Week 2 – Getting started with SAS.

The URL for the project is here: http://research.amnh.org/~rfr/project.htm

At the bottom of this page there is the data file for the project (gosize.egc) and a help file with all the column assignments. You will need this to import your data correctly.

Getting your data into the SAS program and beginning to write your code.

1. Download and unzip the data file. Save it on the desktop or in My Documents. Note the file path. e.g. 'C:\Documents and Settings\user\My Documents\My SAS files\gosize.egc'
Then open the help file at the very bottom of the page. This has the column assignments you will need.

2. Open SAS (start tab, programs, SAS, SAS 9.1). There are three windows – the **Editor window** (where you type your program code), the **Log window** (which tells you what the program has done), and the **Output window** (shows you the results of your analyses). The Log window should have some text in it, the others two will be blank. The log window will alert you to any errors or problems with the code – errors show in red/brown text. Every time you run your program, first check the log, then check your output.

Along the top is your toolbar. The icon of the man is the 'run' command.

3. Start typing in the Editor window to begin your program code. Always begin a new program with a Title, and a comment about what the program is for and any other relevant details.

There are four main types of commands in SAS – Titles, comments, data commands, and procedure commands. Each is written is a certain way and every command is closed with a semicolon. Different types of commands will show in different colours in the Editor window.

SAS commands are case insensitive. Hard returns are ok.

Titles.

Start a title command by typing Title, followed by your title in inverted commas. Titles text shows in purple. Remember to close your title with an inverted comma, and a semicolon.

```
Title 'My SAS program for gosize data';
```

Comments.

Comments are useful for reminding yourself why you are performing a particular command. Your program must contain comments to tell me why you are doing what you're doing. The comment command is also used for de-activating code you might not want to run. This is called 'commenting out' code. Comments show in green. There are two ways to make a comment:

```
* your comment;
```

This opens the comment at the asterisk, and closes it at the next semi-colon. Or:

```
/* your comment*/
```

This turns everything between the asterisks into a comment, even if there are semicolons in there. This is useful for deactivating large parts of your code if you want to only run a small portion of it.

Data commands.

Data commands tell SAS that you are importing data, or making subsets of your data. The first step in your program is to tell SAS what you are naming your data, and where to find it. Data commands show in bold type blue. There are other commands that also go underneath your data statement.

```
Data mydata;
```

Then comes either an input, an infile or an import statement. Sometimes we add a 'set' command if we want to take a subset of an existing data file. For your program you will use infile, but I have given you examples of all three approaches in another handout 'Three ways to get your data into SAS'.

```
Data mydata;
Infile 'C:\ your file path';
```

This line is followed by an options statement:

```
Data mydata;
Infile 'C:\ your file path';
Options nocenter nonumber ls=72 ps=66;
```

Options tell the program how you want your output to look. It is not always necessary to include options, but it makes the output easier to read. I expect Options statements in your program.

Then comes the input statement. This tells SAS what each column in your original data file is called and which position they are in. After the input statement you write the column names and their position across the spreadsheet. Column names show in black, the position numbers in light blue-green. We use a $ to tell the program the column contains non-numerical values. Each name is separated by a space. The help file has all the information you need to get the correct values for your input statement.

```
Data mydata;
Infile 'C:\ your file path';
Options nocenter nonumber ls=72 ps=66;
Input cws 1-8 tagstat $10-13 bndyr 16-17 …….etc;
run;
```

Proc commands.

Proc is short for procedure. Proc commands are used for any analyses on the data. There are many different types of proc command we will use in the program. Proc commands are usually followed by a statement telling SAS which data we are using (if you have multiple data subsets, for example). Proc statements show in bold face blue.

Here is an example of the beginnings of a SAS program using some of these commands.

```
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;
run;
proc sort;
by trialno;
run;
proc means data=spatmem mean median min max range maxdec=2;
var time;
by trialno;
run;
```

In this program fragment we have given the program a title and made a comment about how the data file is set up.
Using the Data command, we have named our data, imported the dataset using Infile, chosen some output Options and designated the columns using Input.
Under the data step there are two different Proc commands. The first (proc sort) sorts the data by one of the variables (trialno, in this case). Then we have calculated some descriptive stats (using proc means) for the variable 'time', and separated our analyses by trial number. These will appear in the output window.

Note that SAS does not show you your spreadsheet automatically, the output window contains only results of your analysis. You can look at your data sets if you want to, by using the panel on the left. Select the 'Explorer' tab. Click 'libraries', then 'work'. Once you have run your code that brings your data in, the spreadsheet will show in the 'work' menu. It will be named whatever you called it in your data step. If you open the spreadsheet, you need to close it again before SAS will run any more procedures on it.

Next: cleaning and screening…..