Using Date Constants to Ranges for Dates

- data constants: two-digit day, 3-character month name, two- or four-digit year, placed in single or double quotes, followed by a lowercase or uppercase 'D'

Lecture 7
July 2, 2003

Kate Cowles
374 SH
kcowles@stat.uiowa.edu

options linesize = 72 ;
LIBNAME CLEAN "/space/kcowles/172/clean/"
;
DATA CLEAN.PATIENTS;
INFILE "/space/kcowles/172/clean/patients.txt" PAD;
INPUT @1 PATNO $3. @4 GENDER $1. @5 VISIT MMDDYY10. @15 HR 3. @18 SBP 3. @21 DBP 3. @24 DX $3. @27 AE $1.;
LABEL PATNO = "Patient Number" GENDER = "Gender" VISIT = "Visit Date" HR = "Heart Rate" SBP = "Systolic Blood Pressure" DBP = "Diastolic Blood Pressure" DX = "Diagnosis Code" AE = "Adverse Event";
FORMAT VISIT MMDDYY10. ;
RUN;

/***************************************************************************/
Program 4-1 Checking that a date is within a specified interval (Data Step approach)
***************************************************************************/
DATA _NULL_;
TITLE "Dates Before June 1, 1998 or After October 15, 1999";
FILE PRINT;
SET CLEAN.patients(KEEP=VISIT PATNO); IF VISIT LT '01JUN1998'D AND VISIT NE . OR VISIT GT '15OCT1999'D THEN PUT PATNO= VISIT= MMDDYY10. ;
RUN;

The SAS System
17:55 Tuesday, July 1, 2003

PATNO=XX5 VISIT=05/07/1998
PATNO=010 VISIT=10/19/1999
PATNO=003 VISIT=11/12/1999
PATNO=028 VISIT=03/28/1998
PATNO=029 VISIT=06/15/1998
Using Proc Print and a Where statement to check date ranges

- easier than previous version because keyword BETWEEN can be used in WHERE statement

Checking for invalid dates

- some dates in PATIENTS data set are missing and others are invalid
- all invalid dates were converted to missing values during the input process
- so to distinguish between invalid and missing dates, we must work with raw data, not the permanent SAS dataset
- error messages in log file occur when we try to read in the invalid date values from the raw data file
How to see what’s in invalid dates: read the date twice!

Program 4-4 Listing missing and invalid dates by reading the date twice, with a date informat and the second time as character data

DATA _NULL_;  
FILE PRINT;  
TITLE "Listing of Missing and Invalid Dates";  
INFILE "/space/kcowles/172/clean/patients.txt" PAD;  
INPUT @1 PATNO $3.  
@5 VISIT MMDDYY10.  
@5 V_DATE $CHAR10.;  
FORMAT VISIT MMDDYY10.;  
IF VISIT = . THEN PUT PATNO= V_DATE=;  
RUN;

Listing of Missing and Invalid Dates 3
17:55 Tuesday, July 1, 2003
PATNO=007 V_DATE=08/32/1998
PATNO=011 V_DATE=13/13/1998
PATNO=015 V_DATE=
PATNO=123 V_DATE=15/12/1999
PATNO=321 V_DATE=
PATNO=020 V_DATE=99/99/9999
PATNO=027 V_DATE=NOTAVAIL

ANother approach to the same task

Program 4-5 Listing missing and invalid dates by reading the date as a character variable and converting to a SAS date with the INPUT function

DATA _NULL_;  
FILE PRINT;  
TITLE "Listing of Missing and Invalid Dates";  
INFILE "/space/kcowles/172/clean/patients.txt" PAD;  
INPUT @1 PATNO $3.  
@5 V_DATE $CHAR10.;  
FORMAT VISIT MMDDYY10.;  
IF VISIT = . THEN PUT PATNO= V_DATE=;  
RUN;

NOTE: Invalid argument to function INPUT at line 92 column 12.
RULE: ----+----1----+----2----+----3----+----4----+----5----+----6--
7 CHAR 007M08/32/1998 88148102 0.  
ZONE 33343323323333233333233223302222222222222222222222222222222222
NUMR 007D08F32F19980881481020000D0000000000000000000000000000000000
63
125
187
249
PATNO=007 V_DATE=08/32/1998 VISIT=. _ERROR_=1 _N_=7
NOTE: Invalid argument to function INPUT at line 92 column 12.

