

- ORIGINALLY A WORKSHOP PRESENTATION
- TOO MANY TOPICS FOR 45 MINUTES
- OVERVIEW (WHAT MOST USERS KNOW)
- NEW POSSIBILITIES
- SPECIFIC EXAMPLES

## ■ BASICS

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' = 'CRANIOTOMY AGE >17 EXCEPT FOR TRAUMA'  
'002' = 'CRANIOTOMY FOR TRAUMA AGE >17'  
'004' = 'SPINAL PROCEDURES'  
'005' = 'EXTRACRANIAL VASCULAR PROCEDURES'  
other = '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

FORMAT NAME: \$DRG      LENGTH: 70      NUMBER OF VALUES: 641 MIN LENGTH: 1      MAX LENGTH: 70      DEFAULT LENGTH 70      FUZZ: 0		
START	END	LABEL (VER. V7 V8 07OCT2007:16:14:23)
001	001	CRANIOTOMY AGE >17 EXCEPT FOR TRAUMA
002	002	CRANIOTOMY FOR TRAUMA AGE >17
004	004	SPINAL PROCEDURES
005	005	EXTRACRANIAL VASCULAR PROCEDURES
006	006	CARPAL TUNNEL RELEASE
007	007	PERIPH & CRANIAL NERVE & OTHER NERV SYST
008	008	PERIPH & CRANIAL NERVE & OTHER NERV SYST
009	009	SPINAL DISORDERS & INJURIES
010	010	NERVOUS SYSTEM NEOPLASMS W CC
011	011	NERVOUS SYSTEM NEOPLASMS W/O CC
012	012	DEGENERATIVE NERVOUS SYSTEM DISORDERS
013	013	MULTIPLE SCLEROSIS & CEREBELLAR ATAXIA
014	014	SPECIFIC CEREBROVASCULAR DISORDERS EXCEP

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 FUSIO
807	807	COMBINED ANTERIOR/POSTERIOR SPINAL FUSIO
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;
```

# BEYOND FORMAT BASICS

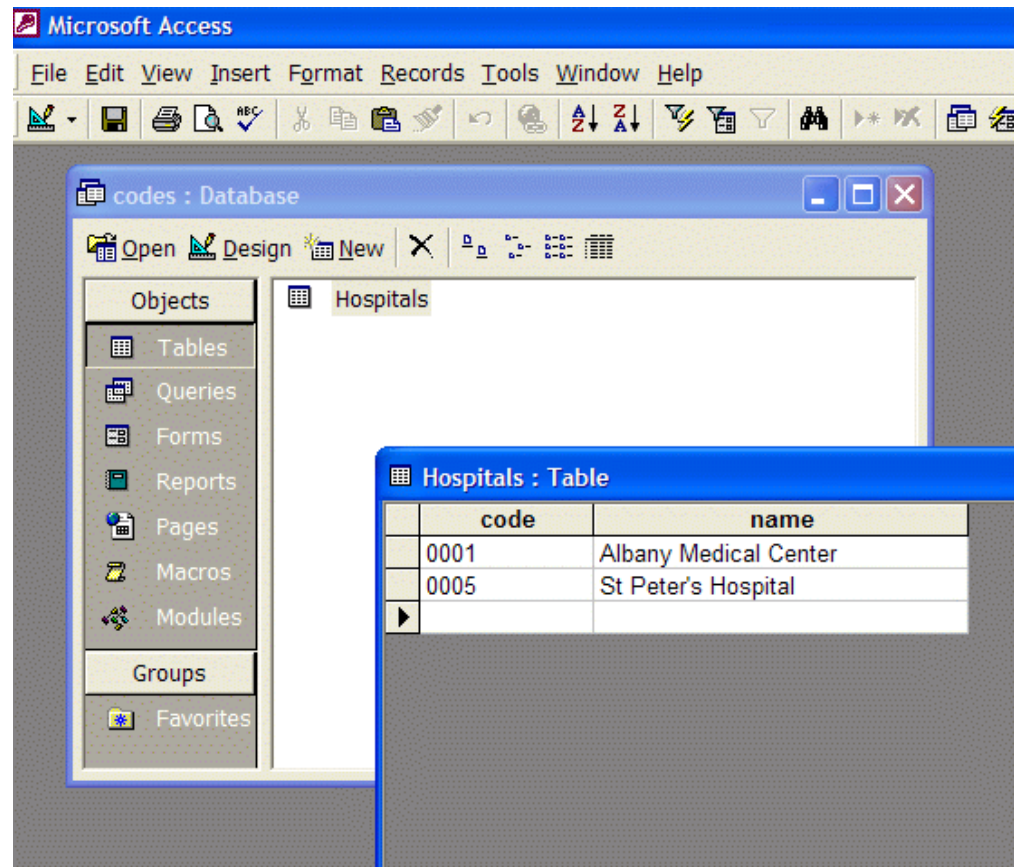
FORMAT NAME: \$DRG      LENGTH: 50      NUMBER OF VALUES: 642 MIN LENGTH: 1      MAX LENGTH: 50      DEFAULT LENGTH 50      FUZZ: 0		
START	END	LABEL (CONT'D)
780	780	ACUTE LEUKEMIA W/O MAJOR O.R. PROCEDURE
781	781	ACUTE LEUKEMIA W/O MAJOR O.R. PROCEDURE
782	782	ACUTE LEUKEMIA W/O MAJOR O.R. PROCEDURE
783	783	ACUTE LEUKEMIA W/O MAJOR O.R. PROCEDURE
784	784	ACQUIRED HEMOLYTIC ANEMIA OR SICKLE CELL
785	785	OTHER RED BLOOD CELL DISORDERS AGE <18
786	786	MAJOR HEAD & NECK PROCEDURES FOR MALIGNA
787	787	LAPAROSCOPIC CHOLECYSTECTOMY W C.D.E
789	789	MAJ JOINT & LIMB REATTACH PROC OF LOW EX
790	790	WOUND DEBRID & SKIN GRT FOR OPEN WND,MS
791	791	WOUND DEBRIDEMENTS FOR OPEN WOUND INJURI
792	792	CRANIOTOMY FOR MULTIPLE SIG TRAUMA WITH
793	793	PROC FOR MUL SIG TRAUMA EXC CRANIOTOMY W
794	794	DIAG FOR MULTIPLE SIGNIFICANT TRAUMA WIT
795	795	LUNG TRANSPLANT
796	796	LOWER EXTREMITY REVASCULARIZATION W CC
797	797	LOWER EXTREMITY REVASCULARIZATION W/O CC
798	798	TUBERCULOSIS W OPERATING ROOM PROCEDURE
799	799	TUBERCULOSIS, LEFT AGAINST MEDICAL ADVIC
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 FUSIO
807	807	COMBINED ANTERIOR/POSTERIOR SPINAL FUSIO
808	808	PERCUATANEOUS CARDIOVASCULAR PROC W AMI,
809	809	OTHER CARDIOTHORACIC PROC W PDX CONGENIT
**OTHER**	**OTHER**	UNKNOWN

## BEYOND FORMAT BASICS

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

example 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 FUNCTIONS 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 ...

FORMAT NAME: \$NUM2NAM LENGTH: 30 NUMBER OF VALUES: 7 MIN LENGTH: 1 MAX LENGTH: 40 DEFAULT LENGTH 30 FUZZ: 0		
START	END	LABEL (VER. V7 V8 07OCT2007:22:14:15)
0001	0001	ALBANY MEDICAL CENTER HOSPITAL
0002	0002	CHILDS HOSPITAL
0005	0005	ST PETERS HOSPITAL
0012	0012	OSWEGO HOSPITAL
0016	0016	CATHOLIC MED CTR
0025	0025	COHOES MEMORIAL HOSPITAL
**OTHER**	**OTHER**	UNKNOWN

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;
```

