

APPENDIX A - SAS OPERATORS

COMPARISON operators are used to construct logical expressions.

use either		to indicate
=	eq	equal
^=	ne	not equal
>	gt	greater than
<	lt	less than
>=	ge	greater than or equal
<=	le	less than or equal
in		determine whether a value is contained in a specified list ¹

LOGICAL operators are used to construct compound logical expressions.

use either		to indicate
&	and	both
	or	either
^	not	not true

ARITHMETIC operators are used to perform calculations with SAS variables or constants.

use	to indicate
+	addition
-	subtraction
*	multiplication
/	division
**	exponentiation

There is a hierarchy that governs which LOGICAL and ARITHMETIC operators are evaluated first.

LOGICAL operators: NOT, AND, OR

ARITHMETIC operators: **, *, /, +, -

If you forget the hierarchy, you can use parentheses in any expression to tell SAS which part of the expression to evaluate first. The only rule to remember is that expressions within parentheses are evaluated first.

¹ Examples of using the IN operator....

```
if xnum in (1 23 56 78 200 567);
```

TRUE if the value of the variable XNUM is found in the given list

```
if xchar in ("MIKE" "KATHY" "SARA" "JESSICA");
```

TRUE if the value of the variable XCHAR is found in the given list

```
if xchar not in ("MIKE" "KATHY" "SARA" "JESSICA");
```

TRUE if the value of the variable XCHAR is NOT found in the given list

```
if age in (10 11 12 13 14 15 16 17 18 19) then group="TEEN ";
else group="OTHER";
```

creates a new variable based on comparing an existing variable to a list of values

APPENDIX B - WHERE STATEMENTS

The BETWEEN-AND operator selects observations in which the values of the variables fall within a range of values. You can specify the limits of the range as constants or expressions. Any range you specify with the BETWEEN-AND operator is an inclusive range, so that a value equal to one of the limits of the range is within the range. The BETWEEN-AND operator has the following form:

WHERE variable BETWEEN value AND value ;

examples:

```
where empnum between 500 and 1000;  
where taxes between salary*0.30 and salary*0.50;
```

You can combine the NOT operator with the BETWEEN-AND operator to select values that fall outside the range.

```
where empnum not between 500 and 1000;
```

The CONTAINS or question mark (?) operator selects observations that include the string specified in the WHERE expression. This operator is available for character variables only. The position of the string in the variable does not matter; however, the operator distinguishes between uppercase and lowercase characters when making comparisons. The following examples select observations containing the values Mobay and Brisbayne for the variable COMPANY, but they do not select the observation containing Bayview:

```
where company ? 'bay';  
where company contains 'bay';
```

The IS NULL or IS MISSING operator selects all observations in which the value of a variable is missing. This operator selects observations with both regular or special missing value characters and can be used for both character and numeric data. The operator has the following form:

```
where idnum is missing;  
where name is null;
```

You can combine the NOT operator with the IS MISSING or IS NULL operators to select nonmissing values, as follows:

```
where salary is not missing;
```

There are other WHERE expressions (e.g. CONTAINS and LIKE). Remember, the use of WHERE statements is LIMITED to SAS data sets, i.e. you can't input raw data via an input statement and use a WHERE statement to restrict observations. When reading raw data, you will have to use an IF statement.

APPENDIX C - DEFAULT SORTING ORDER OF CHARACTERS

There is a default sorting order for characters within SAS. It is known as *ASCII* order and is used on all computers except for the IBM mainframe where *EBCDIC* order is used. The main difference between ASCII and EBCDIC is that numbers come after letters in EBCDIC order. The number to the left of each character shown in the table does represent an actual number in the ASCII sorting sequence. Characters below number 32 and above 126 are mostly 'non-printable', e.g. a TAB (#9) or a FORMFEED/eject-a-page (#12).

#032		#062	>	#092	\	#122	z
#033	!	#063	?	#093]	#123	{
#034	"	#064	@	#094	^	#124	
#035	#	#065	A	#095	_	#125	}
#036	\$	#066	B	#096	`	#126	~
#037	%	#067	C	#097	a		
#038	&	#068	D	#098	b		
#039	'	#069	E	#099	c		
#040	(#070	F	#100	d		
#041)	#071	G	#101	e		
#042	*	#072	H	#102	f		
#043	+	#073	I	#103	g		
#044	,	#074	J	#104	h		
#045	-	#075	K	#105	i		
#046	.	#076	L	#106	j		
#047	/	#077	M	#107	k		
#048	0	#078	N	#108	l		
#049	1	#079	O	#109	m		
#050	2	#080	P	#110	n		
#051	3	#081	Q	#111	o		
#052	4	#082	R	#112	p		
#053	5	#083	S	#113	q		
#054	6	#084	T	#114	r		
#055	7	#085	U	#115	s		
#056	8	#086	V	#116	t		
#057	9	#087	W	#117	u		
#058	:	#088	X	#118	v		
#059	;	#089	Y	#119	w		
#060	<	#090	Z	#120	x		
#061	=	#091	[#121	y		
