Files to download

patients.txt
lab9.sas

Reading in the example data file

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

Program to create SAS data set PATIENTS
***************************************************************
*LIBNAME CLEAN "C:\temp";
*LIBNAME CLEAN "/space/kcoles/172/clean";

DATA CLEAN.PATIENTS;
  INFILE "C:\temp\patients.dat" PAD;
  * INFILE "/space/kcoles/172/clean/patients.dat" PAD;
  INPUT GENDER $3.
    64 GENDER $1.
    65 VISIT MMDYY10.
    615 HR $3.
    618 SBP $3.
    621 DBP $3.
    624 DX $3.
    627 AE $1.;
  LABEL PATNO = "Patient Number"

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 variable is replaced by the assigned value.

Program 2-7 Writing a macro to list out-of-range data values
***************************************************************

DATA CLEAN.PATIENTS;
  INFILE "C:\temp\patients.dat" PAD;
  * INFILE "/space/kcoles/172/clean/patients.dat" PAD;
  INPUT GENDER $3.
    64 GENDER $1.
    65 VISIT MMDYY10.
    615 HR $3.
    618 SBP $3.
    621 DBP $3.
    624 DX $3.
    627 AE $1.;
  LABEL PATNO = "Patient Number"

MACRO RANGE(SSN,VAR,LOW,HIGH,IOVAR);

GENDER = "Gender"
VISIT = "Visit Date"
HR = "Heart Rate"
SBP = "Systolic Blood Pressure"
DBP = "Diastolic Blood Pressure"
DX = "Diagnosis Code"
AE = "Adverse Event?";

FORMAT VISIT MMMDDYY10.;
RUN;

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

Note: The \texttt{IFN} is needed in the 7th line to perform the integer arithmetic. If the value of \texttt{LOWPER} were 20 the value of \texttt{UPPER} without the \texttt{IFN} function would be the text string "100.20" instead of 80.

The data set \texttt{TMP} created by proc univariate contains only one observation and three variables: \texttt{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 \texttt{SET} statement brings in an observation from the \texttt{PATIENTS} dataset, keeping only the variables \texttt{PATNO} and \texttt{HR}. Because all variables brought in with a \texttt{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 \texttt{PROC UNIVARIATE} **********************************************************************************/ ***Solution using \texttt{PROC UNIVARIATE} and Percentiles; ***\texttt{LIBNAME} CLEAN "C:\CLEANING"; ***The two macro variables that follow define the lower and upper percentile cut points; ***Change the value in the line below to the percentile cut-off you want; \LET LOWPER=20; ***Compute the upper cut-off value; \LET UPPER= \texttt{IFN}(100 - \texttt{LOWPER}); ***Choose a variable to operate on; \LET VAR = HR; \texttt{PROC UNIVARIATE} DATA=\texttt{PATIENTS} NOPRINT; \texttt{VAR} \texttt{VAR}; 1 \texttt{PRTNO}; \texttt{OUTPUT} OUT=\texttt{TMP} \\texttt{PCTLPTS} = \texttt{LOWPER} \&UPPER \texttt{PCTLPR} = \texttt{L}; * \texttt{PCTLPR} = \texttt{L} sets prefix of variable names to \texttt{L}_;
5 Turning the previous program into a macro

PROC DATA SETS 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

*--------------------------------------------------------------------------*/
| Program Name: HILOWPER, SAS in C:\CLEANING |
| Purpose: To list the n percent highest and lowest values for |
| a selected variable, |
| Arguments: DSName Data set name |
| VAR Numeric variable to test |
| PERCENT Upper and lower percentile cutoff |
| IDVAR ID variable to print in the report |
| Example: %HILOWPER(DSNAME,VARNAME,PERCENT, IDVAR) |

*--------------------------------------------------------------------------*/

%MACRO HILOWPER(DSNAME,VARNAME,PERCENT, IDVAR);

%let UP_PER = &IDVAR*(100 - &PERCENT);

%LET UP_PER = &IDVAR*(100 - &PERCENT);

PROC UNIVARIATE DATA=&DSN NOPRINT;
   VAR &VAR;
   ID &IDVAR;
   OUTPUT OUT=TMP PCTLPNTS=&PERCENT &UP_PER PCTLPRE = L;
RUN;

DATA HILO;
   SET &DSN(KEEP=&IDVAR &VAR);

004 HIGH 101
017 HIGH 208
008 HIGH 210
321 HIGH 900

The output:

High and Low Values For Variables 2004 Monday, July 14, 2003

<table>
<thead>
<tr>
<th>PATNO</th>
<th>RANGE</th>
<th>NR</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>HIGH</td>
<td>90</td>
<td></td>
</tr>
</tbody>
</table>
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 with 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:

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

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

proc print data = smprank;
  run;

proc rank data = sample out = smprank2 groups = 3;
  var myval ;
  ranks valrank;
  run;

proc print data = smprank2;
  run;
```

Here is the output,

The SAS System
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 SYMFAIL function allows noninteger arithmetic and also provides various conversions (CEIL, FLOOR, INTEGER, or BOOLEAN).

/* Program 2-14: Creating a macro to list the highest and lowest "n" percent of the data, using PROC RANK  
*******************************************************************************/
*==========================================================================*
| Macro Name: HLLOW_P  
| Purpose: To list the upper and lower n% of values  
| Arguments: DSN - Data set name (one- or two-level)  
| VAR - Variable to test  
| PERCENT - Upper and lower n%  
| IDVAR - ID variable  
| Example: %HLLOW_P(CLEAN.PATIENTS,BP,20,PATNO)  
*==========================================================================*/

%MACRO HLLOW_P(DSN,VAR,PERCENT,IDVAR);
  *** Compute number of groups for PROC RANK;
  %LET GRP = %SYMFAIL(100 / &PERCENT,FLOOR);
  *** Value of the highest group from PROC RANK, equal to the number of groups - 1;
  %LET TOP = %SYMFAIL(&GRP - 1);

  PROC FORMAT;
  VALUE RANK 0='Low' &TOP='High';
  RUN;

  PROC RANK DATA=&DSN OUT-NEW GROUPS=&GRP;
    VAR &VAR;
    RANKS RANGES;
  RUN;

  *** Sort and keep top and bottom n%:
  PROC SORT DATA=NEW (WHERE=(RANGE IN (0,&TOP)));
    BY &VAR;
  RUN;

  *** Produce the report:
  PROC PRINT DATA=NEW;
    TITLE "Upper and lower &PERCENT.\% Values for &VAR in &DSN";
    ID &IDVAR;
    VAR RANGE &VAR;
    FORMAT RANGES;
RUN;
PROC DATASETS LIBRARY=WORK MULIST;
  DELETE NEW;
RUN;
QUIT;
%MEND BI_LOW_P;
%HIGH_LOW_P(CLEAN,PATIENTS,SBP,20,PATNO)