Paper 44 - 2018

A Walk through Time: Growing Your SAS® Career

Arthur L. Carpenter
California Occidental Consultants, Anchorage, Alaska

ABSTRACT

The measurement of time is an integral component of most of our data sets. We note the date of the patient's visit and both the date and the time of the administration of a test. The accurate calculation of time intervals, such as the time between drug delivery and onset of an adverse event, becomes critically important. Fortunately SAS ships with a diverse set of tools for working with dates and times.

As SAS programmers and as users of SAS Software, time measurement is important, not only in the successful analysis of our data, but also as a gauge in the growth of our career. To be successful we must be able to measure and work with time values, but time is also a measure of the progression of our career. This paper introduces a number of measures of time, a variety of analytic techniques applied to date and time values, interwoven with a discussion of the Grand Canyon, the Great Wall of China, and the continued growth of our professional knowledge.

KEYWORDS

INTCK, INTNX, MDY, Timeline, SAS Programming Career Development

INTRODUCTION

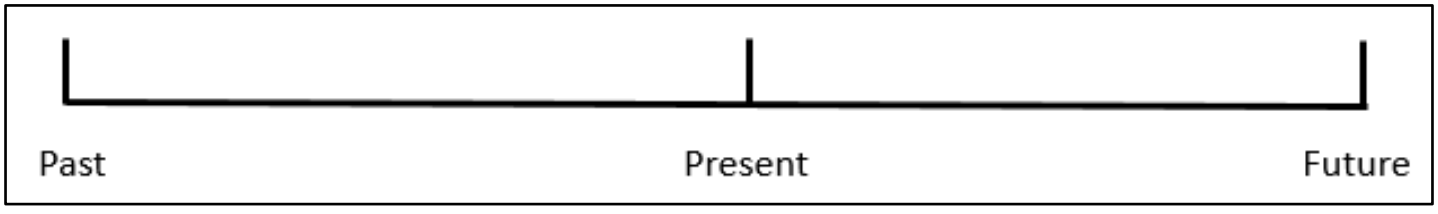
Time is measured in a number of ways and we make use of those measurements using a variety of tools. Without proper measurements and without the proper tools to take advantage of those measurements, we would be unable to analyze a number of important aspects of our data.

Internally SAS stores date, time, and datetime as displacement values using a time line system with an arbitrary zero point. These stored values have no intrinsic meaning relative to our calendar or time keeping systems, however we must be able to use and present these values in such a way so that we and our readers will be able to make sense of them in our reports and analyses.

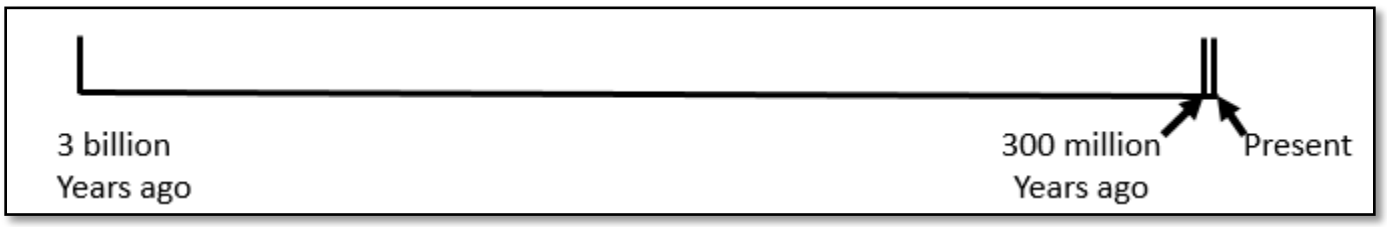
Some of the tools that we make use of when analyzing date and time values using SAS are straightforward while others are more complex and sophisticated. When we are first learning to program in SAS we tend to use the simpler tools, and as we gain in sophistication so do our techniques, but gaining the skills to take full advantage of SAS takes time.

