Function & Call Routine Enhancements in SAS® 9

Presented by
Sue Rakes, PhD
SAS Institute Inc.
Austin TX and Kapolei HI

Call SORTN Routine

Sorts the values of numeric arguments – SAS **9.2**

```
data _null_;
  array x[10] (0, ., .a, 1e-12, -1e-8,
               .z, -37, 123456789, 1e20, 42);
  call sortn(of x[*]);
  put +3 x[*];
run;
```

SAS writes the following output to the log:

```
. A Z -37 -1E-8 0 1E-12 42 123456789 1E20
```

Call SORTC Routine

Sorts the values of character arguments – SAS 9.2

```
data _null_;
  array x{8} $10 ('tweedledum' 'tweedledee'
   'baboon' 'baby' 'humpty' 'dumpty'
   'banana' 'babylon');
  call sortc(of x{*});
  put +3 x{*};
run;
```

SAS writes the following output to the log:

```
   baboon baby babylon banana dumpty humpty
tweedledee tweedledum
```

Call Missing Routine

Sets character or numeric arguments to missing

```
data test;
   a = 1;
   b = 'test';
   call missing(a,b);
   put _all_;
run;
```

SAS log:

```
a=. b=
```

CMISS Function

Counts the number of missing character or numeric arguments – SAS 9.2

```
data _null_;
  x = 1;  y = . ;  z = 'abc';
  n=nmiss(x,y,z); *numeric args only;
  c=cmiss(x,y,z); *char or numeric;
  put n= c=;
run;
```

NOTE: Character values have been converted to numeric…

n=2 c=1

HOLIDAY Function

```
data _null_;
 Easter=holiday('Easter',2011);
 put Easter=weekdate.;
run;
```

```
Partial Log:
Easter=Sunday, April 24, 2011
```

Some other holidays: Columbus, Fathers, Labor, MLK, Memorial, Mothers, Thanksgiving, USPresidents, Veterans, Victoria

STRIP Function

```
data  _null_;
  string = '   Testing   ';
  try1 = strip(string);
  try2 = trim(left(string));
  put string= quote12./
      try1= quote12./
      try2= quote12.;
run;
```

Partial log:
string="   Testing"
try1="Testing"
try2="Testing"

COMPARE Function

Gives position of leftmost character by which two strings differ, or returns 0 if they match

```
pad=compare('abc','abc          ');
differ=compare('abcdef','abc');
truncate=compare('abcdef','abc',':');
blank=compare(' ','abc');
```

pad=0 differ=4 truncate=0 blank= -1

':' argument is like the =: operator, truncates longer string to the length of the shorter

The sign of the result is negative if string-1 precedes string-2 in sort sequence

SUBSTRN Function

Returns a substring, allowing negative or zero start column values

-2	-1	0	1	2	3	4	5
			a	b	c	d	

```
result1 = substrn("abcd", -1, 4);

result2 = substrn("abcd", 0, 4);

put result1=$quote. result2=$quote.;
```

result1="ab " result2="abc"

CHAR Function

Returns a single character from a string – SAS 9.2

```
data _null_;
  result1 = substr("abcd", 3, 1);
  result2 = char("abcd", 3);
  put result1= result2= ;
run;
```

result1=c result2=c

Think SUBSTR with a length of 1

FIRST Function

Returns the first character from a string – SAS 9.2

```
data _null_;
  string='abc';
  result=first(string);
  put result= ;
run;
```

result=a

Think SUBSTR with a start column of 1 and a length of 1

PROPCASE Function

Changes words to proper case

```
data _null_;
  string="sEAn o'brIEn";
  result=propcase(string);
  *define word delimiters;
  result2=propcase(string,"' ");
  put result= result2= ;
run;
```

result=Sean O'brien result2=Sean O'Brien

Default word separators are
Blank / - (. and tab.

LENGTHN and LENGTHC Functions

```
x= '  '; *2 spaces;
len = length(x);
lenn = lengthn(x);
lenc = lengthc(x);
put len= lenn= lenc=;
```

len=1 lenn=0 lenc=2

LENGTHN and LENGTH return the same value for non-blank character strings. LENGTHN returns a value of 0 for blank character strings. LENGTH returns a value of 1. LENGTHC returns the length of a character string, including trailing blanks.

