

**Paper CC03-2011**

**USING Your Age in People Years: Not All Formulas Are the Same**

**Art Carpenter, California Occidental Consultants, Anchorage, AK**

**ABSTRACT**

The calculation of an individual's age can be problematic. SAS® dates are generally measured in terms of days (or seconds if a datetime value is used), so we have to convert the value to years. To some extent, how we calculate age will depend on how we intend to use the value. The society's concept of 'age' is different than the mathematical concept. Age in years is further complicated by the very definition of a year as one rotation of the earth around the sun. This period does not convert to an integer number of days, and it is therefore further complicated by leap years.

Since SAS does not contain an AGE function, we need a formula for the conversion. And there are several to choose from. Which should we use?

Keywords: Age, INTCK, YRDIF, datetime, time, birthday

**INTRODUCTION**

In our society we get 'credit' for a year of life on our birthday. Age, therefore is always an integer which is incremented only on our birthday (this creates what is essentially a step function). When we want to use age as a continuous variable, say as a covariate in a statistical analysis, we would lose potentially valuable information using society's definition. Instead we want a value that has at least a couple of decimal places of accuracy, and takes on the characteristics of a continuous variable rather than those of a step function.

The following examples calculate patient's age, using seven different formulas, at the date of their death (this is all made up data - no one was actually harmed in the writing of this paper). While not strictly true, for the most part the calculated values agree at the units level. But there **are** differences – not all formulas are the same.

**Calculating Age**

DOB	DEATH	age1	age2	age3	age4	age5	age6	age7
21NOV31	13APR86	54.3929	54.4301	55	54	55	54.3918	54
03JAN37	13APR88	51.2745	51.3096	51	51	51	51.2759	51
19JUN42	03AUG85	43.1239	43.1534	43	43	43	43.1233	43
19JAN42	03AUG85	43.5373	43.5671	43	43	43	43.5370	43
23JAN37	13JUN88	51.3867	51.4219	51	51	51	51.3878	51
18OCT33	21JUL87	53.7550	53.7918	54	53	54	53.7562	53
17MAY42	03SEP87	45.2977	45.3288	45	45	45	45.2986	45
07APR42	03AUG87	45.3224	45.3534	45	45	45	45.3233	45
01NOV31	13APR86	54.4476	54.4849	55	54	55	54.4466	54
18APR33	21MAY87	54.0890	54.1260	54	54	54	54.0904	54
18APR43	21MAY87	44.0903	44.1205	44	44	44	44.0904	44

As an aside, if you are going to use the age in years as a continuous variable in an analysis such as a regression or analysis of covariance, there is no real advantage (other than a change in units) in converting from days to years. Consider using age in days to avoid the issues associated with the conversion to years.

## SIMPLE FORMULAS

When you need to determine age in years and you want a fractional age (continuous values), a fairly well accepted industry standard approximates leap years with a quarter day each year.

```
age1 = (death - dob) / 365.25;
```

Depending on how leap years fall relative to the date of death and birth, the approximation could be off by as much as what is essentially two days over the interval. Over a person's lifetime, or even over a period of just a few years, two days will cause an error in at most the third decimal place.

There are several other, somewhat less accurate, variations on this formula for age in years.

## USING FUNCTIONS

The INTCK function counts the number of intervals between two dates. When the selected interval is 'year' it returns an integer number of years. Since by default this function always measures from the start of the interval, the resulting calculation would be the same as if the two dates were both first shifted to January 1. This means that the result will ignore dates of birth and death and could be incorrect by as much as a full year. In SAS9.2 the INTCK function supports an alignment option that mitigates this behavior to some extent.

```
age5 = intck('year',dob,death);
```

Unlike the INTCK function the YRDIF function does not automatically shift to the start of the interval and it partially accounts for leap years. This function was designed for the Securities Industry to calculate interest for fixed income securities based on industry rules, and returns a fractional age.

```
age6 = yrdif(dob,death,'actual');
```

Note the use of the third argument! YRDIF also has the ability to measure intervals using 30 day months and 360 day years. While these standardized intervals have value in the securities industry, for analysis we generally will want to use the actual number of days per year and consequently the third argument is set to 'actual'. Please be careful to notice for the third argument that, since there is more than one possible entry that starts with the letters 'act', 'act' is **not** an acceptable abbreviation for 'actual'.

Because of its specialized intended use, YRDIF does not handle leap days in the way that we would hope for when calculating age. As is shown in the following example the leap day is not counted as a full day.

```

data year;
test2000 = yrdif('07JAN2000'd,'07JAN2001'd,"ACTual");
test2001 = yrdif('07JAN2001'd,'07JAN2002'd,"ACTual");
put test2000=;
put test2001=;
run;

```

Year 2000 was a leap year and year 2001 was not. Correctly calculated we would

```

test2000=1.0000449135
test2001=1

```

expect both TEST2000 and TeST2001 to have a value of 1.0.

Like the formula for AGE1 shown above, the leap day is being averaged across four years. If we were to examine

```

test2004 = yrdif('07JAN2000'd,'07JAN2004'd,"ACTual");
put test2004=;

```

a full four year period (with exactly one leap day), YRDIF returns the correct

age in years (age=4.0).