IN THE BEGINNING

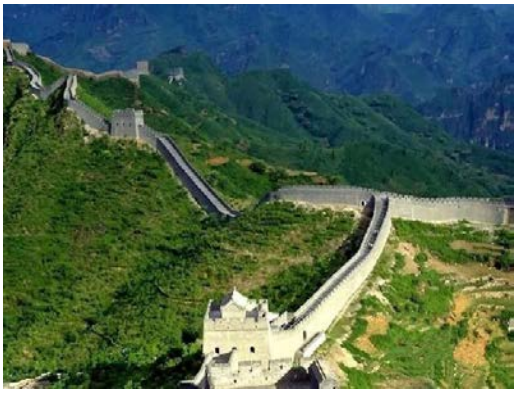
Often when we measure time we think in terms of past – present – future. Analytically we do this with a time line. This device allows us to look both forward and backward in time through the simple device of addition and subtraction.

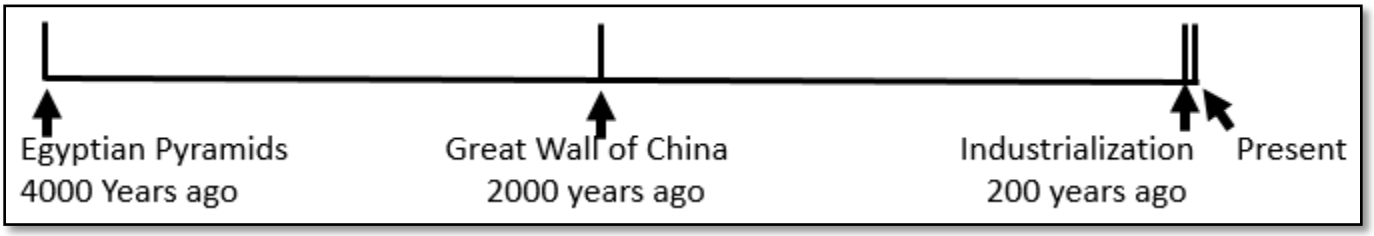


The Grand Canyon of Arizona cuts through rocks that were laid down on an ocean floor over 3 billion years ago. Here some of the oldest rocks on Earth are exposed, and by inspecting the geology of the earth we know that in ages past dinosaurs roamed the earth. Fossil records of these animals, which are over 300 million years old, have been found in rocks on every continent of earth. Compared to the history of mankind this time span is virtually impossible to comprehend. A time line can be used to help us visualize this vast span of time.

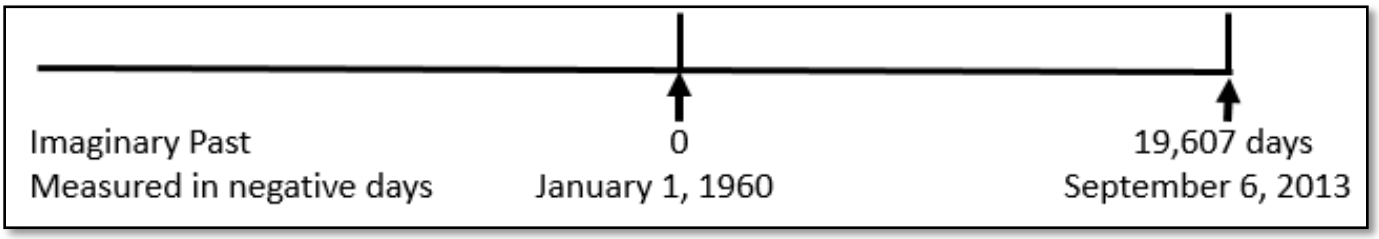


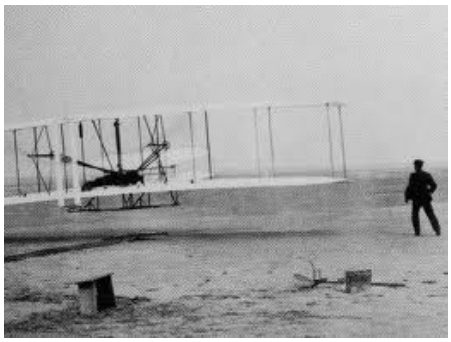
Since very few of our clinical trials, even those studying geriatrics, work with dinosaurs, we can safely restrict our timeline to something a bit more manageable. Early recorded human history includes several major time points, such as the construction of the Egyptian Pyramids around 4000 years ago, and the Great Wall of China which was constructed over 2000 years ago. Placing these events on a time line can help us put the modern age into perspective. One might consider marking the start of the modern age with the industrial revolution which transformed Europe and much of North America in the 19th and early 20th centuries.





