BEYOND FORMAT BASICS

- ORIGINALLY A WORKSHOP PRESENTATION
- TOO MANY TOPICS FOR 45 MINUTES
- OVERVIEW (WHAT MOST USERS KNOW)
- NEW POSSIBILITIES
- SPECIFIC EXAMPLES
provide information about the values of variables ...

```
proc format;
value $gender
  '1' = 'MALES'
  '2' = 'FEMALES'
;
run;
```

group observations based on values of variables ...

```
proc format;
value age
  low-18  = '<19'
  19-64   = '19-64'
  65-high = '65+'
;
run;
```
BEYOND BASICS

CREATE FORMATS FROM DATA SETS

CNTLIN ... 'CONTROL IN' DATA SET

CREATE DATA SETS FROM FORMATS

CNTLOUT ... 'CONTROL OUT' DATA SET

PUTN AND PUTC FUNCTIONS VERSUS PUT

SELECT A FORMAT FOR USE IN A FUNCTION AT EXECUTION TIME BASED ON VALUE OF ONE OR MORE VARIABLES IN A DATA SET
SELECT OBSERVATIONS AND/OR ADD VARIABLES TO A DATA SET

USE FORMATS IN PLACE OF DATA STEP MERGE OR PROC SQL

NESTED FORMATS

USE 'RULES' FROM ONE FORMAT AS THE PORTION OF THE 'RULES' IN ANOTHER FORMAT ... OR, THE FORMATTED VALUE OF A VARIABLE IS ANOTHER FORMAT)
PRELOADED FORMATS

USE LEVELS OF A FORMAT TO CONTROL PROCEDURE OUTPUT RATHER THAN THE RANGE OF VALUES OF VARIABLES IN A DATA SET ... PROCS MEANS/SUMMARY, TABULATE, REPORT

MULTI-LABEL FORMATS

USE OVERLAPPING RANGES OF VALUES

DOES FORMAT STATEMENT LOCATION MATTER

BEGINNING VERSUS END OF A DATA STEP
EXAMPLE #1

USE A DATA SET TO CREATE A FORMAT ...

SIMPLE PROBLEM ... 5 CODES, 5 LABELS ... MANUALLY ENTER SAS CODE ...

```
proc format;
value $drg
  '001' = 'CRANIO TOMY AGE >17 EXCEPT FOR TRAUMA'
  '002' = 'CRANIO TOMY FOR TRAUMA AGE >17'
  '004' = 'SPINAL PROCEDURES'
  '005' = 'EXTRACRANIAL VASCULAR PROCEDURES'
o ther = 'UNKNOWN';
;
run;
```
HOWEVER ... 600+ 'RULES' ...

001  CRANIOTOMY AGE >17 EXCEPT FOR TRAUMA
002  CRANIOTOMY FOR TRAUMA AGE >17
004  SPINAL PROCEDURES

<635 MORE DRGS AND LABELS>

807  COMBINED ANTERIOR/POSTERIOR SPINAL FUSION W/O CC
808  PERCUATANEOUS CARDIOVASCULAR PROC W AMI,HEART FAILURE OR SHOCK
809  OTHER CARDIOTHORACIC PROC W PDX CONGENITAL ANOMALY
COMMON PROBLEM ... CODES AND LABELS IN A RAW DATA FILE OR IN A DATA SET ... HOW TO CREATE A FORMAT ... 

ANSWER ... CNTLIN DATA SET 

MINIMUM CONTENT ... 3 VARIABLES

  FMTNAME (VALUE) 
  START (VALUE RANGE, LEFT SIDE OF =S) 
  LABEL (FORMATTED VALUE, RIGHT SIDE OF =S)
data drg_fmt;
retain fmtname '$drg; 
infile "k:\workshops\hw08\drgs.dat";
input start $3. +1 label $70.; 
run;