As would be generally appropriate, the YRDIF function does not include the last day of the interval (the date of the second argument) when counting the total number of days. If the date represented in the second argument is to be included in the interval an extra day should be added. The equation becomes:

```

age6b = yrdif(dob,death+1,'actual');

```

Although over a person's lifetime I feel that the differences are minor, SAS Institute *strongly* recommends that YRDIF not be used to calculate age. <http://tsdsrv05.unx.sas.com:7777/iw/docs/sasnotes/fusion/36/977.html> They do not, however offer an alternative other than the one discussed next.

## THE WAY OUR SOCIETY MEASURES AGE

Society thinks of age as whole years, with 'credit' withheld until the date of the anniversary of the birth. The following equation (first proposed by William Kreuter) measures age in whole years. It counts the months between the two dates, subtracts one month if the day boundary has not been crossed for the last month, and then converts months to years.

```

age7 = floor(( intck( 'month', dob, death)
               - ( day(death) < day(dob)))/ 12);

```

This formula always yields an integer.

## COMMENTS and CAVEATS

It is real easy to get caught up in the minutia of these equations. Ultimately it comes down to a choice between a continuous age (fractional years) and age as a step function. If you want a continuous age, there is little practical difference between the equations for AGE1 and AGE6. Both will result in values that seldom vary in the second decimal place. The values become even closer in agreement for larger ages. Even ignoring the leap years altogether, as in AGE2, only introduces an inaccuracy of 1 day in four years (1/1461).

The equations shown in this paper, and indeed how we measure age in general, have issues with birthdays that fall on February 29. In non-leap years when is credit to be assigned (when is the birthday celebrated) – on February 28 or March 1? While we can debate the issue, it has little practical application in analysis – the difference is after all one day over the lifetime.

## **SUMMARY**

There is more than one way to measure the age of the patients in the study and it is important that, as analysts we select the correct definition for our purposes. When we use age as a continuous variable there are multiple choices for how we perform the calculations, and the results are NOT the same.

We have choices, but they need to be informed choices.

## **ABOUT THE AUTHOR**

Art Carpenter's publications list includes four books, and numerous papers and posters presented at SUGI, SAS Global Forum, and other user group conferences. Art has been using SAS® since 1977 and has served in various leadership positions in local, regional, national, and international user groups. He is a SAS Certified Advanced Professional programmer, and through California Occidental Consultants he teaches SAS courses and provides contract SAS programming support nationwide.

## **AUTHOR CONTACT**

Arthur L. Carpenter  
California Occidental Consultants  
10606 Ketch Circle  
Anchorage, AK 99515

(907) 865-9167  
[art@caloxy.com](mailto:art@caloxy.com)  
[www.calaxy.com](http://www.calaxy.com)

## **ACKNOWLEDGEMENTS**

The data containing the sample dates used in this paper's calculations was provided by Kirk Lafler of Software Intelligence Corporation. The motivation for writing on this topic came out of conversations with William Kreuter at a PNWSUG conference.

## **REFERENCES**

A well written explanation of the calculation of age and the issues associated with those calculations can be found in Sample Tip 24808 by William Kreuter (2004). <http://support.sas.com/kb/24/808.html>

Further discussions of the limitations of the YRDIF function can be found at:

<http://support.sas.com/kb/3/036.html>  
<http://support.sas.com/kb/36/977.html>

A variety of forum discussions on age calculations have taken place. One discussion which includes a contribution by Mike Zdeb , in which he writes a nice summary and several links.

<http://www.mathkb.com/Uwe/Forum.aspx/sas/44490/Date-difference>.

A discussion of Ian Whitlock's macro function that utilizes William Kreuter's formula can be found at: Chung, Chang Y. and Ian Whitlock, "%IFN – A Macro Function", presented at SUGI 31.

<http://www2.sas.com/proceedings/sugi31/042-31.pdf>. The following references can be found in this paper.

Some of the original discussions of William Kreuter's age calculation formula can be found at:

Kreuter, William (1998). "Sample 271: Accurately Calculating Age with Only One Line of Code." A SAS Sample posted on support.sas.com. <http://support.sas.com/ctx/samples/index.jsp?sid=271>.

Kreuter, William (1993). "Re: Computing Age in years." A SAS-L posting. Archived at Marist.

<http://vm.marist.edu/htbin/wltype?SAS-L.24262>.

Muhlbauer, Lawrence H. (1996). "Re: strictly correct age calculation?" A SAS-L posting on May 7, 1996. Archived at <http://www.listserv.uga.edu/cgi-bin/wa?A2=ind9605A&L=sas-l&P=R9691&D=1&H=0&O=D&T=1>.

#### **TRADEMARK INFORMATION**

SAS, SAS Certified Professional, SAS Certified Advanced Programmer, and all other SAS Institute Inc. product or service names are registered trademarks of SAS Institute, Inc. in the USA and other countries.

® indicates USA registration. Other brand and product names are trademarks of their respective companies.