

Operator, Please?

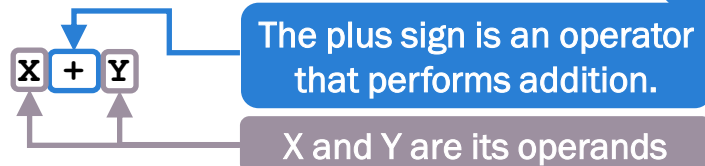
Making the Most of SAS Language Operators

Josh Horstman
Nested Loop Consulting
MWSUG 2024 Conference

1

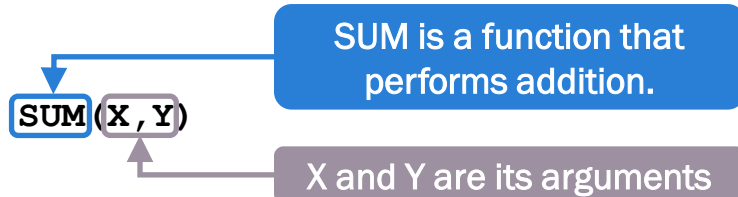
What is an Operator?

- Operator: A character or group of characters defined within a programming language to invoke a specific action



- Comparison to functions:

- Conceptually similar
- Syntactically different



2

Types of Operators

- Styles
 - Binary infix (appears between two operands)
 - Unary prefix (appears before a single operand)
- Functionality
 - Arithmetic operators
 - Comparison operators
 - Logical operators
 - Concatenation operators

3

Arithmetic Operators

Operator	Action	Example	Meaning
+	Addition (binary infix)	$X+Y$	Add Y to X
	Positive (unary prefix)	$+X$	Value of X
-	Subtraction (binary infix)	$X-Y$	Subtract Y from X
	Negative (unary prefix)	$-X$	Arithmetic inverse of X
*	Multiplication	$X*Y$	Multiply X by Y
/	Division	X/Y	Divide X by Y
**	Exponentiation	$X**Y$	Raise X to the Y power

- If one or more operands has a missing value:
 - The result will be a missing value
 - A note will appear in the log

NOTE: Missing values were generated as a result of performing an operation on missing values.

4

Comparison Operators

Operator	Mnemonic	Definition	Example
=	EQ	Equal to	X = Y
^=	NE	Not equal to	X ^= Y
>	GT	Greater than	X > Y
<	LT	Less than	X < Y
>=	GE	Greater than or equal to	X >= Y
<=	LE	Less than or equal to	X <= Y
IN		Equal to one of a list	X IN (1, 2, 3)

- Symbol for NE can vary by operating system
 - Common alternative in most environments: `~=`
- Values in IN list can be separated by spaces or commas

5

Comparison Operators

- Numeric Comparisons
 - Missing values are less than non-missing values
 - Missing values are equal (excluding special missing values .A through .Z)
- Character Comparisons
 - Performed character by character from left to right
 - Blank < period < other characters
 - If unequal lengths, shorter value padded with trailing blanks
- Compound Comparisons

X <= Y <= Z is interpreted as **(X <= Y) and (Y <= Z)**

6

The Colon Modifier

- Appended to comparison operators in character comparisons
- Comparison is truncated to the length of the shorter value

`name =: "Jon"`
 will match "Jon", "Jonny", "Jonathan"
 but also "Jo"

`name in: ("R", "S", "T")`
 will match any name that starts with R, S, or T

7

Logical Operators

Operator	Mnemonic	Type	Definition	Example
&	AND	Binary infix	True if both operands true	<code>X & Y</code>
	OR	Binary infix	True if at least one operand true	<code>X Y</code>
^	NOT	Unary prefix	True if operand is false	<code>^X</code>

- Symbols for OR and NOT can vary by operating system
 - Common alternative OR in most environments: !
 - Common alternative NOT in most environments: ~
- Logical operators are frequently grouped into compound expressions

`(A & B) | (C & D)`

8

MIN and MAX Operators

Operator	Mnemonic	Type	Definition	Example
<>	MIN	Binary infix	Lower of the two values	X <> Y
<>	MAX	Binary infix	Higher of the two values	X <> Y

- **CAUTION #1: These are interpreted differently in WHERE statements!**
 - <> is not supported in a WHERE statement
 - <> is interpreted as “not equal to” in a WHERE statement
- **CAUTION #2: Negation handled after MIN/MAX operator in IF statement**
 - -3<>-6 returns 6
- Consider using the **MIN ()** and **MAX ()** functions instead

9

Concatenation Operator

Operator	Type	Definition	Example
	Binary infix	Concatenates two character values	X Y

- The concatenation operator does not remove leading or trailing blanks.
- Use with **TRIM ()** function to remove trailing blanks
- Use with **STRIP ()** function to remove trailing and leading blanks
- Consider using concatenation functions instead
(**CAT ()**, **CATT ()**, **CATS ()**, **CATX ()**, **CATQ ()**)

10

Order of Operations

Priority	Operators
1	** , + prefix, - prefix, NOT , <>, ><
2	* , /
3	+ , -
4	
5	<, <=, =, ^=, >, >=, >, IN
6	&
7	

- Operations are performed from priority 1 through 7
- Within each priority, operations are performed from left to right
- Expressions within parenthesis are evaluated first

11

Variable Naming Shorthand

- The colon and dash operators can be used as shorthand for referring to a list of variables.
- Variables with a common prefix:
 - `keep x: ;` keeps all variables that start with x.
- Variables with a common prefix and sequential suffix:
 - `var q1-q4 ;` includes the variables q1, q2, q3, and q4.
- Consecutive variables on the Program Data Vector:
 - `drop mrsp--horsepower ;` drops range of variables from PDV.

12

WHERE Statement Operators

- The WHERE statement has its own special operators not available elsewhere:
 - **BETWEEN-AND:** where age between 12 and 15; (inclusive)
 - **CONTAINS:** where model contains "Sedan"; (case-sensitive)
 - **LIKE:** where name like "D_an%";
(underscore = 1 character, percent = 1 or more characters)
 - **SOUNDS LIKE:** where name =* "Kathy";
(uses SOUNDEX phonetic matching algorithm)

13

Conclusion

- The SAS language includes a robust set of operators.
- Understanding how they work and the order of operations aids in accurate and efficient programming.
- In some cases, a SAS function may be preferable

14

Questions?

Josh Horstman
Nested Loop Consulting
josh@nestedloopconsulting.com