FIND Function

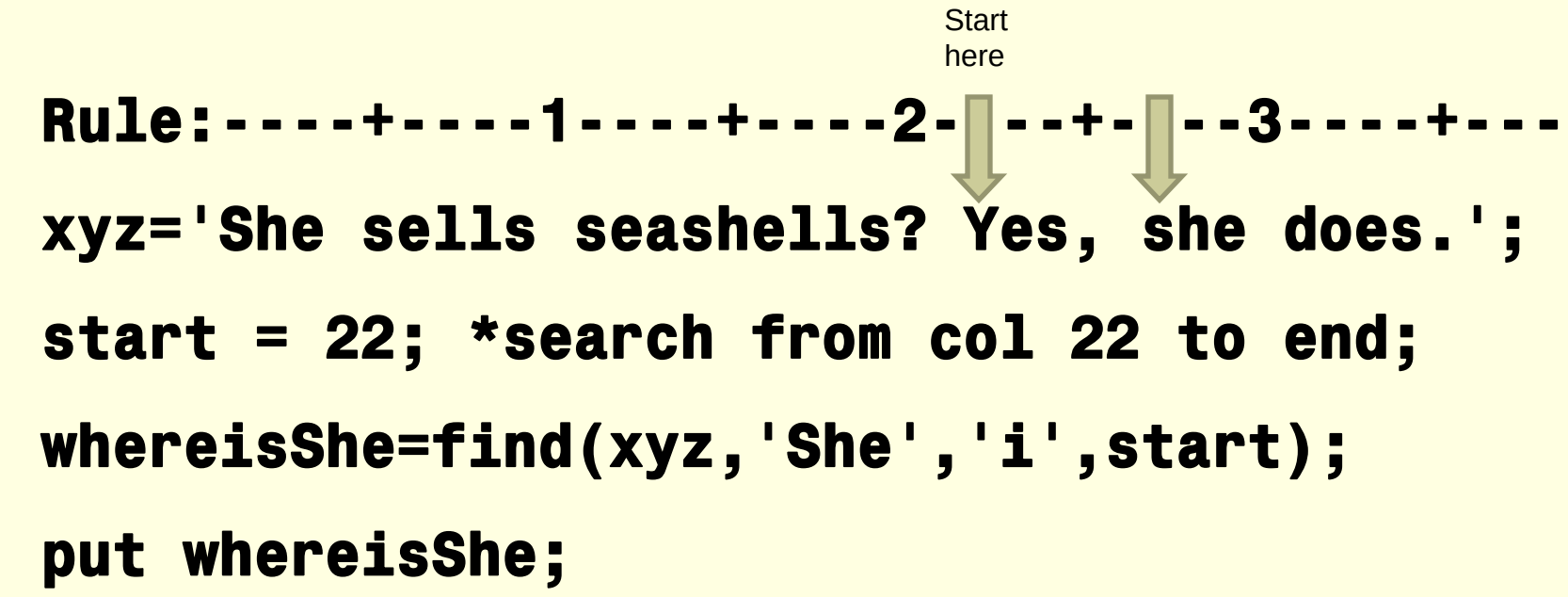
Searches for a substring of characters within a string

```
Rule:----+----1----+----2----+----3----+---
xyz='She sells seashells? Yes, she does.';
whereisShe=find(xyz,'she');
put whereisShe;
```