proc format cntlin=drg_fmt; 
select $drg; 
run;
### BEYOND FORMAT BASICS

<table>
<thead>
<tr>
<th>FORMAT NAME: SDRG</th>
<th>LENGTH: 70</th>
<th>NUMBER OF VALUES: 641</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIN LENGTH: 1</td>
<td>MAX LENGTH: 70</td>
<td>DEFAULT LENGTH: 70</td>
</tr>
<tr>
<td>FUZZ: 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>START</th>
<th>END</th>
<th>LABEL (VER. V7:V8 07OCT2007:16:14:23)</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>001</td>
<td>CRANIOTOMY AGE &gt;17 EXCEPT FOR TRAUMA</td>
</tr>
<tr>
<td>002</td>
<td>002</td>
<td>CRANIOTOMY FOR TRAUMA AGE &gt;17</td>
</tr>
<tr>
<td>004</td>
<td>004</td>
<td>SPINAL PROCEDURES</td>
</tr>
<tr>
<td>005</td>
<td>005</td>
<td>EXTRACRANIAL VASCULAR PROCEDURES</td>
</tr>
<tr>
<td>006</td>
<td>006</td>
<td>CARPAL TUNNEL RELEASE</td>
</tr>
<tr>
<td>007</td>
<td>007</td>
<td>PERIPH &amp; CRANIAL NERVE &amp; OTHER NERV SYST</td>
</tr>
<tr>
<td>008</td>
<td>008</td>
<td>PERIPH &amp; CRANIAL NERVE &amp; OTHER NERV SYST</td>
</tr>
<tr>
<td>009</td>
<td>009</td>
<td>SPINAL DISORDERS &amp; INJURIES</td>
</tr>
<tr>
<td>010</td>
<td>010</td>
<td>NERVOUS SYSTEM NEOPLASMS W CC</td>
</tr>
<tr>
<td>011</td>
<td>011</td>
<td>NERVOUS SYSTEM NEOPLASMS W/O CC</td>
</tr>
<tr>
<td>012</td>
<td>012</td>
<td>DEGENERATIVE NERVOUS SYSTEM DISORDERS</td>
</tr>
<tr>
<td>013</td>
<td>013</td>
<td>MULTIPLE SCLEROSIS &amp; CEREBELLAR ATAXIA</td>
</tr>
<tr>
<td>014</td>
<td>014</td>
<td>SPECIFIC CEREBROVASCULAR DISORDERS EXCEPT</td>
</tr>
</tbody>
</table>

| 800   | 800 | TUBERCULOSIS W CC                      |
| 801   | 801 | TUBERCULOSIS W/O CC                   |
| 802   | 802 | PNEUMOCYSTOSIS                         |
| 803   | 803 | ALLOGENEIC BONE MARROW TRANSPLANT      |
| 804   | 804 | AUTOLOGOUS BONE MARROW TRANSPLANT      |
| 805   | 805 | SIMULTANEOUS KIDNEY/PANCREAS TRANSPLANT|
| 806   | 806 | COMBINED ANTERIOR/POSTERIOR SPINAL FUSION |
| 807   | 807 | COMBINED ANTERIOR/POSTERIOR SPINAL FUSION |
| 808   | 808 | PERCUATANEOUS CARDIOVASCULAR PROC W AMI, |
| 809   | 809 | OTHER CARDIOTHORACIC PROC W PDX CONGENIT |
WHAT HAPPENS IF A CODE IS FOUND THAT IS NOT IN THE VALUE RANGE ... ADD AN 'OTHER' CONDITION ...