For most of our analytic work, even 50 years ago is ancient history. Each of the previous examples assumes a distant past. In fact we now know differently; we now know that the world began on what our calendar calls January 1, 1960.





SAS recognizes this beginning point in time, and uses a similar time line system with the start of time as the zero point. This of course means that, since SAS dates are measured in days since the beginning of time, there really were no dinosaurs.

On September 6, 2013 there had been 19,607 days since the beginning of time. An historian might tell you that the Wright brothers flew at Kitty Hawk, NC on December 17, 1903, however you now know that would be day -20,469 (over 20,000 days before the beginning of time),and therefore this is another event that never actually occurred (fake news). We can put the date on the time line, but that alone does not make it real.

SAS DATES AND THE GREGORIAN CALENDER

There are many ways to determine dates. These competing calendar systems have been developed by different cultures around the world over the last 4000 years. Currently most of the world uses the Gregorian calendar system. Established in 1584 this calendar system allows for leap days to accommodate the length of the Earth's orbit. Conversion between SAS's calendar system (days since January 1, 1960) and our calendar is critical. Few of us have a boss that will find a report date of 15, 437 to be acceptable.

One way to enter dates into SAS is through date constants. The date must be in either DATE 7 (two digit year – not recommended) or DATE9 form. The date string is quoted and immediately followed with a 'D'. The LOG will show the internal value of the date.

```
data _null_;
today='06sep2013'd;
put today= today date9.;
run;
```

```
35   put today= today
date9.;
36   run;

today=19607 06SEP2013
```

You will also encounter dates in raw files. Here the date string is converted into a SAS date through the use of the MMDDYY10. Informat. The colon allows the informat to 'float'. Notice that the VISDT values are now stored as SAS dates. We would need to apply a format such as WORDDATE. if we wanted to see the formatted value.

```
data new;
input id $ visdt :mmddyy10.;
datalines;
A123 05/15/2011
q1 12/31/2012
run;
```

Using an INFORMAT

Obs	id	visdt
1	A123	18762
2	q1	19358

Sometimes our data will have variables that contain the values of the month, day, and year. SAS can calculate the SAS date value through the use of the MDY function. As the name implies, this function accepts the numeric values of the month, the day, and the year. The function then returns the SAS date.

```
data _null_;
day=6;
mon=9;
yr =2013;
date = mdy(mon,day,yr);
put date= date date9.;
run;
```

```
16   put date= date date9.;
17   run;

date=19607 06SEP2013
```

Once we have values stored as SAS dates we can use them to ask time line kinds of questions such as how old is the patient in days? The difference between two dates is a simple subtraction. This makes the calculation both simple and fast. As you become more sophisticated in your programming skills, you will discover a number of other calculation tools. Elapsed days are easy because the base unit of measurement is days, but what if you want number of years between two dates. The AGEYRS variable shown here is one attempt at this measurement, with 365.25 days approximating the length of a year. If you are studying persons with ages of several hundred years, it will be slightly more accurate to divide by 365.2425. As it turns out there are other tools within SAS that can also help us with this type of calculation.

```
data _null_;
today='06sep2013'd;
dob = '04jun1978'd;
agedays = today - dob;
ageyrs = (today-dob)/365.25;
put agedays= ageyrs=;
run;
```

```
54   put agedays= ageyrs=;
55   run;

agedays=12878 ageyrs=35.258042437
```

The INTCK function returns the number of interval boundaries between two dates. Using the same two dates as before, the INTCK function shown here will return the same integer value (35) for all dates of birth in 1978 and all values of TODAY in 2013. Because it counts boundaries and not days within a year, this function effectively always measures from January 1st. Fortunately this function has a number of options that can make it more useful, but we must be careful.

```
yrsINTCK = intck('year',dob,today);
```

A more accurate calculation can be achieved through the use of the YRDIFF function. Although not originally intended for this use, it can now be used to calculate age. When using this function to calculate age in years, be sure that the third argument is specified as 'ACT/ACT', 'ACTUAL', or 'AGE'. 'ACT' is *not* an acceptable abbreviation.

```
yrsActual=yrdif(dob,today,'Actual');
yrsAGE = yrdif(dob,today,'Age');
```

