

Chpt 5: Modifying and Combining SAS Data Sets

The Little SAS Book, 2nd ed., by Lora Delwiche & Susan Slaughter, SAS Publishing

5.1 Modifying a Data Set Using the SET Statement

Example of a SAS program:

```
libname annat 'h:\sas\sas teaching (MEP)\sas seminars';
```

```
data descr;
  set annat.descr_raw;
run;
```

```
proc print data=descr;
run;
```

Obs	patnr	age	sex	height	weight	smoke
1	1	43	2	172	75	0
2	2	30	2	160	55	0
3	3	27	1	180	94	1
4	4	70	2	165	70	0
5	5	25	1	168	61	0
6	6	34	2	192	90	1
7	7	50	1	171	73	0

5.1 Modifying a Data Set Using the SET Statement

```
data descr;
  set annat.descr_raw;
  if age<35 and sex=2;
run;
```

```
proc print data=descr;
run;
```

Obs	patnr	age	sex	height	weight	smoke
1	2	30	2	160	55	0
2	6	34	2	192	90	1

5.1 Modifying a Data Set Using the SET Statement

```
data descr;
  set annat.descr_raw;
  if age<35 and sex=2;
  height_m=height/100;
  bmi=weight/(height_m**2);
  bmi=round(bmi,.1);
run;
```

```
proc print data=descr;
  var patnr height weight bmi;
run;
```

Obs	patnr	height	weight	bmi
1	2	160	55	21.5
2	6	192	90	24.4

5.1 Modifying a Data Set Using the SET Statement

```
data descr_cl;
  set descr;
  if bmi <= 24.9 then bmiclass=1;
  else if (25.0 <= bmi <= 29.9) then bmiclass=2;
  else if (30.0 <= bmi <= 34.9) then bmiclass=3;
  else if bmi => 35.0 then bmiclass=4;
run;
```

```
proc print data=descr_cl;
  var patnr height weight bmi bmiclass;
run;
```

Obs	patnr	height	weight	bmi	bmiclass
1	2	160	55	21.5	1
2	6	192	90	24.4	1

5.2 Stacking Data Sets Using the SET Statement

```
proc print data=descr1;
  var patnr centre age sex height weight;
run;
```

Obs	patnr	centre	age	sex	height	weight
1	11	KS	43	2	172	75
2	12	KS	30	2	160	55
3	13	KS	27	1	180	94

```
proc print data=descr2;
  var patnr centre age sex height weight;
run;
```

Obs	patnr	centre	age	sex	height	weight
1	21	SOS	70	2	165	70
2	22	SOS	25	1	168	61
3	23	SOS	34	2	192	90
4	24	SOS	50	1	171	73

5.2 Stacking Data Sets Using the SET Statement

```
data all;
    set descr1 descr2;
run;

proc print data=all;
    var patnr centre age sex height weight;
run;
```

Obs	patnr	centre	age	sex	height	weight
1	1	KS	43	2	172	75
2	2	KS	30	2	180	55
3	3	KS	27	1	180	94
4	11	SOS	70	2	165	70
5	12	SOS	25	1	168	61
6	13	SOS	34	2	192	90
7	14	SOS	50	1	171	73

Anna Torr ng April 2003

7

5.2 Stacking Data Sets Using the SET Statement

If only one of the data sets (here descr2) includes the variable smoke:

Obs	patnr	centre	age	height	weight	smoke
1	1	KS	43	172	75	.
2	2	KS	30	160	55	.
3	3	KS	27	180	94	.
4	11	SOS	70	165	70	0
5	12	SOS	25	168	61	0
6	13	SOS	34	192	90	1
7	14	SOS	50	171	73	0

Anna Torr ng April 2003

8

5.5 Combining Data Sets Using the MERGE Statement

DESCR

Obs	patnr	age	sex	height	weight	smoke
1	1	43	2	172	75	0
2	2	30	2	160	55	0
3	3	27	1	180	94	1
4	4	70	2	165	70	0

BLOODPR

Obs	patnr	week	sbp
1	1	1	143
2	1	2	140
3	2	1	120
4	2	2	125
5	3	1	135
6	3	2	131
7	4	1	151
8	4	2	146

Anna Torr ng April 2003

9

5.5 Combining Data Sets Using the MERGE Statement

```
proc sort data=descr1;
    by patnr;
run;

proc sort data=bloodpr;
    by patnr;
run;

data all;
    merge descr1 bloodpr;
    by patnr;

proc print data=all;
    var patnr sex height week sbp;
run;
```

Anna Torr ng April 2003

10

5.5 Combining Data Sets Using the MERGE Statement

Cont...

Obs	patnr	sex	height	week	sbp
1	1	2	172	1	143
2	1	2	172	2	140
3	2	2	160	1	120
4	2	2	160	2	125
5	3	1	180	1	135
6	3	1	180	2	131
7	4	2	165	1	151
8	4	2	165	2	146

Anna Torr ng April 2003

11

5.5 Combining Data Sets Using the MERGE Statement

WARNING: Using the MERGE statement without by ...may be very dangerous:

ALL

Obs	patnr	age	sex	height	weight	smoke	week	sbp
1	1	43	2	172	75	0	1	143
2	1	30	2	160	55	0	2	140
3	2	27	1	180	94	1	1	120
4	2	70	2	165	70	0	2	125
5	3	1	135
6	3	2	131
7	4	1	151
8	4	2	146

So don't forget BY!