data drg_fmt;
retain fmtname '$drg';
infile "k:\workshops\hw08\drgs.dat" end=last;
input start $3. +1 label $50.;
output;
if last then do;
    hlo = 'o';
    label = 'UNKNOWN';
    output;
end;
run;
<table>
<thead>
<tr>
<th>START</th>
<th>END</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>780</td>
<td>780</td>
<td>ACUTE LEUKEMIA W/O MAJOR O.R. PROCEDURE</td>
</tr>
<tr>
<td>781</td>
<td>781</td>
<td>ACUTE LEUKEMIA W/O MAJOR O.R. PROCEDURE</td>
</tr>
<tr>
<td>782</td>
<td>782</td>
<td>ACUTE LEUKEMIA W/O MAJOR O.R. PROCEDURE</td>
</tr>
<tr>
<td>783</td>
<td>783</td>
<td>ACUTE LEUKEMIA W/O MAJOR O.R. PROCEDURE</td>
</tr>
<tr>
<td>784</td>
<td>784</td>
<td>ACQUIRED HEMOLYTIC ANEMIA OR SICKLE CELL</td>
</tr>
<tr>
<td>785</td>
<td>785</td>
<td>OTHER RED BLOOD CELL DISORDERS AGE &lt;18</td>
</tr>
<tr>
<td>786</td>
<td>786</td>
<td>MAJOR HEAD &amp; NECK PROCEDURES FOR MALIGNA</td>
</tr>
<tr>
<td>787</td>
<td>787</td>
<td>LAPAROSCOPIC CHOLECYSTECTOMY W C.D.E</td>
</tr>
<tr>
<td>789</td>
<td>789</td>
<td>MAJ JOINT &amp; LIMB REATTACH PROC OF LOW EX</td>
</tr>
<tr>
<td>790</td>
<td>790</td>
<td>WOUND DEBRID &amp; SKIN GRAFT FOR OPEN WND,MS</td>
</tr>
<tr>
<td>791</td>
<td>791</td>
<td>WOUND DEBRIDEMENTS FOR OPEN WOUND INJURY</td>
</tr>
<tr>
<td>792</td>
<td>792</td>
<td>CRANIOTOMY FOR MULTIPLE SIG TRAUMA WITH</td>
</tr>
<tr>
<td>793</td>
<td>793</td>
<td>PROC FOR MULT SIG TRAUMA EXC CRANIOTOMY W</td>
</tr>
<tr>
<td>794</td>
<td>794</td>
<td>DIAG FOR MULTIPLE SIGNIFICANT TRAUMA WIT</td>
</tr>
<tr>
<td>795</td>
<td>795</td>
<td>LUNG TRANSPLANT</td>
</tr>
<tr>
<td>796</td>
<td>796</td>
<td>LOWER EXTREMITY REVASCULARIZATION W CC</td>
</tr>
<tr>
<td>797</td>
<td>797</td>
<td>LOWER EXTREMITY REVASCULARIZATION W/O CC</td>
</tr>
<tr>
<td>798</td>
<td>798</td>
<td>TUBERCULOSIS W OPERATING ROOM PROCEDURE</td>
</tr>
<tr>
<td>799</td>
<td>799</td>
<td>TUBERCULOSIS, LEFT AGAINST MEDICAL ADVIC</td>
</tr>
<tr>
<td>800</td>
<td>800</td>
<td>TUBERCULOSIS W CC</td>
</tr>
<tr>
<td>801</td>
<td>801</td>
<td>TUBERCULOSIS W/O CC</td>
</tr>
<tr>
<td>802</td>
<td>802</td>
<td>PNEUMOCYSTOSIS</td>
</tr>
<tr>
<td>803</td>
<td>803</td>
<td>ALLOGENEIC BONE MARROW TRANSPLANT</td>
</tr>
<tr>
<td>804</td>
<td>804</td>
<td>AUTOLOGOUS BONE MARROW TRANSPLANT</td>
</tr>
<tr>
<td>805</td>
<td>805</td>
<td>SIMULTANEOUS KIDNEY/PANCREAS TRANSPLANT</td>
</tr>
<tr>
<td>806</td>
<td>806</td>
<td>COMBINED ANTERIOR/POSTERIOR SPINAL FUSION</td>
</tr>
<tr>
<td>807</td>
<td>807</td>
<td>COMBINED ANTERIOR/POSTERIOR SPINAL FUSION</td>
</tr>
<tr>
<td>808</td>
<td>808</td>
<td>PERCUATANEOUS CARDIOVASCULAR PROC W AMI,</td>
</tr>
<tr>
<td>809</td>
<td>809</td>
<td>OTHER CARDIOTHORACIC PROC W PDX CONGENIT</td>
</tr>
<tr>
<td><strong>OTHER</strong></td>
<td><strong>OTHER</strong></td>
<td>UNKNOWN</td>
</tr>
</tbody>
</table>
lifetime warranty question ... we keep all our information about screening clinics in an Access database ... how can we always use the latest information from that file ... 

good application for a CNTLIN data set

eexample on the right contains hospital codes and hospital names
libname x access path="k:\monsug\codes.mdb";

data fmt;
retain fmtname 'hosp2name';
set x.hospitals (rename=(code=start name=label))
   end=last;
output;
if last then do;
   hlo = 'o';
   label = 'UNKNOWN';
   output;
end;
run;

proc format cntlin=fmt;
run;

libname x clear;
EXAMPLE #2

USE A PUTN AND PUTC FUNCTION TO ADD VARIABLES TO A DATA SET ...