TO list only invalid dates (not missing ones)

Program 4-6 Removing the missing values from the invalid date listing

DATA _NULL_;  
FILE PRINT;  
TITLE "Listing of Missing and Invalid Dates";  
INFILE "/space/kcowles/172/clean/patients.txt" PAD;  
INPUT @1 PATNO $3.  
@5 V_DATE $CHAR10.;  
FORMAT VISIT MMDDYY10.;  
IF VISIT = . AND V_DATE NE ' ' THEN PUT PATNO= V_DATE=;  
RUN;

Listing of Missing and Invalid Dates 5
17:55 Tuesday, July 1, 2003
PATNO=007 V_DATE=08/32/1998
PATNO=011 V_DATE=13/13/1998
PATNO=015 V_DATE=
PATNO=123 V_DATE=15/12/1999
PATNO=020 V_DATE=99/99/9999
PATNO=027 V_DATE=NOTAVAIL
Working with dates in nonstandard form

- MDY function constructs a date variable from 3 variables containing month, day, and year
- may enable dealing with dates for which there is no SAS informat

Removing missing values from the error listing

- construct a character representation of the date
- use the concatenation operator, ||, to piece together the month, the day, and the year, and the two slashes
Creating a SAS date when the day of the month is missing

- sometimes date values may be missing the *day*, but we would like to create a SAS date by imputing either the 1st or the 15th as the day of the month
- One method: use MONYY informat that reads dates in the form of a 3-character month name and a two- or four-digit year
  - if dates are in this form, SAS will create a SAS date using the 1st of the month as the day value
- Other method: use MDY function, substituting a constant of your choice for the day argument

```sas
DATA NO_DAY;
  INPUT @1 DATE1 MONYY7. @8 MONTH 2. @10 YEAR 4.;
  DATE2 = MDY(MONTH,15,YEAR);
  FORMAT DATE1 DATE2 MMDDYY10.;
DATALINES;
  JAN1998 01/98
;
PROC PRINT DATA=NO_DAY;
  TITLE "Listing of Data Set NO_DAY";
RUN;
```

Listing of Data Set NO_DAY
17:55 Tuesday, July 1, 2003

<table>
<thead>
<tr>
<th>Obs</th>
<th>DATE1</th>
<th>MONTH</th>
<th>YEAR</th>
<th>DATE2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01/01/1998</td>
<td>1</td>
<td>1998</td>
<td>01/15/1998</td>
</tr>
</tbody>
</table>

Suspending error checking for known invalid dates

- if you wish to prevent automatic listing of date errors in the SAS log, use double question mark (??) modifier in INPUT statement or with INPUT function
- prevents the NOTES and data listings from being printed in SAS log, and keepsSAS internal variable _ERROR_ at 0

```sas
DATA MISS_DAY;
  INPUT @1 PATNO $3. @4 MONTH 2. @6 DAY 2. @8 YEAR 4.;
  IF DAY NE . THEN DATE = MDY(MONTH,DAY,YEAR);
  ELSE DATE = MDY(MONTH,15,YEAR);
  FORMAT DATE MMDDYY10.;
DATALINES;
  00110211998
  002051998
  00344 .1998
;
PROC PRINT DATA=MISS_DAY;
  TITLE "Listing of Data Set MISS_DAY";
RUN;
```

Listing of Data Set MISS_DAY
17:55 Tuesday, July 1, 2003

<table>
<thead>
<tr>
<th>Obs</th>
<th>PATNO</th>
<th>MONTH</th>
<th>DAY</th>
<th>YEAR</th>
<th>DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>001</td>
<td>10</td>
<td>21</td>
<td>1998</td>
<td>10/21/1998</td>
</tr>
<tr>
<td>2</td>
<td>002</td>
<td>5</td>
<td></td>
<td>1998</td>
<td>05/15/1998</td>
</tr>
<tr>
<td>3</td>
<td>003</td>
<td>44</td>
<td></td>
<td>1998</td>
<td></td>
</tr>
</tbody>
</table>
Program 4-11 Suspending error checking for known invalid dates, using the **informat** modifier

```
DATA DATES;
  INFILE "/space/kcowles/172/clean/patients.txt" PAD;
  INPUT @5 VISIT ?? MMDDYY10.;
  FORMAT VISIT MMDDYY10 ;
RUN;
```

Program 4-12 Demonstrating the ?? informat modifier with the INPUT function

```
DATA _NULL_;  
FILE PRINT;  
INFILE "/space/kcowles/172/clean/patients.txt" PAD;  
INPUT @1 PATNO $3.;  
@5 V_DATE $CHAR10.;  
VISIT = INPUT(V_DATE, ?? MMDDYY10.);  
FORMAT VISIT MMDDYY10 ;  
IF VISIT = . THEN PUT PATNO= V_DATE=;  
RUN;
```

Listing of Data Set MISS_DAY

<table>
<thead>
<tr>
<th>PATNO</th>
<th>V_DATE</th>
</tr>
</thead>
<tbody>
<tr>
<td>007</td>
<td>08/32/1998</td>
</tr>
<tr>
<td>011</td>
<td>13/13/1998</td>
</tr>
<tr>
<td>015</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>15/12/1999</td>
</tr>
<tr>
<td>321</td>
<td></td>
</tr>
<tr>
<td>020</td>
<td>99/99/9999</td>
</tr>
<tr>
<td>027</td>
<td>NOTAVAIL</td>
</tr>
</tbody>
</table>

17:55 Tuesday, July 1, 2003