MWSUG 2016 – PO04

When ANY Function Will Just NOT Do

Richann Watson, Experis; Karl Miller, inVentiv Health

Function Search Criteria	ANY Function	NOT Function
alpha-numeric character	ANYALNUM(*string, <start>*)	NOTALNUM(*string, <start>*)
alphabetic character	ANYALPHA(*string, <start>*)	NOTALPHA(*string, <start>*)
digit	ANYDIGIT(*string, <start>*)	NOTDIGIT(*string, <start>*)
character that is valid as the first character of a SAS variable*	ANYFIRST(*string, <start>*)	NOTFIRST(*string, <start>*)
lowercase letter	ANYLOWER(*string, <start>*)	NOTLOWER(*string, <start>*)
character that is valid in a SAS variable*	ANYNAME(*string, <start>*)	NOTNAME(*string, <start>*)
punctuation character	ANYPUNCT(*string, <start>*)	NOTPUNCT(*string, <start>*)
white-space character: blank, horizontal and vertical tab, carriage return, line feed, and form feed	ANYSPACE(*string, <start>*)	NOTSPACE(*string, <start>*)
uppercase letter	ANYUPPER(*string, <start>*)	NOTUPPER(*string, <start>*)

* SAS variable name under VALIDVARNAME=V7

Start:

- optional.
- determine search direction
  - Start > 0, search is from left to right
  - Start < 0, search is from right to left
  - Start < negative length of string, search starts at the end of the string

Functions yield the position of the first encounter of the desired search. It returns a zero when one of the following is true:

- Search character is not found
- Start > length of the string
- Start = 0

Convert individual character date/time components into ISO 8601 format.

1. Determine which components has values other than a number. If it has a value other than a number, then it is assumed it is missing and denoted with a single dash.

```
if not(notdigit(&dttmvar.)) then _&dttmvar. = &dttmvar.;
else _&dttmvar. = '-';
```

2. Create date and time variables

```
isodt = catx("-", _year, _mon, _day);
isotm = catx(":", _hr, _min, _sec);
```

3. Determine if the time is completely missing (i.e., isotm = '-:-:-')

```
if notpunct(strip(isotm)) > 0 then _isotm =
   substr(isotm, 1, notpunct(strip(isotm), -length(isotm)));
else _isotm = ' ';
```

4. Combine date with the new time variable

```
   newdttm = catx("T", isodt, _isotm);
```

5. Determine if the date is complete. If ANYALPHA returns a value greater than 0, then ISO 8601 date is complete and no further processing

```
if anyalpha(strip(newdttm)) > 0 then NEWDTC = newdttm;
```

6. ANYALPHA in step 5 returns 0, then there is no time so need report up to last non-missing date component

```
… if notpunct(strip(newdttm))>0 then NEWDTC = substr(newdttm, 1,
notpunct(strip(newdttm), -length(newdttm)));
```

MON	DAY	YEAR	HR	MIN	SEC	…	ISODT	_ISOTM	…	NEWDTC
01	01	2016	01	30	45		2016-01-01	01:30:45		2016-01-01T01:30:45
01	01	2016	01	30	--		2016-01-01	01:30		2016-01-01T01:30
01	01	2016	01	30	UN		2016-01-01	01:30		2016-01-01T01:30
01	01	2016	01	--	UN		2016-01-01	01		2016-01-01T01
01	01	2016	UN	30			2016-01-01	-:30		2016-01-01T-:30
01	01	2016	UN	UK	NA		2016-01-01			2016-01-01
01	--	2016	UN	UK	NA		2016-01--			2016-01
NA	--	2016	UN	UK	NA		2016----			2016
UK	01	2016	o1	30			2016---01	-:30		2016---01T-:30
NA	01	2016	UK	30			2016---01	-:30		2016---01T-:30
--	01	2016	NA	30			2016---01	-:30		2016---01T-:30
01	--	2016	O1	30			2016-01--	-:30		2016-01--T-:30
01	NA	2016	01	30			2016-01--	01:30		2016-01--T01:30
UK	UK	2016	01	30			2016----	01:30		2016----T01:30
01	01	UNK	01	30			--01-01	01:30		--01-01T01:30
0I	01	NA		30			----01	-:30		----01T-:30
--	UK	UNK	01	30	45		-----	01:30:45		-----T01:30:45
--	UK	UNK		NA			-----			
11	19	UNK	UK	NA	--		--11-19			--11-19

It is very important to keep in mind what is actually being searched by the functions. For example, if you want to determine if a character can be converted to numeric , then you will need to ensure that there are no alphabetic characters in the value. You may be tempted to use **NOTALPHA**. However, **N**OTALPHA will return the position of the ***first non-alphabetic*** character and due to the case that some character results can contain both alphabetic and numeric characters, the use of NOTALPHA would yield a non-zero value for results that are alphanumeric and not strictly numeric.

Week Number and Day Number from Text

Building upon previous example we can extract week and day. Data comes in a variety of formats but what is consistent is that there is a number that represents the week that is preceded by characters, punctuation, and/or white space. This week number is then followed by more characters, punctuation, and/or white space, with the last character in the string being a number which represents the day.

1. Find the location of the first number when searching from the left.

```
firstnumloc = anydigit(visit);
```

2. Find the location of first alpha character when searching from left starting the search at the position of first number.

```
secalploc = anyalpha(visit, firstnumloc);
```

3. Find the location of first number when searching from right

```
lastnum = anydigit(visit, -length(visit));
```

4. Extract the week portion using the location of first number when searching from left (a.) and the location of first alpha character after the first number when searching from left (b.).

```
WEEK = input(substr(visit, firstnumloc,
                    secalploc - firstnumloc), best.);
```

5. Extract the day portion using the location of first number when searching from right (c.).

```
DAY = input(substr(visit, lastnum), best.);
```

VISIT	FIRSTNUMLOC	SECALPLOC	LASTNUM	WEEK	DAY
W4D1	2	3	4	4	1
w 4 d 1	3	5	7	4	1
W4 D1	2	4	5	4	1
w-4 d-1	3	5	7	4	1
wk4 d1	3	5	6	4	1
wk4 dy1	3	5	7	4	1
wk 4 dy 1	4	6	9	4	1
WK: 4 DY: 1	5	7	11	4	1
WEEK 4 DAY 1	6	8	12	4	1
week:4 day:1	6	8	12	4	1

For more details on the ANY and NOT functions and for the complete code on creating ISO 8601 dates using these functions refer to the following paper:

http://www.lexjansen.com/pharmasug/2016/QT/PharmaSUG-2016-QT16.pdf