FIRST ... SIMPLE EXAMPLE ... DATA SET CONTAINS THE VARIABLE BMI ... IS A PERSON NORMAL, AT RISK, OR OVERWEIGHT

ASSUME RANGES FOR ALL THREE CATEGORIES ARE THE SAME REGARDLESS OF AGE ... USE A FORMAT PLUS A PUT FUNCTION TO CREATE A NEW VARIABLE
proc format;
value bmi2group
  low   - 17.9 = 'normal'
  18    - 19.0 = 'risk'
  19.1  - high = 'overweight'
;
run;

data bmi;
input age_group : $1. bmi @@;
bmi_group = put(bmi,bmi2group.);
datalines;
2 17.8 5 16.7 2 18.6 3 17.6 4 17.0 3 18.2 4 18.0
;
run;
ANOTHER LIFETIME WARRANTY QUESTION ... THE RULES FOR BMI VARY BY AGE GROUP ...

proc format;
value age2_  low-17.9='normal'  18.0-19.0='at risk'
19.1-high='overweight';

value age3_  low-17.1='normal'  17.2-18.1='at risk'
18.2-high='overweight';

value age4_  low-16.7='normal'  16.8-17.9='at risk'
18.0-high='overweight';

value age5_  low-16.7='normal'  16.8-18.1='at risk'
18.2-high = 'overweight';
run;
data bmi;
length bmi_group $10; ◄
input age_group : $1. bmi @@;

* PUTN accepts a variable as the 2nd argument;
* format varies by age group;

bmi_group=putn(bmi,cat('age',age_group,'_'));

datalines;
2 17.8 5 16.7 2 18.6 3 17.6 4 17.0
3 18.2 4 18.0
;
run;
WHY THE LENGTH STATEMENT ...  

VARIABLE CREATED WITH A PUT FUNCTION IS A CHARACTER VARIABLE WITH A LENGTH EQUAL TO THE LENGTH OF THE LONGEST LABEL IN THE FORMAT USED AS THE 2ND ARGUMENT

VARIABLE CREATED WITH A PUTN FUNCTION IS A CHARACTER VARIABLE WITH A LENGTH OF 200

WHY THE CAT FUNCTION ...

ELIMINATES THE NEED TO CREATE A NEW VARIABLE THAT CONTAINS THE FORMAT NAME
PASS / FAIL GRADES VARY BY YEAR ...

proc format;
  value $FR 'A'-'D' = 'PASS' other = 'FAIL';
  value $SO 'A'-'C' = 'PASS' other = 'FAIL';
  value $JR 'A'-'B' = 'PASS' other = 'FAIL';
  value $SR 'A' = 'PASS' other = 'FAIL';
run;

data students;
  length pass_fail $4.; ▼
  input class : $2. grade : $1. @@;
  pass_fail = putc(grade,cat('$',class)); ▼
  datalines;
  FR D SO A SO D JR B JR C SR A SR B
  ;
run;
WHY THE LENGTH STATEMENT ...

VARIABLE CREATED WITH A PUTC FUNCTION IS A CHARACTER VARIABLE WITH A LENGTH EQUAL TO THE LENGTH OF THE VARIABLE USED AS THE 1ST ARGUMENT

NOTE: DIFFERENT LENGTHS THAT RESULT FROM PUT, PUTN, AND PUTC

WHY THE CAT FUNCTION ...

ELIMINATES THE NEED TO CREATE A NEW VARIABLE THAT CONTAINS THE FORMAT NAME
EXAMPLE #3

NESTED FORMATS ... OR USE A PRE-EXISTING FORMAT AS A LABEL IN ANOTHER FORMAT ...