14, case sensitive
Zero means no match

FIND Function

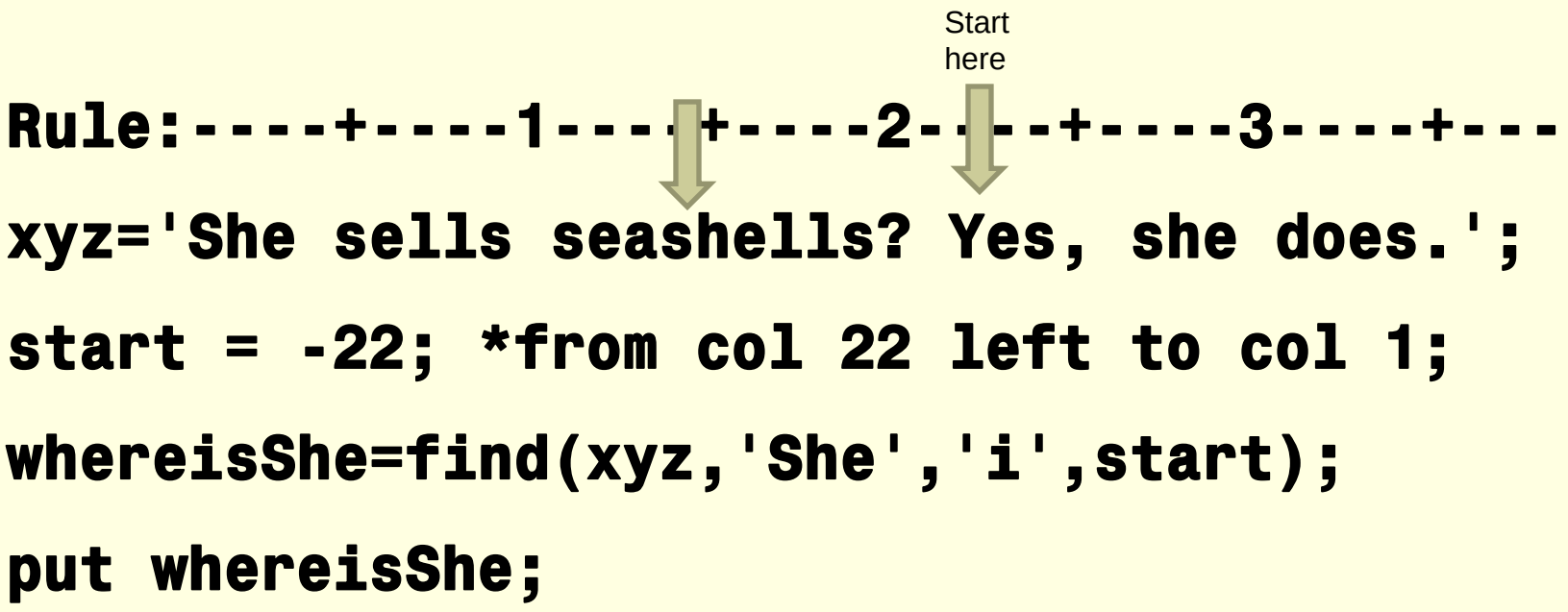
Modifiers and start column arguments optional



27, “i” modifier ignores case

FIND Function

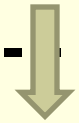
Negative start column searches right to left



14

FINDC Function

Searches for any one character in a list of characters



```
Rule:----+----1----+----2----+----3----+---
xyz='She sells seashells? Yes, she does.';
*search for an s, an h or an e;
whereisShe=findc(xyz,'she');
put whereisShe;
```

2

FINDW Function

Searches a string for a word - SAS 9.2

```
xyz='She sells seashells? Yes, she does.';
*search for the word she;
whereisShe=findw(xyz,'she');
put whereisShe;
```

27, default delimiters
blank ! $ % & ( ) * + , - . / ; < ^ |

FINDW Function

Counts words until finding a match - SAS 9.2

```
xyz='She sells seashells? Yes, she does.';
whereisShe=findw(xyz,'she', ' ', 'E');
*use a blank delimiter;
*E counts words until the word she;
put whereisShe;
```

5, fifth word; E modifier counts the words that are scanned until the specified word is found

