Data Cleaning involving duplicate IDs and duplicate records

Lecture 8
July 14, 2003

Kate Cowles
374 SH
kcowles@stat.uiowa.edu

Patients file: duplicate records added

02/11/11/1998 88140 80 10
00/2/11/13/1998 84120 78 10
00/2/11/13/1998 84120 78 10
003/10/21/1998 88190100 31
003/11/12/1999 58112 74 0
04/0/1/1999101200120 5A
11/0/5/07/1998 68120 80 10
06/06/15/1999 72102 68 61
06/07/07/1999 82418 84 10
007/08/32/1998 88148102 0
M1/1/1/1998 89190100 0
007/08/08/199210 70
008/09/25/1999 86240180 41
01/0/10/19/1999 40120 10
01/1/11/13/1999 88000 20 41
01/12/10/12/98 60122 74 0
01/3/1/3/23/1999 74108 64 1
01/4/1/2/02/1999 22130 90 1
002/1/11/13/1998 84120 78 10
003/11/12/1999 88112 74 0
01/5/7 82418 88 31
01/7/7/4/05/1999208 84 20
01/9/06/07/1999 58118 70 0
12/10/15/12/1999 60 10
321/7 900400200 51
02/0/9/99/199910 20 8 0
02/11/10/11/1998 48114 82 21
02/11/12/31/1998 22 34 78 0
02/11/12/31/1998 22 34 78 0
12/11/01/1999 74102 68 51
02/17/21/11/1999 58106 70
02/19/23/1998 80150 90 30
Eliminating duplicates using PROC SORT

- NODUPKEY option on PROC SORT eliminates multiple observations where the BY variables have same value
- OUT= option is used to create new data set, leaving original data set unchanged
- when NODUPKEY removes multiple observations, the only indication is in the NOTE in the SAS log
- looking for duplicate IDs with NODUPKEY in PROC SORT is useful only if the SAS log shows that NO duplicates were removed
  — otherwise, you need to see which IDs had duplicate data and the nature of the data
- if NODUPKEY is used with more than one BY variable, only those observations with identical values on all the BY variables will be deleted

```
* lect8.sas ;
/*****************************************************************************/
options linesize = 72 ;
libname CLEAN '/spac/kcovies/172/clean/' ;
/*****************************************************************************/
Program S-1 Demonstrating the NODUPKEY option of PROC SORT
*****************************************************************************
PROC SORT DATA=CLEAN.PATIENTS OUT=SINGLE NODUPKEY;
  BY PATNO;
RUN;
PROC PRINT DATA=SINGLE;
  TITLE "Data Set SINGLE - Duplicated ID's Removed from PATIENTS";
  ID PATNO;
RUN;
*****************************************************************************
```

The NODUP option of PROC SORT

- NODUP also deletes duplicates, but only for two observations where ALL the variables have identical values
- you have to sort by all of the variables to make this work correctly
- note use of _ALL_ as the by variable in the example for this purpose

```
Program S-2 Demonstrating the NODUP option of PROC SORT
*****************************************************************************
PROC SORT DATA=CLEAN.PATIENTS OUT=SINGLE NODUP;
  BY _ALL_ ;
RUN;
PROC PRINT DATA = SINGLE ;
  TITLE 'SINGLE from NODUP' ;
run ;
*****************************************************************************
```
Data step method for identifying and listing duplicate IDs

What happens if you use NODUP and sort on only one variable

A new dataset for illustration purposes
### Detecting duplicates having same visit date for same patient

Program 5-6 Identifying patient ID's with duplicate visit dates

```sas
PROC SORT DATA=CLEAN.PATIENT2 OUT=TMP;
   BY PATNO VISIT;
RUN;

DATA DUP;
   SET TMP;
   BY PATNO VISIT;
   IF FIRST.VISIT AND LAST.VISIT THEN DELETE;
RUN;

PROC PRINT DATA=DUP;
   TITLE "Listing of Duplicates from Data Set CLEAN.PATIENT2";
   ID PATNO;
RUN;

Listing of Duplicates from Data Set CLEAN.PATIENT2