THE PRE-EXISTING FORMAT CAN BE USER-WRITTEN OR IT CAN BE SAS-SUPPLIED ...
ASSUME THAT YOU HAVE A FORMAT LIBRARY ... IT CONTAINS A FORMAT NAMED $NUM2NAM ... THAT FORMAT HAS HOSPITAL CODES AND HOSPITAL NAMES ...

<table>
<thead>
<tr>
<th>START</th>
<th>END</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0001</td>
<td>0001</td>
<td>ALBANY MEDICAL CENTER HOSPITAL</td>
</tr>
<tr>
<td>0002</td>
<td>0002</td>
<td>CHILD S HOSPITAL</td>
</tr>
<tr>
<td>0005</td>
<td>0005</td>
<td>ST PETERS HOSPITAL</td>
</tr>
<tr>
<td>0012</td>
<td>0012</td>
<td>OSWEGO HOSPITAL</td>
</tr>
<tr>
<td>0016</td>
<td>0016</td>
<td>CATHOLIC MED CTR</td>
</tr>
<tr>
<td>0025</td>
<td>0025</td>
<td>COHOES MEMORIAL HOSPITAL</td>
</tr>
<tr>
<td><strong>OTHER</strong></td>
<td><strong>OTHER</strong></td>
<td>UNKNOWN</td>
</tr>
</tbody>
</table>

YOU HAVE ONE ADDITION (A NEW HOSPITAL) AND ONE CHANGE (ONE HOSPITAL HAS CLOSED) ... YOU DO NOT WANT TO MODIFY THE FORMAT IN THE FORMAT LIBRARY
USE A FORMAT NAME AS THE LABEL FOR OTHER ...

proc format;
value $numplus ▶
'0004' = 'MEMORIAL HOSPITAL'
'0016' = 'CATHOLIC MED CTR (**CLOSED***)
other = [$num2nam35.] ▶
;
run;

<table>
<thead>
<tr>
<th>FORMAT NAME: $NUMPLUS LENGTH: 35 NUMBER OF VALUES: 3</th>
<th>LABEL (VER. V7\V8 07OCT2007:22:19:11)</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>END</td>
</tr>
<tr>
<td>0004</td>
<td>0004</td>
</tr>
<tr>
<td>0016</td>
<td>0016</td>
</tr>
<tr>
<td><strong>OTHER</strong></td>
<td><strong>OTHER</strong></td>
</tr>
</tbody>
</table>
TASK ... ALL DATES WITH A GIVEN RANGE ARE TO BE Labeled AS 'OK' ... ALL OTHER DATES SHOULD BE DISPLAYED WITH A DATE9. FORMAT ...

proc format;
value chk_date '01FEB1999'd - '31JUL1999'd = 'OK' other = [date9.];
run;

| FORMAT NAME: CHK_DATE LENGTH: 9 NUMBER OF VALUES: 2 |
| MIN LENGTH: 1 MAX LENGTH: 40 DEFAULT LENGTH 9 FUZZ: STD |
| START | END | LABEL (VER. V7|V8 07OCT2007:22:26:41) |
| **OTHER** | 14276 | **OTHER** | 14456 | OK [DATE9.] |
EXAMPLE #4

SAS PROCS ONLY COUNT "WHAT'S THERE" ... THEY DO NOT PRODUCE COUNTS OF "WHAT'S NOT THERE" ...

... YOU CAN FORCE PROCS TO CREATE TABLES BASED ON FORMAT VALUES, NOT JUST ON DATA CONTENTS ...

USE DATA SET SASHELP.CLASS ... THERE ARE NO FEMALES AGE 16+ IN THE DATA SET ...
proc format;
value age low-15 = '<16' 16-high = '16+';
run;

proc means data=sashelp.class mean maxdec=1;
var weight;
class sex age;
format age age.;
run;

<table>
<thead>
<tr>
<th>SEX</th>
<th>AGE</th>
<th>N</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>&lt;16</td>
<td>9</td>
<td>90.1</td>
</tr>
<tr>
<td>M</td>
<td>&lt;16</td>
<td>9</td>
<td>104.4</td>
</tr>
<tr>
<td></td>
<td>16+</td>
<td>1</td>
<td>150.0</td>
</tr>
</tbody>
</table>
proc means data=sashelp.class mean maxdec=1 completetypes;
 var weight;
 class sex age / preloadfmt;
 format age age.;
 run;