Dates can also be stored and used by the macro language. Since macro variables store text, it is their usage that determines how the stored value will be interpreted. Although either of the values shown here could be a SAS date neither will be until the macro variable is put to use. Since is the usage of the macro variable that determines what it should contain, it makes sense to store the most flexible value. Here the macro variable &DAYVAL will ultimately be more useful.

```
%let today=06sep2013;
%let dayval=19607;

%let convert = %sysfunc(putn("&today"d,6.));
%put &=today &=dayval &=convert;
```

```
7    %put &=today &=dayval &=convert;
TODAY=06sep2013 DAYVAL=19607 CONVERT=19607
```

The PUTN function has been used to convert the text string contained in &TODAY to its SAS date equivalent. The macro language does not inherently know how to interpret date constants, fortunately the PUTN function can be used to force the conversion. The LOG shows the result of the %PUT statement.

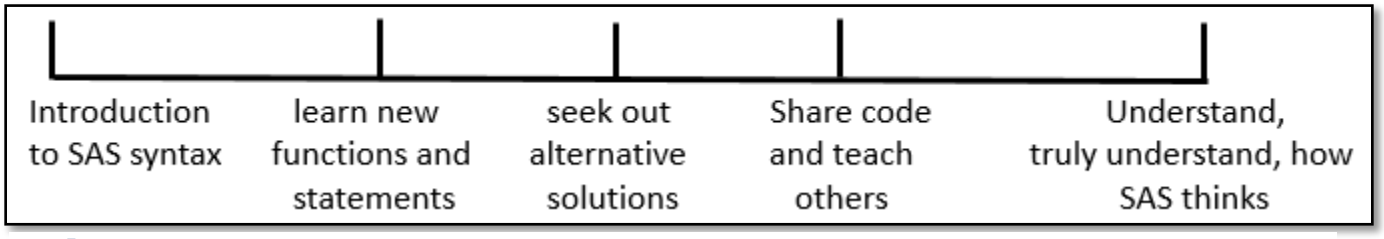
We might also want to put the current date in the title of a report. The %CURRDATE macro is a macro function that returns the current date using the WORDDATE. format. It utilizes the %QSYSFUNC macro function to access the DATA step function DATE, which returns the current date as a SAS date. This SAS date is then formatted for display.

```
%macro currdate;
%qtrim(%qleft(%qsysfunc(date(),worddate18.)))
%mend currdate;
title "Today is %currdate";
```

The examples in this section show a progression from simple usages of dates, to the use of interval functions, to more complex usages in the macro language. This progression mimics what will likely be the progression that a new SAS programmer takes as they start to learn the language. The simple assignments and date constants are easily understood. Then the programmer can advance to more and more complex functions, and eventually to the use of dates within the macro language.

THE TIMELINE AND A SAS CAREER

Although some SAS programmers and analysts learn the SAS language while in school, most learn while working, On the Job. This is not necessarily bad, but may not be the most efficient way to learn a programming language. Hopefully the new programmer will have a chance to attend some classes and perhaps papers at a user conference. Generally your learning 'time line' will start with syntax basics, from there you will add new functions, statements, options and such to your programming skills. At this point you will be solving most programming problems that come your way. You will be a 'good' programmer. But there is a trap for the unwary. If you become complacent, it is easy to stop learning. At this point you will need to force yourself to seek new knowledge, to discover alternative approaches, and to formulate new solutions to old problems, while sharing what you have learned. Those who continue to learn and continue to explore the language are the programmers who become known for their expertise.

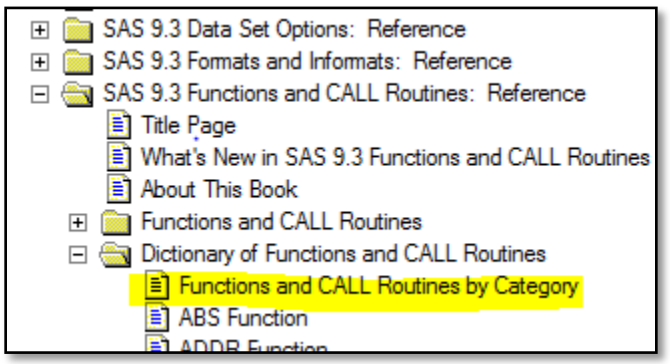


➔ Time line to SAS programming expertise

There are pitfalls to this approach of informal learning, not the least of which could be that the new programmer will be learning from someone else's code. When this happens it is often difficult for the new programmer to learn more than was known by the original programmer. The new programmer will need to take definitive steps to avoid getting trapped into only knowing what the other programmer knows.

