

Top 10 SAS® Functions in 2017

A brief summary of SAS Communities Survey

- by Flora Fang Liu

What are SAS Functions? Why use SAS Functions?

- **What?**

- SAS functions perform computations, data manipulation, and enhancement
- •Character
- •Numeric
- •Dates/temporal
- •Mathematics

- **Why?**

- You need to transform raw/operational data into report-and analytics-ready structures
- •DATA step programming
- •SQL

Where to find SAS Functions

Function Dictionary

<http://support.sas.com/documentation/cdl/en/lefunctionsref/69762/PDF/default/lefunctionsref.pdf>

Dictionary	93
ABS Function	93
ADDR Function	94
ADDRLONG Function	95
AIRY Function	96
ALLCOMB Function	97
ALLPERM Function	99
ANYALNUM Function	101
ANYALPHA Function	104
ANYCNTRL Function	106
ANYDIGIT Function	107
ANYFIRST Function	109
ANYGRAPH Function	110
ANYLOWER Function	113
ANYNAME Function	114
ANYPRINT Function	116
ANYPUNCT Function	118
ANYSACE Function	120
ANYUPPER Function	122
ANYXDIGIT Function	124
ARCOS Function	126
ARCOSH Function	126
ARSIN Function	127
ARSINH Function	128
ARTANH Function	129
ATAN Function	130
ATAN2 Function	131
ATTRC Function	132
ATTRN Function	134
BAND Function	139
BETA Function	140
BETAINV Function	140
BLACKCLPRC Function	141
BLACKPTPRC Function	143
BLKSHCLPRC Function	145
BLKSHPTPRC Function	147
BLSHIFT Function	149
BNOT Function	149

DCLOSE Function	386
DCREATE Function	388
DEPDB Function	389
DEPDBSL Function	390
DEPSL Function	391
DEPSYD Function	392
DEPTAB Function	392
DEQUOTE Function	393
DEVIANCE Function	395
DHMS Function	399
DIF Function	400
DIGAMMA Function	402
DIM Function	402
DINFO Function	404
DIVIDE Function	410
DNUM Function	412
DOPEN Function	413
DOPTNAME Function	414
DOPTNUM Function	416
DOSUBL Function	417
DREAD Function	419
DROPNOTE Function	420
DSNAME Function	421
DSNCATLGD Function	422
DUR Function	423
DURP Function	424
EFFRATE Function	425
ENVLEN Function	426
ERF Function	427
ERFC Function	428
EUCLID Function	429
EXIST Function	430
EXP Function	433
FACT Function	433
FAPPEND Function	434
FCLOSE Function	436

Character manipulation

SUBSTR

Do you use it?
Yes | No

Extracts a substring from an argument -returns the characters from start to end
SUBSTR (string, position<, length>)

- string specifies any SAS character expression.
- position specifies a numeric expression that is the beginning character position.
- length specifies a numeric expression that is the length of the substring to extract.
- If you omit length, SAS extracts the remainder of the expression.

```
datanewinfo;
set info;
idnoC=put(idno, 4.);
divid=substr(idnoc, 4, 1);
```

idnoC	divid
1231	1
1251	1
1362	2
1751	1
1833	3
1543	3

SCAN

Do you use it?
Yes | No

Selects a given word from a character expression
SCAN (string, n<, delimiter(s)>)

- string specifies any character expression.
- n specifies a numeric expression that produces the number of the word in the character string you want SCAN to select.
- delimiter specifies a character expression that produces characters that you want SCAN to use as a word separator in the character string.

```
lname= scan(name, 1, ', ');
fname= scan(name, 2, ', ');
```

name	lname	fname
Jones, Sally	Jones	Sally
Miller, Barry	Miller	Barry
Smith, Laura	Smith	Laura
McAllister, Sean	McAllister	Sean
Johnson, Cynthia	Johnson	Cynthia
Schwarz, Kate	Schwarz	Kate

CATX

Do you use it?
Yes | No

Concatenates character strings, removes leading and trailing blanks, and inserts separators
CATX (separator, string-1 <, ...string-n>)

- separator specifies a character string that is used as a separator between concatenated strings.
- string specifies a SAS character string.
- The CATX function returns a value to a variable, or returns a value in a temporary buffer.

```
newname= catx(' ', fname, lname);
lname=scan(name, 1, ', ');
fname=scan(name, 2, ', ');
newname=catx(' ', fname, lname);
```

lname	fname	newname
Jones	Sally	Sally Jones
Miller	Barry	Barry Miller
Smith	Laura	Laura Smith
McAllister	Sean	Sean McAllister
Johnson	Cynthia	Cynthia Johnson
Schwarz	Kate	Kate Schwarz

FIND

Do you use it?
Yes | No

Searches for a specific substring of characters within a character string that you specify
FIND (string, substring<, modifiers>)

- string specifies a character constant, variable, or expression that will be searched for substrings.
- substring is a character constant, variable, or expression that specifies the substring of characters to search for in string.
- Modifiers is a character constant, variable, or expression that specifies one or more modifiers. The following modifiers can be in uppercase or lowercase:
 - I ignores character case during the search. If this modifier is not specified, FIND only searches for character substrings with the same case as the characters in substring.

