

THE DATE GAME

SAS Winnipeg Users Group

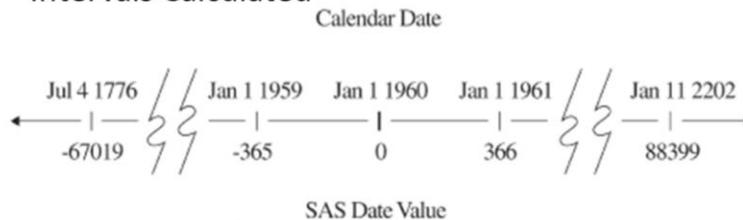
March 13, 2015

Charles Burchill



SAS Date Defined

- Integer (numeric) variable
- Days since January 1, 1960
e.g. 20161- represents March 14, 2015
0 – represents Jan 1, 1960.
- Can be displayed in many ways
- Portions of dates extracted
(e.g. day/week/month)
- Intervals Calculated



- SAS DateTime and Time in Seconds.

Why Jan 1, 1960? See:

<http://support.sas.com/community/newsletters/news/insider/dates.html>

Tony Barr was looking for a timestamp that would pre-date most electronic records that were available in the early seventies, so he selected the 1960 date. He also wanted to ensure that the timestamps would fit into less storage space, and selecting a more recent base date than the IBM base January 1, 1900, would allow this. The date January 1, 1960 was also close to the release date of the IBM System 360 architecture (which is recognized to be in April 1964). Remember that, at the time, SAS was only running on the IBM mainframes, hence the 360-centric mind-set.

Useful Range?



- November 1, 1582 (-137757)
Jan 1, 1582 returns a date but actual transition was Oct 15, 1582.
But only for France, Italy, Luxembourg, Portugal, and Spain. Other places moved to using the current Gregorian Calendar in different years e.g. UK 1752, and Soviet Union 1918.

- December 31, 20000 (6589335)
SAS 9.4 docs say A.D. 19900 (6552811)



SAS can perform calculations on dates ranging from A.D. November 1582 to A.D. 19,900. – SAS 9.4 online documentation. Different articles have different ranges. September 1752 is the earliest that you can get reliable day of the week information – SAS 9.4 online documentation.

You can actually get ‘dates’ outside of this range but there may be validity issues.

A little side note: The earliest date that is valid in SAS software is January 1, 1582. The last valid date is December 21, 20000.

--

SAS software uses the Gregorian calendar. That is the year that France, Italy, Luxembourg, Portugal, and Spain replaced the Julian calendar with the Gregorian. (The Gregorian calendar was first implemented so that the day after October 4, 1582 was October 15, 1582. Nevertheless, SAS software recognizes 31 days in the month of October, 1582.) While the rest of Roman Catholic Europe switched shortly after 1582, the United Kingdom and its colonies did not move to the Gregorian calendar until 1752. Many other countries switched even later, including the Soviet Union in 1918 and Greece in 1923. Some historic dates therefore might be handled in a misleading manner -- a problem which, it should be noted, is true of any use of SAS

dates in such instances.

Display of Dates?

18724 is not very helpful

- SAS Date Values can be displayed in various ways using built in formats.
- The following formats will Display 18724 as:
 - DATE. 07APR11
 - DDMMYY10. 07/04/2011
- Some formats have letter modifiers
 - B – blank, C – colon, D - dash, and P - period
 - DDMMYYC10. 07:04:2011

SAS Date Format	Displayed Date
DATE.	07APR11
DATE9.	07APR2011
DAY.	7
DDMMYY.	07/04/11
DDMMYY10.	07/04/2011
DDMMYYB10.	07 04 2011
DDMMYYC10.	07:04:2011
DOWNAME	Thursday
MMDDYY.	04/07/11
MMDDYY10.	04/07/2011
MMDDYYD10.	04-07-2011
MMDDYYP10.	04.07.2011
MMYYD.	04-2011
MONNAME.	April
MONTH.	4
MONYY.	APR11
WEEKDATE.	Thursday, April 7, 2011
WEEKDAY.	5
WORDDATE.	April 7, 2011
WORDDATX.	07 April 2011

SAS Date or String?

15OCT2034

20121030

April 7, 2011

Check the contents – numeric character with a Date format – it is a SAS date.

Many people see the a date in a proc print or view table and then try to use string functions – string functions will usually generate a numeric to character conversion note but actually return a [nonsensical] value.

The other issue is people think a string is a date value and try to use SAS functions – fortunately these usually generate errors.

Reading External Data (1)

- Use of Informats when reading raw data

```
data test ;
    infile datalines dlm=' ' dsd ;
    informat birth yymmdd8.
           death date9. ;
    Format birth death yymmdd10. ;
    input birth death;
    datalines ;
19670421,01aug2011
19450512,15jul2012
    ;
```

SAS Date Informat	Calendar Date	Result
DATE.	07APR2011	18724
DATE9.	07APR2011	18724
DDMMYY.	070411	18724
DDMMYY8.	07042011	18724
MMDDYY.	04072011	18724
MMDDYY10.	04/07/2011	18724
MONYY.	APR11	18718
YYMMDD.	110407	18724
YYMMDD8.	20110411	18724
YYQ.	1104	18718
YYQ6.	201104	18718

There is also an informat that can be used for any of the above calendar date formats ANYDTEw. (Use with caution!)

