IMS Health & Quintiles are now

IQVIA™

PROC FCMP

A Functional Replacement for Macros

Wouter van Wyk

Table of Contents

Function vs Subroutine

- Function
  - Named section of a program performing a specific task
- SAS Function
  - Performs computation on arguments
  - Returns a value
- CALL Routine/subroutine
  - Alters variables
  - Cannot use them in assignment statement or expressions

What is Proc FCMP?

- Create, Test and Store SAS functions (since SAS v9.1)
  - DATA step syntax

```
proc fcmp <outlib=Library.Dataset.Package> <inlib=Library.Dataset> <Options>;
  FUNCTION function-name(argument-1 <, argument-2, ...>) <VARARGS> <$> <length>
  <KIND | GROUP='string' ><LABEL='string-2'>;
  ... more-program-statements...
  RETURN (expression);
  ENDSUB;

  SUBROUTINE subroutine-name (argument-1 <, argument-2, ...>) <VARARGS>
  <KIND | GROUP='string'>;
  OUTARGS out-argument-1 <, out-argument-2, ...>;
  ... more-program-statements ...
  ENDSUB;
run;
```

Example 1: Study Day

```
options cmplib=work.funcs;
```

```
proc fcmp outlib=work.funcs.trial;
  FUNCTION study_day(reference_date, event_date);
    day = event_date - reference_date;
    if day >= 0 then day = day + 1;
    RETURN (day);
  ENDSUB;
run;
```

```
data example1;
   set example
   ady = STUDY_DAY(trtsdt, adt);
run;
```

Example 2: String functions in a where clause

```
options cmplib=work.functions;
```

```
proc fcmp outlib=work.functions.conv;
  FUNCTION str_eq(str1 $,str2 $) ;
    if UPCASE(STRIP(str1)) eq UPCASE(STRIP(str2)) then RETURN(1);
    if UPCASE(STRIP(str1)) ne UPCASE(STRIP(str2)) then RETURN(0);
  ENDSUB;
run;
```

```
data example1;
  set example;
  where str_eq(studyid,'r033812Gts3001'); /*Not Case Sensitive anymore*/
run;
```

Example 3: SQL in a DATA step

```
%Macro GetSubjDate;
  Proc sql;
     Select 'Select input(' || strip(name) || ',yymmdd10.) as lstDate from SDTM.' ||
              strip(memname) || " where usubjid =&Usubjid" as qry
     into :Runit Separated by " Union all "
     from dictionary.columns
     where LIBNAME = 'SDTM' and name ? 'ENDTC';

     Select put(Max(lstDate),Date9.) into :LstDate
     from (&Runit);
  quit;
%Mend GetSubjDate;
```

```
Proc fcmp outlib=work.functions.conv;
  FUNCTION Get_LastSubjectDate(usubjid $) $;
    length LstDate $9;
    rc=run_macro('GetSubjDate',usubjid,LstDate); /*THE RUN_MACRO STATEMENT CALLS GetSubjDate Macro*/
    return(LstDate);
  ENDSUB;
run;
```

```
data example1;
  set example;
  lstdat = Get_LastSubjectDate(usubjid);
run;
```

Example 4: Function vs Subroutine

```
proc fcmp outlib=funcsol.functions.conversions;
  function BMI(lb,ht) ;
    return((lb*703)/(ht*ht));
  endsub;
run;
```

```
options cmplib=(funcsol.functions);
```

```
data bmi;
  set sashelp.class(keep=name age weight height);
  BMI = bmi(weight,height);
run;
```

Example 4: Function vs Subroutine

```
proc fcmp outlib=funcsol.functions.conversions;
  subroutine BiomassIndex(w,h,b) ;
    outargs b;
    b=((w*703)/(h*h));
  endsub;
run;
```

```
options cmplib=(funcsol.functions);
```

```
data bmi;
  set sashelp.class(keep=name age weight height);
  BMIndex=.;
  call biomassindex(weight,height,BMIndex);
run;
```