Use the FIND function to identify a misspelled name and write a message to the SAS log.

```
if find(newname, 'Lau', 'i') > 0 then
  put 'value found ' newname;
```

Partial SAS Log

```
309 run;
value found Laura Smith
NOTE: There were 6 observations read from the data set WORK.INFO.
NOTE: The data set WORK.NEWINFO has 6 observations and 14 variables.
NOTE: DATA statement used (Total process time):
      real time          0.03 seconds
      cpu time           0.01 seconds
```

Temporal / date / time

DATEPART

Extracts the date from a SAS datetime value

DATEPART (datetime)

- Datetime specifies a SAS expression that represents a SAS datetimevalue.

Example:

Extract the SAS date portion of HIREDATE using the DATEPART function

```
newhire= datepart(hiredate);
```

```
newhire= datepart(hiredate);
```

hiredate	New SAS Date Value	New SAS Date Value with Format
01JAN01:12:09:03	14976	01JAN2001
15FEB05:09:34:42	16482	15FEB2005
01JUN10:10:06:05	18414	01JUN2010
23SEP15:02:39:14	20354	23SEP2015
15JAN00:01:03:44	14624	15JAN2000
22NOV98:03:56:03	14205	22NOV1998

Do you use it?
Yes | No

DATEDIF

Returns the number of days between two dates

DATDIF (sdate,edate,basis)

- sdate specifies a SAS date value that identifies the starting date.
- edate specifies a SAS date value that identifies the ending date.
- basis identifies a character constant or variable that describes how SAS calculates the date difference.
 - '30/360' or '360' specifies a 30 day month and a 360 day year.
 - 'ACT/ACT' or 'Actual' uses the actual number of days between dates.

Example:

Calculate the number of days (years) between the 2 new hire and termination dates

```
if newtermne .then do;
  dayofserv=datdif(newhire,newterm, 'ACT/ACT');
  yearofserv= yrdif(newhire, newterm, 'ACT/ACT');
end;
```

newterm	newhire	dayofserv	yearofserv
15848	14976	872	2.38904
.	16482	.	.
.	18414	.	.
20528	20354	174	0.47616
.	14624	.	.
.	14205	.	.

Do you use it?
Yes | No

Data conversion

PUT

PUT (source, format)

converts the expression to a character string -always returns a character value
- source argument can be character or numeric.

- format contains the SAS format that you want applied to the variable or constant that is specified in the source.

-To override the default alignment, you can add an alignment specification to a format:

- L left aligns the value.
- C centers the value.
- R right aligns the value.

Example:

Convert Numeric to Character

Since IDNO is stored as numeric, it must be converted to character via the PUT function.

```
idnoC= put(idno, 4.);

if find(newname, 'Lau', 'i')>0then
put 'value found ' newname;
```

idno	idnoC
1231	1231
1251	1251
1362	1362
1751	1751
1833	1833
1543	1543

Copyright © SAS Institute Inc. All rights reserved.

Do you use it?
Yes | No

INPUT

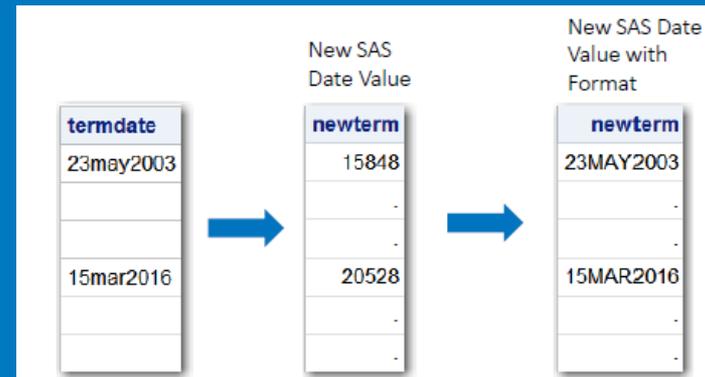
INPUT (character-expression, informat)

- Converts a string expression using the specified informat
- Often used to **convert character to numeric**

Example:

Convert TERMDATE to a SAS date using the INPUT function

```
newterm= input(termdate, date9.);
```



Do you use it?
Yes | No

Numeric

SUM

SUM Syntax

SUM (argument1, argument2, ...)

- Argument(s) are numeric.
- The argument list can consist of a variable list, which can be preceded by OF.
- If all the arguments have missing values, the result is a missing value.

```
totalcomp = sum(salary, bonus, merit);
```

salary	bonus	merit
35255.23	1250	1000
64159.68	675	.
44000.22	.	1500
55023.57	500	900
48000.68	.	500
53000.99	950	750



totalcomp
37505.23
64834.68
45500.22
56423.57
48500.68
54700.99

Do you use it?
Yes | No

ROUND

The ROUND function will express the total compensation as a whole number without decimal positions.

ROUND Syntax

Rounds the first argument to the nearest multiple of the second argument, or to the nearest integer when the second argument is omitted

ROUND (argument <,rounding-unit>)

- argument is a numeric constant, variable, or expression to be rounded.
- rounding-unit is a positive, numeric constant, variable, or expression that specifies the rounding unit.

```
totalcomp=sum(salary, bonus, merit);
totalcomp=round(totalcomp);
```

totalcomp
37505.23
64834.68
45500.22
56423.57
48500.68
54700.99



totalcomp
37505
64835
45500
56424
48501
54701

Do you use it?
Yes | No