APPENDIX A - SAS OPERATORS

The following are all taken from SAS online help ...

Logical (Boolean) Operators and Expressions

Logical operators, also called **Boolean operators**, are usually used in expressions to link sequences of comparisons. The logical operators are shown in the following table:

Logical Operators

Symbol	Mnemonic Equivalent	Example
&	AND	`(a>b & c>d)`
|	OR (table note 1)	`(a>b or c>d)`
!	OR	
¦	OR	
¬	NOT (table note 2)	`not(a>b)`
^	NOT	
~	NOT	

TABLE NOTE 1: The symbol you use for OR depends on your operating environment.

TABLE NOTE 2: The symbol you use for NOT depends on your operating environment.

Arithmetic Operators

Arithmetic operators indicate that an arithmetic calculation is performed, as shown in the following table:

Arithmetic Operators

Symbol	Definition	Example	Result
**	exponentiation	`a**3`	raise A to the third power
*	multiplication (table note 1)	`2*y`	multiply 2 by the value of Y
/	division	`var/5`	divide the value of VAR by 5
+	addition	`num+3`	add 3 to the value of NUM
-	subtraction	`sale-discount`	subtract the value of DISCOUNT from the value of SALE

TABLE NOTE 1: The asterisk (*) is always necessary to indicate multiplication; `2Y` and `2(Y)` are not valid expressions.

Comparison Operators

Comparison operators set up a comparison, operation, or calculation with two variables, constants, or expressions. If the comparison is true, the result is 1. If the comparison is false, the result is 0.

Comparison operators can be expressed as symbols or with their mnemonic equivalents, which are shown in the following table:

Comparison Operators

Symbol	Mnemonic Equivalent	Definition	Example
=	EQ	equal to	`a=3`
^=	NE	not equal to (table note 1)	`a ne 3`
¬=	NE	not equal to	
~=	NE	not equal to	
>	GT	greater than	`num>5`
<	LT	less than	`num<8`
>=	GE	greater than or equal to (table note 2)	`sales>=300`
<=	LE	less than or equal to (table note 3)	`sales<=100`
	IN	equal to one of a list	`num in (3, 4, 5)`

The IN Operator in Numeric Comparisons

In a DATA step, you can use a shorthand notation to specify a range of sequential integers to search. The range is specified by using the syntax M:N as a value in the list to search, where M is the lower bound and N is the upper bound. M and N must be integers, and M, N, and all the integers between M and N are included in the range. For example, the following statements are equivalent.

- `y = x in (1, 2, 3, 4, 5, 6, 7, 8, 9, 10);`
- `y = x in (1:10);`

You can use multiple ranges in the same IN list, and you can use ranges with other constants in an IN list. The following example shows a range that is used with other constants to test if X is 0, 1, 2, 3, 4, 5, or 9.

The IN Operator in Character Comparisons

You can use the IN operator with character strings to determine whether a variable's value is among a list of character values. The following statements produce the same results:

- `if state in ('NY','NJ','PA') then region+1;`
- `if state='NY' or state='NJ' or state='PA' then region+1;`