```

### Using PROC FREQ to detect duplicate IDs

- can use PROC FREQ to count the number of observations for each value of patient ID variable
- use patient ID variable and OUT= option of TABLES statement to create a SAS data set containing the value of the PATNO variable and the frequency count
  - PROC FREQ uses the variable name COUNT to hold the frequency information
- after you have this information, you can use it to select the original duplicate observations from your data set

<table>
<thead>
<tr>
<th>Obs</th>
<th>PATNO</th>
<th>VISIT</th>
<th>HR</th>
<th>SBP</th>
<th>DBP</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>001</td>
<td>06/12/98</td>
<td>80</td>
<td>130</td>
<td>80</td>
</tr>
<tr>
<td>2</td>
<td>001</td>
<td>06/15/98</td>
<td>78</td>
<td>128</td>
<td>78</td>
</tr>
<tr>
<td>3</td>
<td>002</td>
<td>01/01/99</td>
<td>48</td>
<td>102</td>
<td>66</td>
</tr>
<tr>
<td>4</td>
<td>002</td>
<td>01/10/99</td>
<td>70</td>
<td>112</td>
<td>82</td>
</tr>
<tr>
<td>5</td>
<td>002</td>
<td>02/02/99</td>
<td>74</td>
<td>118</td>
<td>78</td>
</tr>
<tr>
<td>6</td>
<td>003</td>
<td>10/21/98</td>
<td>68</td>
<td>120</td>
<td>70</td>
</tr>
<tr>
<td>7</td>
<td>004</td>
<td>03/12/98</td>
<td>70</td>
<td>102</td>
<td>66</td>
</tr>
<tr>
<td>8</td>
<td>004</td>
<td>03/13/98</td>
<td>70</td>
<td>106</td>
<td>68</td>
</tr>
<tr>
<td>9</td>
<td>005</td>
<td>04/14/98</td>
<td>72</td>
<td>118</td>
<td>74</td>
</tr>
<tr>
<td>10</td>
<td>005</td>
<td>04/14/98</td>
<td>74</td>
<td>120</td>
<td>80</td>
</tr>
<tr>
<td>11</td>
<td>006</td>
<td>11/11/98</td>
<td>100</td>
<td>180</td>
<td>110</td>
</tr>
<tr>
<td>12</td>
<td>007</td>
<td>09/01/98</td>
<td>68</td>
<td>138</td>
<td>100</td>
</tr>
<tr>
<td>13</td>
<td>007</td>
<td>10/01/98</td>
<td>68</td>
<td>140</td>
<td>98</td>
</tr>
</tbody>
</table>

Program 5-7 Using PROC FREQ and an output data set to identify a duplicate ID

```sas
PROC FREQ DATA=CLEAN.PATIENTS NOPRINT;
   TABLES PATNO / OUT=DUP KEEP=PATNO COUNT WHERE=(COUNT GT 1);
RUN;

PROC SORT DATA=DUP NO;
   BY PATNO;
RUN;

PROC SORT DATA=DUP_NO;
   BY PATNO;
RUN;

DATA DUP;
   MERGE TMP DUP_NO(IN=YES DUP DROP=COUNT);
   BY PATNO;
   IF YES_DUP;
RUN;

PROC PRINT DATA=DUP;
   TITLE "Listing of Data Set DUP";
RUN;
```
More efficient program to accomplish similar task

等奖程

Identifying subjects with "n" observations each using data step

- sometimes we know how many observations there should be for each subject and we need to verify that the correct number are there in data set

- example program lists all patient IDs that do not have exactly two observations each
Accomplishing same task using PROC FREQ

• usually easier to have a SAS procedure do the work than to code up a fancy DATA step

/~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Program 5-12 Using PROC FREQ to list all ID's for subjects who do not have exactly two observations
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
PROC FREQ DATA=CLEAN.PATIENT2 NOPRINT;
    TABLES PATNO / OUT=DUP_NOKEEP=PATNO COUNT
        WHERE=(COUNT NE 2));
RUN;

DATA _NULL_;
    TITLE "Patient ID's for Patients with Other than Two Observations";
    FILE PRINT;
    PUT "Patient number " PATNO " has " COUNT "observation(s).";
RUN;

Patient ID's for Patients with Other than Two Observations 12
08:41 Monday, July 7, 2003
Patient number 002 has 3 observation(s).
Patient number 003 has 1 observation(s).
Patient number 006 has 1 observation(s).