Week 2 – Cleaning and screening your data file

When working with data files that have been imported from elsewhere, it is likely that the dataset will contain some errors. Rather than changing values in the raw dataset (unadvisable!) it is better to have SAS remove or edit these problematic observations during your analysis. This means your original dataset will be unchanged.

There are two choices when it comes to screening out observations containing errors. You can either remove the whole observation (i.e., delete the whole row), or you can replace a single incorrect variable value with a 'missing' observation. This keeps the rest of the observation as it is.

In general, if it appears that a single value was entered incorrectly, then it should be converted to a 'missing' value. The entire observation should not be deleted because there still may be usable data in other variables. For continuous variables, you must decide whether especially large or small values are outliers (and should be converted to 'missing') or whether these values are plausible measurements that reflect the 'normal' variation in the data. This is a judgment call that you as the biologist must make. There is no correct answer and it is expected that final results will differ between projects. However, each conditional statement used to clean the data should have a comment with it explaining why you decided to make the changes that you did.

In the gosling dataset there are a few observations that were recorded for Ross goslings (as opposed to snow goslings) – these should be permanently deleted from the dataset. The following includes some facts about the data that you will help you decide what to delete and what to convert to 'missing':

1. For gosling color at the time of banding (COLOR), 'B' and 'W' mean blue and white plumage, respectively. Values of 'R' and '7' are recorded for Ross geese and a value of '1' means the bird was a white ('W') snow gosling.
2. For SEX, 'F', 'M' and 'U' denote female, male and unknown, respectively.
3. Gosling colors at hatching (GCOLOR) are recorded as either 'B', 'W' and '0'. A value of '0' means the color was not determined.
4. The conditions of goslings in each nest were checked simultaneously whether or not all birds were hatched. Six different conditions (GOS_COND) were recorded:
    a. 'E' – egg is not yet hatched
    b. 'P' – egg is pipped (chipped – when hatchling is starting break through)
    c. '5' – half of egg shell is broken away as hatchling is emerging
    d. 'W' – hatchling is wet
    e. 'D' – hatchling is damp
    f. 'F' – hatchling is fluffy
5. Development of feathers can be highly dependent on environmental factors, therefore, they can vary significantly from year to year. In the gosling dataset, measurements of both the primary and midtail feathers reflect this variation and both exhibit bimodal distributions with a very small second peak. Keep this in mind when screening for outliers.
6. Banding of goslings began about 3-4 weeks after hatching. Therefore, banding age (BNDAGE), which is recorded in estimated days after hatching, should not be much less than 21 days. Smaller values are either errors or a coding for 'missing' values.

SAS Code for cleaning, screening and identifying outliers.

Using the same fictional dataset as previously:

```
Title 'Robyn's SAS program for analysing spatial memory data';
* This is experimental data from spatial memory experiment run in March
07. Times are in seconds, animals are named A to F, trials are numbered
1-5;
data  spatmem;
infile 'C:\Documents and Settings\user\My Documents\My SAS
files\spatialtimes.egc';
options nocenter nonumber ls=72 ps=66;
input animal $1 trialno 3-4 time 6-9;
```

After the input statement we can use conditional statements to convert errors and resolve other problems, before proceeding with the analysis.

In this dataset, we know that the animals are named A-F, and there are a maximum of 5 trials. Let's imagine we had an experimenter who numbered the animals rather than used their letter-names, so we want to make sure the program knows that Animal A is the same as Animal 1 etc.

So, after the input statement, we can place a comment, then our if/then statements.

```
input animal $1 trialno 3-4 time 6-9;
* Re-coding incorrect animal names and removing incorrect trial numbers;
if animal='1' then animal='A';
if animal='2' then animal='B';
if trialno>5 then delete;
run;
```

Here we have recoded our animal names, but we have deleted the whole observation for trials numbered above 5. We also could have recoded those trial numbers as missing values.

Finding outliers in continuous data is more involved. In our case, our continuous variable is 'time'. Let us assume we know that in *most* trials, animals took between 2 and 10 minutes to finish a trial (120 - 600 seconds). But, we want to be sure we only remove the extreme values or errors. In this case it is helpful to look at a histogram of the data. We use the command proc univariate for this.

```
if animal='1' then animal='A';
if animal='2' then animal='B';
if trialno>5 then delete;
run;
proc univariate data=spatmem noprint;
histogram time;
run;
```

This will allow us to look at the distribution of the data on a graph and decide for ourselves where the outliers are, and we can delete observations that show time as higher or lower than what we think is reasonable. Or we can simply recode them as missing.

So, we can then go back above our histogram command, and add additional if/then statements:

```
if animal='1' then animal='A';
if animal='2' then animal='B';
if trialno>5 then delete;
if time<70 then time=' ';
if time>750 then time=' ';
run;
proc univariate data=spatmem noprint;
histogram time;
run;
```

Now when we run the code, the histogram will show the modified distribution. Next we can calculate descriptive stats for our data using our clean data set.

Descriptive statistics using SAS.

Now that we have a clean data set, we can start to gather basic statistics about each variable. For continuous variables, we are interested in means, medians etc. We use proc means for this. For discrete measures we can look at frequencies. We use the command proc freq for this.

Our complete program will now look like this:

```
Title 'Robyn's SAS program for analysing spatial memory data';
* This is experimental data from spatial memory experiment run in March
07. Times are in seconds, animals are named A to F, trials are numbered
1-5;
data  spatmem;
infile 'C:\Documents and Settings\user\My Documents\My SAS
files\spatialtimes.egc';
options nocenter nonumber ls=72 ps=66;
input animal $1 trialno 3-4 time 6-9;
if animal='1' then animal='A';
if animal='2' then animal='B';
if trialno>5 then delete;
if time<70 then time=' ';
if time>750 then time=' ';
run;
proc univariate data=spatmem noprint;
histogram time;
run;
title 'Descriptive stats for time';
* maxdec tells the program to show results to two decimal places;
proc means data=spatmem mean median min max range stderr maxdec=2;
var time;
run;
title 'Frequency tables for animal and trialno';
* nocum and nopercent stop the output showing columns in the table for
cumulative frequencies and percentages (we don't need them here);
proc freq data=spatmem;
tables animal trialno/nocum nopercent;
run;
```