%Assert Your Way to Sleep-Filled Nights: A One-Line Data Validation Macro

Boston Area SAS User Group
December 3, 2015

Quentin McMullen
quentin.mcmullen@siemens.com
Siemens Healthcare Diagnostics, Inc.

Simple DATA Step

```
data want ;
  set have ;
  Age = (ConsentDt - DOB) / 365.25 ;
run ;
```

Check Tables

```
proc freq data=want ;
  tables Age*ConsentDt*DOB /
    missing list ;
run ;

proc means data=want n min mean max ;
  var Age ConsentDt DOB ;
run ;
```

Logical Assertion

```
data want ;
set have ;
if NOT
 ("01JAN2011"d<=ConsentDt<=today() )
 then putlog "ERROR: bad ConsentDt" ;
if NOT
 ("01JAN1900"d <= DOB <= today() )
 then putlog "ERROR: bad DOB" ;
Age = (ConsentDt - DOB) / 365.25 ;
if NOT (18 <= Age <= 100 )
 then putlog "ERROR: bad Age" ;
run ;
```

Logical Assertion

```
data want ;
set have ;
if NOT
 ("01JAN2011"d<=ConsentDt<=today() )
 then putlog "ERROR: bad ConsentDt" ;
if NOT
 ("01JAN1900"d <= DOB <= today() )
 then putlog "ERROR: bad DOB" ;
Age = (ConsentDt - DOB) / 365.25 ;
if NOT (18 <= Age <= 100 )
 then putlog "ERROR: bad Age" ;
run ;
```

Logical Assertion

```
data want ;
set have ;
if NOT
 ("01JAN2011"d<=ConsentDt<=today() )
 then putlog "ERROR: bad ConsentDt" ;
if NOT
 ("01JAN1900"d <= DOB <= today() )
 then putlog "ERROR: bad DOB" ;
Age = (ConsentDt - DOB) / 365.25 ;
if NOT (18 <= Age <= 100 )
 then putlog "ERROR: bad Age" ;
run ;
```

Logical Assertion

```
data want ;
set have ;
if NOT
 ("01JAN2011"d<=ConsentDt<=today() )
 then putlog "ERROR: bad ConsentDt" ;
if NOT
 ("01JAN1900"d <= DOB <= today() )
 then putlog "ERROR: bad DOB" ;
Age = (ConsentDt - DOB) / 365.25 ;
if NOT (18 <= Age <= 100 )
 then putlog "ERROR: bad Age" ;
run ;
```

Log

```
113     Age = (ConsentDt - DOB) / 365.25 ;
114     if NOT (18 <= Age <= 100 )
115      then putlog "ERROR: bad Age" ;
116   run ;
```

```
ERROR: bad Age
NOTE: There were 100 observations read
   from the data set WORK.HAVE.
NOTE: The data set WORK.WANT has 100
   observations and 4 variables.
```

Logical Assertion

```
data want ;
set have ;
if NOT
%assert
 ("01JAN2011"d<=ConsentDt<=today())
('01JAN2011"d<= ConsentDt<= today()")
 then putlog "ERROR: bad ConsentDt" ;
if NOT
%assert
 ("01JAN1900"d <= DOB <= today() )
('01JAN1900"d <= DOB <= today() )
 then putlog "ERROR: bad DOB" ;
Age = (ConsentDt - DOB) / 365.25 ;
if NOT (18 <= Age <= 100 )
%assert ( 18 <= Age <= 100 )
 then putlog "ERROR: bad Age" ;
run ;
```

%Assert()

```
%macro assert (assertion) ;

if NOT (&assertion) then putlog
 "ERROR: Assertion
 (%superq(assertion)) is FALSE." ;

%mend assert ;
```

%Assert Calls

```
data want ;
set have ;

%assert
("01JAN2011"d<= ConsentDt<= today() )

%assert
("01JAN1900"d<= DOB <= today() )

Age = (ConsentDt - DOB) / 365.25 ;

%assert ( 18 <= Age <= 100 )
run ;
```

Log

```
48   %assert
49   ("01JAN2011"d<= ConsentDt<= today() )
50   %assert
51   ("01JAN1900"d<= DOB <= today() )
52   Age = (ConsentDt - DOB) / 365.25;
53   %assert ( 18 <= Age <= 100 )
54 run ;
ERROR: Assertion (18 <= Age <= 100) is
FALSE.
NOTE: There were 100 observations read
from the data set WORK.HAVE.
```

%Assert Examples

```
data …;
%assert( 18<=age<=100 )
%assert( put(age,agerange.) ne "INVALID" )
%assert( not missing(salary) )
%assert( Gender IN ('M','F') )
%assert( a=1 )
%assert( (a=1) )
%assert( cmiss (of _all_)=0 )
%assert( exist("mylib.mydata") )
%assert( first.id and last.id )
%assert( (BMI=Weight/Height**2)  )
%assert( (Gender=M) = (ProstCancer IN (0,1) )
```

Communicative Code

```
data c ;
  merge a
        b
  ;
  by id ;
run ;
```

Communicative Code

```
data c ;
  merge a (in=a)
        b (in=b)
  ;
  by id ;

  %assert(a and b)
  %assert(first.id and last.id)

run ;
```

Enhanced %Assert()

- MSG=

```
%assert(18 <= Age <= 100
        ,msg="Bad Age" id= age=
         )
```

- Errors=
- Errorflag=
- Global macro var switch on/off

Conclusion

%ASSERT() is:

- A debugging tool
- A run-time validation tool
- A communication tool

All of which lead to greater confidence in your results!

NESUG 2012 paper with full macro code:
http://www.lexjansen.com/nesug/nesug12/cc/cc31.pdf

Acknowledgement