COUNT Function

Counts the number of times that a substring appears in a character string

```
string = 'Baboons Eat Bananas ';
b = count(string,'ba');
b_i = count(string,'ba','i'); *ignore case;
abc_i = count(string,'abc','i');
```

b=0 b_i=2 abc_i=0

COUNTC Function

Counts the number of characters in a string

```
string = 'Baboons Eat Bananas ';
b = countc(string,'b');
b_i = countc(string,'b','i');

  *Count a or b or c;
abc_i = countc(string,'abc','i');

/* Scan string for characters that are not "a","b",
   and "c"*, ignore case, and include blanks. */
abc_iv = countc(string,'abc','iv');
```

b=1 b_i=3 abc_i=8 abc_iv=12

*V counts characters that do not appear in the second argument

COUNTW Function

Counts the number of words in a string – SAS 9.2

```
string = 'Baboons Eat Bananas ';
blank = countw(string,' '); *blank delimiter;
b_i = countw(string,'b','i'); *b delimiter;
bl_b_i = countw(string,' b','i'); *either;
put  blank=  b_i=  bl_b_i=;
```

blank=3 b_i=3 bl_b_i=4

Modifiers on SCAN, COMPRESS, COUNTC&W , FINDC&W – SAS 9.2

SCAN(string, count<,charlist <,modifier(s)>>)

Selected Modifiers:

B scans backward from right to left

D adds digits to the list of characters

I ignores the case

K characters that are in the list of characters are kept

M multiple consecutive delimiters, and delimiters at the beginning or end of the string argument, refer to words with a length of zero

Q ignores delimiters that are inside of substrings enclosed in quotation marks

T trims trailing blanks

V counts characters that do not appear in the list of characters

Modifiers on SCAN, COMPRESS, COUNTC&W , FINDC&W – SAS 9.2

```
x=',leading, trailing,and multiple,,delimiters,,';

nwords = countw(x, ',', 'm'); *like DSD;
do count = 1 to nwords;
  word = scan(x, count,',', 'm'); *like DSD;
  put count= word=$quote. @;
end;
```

count=1	word=""	count=2	word="leading"
count=3	word=“ trailing"	count=4	word="and multiple"
count=5	word=""	count=6	word="delimiters"
count=7	word=""	count=8	word=""

Modifiers on SCAN, COMPRESS, COUNTC&W, FINDC&W– SAS 9.2

```
x='Math A English B Physics A';
  *keep ABCD: ;
y=compress(x,'ABCD','k');

w= 'a2B1-77G5y';
  *keep digits: ;
z=compress(w,,'kd');
```

y=ABA z=21775

CAT Functions

Function	Equivalent Code
CAT(OF X1-X4)	X1||X2||X3||X4
CATS(OF X1-X4)	TRIM(LEFT(X1))||TRIM(LEFT(X2))|| TRIM(LEFT (X3))|| TRIM(LEFT(X4))
CATT(OF X1-X4)	TRIM(X1)||TRIM(X2)||TRIM(X3)||TRIM(X4)
CATX(SP, OF X1-X4)	TRIM(LEFT(X1))||SP||TRIM(LEFT(X2))||SP|| TRIM(LEFT(X3))||SP||TRIM(LEFT(X4))

CATQ Function – SAS 9.2

Concatenates character or numeric values by using a delimiter to separate items and adding quotation marks to strings that contain the delimiter

```
*D modifier indicates optional second
  argument is the delimiter;
text1=CATQ('D',' ', 'noblanks','one blank',
  12345,'  lots of blanks  ');
*Add quotes to all arguments;
text2=CATQ('A', 'noblanks','one blank',
  12345,'  lots of blanks  ');
```

text1=noblanks "one blank" 12345 " lots of blanks "
text2="noblanks" "one blank" "12345" " lots of blanks "

WhichC and WhichN Functions – SAS 9.2

Searches for a value that is equal to the first argument, and returns the index of the first matching value

```
array fruit (*) $12 fruit1-fruit3
  ('watermelon' 'apple' 'banana');
c1=whichc('watermelon', of fruit[*]);
c2=whichc('banana', of fruit[*]);
c3=whichc('orange', of fruit[*]);
n1=whichn(1492,1066,1215,1492,1776);
*Hastings, Magna Carta, Columbus, US;
```