If you do not set aside 20 minutes a day to learn something new, it is very difficult to hone your SAS knowledge. In order to use those 20 precious minutes successfully plan out what you are going to learn and how you will spend that time. Consider some of the following proven ideas for increasing your SAS expertise.

- Read a SAS tip each day, follow the sasCommunity.org tip of the day project. Consider contributing by submitting a tip.
- Read a SAS blog. There are several and many of the better ones can be found on the blog consolidator Blog Planet on sasCommunity.org.
- Scan the brief one sentence description of each function and call routine mentioned in the documentation's "Functions and CALL Routines by Category". Repeat for Formats, Informats, and Options.
- To get another programmer's perspective, read user written papers on topics of interest. Most papers are indexed on a site hosted by Lex Jansen.



- Read and participate in the SAS discussion forums at communities.sas.com. A number of world class programmers contribute and their solutions to common problems can be very innovative.

SUMMARY

Using SAS dates can be thought of as a metaphor for how we learn SAS. We start with simple ideas and a simple understanding both of the language and the use of SAS dates. As we grow in our understanding of the language we can take on more and more complex programming tasks. These include the use of SAS dates in a variety of situations.

SAS is a complex language. Learning SAS well takes time and successful learning will not be random. Plan which classes you want to take and which conferences you want to attend. Find sources of information that you can understand and use them. Investing time in the learning process will pay dividends in the end. Set aside at least 20 minutes a day to just learn something new. Read about new functions. Read a SAS blog. Follow the sasCommunity.org tip of the day. Share what you learn with others. Present and teach what you know.

ABOUT THE AUTHOR

Art Carpenter's publications list includes five 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
www.caloxy.com

REFERENCES

Some of the examples in this paper have been borrowed (with the author's permission) from the book *Carpenter's Guide to Innovative SAS® Techniques* by Art Carpenter (SAS Press, 2012).

The most complete reference on SAS dates is the book *Essential Guide to SAS® Dates and Times* by Derek Morgan (SAS Press, 2006).

A paper describing the creation of expanded holiday functions for a number of calendar systems was presented by Art Tabachneck.
Tabachneck, Arthur S., Matthew Kastin, and Xia Ke Shan, 2012, "Sometimes One Needs an Option with Unusual Dates", proceedings of the 2012 SAS Global Forum Conference, SAS Institute Inc., paper 040-2012.
http://support.sas.com/resources/papers/proceedings12/040-2012.pdf

Other User Group papers:
Carpenter, Arthur L., 2000, "Placing Dates in Your Titles: Do It Dynamically", proceedings of the Twenty-Fifth SAS User Group International Conference (SUGI), 2000, Cary, NC: SAS Institute Inc., paper 93-25.
http://www2.sas.com/proceedings/sugi25/25/cc/25p093.pdf

Carpenter, Arthur L., 2004, “Looking for a Date? A Tutorial on Using SAS® Dates and Times”, proceedings of the Thirtieth SAS User Group International Conference (SUGI), 2005, Cary, NC: SAS Institute Inc., paper 255-30. http://www2.sas.com/proceedings/sugi30/255-30.pdf

Finley, Wayne, 2000, “A Beginners Guide to SAS© Date and Time Handling”, Proceedings of the Twenty-Fifth Annual SAS® Users Group International Conference, Paper 58-25. http://www2.sas.com/proceedings/sugi25/25/btu/25p058.pdf

Karp, Andrew H., 1999, “Working with SAS® Date and Time Functions,” Proceedings of the Twenty-Fourth Annual SAS® Users Group International Conference, Paper 58-24. http://www2.sas.com/proceedings/sugi24/Begtutor/p58-24.pdf

Smiley, Christine A., 1999, “Working with Dates in SAS® Software”, proceedings of the 24th SUGI Conference, SAS Institute Inc., paper 73. http://www2.sas.com/proceedings/sugi24/Coders/p073-24.pdf

TRADEMARK INFORMATION