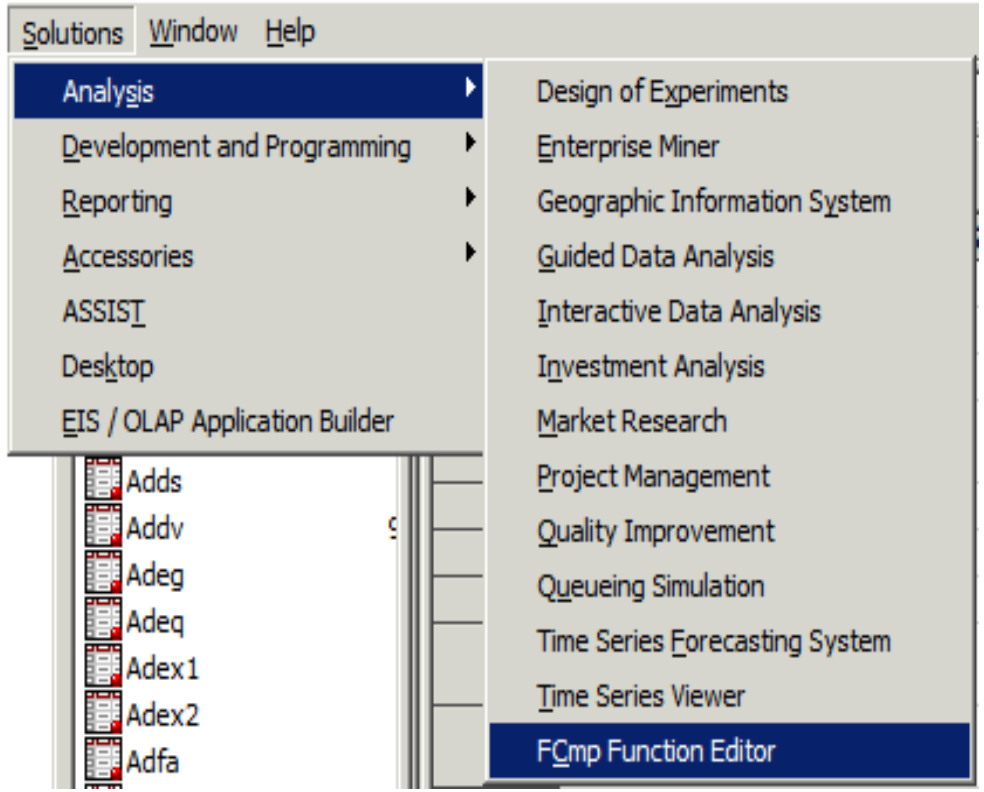
Options

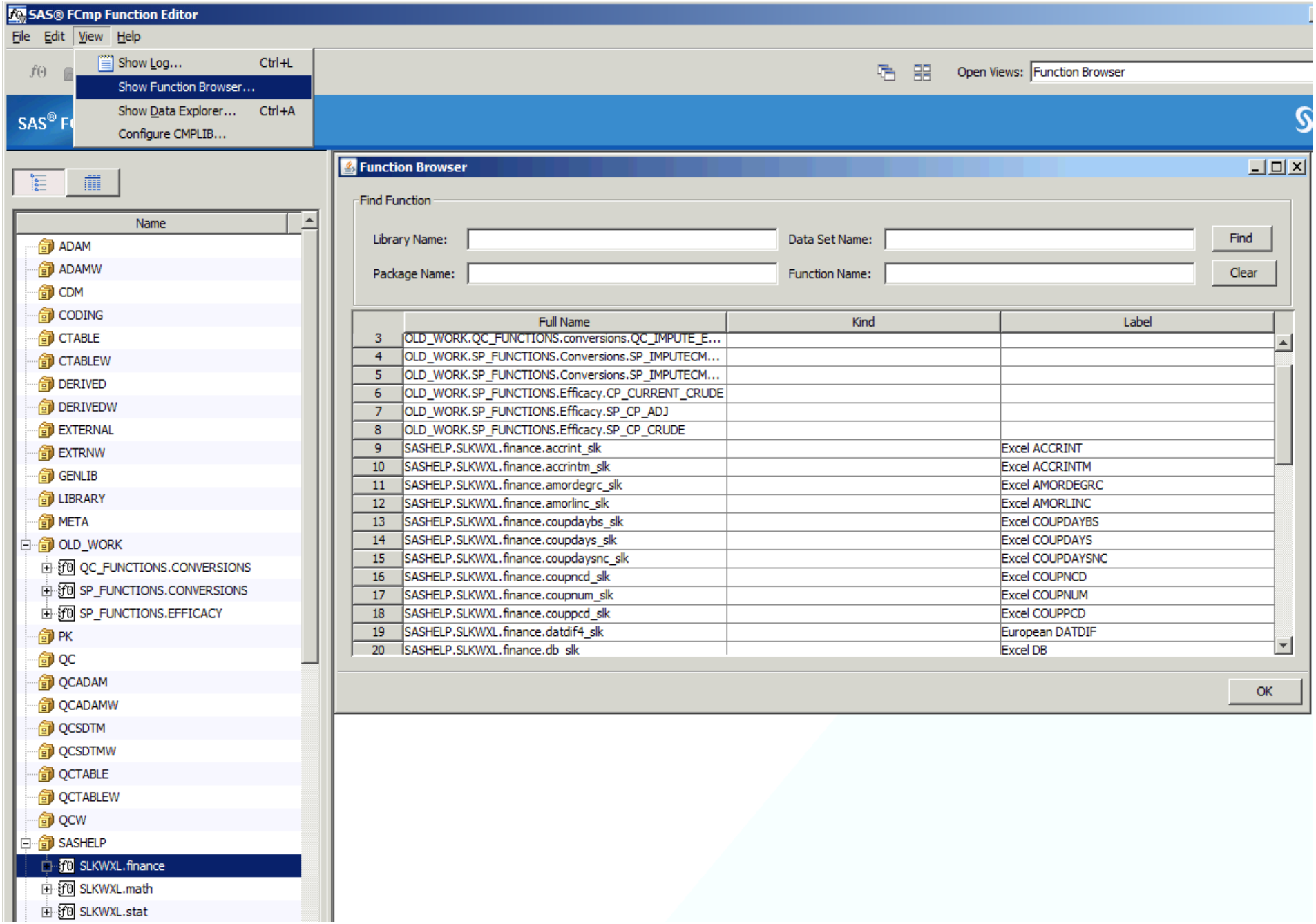
- ENCRYPT
- HIDE
  - specifies to encode the source code in a data set
- FLOW
  - specifies printing a message for each statement in a program as it is executed. This option produces extensive output
- LISTFUNCS
  - Specifies that prototypes for all visible FCMP functions or subroutines be written to the SAS listing.

