Lab session 9
Macros for data cleaning

July 20, 2005

1 Files to download

patients.txt
lab9.sas

2 Reading in the example data file

options linesize = 72 mprint ; * mprint option puts macro debugging info in the log file ;

 /*-----------------------------------------------------------------
 Program to create SAS data set PATIENTS
 *------------------------------------------------------------------- */
*LIBNAME CLEAN "C:\TEMP";
LIBNAME CLEAN "space/kcowles/172/clean";
DATA CLEAN.PATIENTS;
INFILE "C:\TEMP\patients.dat" PAD;
* INFILE "space/kcowles/172/clean/patients.dat" 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;

3 A macro to list out-of-range data values

Recall that a macro program is a piece of SAS code in which parts of the code are replaced by variable information by the macro processor before the code is processed in the usual way by the SAS compiler. When the macro is called, the macro processor replaces each of the arguments with the values you specify. Then, in the macro program, every macro variable is replaced by the assigned value.

/*============================================================================
 Program 2-7 Writing a macro to list out-of-range data values
============================================================================*/
%MACRO RANGE(DSN,VAR,LOW,HIGH,IDVAR);
| Program Name: RANGE.SAS in C:\CLEANING |
| Purpose: Macro that takes lower and upper limits for a numeric variable, and an ID variable to print out an exception report to the output window |
| Arguments: DSN - Data set name |
| VAR - Numeric variable to test |
| LOW - Lowest valid value |
| HIGH - Highest valid value |
| IDVAR - ID variable to print in the exception report |
| Example: %RANGE(CLEAN.PATIENTS,HR,40,100,PATNO) |
/*============================================================================*/

%MACRO RANGE(DSN,VAR,LOW,HIGH,IDVAR);

| Program Name: RANGE.SAS in C:\CLEANING |
| Purpose: Macro that takes lower and upper limits for a numeric variable, and an ID variable to print out an exception report to the output window |
| Arguments: DSN - Data set name |
| VAR - Numeric variable to test |
| LOW - Lowest valid value |
| HIGH - Highest valid value |
| IDVAR - ID variable to print in the exception report |
| Example: %RANGE(CLEAN.PATIENTS,HR,40,100,PATNO) |
/*============================================================================*/

%MACRO RANGE(DSN,VAR,LOW,HIGH,IDVAR);
4 A program to use PROC UNIVARIATE to look for highest and lowest values by percentiles

Note: the %EVAL is needed in the 7th line to perform the integer arithmetic. If the value of LOW_PER were 20, the value of UP_PER without the %EVAL function would be the text string "100 - 20" instead of 80.

The data set TMP created by proc univariate contains only one observation and three variables: PATNO, L_20 and L_80.

We want to add the two values of L_20 and L_80 to every observation in the original dataset.

The SET statement brings in an observation from the PATIENTS dataset, keeping only the variables PATNO and HR. Because all variables brought in with a SET statement are automatically retained, the values for L_20 and L_80 are added to every observation in the dataset HILO.

/***************************************************************************
***Program 2-12 Program to print the top and bottom "n" percent of data values, using PROC UNIVARIATE
***************************************************************************/
***Solution using PROC UNIVARIATE and Percentiles;
LIBNAME CLEAN "C:\CLEANING";
***The two macro variables that follow define the lower and upper percentile cut points;
%LET LOW_PER=20;
***Compute the upper cut-off value;
%LET UP_PER= %EVAL(100 - &LOW_PER);
***Choose a variable to operate on;
%LET VAR = HR;
PROC UNIVARIATE DATA=CLEAN.PATIENTS NOPRINT;
VAR &VAR;
ID PATNO;
OUTPUT OUT=TMP PCTLPTS=&LOW_PER &UP_PER PCTLPRE = L_;
  * PCTLPRE = L sets prefix of variable names to L_;
RUN;

DATA HILO;
    SET CLEAN.PATIENTS(KEEP=PATNO &VAR);
    ***Bring in upper and lower cutoffs for variable;
    IF _N_ = 1 THEN SET TMP;
    IF &VAR LE L_&LOW_PER THEN DO;
        RANGE = 'LOW ';
        OUTPUT;
    END;
    ELSE IF &VAR GE L_&UP_PER THEN DO;
        RANGE = 'HIGH';
        OUTPUT;
    END;
RUN;

PROC SORT DATA=HILO( WHERE=(&VAR NE .));
    BY DESCENDING RANGE &VAR;
RUN;
PROC PRINT DATA=HILO;
    TITLE "High and Low Values For Variables";
    ID PATNO;
    VAR RANGE &VAR;
RUN;

