Robyn Crook - 2008

Week 5 – Merging datasets by linking variables

The first section of Component 2 in the Project Description requires you to create and then merge two datafiles together. When we merge two files, we need to merge by a particular variable. This variable (our 'linking' variable) is what the program uses to identify which rows contain duplicate observations which will be merged together. Note that there is a different example embedded as a link in project description.

We will use the same example dataset as last week to demonstrate the merging procedure. The data file was set up as follows:

```
Title 'Code for merging two files';
Data cattle;
Infile 'U:\cattle_measurements.egc';
Input indiv 1 length 3-5 weight 7-9 height 11-12 age 14
colour $16 sex $18;
run;
```

Your code for merging files will continue after the end of your code for Component 1. I have included the Data step so you can see the set-up of this fictional dataset.

Let's assume we want to create two secondary files from 'cattle' that we will then merge. One secondary file contains observations where 'length' is not missing, and the other contains observations where 'weight' is not missing. We will then merge these two together, such that our final file contains observations that have values for either weight or height for each individual, but data for individuals that are present in both secondary sets are not duplicated in out final dataset, but are merged into a single row. This is general principle of 'merging', rather than 'combining' two datasets.

First we need to select observations from 'cattle' that have no missing 'length' values, and place these in a secondary dataset. This is the same code you have used in the past to create subsets. The 'keep' statement is not necessary but is required in the equivalent stage of the program project.

```
Title 'Secondary file containing lengths';
data cattle2;
set cattle;
if length;
keep indiv length;
run;
```

We need to sort 'cattle2' by individual (our linking variable).

```
proc sort data=cattle2;
by indiv;
run;
```

Now we can create the other secondary file, containing no missing 'weight' values, and then sort, again, by our linking variable.

```
Title 'Secondary file containing weights';
data cattle3;
set cattle;
if weight;
keep indiv weight;
run;
proc sort data=cattle3;
by indiv;
run;
```

Now we can merge the two datasets together.

```
Title 'Merging two secondary files cattle2 and cattle3';
data cattle4;
merge cattle2 cattle3;
by indiv;
run;
```

We could also have merged in the reverse order (merge cattle3 cattle2;). Try this with your code and see if the resulting file is the same.

Now in the final file 'cattle4', each individual that has either a length observation or a weight observation is represented, individuals who had neither have been excluded, and individuals who had observations for both length and weight (so were present in both subsets) are only represented in the final merged set once, in a single row containing indiv, length and weight values.

Next week – Proc ANOVA.