c1=1 c2=3 c3=0
n1=3

SMALLEST and LARGEST Functions

Returns the nth smallest or largest nonmissing value*

```
smallest = smallest(1,456,789,.Q,123);
minimum = min(456, 789, .Q, 123);
secondsmall = smallest(2,456,789,.Q,123);
largest = largest(1,456,789,.Q,123);
maximum = max(456, 789, .Q, 123);
secondlarge = largest(2,456,789,.Q,123);
```

smallest=123 minimum=123 secondsmall=456
largest=789 maximum=789 secondlarge=456

*Ordinal function returns nth smallest missing or non-missing value

GCD and LCM Functions – SAS 9.2

Returns the greatest common divisor and lowest common multiple

```
data _null_;
  GCD=gcd(0,10,15);
  LCM=lcm(5,10,15);
  put GCD= LCM=;
run;
```

GCD=5 LCM=30

SUMABS Functions – SAS 9.2

Returns the sum of the absolute values of the arguments

```
data _null_;
   Total=sumabs(.,0,10,-15);
   put Total=;
run;
```

Total = 25

RAND Function

Generates random numbers from any distribution

RAND (dist, parm-1,...,parm-k)

Distributions: Bernoulli, Beta, Binomial, Cauchy, Chi-Square, Erlang, Exponential, F, Gamma, Geometric, Hypergeometric, Lognormal, Negative binomial, Normal, Poisson, T, Tabled, Triangular, Uniform, Weibull

ANYDIGIT Function

Searches a character string for a digit

```
col1 = anydigit('abc123xyz');

   *start in col 5;
col2 = anydigit('abc123xyz',5);

   *start in col 7;
col3 = anydigit('abc123xyz',7);

   *search right to left: ;
col4 = anydigit('abc123xyz',-7);
```

col1=4 col2=5 col3=0 col4=6

ANYALPHA Function

Searches a string for an alphabetic character

```
col1 = anyalpha('abc123xyz');

   *start in col 5: ;
col2 = anyalpha('abc123xyz',5);

   *start in col 7;
col3 = anyalpha('abc123xyz',7);

   *search right to left: ;
col4 = anyalpha('abc123xyz',-7);
```

col1=1 col2=7 col3=7 col4=7

ANYALNUM, ANYPUNCT, ANYSPACE Functions

Same syntax as ANYDIGIT and ANYALPHA

ANYALNUM searches a character string for an alphanumeric character

ANYPUNCT searches a character string for a punctuation character

ANYSPACE searches a character string for a white-space character: blank, horizontal and vertical tab, carriage return, line feed, and form feed

NOTDIGIT Function

Searches a string for any character that is not a digit

```
col1 = notdigit('abc123xyz');

   *start in col 5: ;
col2 = notdigit('abc123xyz',5);

   *start in col 7;
col3 = notdigit('abc123xyz',7);

   *search right to left: ;
col4 = notdigit('abc123xyz',-7);
```

col1=1 col2=7 col3=7 col4=7

NOTALPHA Function

Searches a string for a non-alphabetic character

```
col1 = notalpha('abc123xyz');

   *start in col 5: ;
col2 = notalpha('abc123xyz',5);

   *start in col 7;
col3 = notalpha('abc123xyz',7);

   *search right to left: ;
col4 = notalpha('abc123xyz',-7);
```

col1=4 col2=5 col3=0 col4=6

NOTALNUM, NOTPUNCT, NOTSPACE Functions

Same syntax as NOTDIGIT and NOTALPHA

NOTALNUM searches a character string for a non-alphanumeric character

NOTPUNCT searches a character string for a non-punctuation character

NOTSPACE searches a character string for a non-white-space character: blank, horizontal and vertical tab, carriage return, line feed, and form feed

PERL Regular Expressions

PRXPARSE Function

Compiles a Perl regular expression (PRX) that can be used for pattern matching of a character value.

PRXMATCH Function

Searches for a pattern match and returns the position at which the pattern is found

```
address='Ewa Beach, HI 96707-2235';
rc=prxparse('/\d{5}-\d{4}/');
col=prxmatch(rc,address);
if col>0 then put 'Zip +4 at col ' col;
```

Log: Zip +4 at col 15

Q & A