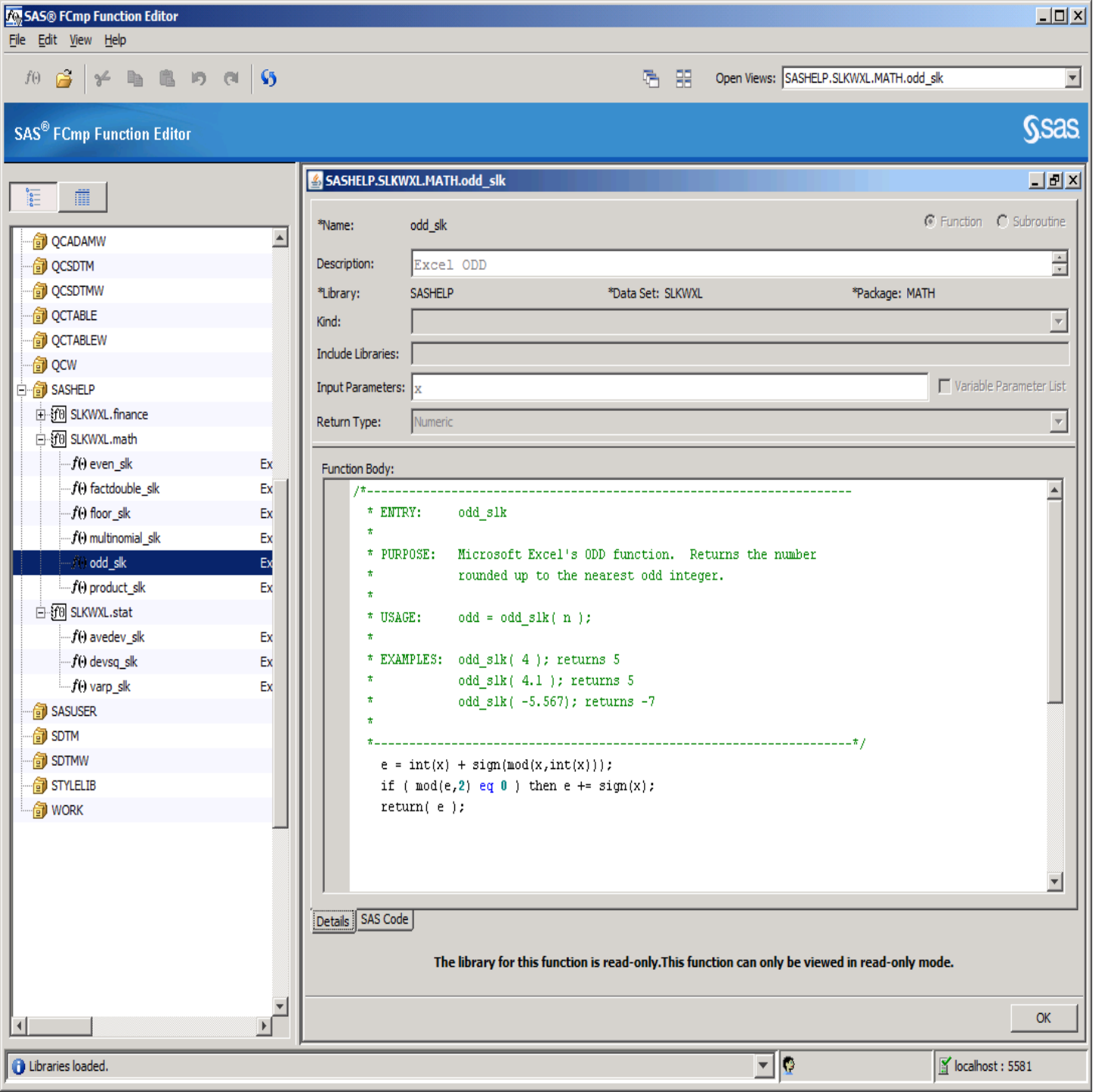
Use

- You can use the functions and subroutines that you create in PROC FCMP with the DATA step, the WHERE statement, the Output Delivery System (ODS), and among the following procedures:
  - PROC FORMAT,
  - PROC PHREG,
  - PROC REPORT COMPUTE blocks,
  - PROC SQL (functions with array arguments are not supported)