Anna Torr ng April 2003

12

5.9 Using SAS Data Set Options

The most frequently used data set options:

Options:	Tells SAS:
KEEP = <i>variable list</i>	variables to keep
DROP = <i>variable list</i>	variables to drop
RENAME = (<i>oldvar</i> = <i>newvar</i>)	rename variable
FIRSTOBS = <i>n</i>	start reading at obs <i>n</i>
OBS = <i>n</i>	stop reading at obs <i>n</i>
IN = <i>newvar-name</i>	temporary variable for tracking whether that data set contributed to the current observation.

Anna Torr ng April 2003

13

5.9 Using SAS Data Set Options

```
data in1;
  set annat.descr_raw (keep = patnr age sex smoke);

data in2;
  set annat.descr_raw (drop = height weight);

data in3;
  set annat.descr_raw (rename = (height = heightcm));

proc print data=in3 (firstobs = 2 obs = 4);
run;
```

Anna Torr ng April 2003

14

5.9 Using SAS Data Set Options

Tracking observations using the IN= option (IN creates a new temporary variable)

```
data all;
  merge descr1 (in=indesc) bloodpr (in=bp);
  by patnr;
```

These variables exist only for the duration of current data step and are not added to the data set being created.

SAS gives IN=variables a value of 0 if that data set did not contribute to the current observation and a value of 1 if it did.

Anna Torr ng April 2003

15

5.10 Tracking and selecting observations with the IN=option

Inclusion criteria: Age \geq 18

DESCR			BLOODPR		
Obs	patnr	age	Obs	patnr	sbp
1	12	43	1	12	150
2	13	30	2	13	128
3	14	27	3	14	122
4	15	17	4	15	120

```
INCL_PP
-----
data ppanalys;
  merge incl_pp(in=pp) bloodpr;
  by patnr;
  if pp=1;
run;
/* Only patients that fulfilled the
inclusion criteria are included*/
```

Anna Torr ng April 2003

16

5.10 Tracking and selecting observations with the IN=option

PPANALYS

Obs	patnr	age	sbp
1	12	43	150
2	13	30	128
3	14	27	122

Anna Torr ng April 2003

17

5.11 Writing Multiple Data Sets Using the OUTPUT Statement

RESULTS

```
Obs stud_nr test scores
1 100 1 56
2 100 2 40
3 100 3 89
4 101 1 60
5 101 2 55

data score1 score2 score3;
  set results;
  if test=1 then output score1;
  if test=2 then output score2;
  if test=3 then output score3;

run;
```

Obs	stud_nr	test	scores
1	100	1	56
2	101	1	60

Anna Torr ng April 2003

18

5.12 Making Several Observations from One Using the OUTPUT Statement

Usually SAS writes an observation to a data set at the end of the DATA step, but you can override this default using the OUTPUT statement. This statement gives you control over when an observation is written to a SAS data set.

```
data generate;
  do x=1 to 6;
    y=x**2;
    output;
  end;
run;
```

DO loop with 6 iterations.
Without the OUTPUT, SAS would have written only 1 obs. at the end of the data step.

Obs	x	y
1	1	1
2	2	4
3	3	9
4	4	16
5	5	25
6	6	36

Anna Torr ng April 2003

19

5.13 Changing Observations to Variables Using PROC TRANSPOSE

PROC TRANSPOSE transposes SAS data set (turning observations into variables or variables into observations).

```
proc transpose data = oldname out = newname;
  by variable-list;
  id variable;
  var variable-list;
run;
```

BY statement: for grouping variables you want to keep as variables

ID statement: names variable whose formatted values will become the new variable names

VAR statement: names the variables whose values you want to transpose

Anna Torr ng April 2003

20

5.13 Changing Observations to Variables Using PROC TRANSPOSE

BLOODPR

Obs	patnr	week	sbp
1	1	1	143
2	1	2	140
3	2	1	120
4	2	2	125
5	3	1	135
6	3	2	131
7	4	1	151
8	4	2	146

?

Anna Torr ng April 2003

21

5.13 Changing Observations to Variables Using PROC TRANSPOSE

BLOODPR

Obs	patnr	week	sbp
1	1	1	143
2	1	2	140
3	2	1	120
4	2	2	125
5	3	1	135
6	3	2	131
7	4	1	151
8	4	2	146

WISH

patnr	sbp_w1	sbp_w2
1	143	140
2	120	125
3	135	131
4	151	146

Anna Torr ng April 2003

22

5.13 Changing Observations to Variables Using PROC TRANSPOSE

```
proc transpose data=bloodpr out=bp;
  by patnr;
  id week;
  var sbp;
run;
```

BP

Obs	patnr	_NAME_	_1	_2
1	1	sbp	143	140
2	2	sbp	120	125
3	3	sbp	135	131
4	4	sbp	151	146

Anna Torr ng April 2003

23

5.14 Using SAS Automatic Variables

N: indicates the number of times SAS has looped through the DATA step

ERROR: value of 1 if a data error for that observation and 0 otherwise

FIRST.variable and LAST.variable: Available only in special circumstances i.e when you are using the BY statement in a data step.

Anna Torr ng April 2003

24

5.14 Using SAS Automatic Variables

Example: FIRST.variable

CANCER

```
-----  
Obs  lopnr  diadat  icd7  
1    4567   19870506  170  
2    6853   19640115  190  
3    6853   19681108  170
```

```
proc sort data=cancer;  
  by lopnr diadat;
```

```
/* Include only the first diagnosis of cancer*/
```

```
data firstcan;  
  set cancer;  
  by lopnr diadat;  
  if first.lopnr;
```

```
run;
```

Anna Torr ng April 2003

25

5.14 Using SAS Automatic Variables

FIRSTCAN

```
-----  
1    4567   19870506  170  
2    6853   19640115  190
```

Anna Torr ng April 2003

26