<table>
<thead>
<tr>
<th>SEX</th>
<th>AGE</th>
<th>N</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;16</td>
<td>9</td>
<td>90.1</td>
</tr>
<tr>
<td></td>
<td>16+</td>
<td>0</td>
<td>.</td>
</tr>
<tr>
<td>M</td>
<td>&lt;16</td>
<td>9</td>
<td>104.4</td>
</tr>
<tr>
<td></td>
<td>16+</td>
<td>1</td>
<td>150.0</td>
</tr>
</tbody>
</table>
EXAMPLE #5

RANGES THAT OVERLAP NORMALLY CAUSE ERRORS IN PROC FORMAT ... HOWEVER, YOU CAN CREATE FORMATS WITH RANGES THAT OVERLAP USING THE MULTILABEL OPTION ...
PROC FORMAT;
value age (multilabel notsorted)
   11 - 13 = '11-13'
   11   = '   11'
   12   = '   12'
   13   = '   13'
   14 - 16 = '14-16'
   14   = '   14'
   15   = '   15'
   16   = '   16'
low - high = 'TOTAL';
run;
proc means data=sashelp.class mean maxdec=1;
var weight;
class age /mlf preloadfmt order=data;
format age age.;
run;

<table>
<thead>
<tr>
<th>Age</th>
<th>N</th>
<th>Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>11-13</td>
<td>10</td>
<td>87.4</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>67.8</td>
</tr>
<tr>
<td>12</td>
<td>5</td>
<td>94.4</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>88.7</td>
</tr>
<tr>
<td>14-16</td>
<td>9</td>
<td>114.1</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>101.9</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>117.4</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>150.0</td>
</tr>
<tr>
<td>TOTAL</td>
<td>19</td>
<td>100.0</td>
</tr>
</tbody>
</table>
LAST ... DOES LOCATION OF THE FORMAT STATEMENT IN A DATA STEP MATTER ...

proc format;
value $answer '1' = "YES" '2' = "NO"
              '3' = "DON'T KNOW" '4' = "OTHER";
run;

data results;
format answer $answer.;
input answer : $1. @@;
datalines;
1 2 3 4
;
run;

data results;
input answer : $1. @@;
format answer $answer.;
datalines;
1 2 3 4
;
run;
FORMAT PRIOR TO INPUT ...

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSWER</td>
<td>Char</td>
<td>10</td>
<td>$ANSWER.</td>
</tr>
</tbody>
</table>

FORMAT AFTER INPUT ...

<table>
<thead>
<tr>
<th>Variable</th>
<th>Type</th>
<th>Len</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANSWER</td>
<td>Char</td>
<td>1</td>
<td>$ANSWER.</td>
</tr>
</tbody>
</table>

ONLY A CONCERN WITH CHARACTER VARIABLES
BEYOND FORMAT BASICS

- PAPER ON WEB SITE HAS HYPERLINKS TO OTHER FORMAT RELATED PAPERS
- ALL SAS CODE FROM PAPER ON THE WEB SITE

Beyond FORMAT Basics
Mike Zdeb, University at Albany School of Public Health

Files associated with this paper can be found in the \nesug05\code folder on this CD. Look for file HOW8.ZIP.

The Power of PROC FORMAT
By: Jonas V. Bilanas
List price: 29.95 USD
124 pages

ISBN: 978-1-59047-573-7**
Publisher: SAS Press
Copyright Date: March 2005

Description:
Are you a programmer, statistician, or data analyst tasked with generating reports? Discover how you can put the powerful FORMAT procedure to work for you with The Power of PROC FORMAT. Written in an easy-to-follow tutorial style and illustrated with real-world examples and solutions, this handy guide introduces beginning to intermediate SAS users to the functionality of the FORMAT procedure. Learn how the FORMAT procedure can recategorize data values while doing a variety of tasks, including building user-defined formats and informats, implementing a table lookup in SAS, using the DATA step and other SAS procedures, assigning descriptive labels to data values, creating new variables and finding unexpected values, generating data extracts, and merging data sets.