FCMP Function Editor




SAS® FCmp Function Editor
File Edit View Help
Show Log... Ctrl+L
Show Function Browser...
Show Data Explorer... Ctrl+A
Configure CMPLIB...
Open Views: Function Browser
Function Browser
Find Function
Library Name:
Data Set Name:
Find
Package Name:
Function Name:
Clear
Full Name
Kind
Label
3 OLD_WORK.QC_FUNCTIONS.conversions.QC_IMPUTE_E...
4 OLD_WORK.SP_FUNCTIONS.Conversions.SP_IMPUTECM...
5 OLD_WORK.SP_FUNCTIONS.Conversions.SP_IMPUTECM...
6 OLD_WORK.SP_FUNCTIONS.Efficacy.CP_CURRENT_CRUDE
7 OLD_WORK.SP_FUNCTIONS.Efficacy.SP_CP_ADJ
8 OLD_WORK.SP_FUNCTIONS.Efficacy.SP_CP_CRUDE
9 SASHELP.SLKWXL.finance.accrint_slk Excel ACCRINT
10 SASHELP.SLKWXL.finance.accrintm_slk Excel ACCRINTM
11 SASHELP.SLKWXL.finance.amordegrc_slk Excel AMORDEGRC
12 SASHELP.SLKWXL.finance.amorlinc_slk Excel AMORLINC
13 SASHELP.SLKWXL.finance.coupdaybs_slk Excel COUPDAYBS
14 SASHELP.SLKWXL.finance.coupdays_slk Excel COUPDAYS
15 SASHELP.SLKWXL.finance.coupdaysnc_slk Excel COUPDAYSNC
16 SASHELP.SLKWXL.finance.coupncd_slk Excel COUPNCD
17 SASHELP.SLKWXL.finance.coupnum_slk Excel COUPNUM
18 SASHELP.SLKWXL.finance.couppcd_slk Excel COUPPCD
19 SASHELP.SLKWXL.finance.datdif4_slk European DATDIF
20 SASHELP.SLKWXL.finance.db_slk Excel DB
OK
Name
ADAM
ADAMW
CDM
CODING
CTABLE
CTABLEW
DERIVED
DERIVEDW
EXTERNAL
EXTRNW
GENLIB
LIBRARY
META
OLD_WORK
QC_FUNCTIONS.CONVERSIONS
SP_FUNCTIONS.CONVERSIONS
SP_FUNCTIONS.EFFICACY
PK
QC
QCADAM
QCADAMW
QCSDTM
QCSDTMW
QCTABLE
QCTABLEW
QCW
SASHELP
SLKWXL.finance
SLKWXL.math
SLKWXL.stat


SAS® FCmp Function Editor
File Edit View Help
Open Views: SASHELP.SLKWXL.MATH.odd_slk
SAS® FCmp Function Editor
QCADAMW
QCSDTM
QCSDTMW
QCTABLE
QCTABLEW
QCW
SASHELP
SLKWXL.finance
SLKWXL.math
even_slk
factdouble_slk
floor_slk
multinomial_slk
odd_slk
product_slk
SLKWXL.stat
avedev_slk
devsq_slk
varp_slk
SASUSER
SDTM
SDTMW
STYLELIB
WORK
SASHELP.SLKWXL.MATH.odd_slk
*Name: odd_slk
Function
Subroutine
Description: Excel ODD
*Library: SASHELP
*Data Set: SLKWXL
*Package: MATH
Kind:
Include Libraries:
Input Parameters: x
Variable Parameter List
Return Type: Numeric
Function Body:
/*----------
* ENTRY: odd_slk
*
* PURPOSE: Microsoft Excel's ODD function. Returns the number
* rounded up to the nearest odd integer.
*
* USAGE: odd = odd_slk(n);
*
* EXAMPLES: odd_slk(4); returns 5
* odd_slk(4.1); returns 5
* odd_slk(-5.567); returns -7
*
----------/
e = int(x) + sign(mod(x,int(x)));
if (mod(e,2) eq 0) then e += sign(x);
return(e);
Details
SAS Code
The library for this function is read-only.This function can only be viewed in read-only mode.
OK
Libraries loaded.
localhost : 5581

Advantages

- Function archive can be built that can be used across studies to streamline efficiency.
- Complex programs can be simplified by using user defined functions.
- User defined functions are independent from their use.
- You can still use macros with functions.
- Functions help to keep code clean.
- Better debugging than MACROS
- More intuitive
- Disadvantage: Has not been widely adopted yet, limited information available compared to other SAS topics

Function vs Macros

- Another tool in your SAS toolbox
- Using macros inappropriately
- Function does not alter any variables in a dataset
- Possibly use functions as some of the building blocks in a macro
- Functions more intuitive to new SAS users.