

Creating Date and Time Intervals

Judith Boura, William Beaumont Hospital, Royal Oak, Michigan

ABSTRACT:

This course teaches you how to create time intervals using date and time fields entered in various formats with the SAS(system. You will learn how to use the SUBSTR and INPUT functions to convert character data into a date or time field. You will also learn how to create time intervals taking all time and date possibilities into account.

INTRODUCTION:

When working with date and time fields, the number of hours, days or minutes between dates and times may need to be calculated. This may be a simple statement or a long involved series of statements which take every possibility into account. Dates and times may be entered as two separate fields or as one field. These fields may be entered in various formats and all these possibilities will be considered. Sample programs will be distributed that should be easy to reproduce.

Dates:

There are many different ways to enter dates into a field. Various forms of the date 10/3/56 can look like:

October 3, 1956
10/03/56
100356
10/03/1956
03/10/56 (European)
Oct 3 1956
10-3-56, etc.

If the date is entered as character, and you want to calculate a time interval, the date must be changed to a recognizable SAS date informat. SAS has 17 date and/or time informats already built in to the SAS system. (See page 63 of the SAS Language manual.) Assigning an informat or creating a new variable may change the way the date is handled. Pages 635 through 672 of the SAS Language manual describe these informats in detail and give examples of each.

All variables can be changed at time of input into the data set by assigning these informats in the DATA step INPUT statement.

```
input var1 mmddyy8. var2 time5.;
```

However, if you are using a data set that has the variables already coded as character, you may need to create new variables.

Example:

If date1 is entered as a character 100356, we can convert it to a new date variable, date2, using an INPUT statement.

```
date2=input(date1,mmddyy6.);
```

You'll notice the "6." following the "mmddyy" The 6 tells SAS that the date is contained in 6 spaces and the period tells SAS that this is an informat. If date1 had been entered as 10/03/56, the formula for date2 would be;

```
date2=input(date1,mmddyy8.);
```

Dates entered as characters, 03 Oct 1956 can be converted using

```
date2=input(date1,date11.);
```

Suppose you have a European date (date1=031056) and you want it converted to a 100356 date. You could use the SUBSTR function along with an INPUT statement to convert the date.

Example:

```
date2=input(substr(date1,3,2)||substr(date1,1,2)  
||substr(date1,5,2),mmddyy6);
```

The SUBSTR function selects part of a variable. The function name "SUBSTR" is followed by a parentheses. The variable being taken from is first in the parentheses, followed by a comma and the number of the place where you want to begin and then a comma and the number of characters you want included. The statement substr(date1,5,2) goes to the 5th spot of variable date1 which is "5" and selects the next two characters, "56". The || joins the substrings together.

For 2-digit years, SAS assumes the 1900s. A four-digit year must have all 4 places typed in (1895 vs. 95 or 2001 vs. 01).

Time:

There are also different ways that times are entered into a data set. For the time 3:25pm you could have:

03:25 PM or 0325 PM or 1525 etc.

If the time has been entered as a character such as time1=1525 then you could use an input

statement:

Example:

```
time2=input(substr(time1,1,2)||':'  
            ||substr(time2,3,2),time5.);
```

For times entered not in military time but as characters such as time1=03:25 PM, a more complicated program may be needed.

Example:

```
AMPM=substr(time1,7,2); *(reads in AM or PM);  
hour=substr(time1,1,2); *(reads in hour only);  
if AMPM='PM' and hour='12' then  
    time2=(input(substr(time1,1,5),time5.));  
else if AMPM='AM' and hour='12' then  
    time2=(input(substr(time1,1,5),time5.))-43200;  
else if AMPM='PM' then  
    time2=(input(substr(time1,1,5),time5.))+43200;  
else if AMPM='AM' then  
    time2=input(substr(time1,1,5),time5.);
```

The first two lines create new variables for checking whether a time is AM or PM and one for the hour. Line 3 states if it is PM and the time is 12 something PM then the time should be 12something. The next check is to see if the time is AM and the hour is 12. If it is then the 43200 is subtracting 12 hours (in seconds) from the time to make the hour 00. The time needs to be in military time and 12:00 AM is time 00:00. The next line states that if the time is PM (and not hour=12) then 43200 (12 hours in seconds) needs to be added to the time. Otherwise the time is simply the time1 if AM.

Time Intervals:

Once the dates and times have been entered so that SAS recognizes them as date and time fields, time intervals can be calculated. If the date and time are entered as one field with datetim14. informat in SAS, then the interval is easy to calculate. Datetim1 is the 1st date and time while datetim2 is the second date and time. To calculate an interval you simply subtract the later date from the earlier date.

```
interval=datetim2-datetim1;
```

This will give you the time interval in seconds. For the time interval in minutes, you simply divide by 60.

```
interval=(datetim2-datetim1)/60;
```

For the time interval in hours, you simply divide by 3600.

```
interval=(datetim2-datetim1)/3600;
```

Dates and times are more likely entered separately. This creates a more lengthy program but still very able to do. For the next example, DATE1 and TIME1 will be the first date and time considered. The second date and time will be DATE2 and TIME2 and the interval being calculated will be INTERVAL. It is assumed these are dates and times that SAS already recognizes.

Example:

```
*** Interval calculation *****;  
if time1^=' ' and time2^=' ' and date1^=' ' and  
date2^=' ' then do;  
if date2=date1 and time2>time1 then  
    interval=time2-time1;  
else if date2=date1 and time2=time1 then  
    interval=0;  
else if date2>date1 then do;  
    if time2=time1 then  
        interval=(date2-date1)*24*3600;  
    else  
        if time2>time1 then  
            interval=(date2-date1)*24*3600-(time1-time2);  
        else  
            if time2<time1 then  
                interval=(date2-date1)*24*3600-(time2-time1);  
            end;  
        else interval=' '  
    end;  
end;
```

If any date and/or time is missing, the interval obviously cannot be completed. The first statement eliminates those observations. The second statement calculates the time where it is the same day but not the same time and time2 follows time 1. If it is the same day and time, the interval=0. This is calculated in the next statement. If the 2nd date follows the third date then there are 3 possibilities. The times can be the same so the interval would equal the difference in dates multiplied by 24 hours and again by 3600 seconds (in each hour). If time2 is after (>) time 1 then the interval is the difference in dates(multiplied by 24 hours and again by 3600 seconds) plus time2 –time1. If time2 is before (<) time 1 then the interval is the difference in dates(multiplied by 24 hours and again by 3600 seconds) plus time1 –time2.

This has calculated the interval in seconds. For the interval in minutes (INTmin) then the statement "intmin=interval/60;" needs to be added before the "end;".

The program could be even longer if you need to add the previous types of statements to format the date and time fields properly.

The easiest way to handle dates and times is to make sure they are entered properly as dates and times in to the original data base that you are working with. That can't always happen, especially if the data is collected by someone else. As you can see, there is almost always a way to manipulate those variables to be able to use them as needed. The programs may be simple or quite cumbersome but the possibility is still there.

References:

SAS Institute Inc., SAS(Language: Reference, Version 6, First Edition, Cary, NC: SAS Institute Inc., 1990. 1042pp.

SAS Institute Inc., SAS(Procedures Guide, Version 6, Third Edition, Cary, NC: SAS Institute Inc., 1990. 705pp.

SAS is a registered trademark of SAS Institute Inc. in the USA and other countries.

Contact:

Judith Boura, Cardiology Research,
William Beaumont Hospital, 3601 W. 13 Mile, Rd.
Royal Oak, MI 48073
(248) 551-0789 jboura@beaumont.edu