

Computing Age Correctly

William Kreuter

University of Washington

This macro returns an integer representing an age calculated according to common usage—*i.e.*, incremented exactly on a person's birthday, and always correct on all days in all leap years and non-leap years. The macro could also be used to return any other number of anniversaries which have passed since some event.

The birthday should be the first argument and the date of interest, on which the age will be reckoned, should be the second. The macro call should be on the right-hand side of an assignment statement. There is no error checking, so the user should ensure that the first argument is always chronologically earlier than the second.

In the following example, the variable AGE will be assigned 42. Either or both arguments may, of course, be variables, expressions or a function (*e.g.*, `today()`), instead of constants—but both arguments must yield SAS date values.

```
age = %age('24mar55'd, '01oct97'd);
```

The macro is my original work. **I can supply this in machine-readable form: inquire at billyk@u.washington.edu.** Code follows.

```
/* returns age in integer years on &DATE elapsed since &BIRTH */  
%macro age(date,birth);  
  floor(({intck('month',&birth,&date)-  
    (day(&date)<day(&birth)))/12})  
%mend age;
```