<http://support.sas.com/resources/papers/proceedings11/117-2011.pdf>

The anydtew. Format has been really very helpful in some instances but it must be used with some caution:

<http://support.sas.com/resources/papers/proceedings11/117-2011.pdf>

Reading External Data (2)

- SAS will understand PC date formats such as Excel or Access (Interface to PC Files)
 - PCFiles Library Engine
 - PROC IMPORT
- If Importing Date Fields always check!
Especially from Excel

String to Date

- SAS Date Constant: 'DDMMMYYYY'd
'01APR2011'D
- Use of INPUT(VAR,INFORMAT.) function
dt_string= '20110401' ;
sasdt = INPUT(dt_string, <?|??>yymmdd8.) ;

Also Date to String:

Use of PUT(VAR,FORMAT.) function
STRING= PUT(sasdt,yymmddn8.) ;

When using the SAS informat if a string does not fit the required structure SAS will return an invalid data error message and associated input lines. These errors can be suppressed (at your risk). I recommend not using these options unless you understand the errors and will be correcting/noting them in some other way. The input function returns a missing value if the input data is invalid.

SAS documentation about ? Or ?? Options:

? or ?? specifies the optional question mark (?) and double question mark (??) modifiers that suppress the printing of both the error messages and the input lines when invalid data values are read. The ? modifier suppresses the invalid data message. The ?? modifier also suppresses the invalid data message and, in addition, prevents the automatic variable _ERROR_ from being set to 1 when invalid data are read.

In this example, the question mark (?) modifier tells SAS not to print the invalid data error message if it finds data errors. The automatic variable _ERROR_ is set to 1 and input data lines are written to the SAS log.

```
y=input(x,? 3.1);
```

Because the double question mark (??) modifier suppresses printing of error messages and input lines and prevents the automatic variable _ERROR_ from being set to 1 when invalid data are read, the following two examples produce the same result:

```
y=input(x,?? 2.);  
y=input(x,? 2.); _error_=0;
```

Extracting Date Parts

SAS Date Function	Example	Result
MONTH(SAS date variable name)	MONTH(EXP_DATE)	4 (4 th month of date = April)
DAY(SAS date variable name)	DAY(EXP_DATE)	7 (7 th day of month)
WEEKDAY(SAS date variable name)	WEEKDAY(EXP_DATE)	5 (5 th day of week = Thursday)
YEAR(SAS date variable name)	YEAR(EXP_DATE)	2011 (year of date)
QTR(SAS data variable name)	QTR(EXP_DATE)	2 (2 nd quarter of year)

What is TODAY?

- The today function returns the current date as a numeric.
SAS_DATE = TODAY();

DateTime to Date?

- If you have a DateTime Variable
SAS_DATE = DATEPART(date_time_var);

See:

http://support.sas.com/documentation/cdl/en/etsug/63348/HTML/default/viewer.htm#etsug_intervals_sect014.htm

In the SAS Output window, the date and time automatically prints at the top of each page of output. However, the date and time printed is the date and time that the current SAS session was invoked. Instead, you might want to print the date and time that the SAS code is submitted. You can accomplish this by adding the following code to the top of each SAS program: `options nodate; %let timenow=%sysfunc(time(), time.); %let datenow=%sysfunc(date(), date9.);`

Intervals?

- Just Subtract ☺
- YRDIFF(start_dt, end_dt, <basis>)
 - Basis default is 'AGE'
- Interval Boundaries Between Two Dates: INTCK()
 - INTCK(interval<multiple> <.shift-index>, start-date, end-date, <'method'>)
 - mon_dif = intck('month', '01jan97'd, '31mar97'd);
 - weekvar=intck('week2.2','01jan97'd, '31mar97'd);
- Increment date by defined interval: INTNX()
 - INTNX(interval<multiple> <.shift-index> , start-from, increment<, 'alignment'>)
 - date2=intnx('month','01aug11'd,1);
 - Week2=intnx('week.2','01aug11'd,2);

NOTE: The WEEKDAY function produces an integer that represents the day of the week, where 1=Sunday, 2=Monday, ..., 7=Saturday. This date sequence covers other intervals as well. In the above example the interval for the week2.2 means how many bi-week boundaries are crossed starting on Monday.

Method – C – Continuous, D – Discrete

Alignment – BEGINNING|B, MIDDLE|M, or END|E.

Within the interval the returned date is the date that represents the point in the range of the interval.

The default is the beginning if not specified.

Intervals

The interval names that are used with SAS date values are :

YEAR, SEMIYEAR, QTR, MONTH, SEMIMONTH, TENDAY, WEEK, WEEKDAY, and DAY.

See:

<http://support.sas.com/documentation/cdl/en/lefunctionsref/67398/HTML/default/viewer.htm#p0syn64amroombn14vrdzksh459w.htm>

NOTE: The WEEKDAY function produces an integer that represents the day of the week, where 1=Sunday, 2=Monday, ..., 7=Saturday. This date sequence covers other intervals as well. In the above example the interval for the week2.2 means how many bi-week boundaries are crossed starting on Monday.

Storage

- Numeric Length 8 – Full Range
- Numeric length 4 - 1582-11-01 to 7701-10-23

Source Info

- 244-2012: SAS® Dates: Facts, Formats, and Functions (<http://support.sas.com/resources/papers/proceedings10/134-2010.pdf>)
- 134-2010: Working with SAS® Date and Time Functions (<http://support.sas.com/resources/papers/proceedings10/134-2010.pdf>)
- 226-2007: Using SAS® Dates and Times—A Tutorial (<http://www2.sas.com/proceedings/forum2007/226-2007.pdf>)
- 336-2013: We All Have Bad Dates Once in a While... (<http://support.sas.com/resources/papers/proceedings13/336-2013.pdf>)
- Dates, Times, and Intervals
(<http://support.sas.com/documentation/cd/en/lrcon/67885/HTML/default/viewer.htm#n0q9ylcaccjgirn19hvtqnd9cte8p.htm>)

**Thanks
for
Playing**