wqtrans** in the Work directory, and confirm that appropriate names and labels have been assigned to each of the variables. Note that we only need to name and label the variables taken from **ecoli**, and they will be applied to the entire column.

proc sql;

create table wqtrans as select station label='Station', collection\_date label='Collection Date' format=mmddyy10., 'EColi' label='Bacteria' as Bacteria, Ecoli label='Count per 100 ml' as count from ecoli

union select station,collection\_date, 'Fecal Coliform', fecalcoli from fcoli

union select station,collection\_date, 'Enterococci', enterococci from entero;

quit;

Here is an attempt to try the same recombination of the data using **OUTER UNION**. How well does it work?

proc sql; select \* from Ecoli

outer union corr select \* from Fcoli

outer union corr select \* from Entero;

quit;

Download the Fall 2008 and Fall 2010 enrollment data. In this first example, we want to identify the 2008 enrollees who have changed their enrollment status in 2010 from part-time or full-time in 2008. How stable does enrollment status seem? Note that many of these switches actually include students who may have left USC.

/\* Surprising number of switchers \*/

proc sql; select noobs, enroll from fall08 except select noobs, enroll from fall10;

quit;

The code below actually counts the total number of enrollment switches using an in-line view. Repeat the same exercise with **Housing** to see if type of residence is stable.

proc sql;

title "Enrollment Status switch count with in-line view";

select count(\*) from (select noobs, enroll from fall08

except select noobs, enroll from fall10);

quit;

This code reports the average Fall 2008 GPA of switchers by enrollment status—note the use of the inner join. You may have to replace “avg(cltotgpa)” with “avg(input(cltotgpa,5.))” if cltotgpa is uploaded as a character variable rather than a numeric variable. Graduate students should write code to also report the number of switchers by Fall 2008 enrollment status.

proc sql;

select fall08.enroll, avg(cltotgpa) as avggpa from fall08, (select noobs, enroll from fall08 except select noobs, enroll from fall10) as switch

where fall08.noobs=switch.noobs

group by fall08.enroll;

quit;