

SAS Macro #5

Round #5

JC Wang



%EVAL and %SYSEVALF Functions

- ▶ %EVAL takes 'numerical expression' and returns 'booleans' (0 or 1) or 'integers', e.g.
 - ▶ `%LET i = %EVAL(3+2);`
 - ▶ `%LET j = %EVAL(5/2)`
 - ▶ `%LET a = 3;`
`%LET b = %EVAL(a>0);`
`%LET c = %EVAL(a=);`
- ▶ %SYSEVALF takes 'numerical expression' and returns 'floating point numerical value', e.g.
 - `%LET r = 2;`
 - `%LET pi = 3.14159;`
 - `%LET area = %SYSEVALF(&pi * &r**2);`



Selected Character Functions

%func or *%Qfunc*

- ▶ **%LENGTH** (*character-string* | *text-expression*): string length. E.g., `%LENGTH(&a)`; `%LENGTH(&a and &b)`;
- ▶ **%INDEX** (*source*, *string*): searches *source* for the first occurrence of *string* and returns the position of its first character. A 0 is returned if *string* is not found. E.g., `%INDEX(&a, jo)`;
- ▶ **%UPCASE** and **%QUPCASE**: convert values to uppercase
- ▶ **%SCAN** and **%QSCAN**: search for a word that is specified by its position in a string
- ▶ **%SUBSTR** and **%QSUBSTR**: produce a substring of a character string



Selected Quoting Functions

- ▶ **%STR** and **%NRSTR**: mask special characters and mnemonic operators in constant text at macro compilation. Unmatched quotation marks (" ") and parentheses (()) must be marked with a preceding %.
- ▶ **%BQUOTE** and **%NRBQUOTE**: mask special characters and mnemonic operators in a resolved value at macro execution. E.g., `%IF %BQUOTE(&statename) NE %THEN . . .`
They do not require that unmatched quotation marks and parentheses be marked.
- ▶ **%SUPERQ**: masks all special characters and mnemonic operators at macro execution but prevents resolution of the value.
- ▶ **%UNQUOTE**: unmarks all special characters and mnemonic operators for a value.



%SYSFUNC and %QSYSFUNC Functions

Execute SAS functions or user-written functions (created with SAS/TOOLKIT)

```
%SYSFUNC ( function ( argument ( s ) ) < , format > )
```

where

- ▶ *function*:
 - ▶ cannot be a macro function
 - ▶ cannot be any of the following: INPUT/PUT, DIF/LAG, DIM/LBOUND/HBOUND, IORCMMSG, MISSING, RESOLVE, SYMGET, and all variable information functions.
 - ▶ You cannot nest functions to be used with a single %SYSFUNC. However, you can nest %SYSFUNC calls such as

```
%LET x=%SYSFUNC ( TRIM ( %SYSFUNC ( LEFT ( &iind ) ) ) ) ;
```
- ▶ *arguments*: function arguments
- ▶ *format*: optional format to apply to the result of *function*



Example 1

Checking Existence of SAS Data Set

within a macro definition, assuming that `data` is a parameter of the macro

```
%IF %SYSFUNC ( exist ( &data ) ) %THEN %DO;  
...  
%END;
```

