SAS Work Shop **Statistical Programs**
DATA STEP **College of Agriculture**
Handout #10 HTTP://WWW.UIDAHO.EDU/AG/STATPROG

SAS FUNCTIONS AND OPERATORS

In order to do calculations in SAS we need to be familiar with items such as addition (**+**), subtraction (**-**), division (**/**), and multiplication (*****), which are called **operators**. They perform the indicated operations between SAS variables and/or constants. The flow of any expression (order of execution) can be controlled in the normal algebraic manner using parentheses. SAS also provides **functions** for common mathematical expressions such as square root (**SQRT()**), natural log (**LOG()**), and many more.

EXAMPLE 1

SAS Code:

```
DATA EXP;
      INFILE 'A:\EXP.DAT';
      INPUT SOIL $ TRT COUNT1 COUNT2;

      AVGCNT = (COUNT1 + COUNT2)/2;
      RESPONSE = SQRT(COUNT1 * COUNT2) - LOG(COUNT2);
```

The new variable AVGCNT is the computed average of COUNT1 and COUNT2. A new operator is used here which is the assignment operator **=**. This operator takes the expression on the right and puts the result into the variable on the left. The other operators used here are the obvious addition and division. It is important to note the use of parentheses. If they had been left out i.e.

```
AVGCNT = COUNT1 + COUNT2 / 2;
```

SAS FUNCTIONS AND OPERATORS (cont.)

the result would be COUNT2 divided by 2 and then added to COUNT1. This occurs because division has a higher priority than addition, and hence it will be done first. By using parentheses, SAS is forced to add COUNT1 and COUNT2 first before dividing by 2. This system of priorities for operators is outlined in the SAS manual. However, rather than relying on priorities, it is best to always use parentheses to insure the order of execution is correct.

The variable RESPONSE illustrates the use of two common SAS functions, **SQRT()** and **LOG()**. Functions require **arguments** which are the items between the associated parentheses. The arguments must be valid values for the specified function, i.e. for **SQRT()** the argument should be non-negative. Notice also that operators and functions can be mixed freely.

Also available in SAS are comparison operators. For example, these might be **<, >,** or **=**, and can be used alone or in combinations such as **<=** or **>=**. Comparison operators are often used with the **IF - THEN** structures discussed earlier.

SAS FUNCTIONS AND OPERATORS - MISSING VALUES

How do SAS functions and operators react to missing values? Examine the AVGCNT variable in the previous example again.

AVGCNT = (COUNT1 + COUNT2)/2;

What would happen if COUNT2 were missing? In that case SAS would not know how to add a missing value to a real number so it would set AVGCNT to a missing value as well. Anytime SAS encounters an operation or function that is not defined, the result will be a missing value. This is called **propagation of missing values** because one missing value causes others to occur as well. In the SAS LOG this will appear as " *X number* of missing values were generated at line *Y*".

This situation may be a problem especially if we want to ignore missing values. Special SAS functions are available to handle this, although they may seem to duplicate existing operators or groups of operators. For example the SAS function **SUM()** does addition like the operator **+**. The difference is in how the two handle missing values. If the SAS function encounters a missing value it will skip it and work on only those values that are not missing. So if our example is written as:

AVGCNT = SUM(COUNT1 COUNT2)/2;

and COUNT2 is missing, the function **SUM()** will only use COUNT1 and AVGCNT will equal COUNT1 divided by 2. This is clearly not right for the average, but the addition

SAS FUNCTIONS AND OPERATORS - MISSING VALUES (cont.)

part has worked correctly. If we used the **SUM()** function on 10 variables and 2 were missing, it would only add the 8 that were not missing. Now, back to AVGCNT: SAS goes further to help compute things like means, so that AVGCNT could be done correctly by the following:

```
AVGCNT = MEAN(COUNT1 COUNT2);
```

Here the function **MEAN()** does all the computation including accounting for the number of non-missing values that have to be added. So if COUNT2 is missing AVGCNT will be COUNT1 divided by 1. Many similar summarizing functions are available in SAS such as **STD(), VAR(), N(),** etc and operate in the same way. That is, all will skip missing values when necessary and compute the required function accordingly.

As always, be careful and make sure SAS is computing what you think it is! Just because the program ran without errors does not mean it did things right!! Check your OUTPUT and LOG screens to verify the results.