Chapters 2, 19 WORKING WITH DATES AND TIMES

HOW SAS STORES DATE AND TIME VALUES

* When SAS reads a date, it converts it into the number of days from January 1, 1960. For example, January 5, 1960 SAS stores as number 4; December 16, 1989 as 10945; January 1, 1950 as -3652.
* When SAS reads time, it converts it into the number of seconds since midnight. For example, 12:15am is stored as (15)(60)=900; 1am is stored as (60)(60)=3600.
* When SAS reads date and time, it converts it into the number of seconds from the midnight on January 1, 1960. For example, April 22, 1989, 16:10:45 is stored as 92488384; July 4, 1776, 11:30:23 is stored as

-5790400177.

DATE FORMATS

Here are some examples of date expressions and the corresponding formats.



TIME FORMATS

Here are some examples of time expressions and the corresponding formats.



* Note that the smallest admissible field width in TIME formats is 5, therefore, even though the time expression 2:34 is only 4 columns wide, the width 5 must be specified.

DATE AND TIME FORMATS

Here are some examples of date and time expressions and the corresponding formats.



HANDLING TWO-DIGIT YEAR VALUES

SAS represents dates correctly from 1582 A.D. to 20,000 A.D. However, two-digit years may not be interpreted correctly due to the turns of the centuries. To fix this problem, option **YEARCUTOFF** may be used. When a two-digit year value is specified, SAS interprets it based on a 100-year span that starts with the YEARCUTOFF value. By default, YEARCUTOFF=1940, and the default 100-year span is between 1940 and 2039.

Example. By default, SAS reads the date expression 10/08/69 as 10/08/1969; 04/11/05 as 04/11/2005; and 04/11/22 as 04/11/2022.

Example. Suppose the starting year is specified as

options YEARCUTOFF=1950

Then the 100-year span is between 1950 and 2049, and SAS reads the date 04/11/42 as 04/11/**2042**.

USING DATES AND TIMES IN CALCULATIONS

Example. A data file contains patient ID, foot surgery date (Surgery\_Date), and return to work/preinjury activities date (RTW\_Date). The following SAS code reads the data correctly, computes the length of time on disability (in months), and outputs the data.

data footsurgery;

input @1 ID $3. @ 5 Surgery\_Date mmddyy10. @15 RTW\_Date

mmddyy10.;

disability=(RTW\_Date-Surgery\_Date)/30;

cards;

001 1/10/2006 5/15/2006

002 2/9/2006 6/20/2006

003 3/9/2006 6/26/2006

004 3/16/2006 6/1/2006

005 4/9/2006 9/22/2006

006 6/22/2006 8/21/2006

007 6/22/2006 8/16/2006

008 8/1/2006 12/15/2006

;

proc print;

run;

Note: To convert days into months divide by 30, into weeks – divide by 7, in years – divide by 365.25

The printout looks like this:

Surgery\_

Obs ID Date RTW\_Date disability

1 001 16811 16936 4.16667

2 002 16841 16972 4.36667

3 003 16869 16978 3.63333

4 004 16876 16953 2.56667

5 005 16900 17066 5.53333

6 006 16974 17034 2.00000

7 007 16974 17029 1.83333

8 008 17014 17150 4.53333

proc print;

format Surgery\_Date date7. RTW\_Date date7.;

run;

The printout looks like this:

Surgery\_

Obs ID Date RTW\_Date disability

1 001 10JAN06 15MAY06 4.16667

2 002 09FEB06 20JUN06 4.36667

3 003 09MAR06 26JUN06 3.63333

4 004 16MAR06 01JUN06 2.56667

5 005 09APR06 22SEP06 5.53333

6 006 22JUN06 21AUG06 2.00000

7 007 22JUN06 16AUG06 1.83333

8 008 01AUG06 15DEC06 4.53333

proc print;

format Surgery\_Date date7. RTW\_Date date9.

disability 4.2;

run;

Surgery\_

Obs ID Date RTW\_Date disability

1 001 10JAN06 15MAY2006 4.17

2 002 09FEB06 20JUN2006 4.37

3 003 09MAR06 26JUN2006 3.63

4 004 16MAR06 01JUN2006 2.57

5 005 09APR06 22SEP2006 5.53

6 006 22JUN06 21AUG2006 2.00

7 007 22JUN06 16AUG2006 1.83

8 008 01AUG06 15DEC2006 4.53

Example. The following data contain patient ids, dialysis start date (Dialysis\_Start\_Date), and kidney implant date (Implant\_Date). Time on dialysis (Dialysis\_Duration) is calculated (in years), and the variables are printed.

data kidney;

input @1 ID $1. @3 Dialysis\_Start\_Date MMDDYY8. @12

Implant\_Date MMDDYY10.;

Dialysis\_Duration=(Implant\_Date-Dialysis\_Start\_Date)/365.25;

cards;

1 3/1/05 9/21/2006

2 3/1/01 6/26/2006

3 9/1/98 6/25/2004

4 9/1/00 2/1/2001

5 11/23/05 1/8/2007

6 11/23/05 2/8/2007

;

proc print;

run;

The output is

Dialysis\_ Implant\_ dialysis\_

Obs ID Start\_Date Date duration

1 1 16496 17065 1.55784

2 2 15035 16978 5.31964

3 3 14123 16247 5.81520

4 4 14854 15007 0.41889

5 5 16763 17174 1.12526

6 6 16763 17205 1.21013

proc print;

format Dialysis\_Start\_Date mmddyy10. Implant\_Date mmddyy10.;

run;

The output is

Dialysis\_ Implant\_ dialysis\_

Obs ID Start\_Date Date duration

1 1 03/01/2005 09/21/2006 1.55784

2 2 03/01/2001 06/26/2006 5.31964

3 3 09/01/1998 06/25/2004 5.81520

4 4 09/01/2000 02/01/2001 0.41889

5 5 11/23/2005 01/08/2007 1.12526

6 6 11/23/2005 02/08/2007 1.21013

Example. Patient’s medical record number (Record\_Number), check in time for an appointment (Check\_In\_Time), and check out time (Check\_Out\_Time) are recorded. The length of the appointment is computed (in minutes).

data doctor\_visit;

input @1 Record\_Number $4. @6 Check\_In\_Time time5. @12

Check\_Out\_Time time8.;

appt\_length=(Check\_Out\_Time-Check\_In\_Time)/60;

cards;

9956 9:00 10:13:00

3442 11:15 12:15:00

8113 10:45 11:26:00

9008 13:55 15:07:00

1139 14:30 15:48:00

;

proc print;

run;

The SAS output is as follows:

Record\_ Check\_ Check\_ appt\_

Obs Number In\_Time Out\_Time length

1 9956 32400 36780 73

2 3442 40500 44100 60

3 8113 38700 41160 41

4 9008 50100 54420 72

5 1139 52200 56880 78

proc print;

format Check\_In\_Time time5. Check\_Out\_Time time5.;

run;

The SAS output is as follows:

Record\_ Check\_ Check\_ appt\_

Obs Number In\_Time Out\_Time length

1 9956 9:00 10:13 73

2 3442 11:15 12:15 60

3 8113 10:45 11:26 41

4 9008 13:55 15:07 72

5 1139 14:30 15:48 78