The output:
High and Low Values For Variables  2
20:04 Monday, July 14, 2003

<table>
<thead>
<tr>
<th>PATNO</th>
<th>RANGE</th>
<th>HR</th>
</tr>
</thead>
<tbody>
<tr>
<td>020</td>
<td>LOW</td>
<td>10</td>
</tr>
<tr>
<td>014</td>
<td>LOW</td>
<td>22</td>
</tr>
<tr>
<td>023</td>
<td>LOW</td>
<td>22</td>
</tr>
<tr>
<td>022</td>
<td>LOW</td>
<td>48</td>
</tr>
<tr>
<td>003</td>
<td>LOW</td>
<td>58</td>
</tr>
<tr>
<td>019</td>
<td>LOW</td>
<td>58</td>
</tr>
<tr>
<td>001</td>
<td>HIGH</td>
<td>88</td>
</tr>
<tr>
<td>007</td>
<td>HIGH</td>
<td>88</td>
</tr>
<tr>
<td>321</td>
<td>HIGH</td>
<td>900</td>
</tr>
<tr>
<td>004</td>
<td>HIGH</td>
<td>101</td>
</tr>
<tr>
<td>017</td>
<td>HIGH</td>
<td>208</td>
</tr>
<tr>
<td>008</td>
<td>HIGH</td>
<td>210</td>
</tr>
</tbody>
</table>

5  Turning the previous program into a macro

PROC DATASETS is used to delete the two temporary datasets TMP and HILO.

/***************************************************************************/
Program 2-13 Creating a macro to list the highest and lowest "n" percent of the
data, using PROC UNIVARIATE
***************************************************************************/

%MACRO HILOWPER(DSN,VAR,PERCENT,IDVAR);
***Compute upper percentile cutoff;
%LET UP_PER = %EVAL(100 - &PERCENT);
PROC UNIVARIATE DATA=&DSN NOPRINT;
    VAR &VAR;
    ID &IDVAR;
    OUTPUT OUT=TMP PCTLPTS=&PERCENT &UP_PER PCTLPRE = L_;
RUN;
DATA HILO;
    SET &DSN(KEEP=&IDVAR &VAR);

%MACRO HILOWPER(DSN,VAR,PERCENT,IDVAR);
IF _N_ = 1 THEN SET TMP;
IF _VAR_ LE L_&_PERCENT THEN DO;
  RANGE = 'LOW ';
  OUTPUT;
END;
ELSE IF _VAR_ GE L_&_UP_PER THEN DO;
  RANGE = 'HIGH';
  OUTPUT;
END;
RUN;

PROC SORT DATA=HILO(WHERE=(&VAR NE .));
  BY DESCENDING RANGE &VAR;
RUN;

PROC PRINT DATA=HILO;
  TITLE "Low And High Values For Variables";
  ID &IDVAR;
  VAR RANGE &VAR;
RUN;

PROC DATASETS LIBRARY=WORK NOLIST;
  DELETE TMP;
  DELETE HILO;
RUN;
QUIT;
%MEND HILOWPER;

%HILOWPER(CLEAN.PATIENTS,HR,20,PATNO)
%HILOWPER(CLEAN.PATIENTS,SBP,20,PATNO)
%HILOWPER(CLEAN.PATIENTS,DBP,20,PATNO)

6 Introduction to PROC RANK

PROC RANK produces a new variable (or replaces the values of an existing variable) with values equal to the ranks of another variable. The GROUPS= option on PROC RANK allows you to group your data values. For example, if you set GROUPS = 4, the new variable that usually holds rank values will instead have values of 0, 1, 2, and 3, with those observations in group 0 being in the bottom quartile, etc.

Here is sample code to illustrate how PROC RANK works.

data sample ;
  input myval ;
datalines ;
12
72
46
192
13
81
0
42
189
16
3
;
run ;

proc rank data = sample out = smplrank;
  var myval ;
  ranks valrank ;
run ;

proc print data = smplrank ;
run ;

proc rank data = sample out = smplrank2 groups = 3;
  var myval ;
  ranks valrank ;
run ;

proc print data = smplrank2 ;
run ;

Here is the output.
7 Using PROC RANK to look for highest and lowest values by percentage

This method is simpler and more efficient, but slightly less accurate than the method using PROC UNIVARIATE.

The %SYSEVALF function allows noninteger arithmetic and also provides various conversions (CEIL, FLOOR, INTEGER, or BOOLEAN).
PROC DATASETS LIBRARY=WORK NODLIST;
    DELETE NEW;
RUN;
QUIT;
%MEND HI_LOW_P;

%HlOw_P(CLEAN.PATIENTS,SBP,20,PATNO)