FORMAT NAME: \$NUMPLUS LENGTH: 35 NUMBER OF VALUES: 3 MIN LENGTH: 1 MAX LENGTH: 40 DEFAULT LENGTH 35 FUZZ: 0		
START	END	LABEL (VER. V7 V8 07OCT2007:22:19:11)
0004	0004	MEMORIAL HOSPITAL
0016	0016	CATHOLIC MED CTR (***CLOSED***)
**OTHER**	**OTHER**	[\$NUM2NAM35.]

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)
14276 **OTHER**	14456 **OTHER**	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;
```

SEX	AGE	N Obs	Mean
F	<16	9	90.1
M	<16	9	104.4
	16+	1	150.0

```
proc means data=sashelp.class mean maxdec=1
           completetypes; ◀
var weight;
class sex age / preloadfmt; ◀
format age age.;
run;
```

SEX	AGE	N Obs	Mean
F	<16	9	90.1
	16+	0	.
M	<16	9	104.4
	16+	1	150.0

## EXAMPLE #5

RANGES THAT OVERLAP NORMALLY CAUSE ERRORS  
IN PROC FORMAT ... HOWEVER, YOU CAN CREATE  
FORMATS WITH RANGES THAT OVERLAP USING THE  
MULTILABEL OPTION ...

MULTILABEL  
OPTION TELLS  
PROC FORMAT  
THAT RANGES  
OVERLAP

NOTSORTED  
OPTION  
PREVENTS  
SORTING OF  
THE FORMAT IN  
RANGE ORDER  
(DEFAULT)

```
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;
```

Analysis Variable : Weight		
Age	N Obs	Mean
11-13	10	87.4
11	2	67.8
12	5	94.4
13	3	88.7
14-16	9	114.1
14	4	101.9
15	4	117.4
16	1	150.0
TOTAL	19	100.0

### 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 ...

Alphabetic List of Variables and Attributes				
#	Variable	Type	Len	Format
1	ANSWER	Char	10	\$ANSWER.

## FORMAT AFTER INPUT ...

Alphabetic List of Variables and Attributes				
#	Variable	Type	Len	Format
1	ANSWER	Char	1	\$ANSWER.

ONLY A CONCERN WITH CHARACTER VARIABLES

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

	<b>Beyond FORMAT Basics</b> Mike Zdeb, <i>University at Albany School of Public Health</i>	HOW8
Files associated with this paper can be found in the \nesug05\code folder on this CD. Look for file HOW8.ZIP.		HOW9
		HOW10

	<b>The Power of PROC FORMAT</b>
	By: Jonas V. Bilenas
	List price: 29.95 USD
	124 pages
	ISBN: 978-1-59047-573-7**
	ISBN 10: 1-59047-573-9**
	Publisher: SAS Press
	Copyright Date: March 2005
	<b>Description:</b>
	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.
<b>More About This Book:</b> <a href="#">➤ Book Reviews</a> <a href="#">➤ Table of Contents (html format)</a> <a href="#">➤ Table of Contents (.pdf format, 35415 bytes)</a> <a href="#">➤ Sample Chapter (.pdf format, 157668 bytes)</a> <a href="#">➤ SAS Code Used</a>	
<b>About the Author:</b>	