

Dating Tips for SAS Users

Keith J. Brown
Office of Information Technology
University of North Carolina at Chapel Hill

Time and the measurement of its passage have fascinated humans for millenia. This fascination has not waned with the computerization of our society. As the dates of various milestones in our lives -- birth, graduation, employment, marriage, and death -- become nothing more than bits in a computer's memory, techniques for turning those bits back into useable information become more important. The date and time formats, informats, and functions in SAS allow us to retrieve that information and make use of it in reports and calculations.

Unlike some computer languages, SAS does not have a separate variable type for dates; these are stored as regular numeric data. The world, according to SAS, began on January 1, 1960 (Day 0 in SAS). Each day since then is represented as a positive integer, while dates before that are stored as negative numbers. Fortunately, we are not required to do the calculations when inputting dates into SAS; the array of informats makes this job much simpler. For instance, all the date variables in the following example will have the value of -24, despite the differences in the raw data representation of that date.

```
DATA DATES1;
INPUT      @1  DATE1      MMDDYY6.
           @10 DATE2      DDMMYY8.
           @20 DATE3      DATE9.
           @40 DATE4      JULIAN5.
           @50 DATE5      NENG010.
;
CARDS;
120859      08/12/59      08DEC1959      59342      S.34/12/08
;
```

(Julian dates are often used by computers internally; Nengo dates are based on the reigns of Japanese emperors).

All these informats hinge on having the complete date value in a single field. The MONYY and YYQ informats take partial date information and expand them into full SAS dates. 'DEC59' becomes 'December 1, 1959', while '59Q4' is transformed into 'October 1, 1959'.

The requirement of complete date values in a single field is not always met; indeed, in real world data, parts of the date often may be separated or missing. The MDY function allows you to take the various components of a SAS date from variables or constants to produce a usable SAS date variable, as shown in the example below.

```
DATA DATES2;
INFILE IN;
INPUT      @150      YEAR
           @325      MONTH
           @999      DAY
;
IF DAY > 0 THEN DATE1=MDY(MONTH,DAY,YEAR);
ELSE DATE1=MDY(MONTH,15,YEAR);
```

Dates, along with other numeric data, are often stored in compressed form in large datasets to conserve space. Since SAS treats dates as nothing more than numbers, regular numeric functions can also be used to handle date values. By using functions such as PUT and INPUT, we can produce a usable SAS date from the most recalcitrant data.

```
DATA DATES3;
INFILE IN;
INPUT      @1200      BIRTH      PD4.;
BIRTH=INPUT(PUT(BIRTH,Z6.),MMDDYY6.);
```

While a single date in a report conveys some information, dates are more often used in pairs in calculating age, duration, or elapsed time. Using date variables, date literals (constants), and the numeric properties of SAS dates, we can make calculations of the length of time in several ways.

- a) Between two date variables:
`AGE=(DEATH-BIRTH)/365;`
- b) Between a fixed date and a variable:
`SESUGAGE=('18SEP94'D - BIRTH)/365;`
- c) Between two fixed dates:
`MYSUGAGE=('18SEP94'D - '08DEC59'D)/365;`
- d) Between a variable and a fixed distance:
`WRITING=DEADLINE-2;`

Dates and the things they measure are an important part of our life and the data that we accumulate. With the array of tools available for handling them in SAS, we can easily make use of the information they provide.