Chapter 4 Combining SAS Data Sets

This chapter introduces combining datasets vertically (adding more cases) and horizontally (adding more variables).

4.1 Stack Datasets Vertically (add more cases)

a. Use SET statement in a DATA step to combine the datasets vertically

A few things to note beforehand.

- It is not necessary for two datasets to have same variables or variables in the same order.
- It is critical that if the same variable appears in both datasets, it should be of the same type (character or numeric).
- If a variable is present in only one dataset, after merging, the values for that variable will be missing for all cases in the other dataset.
- The order of the variables in the first dataset will be used in the merged dataset.

Now let's see an example.

First, we input the data responses directly using DATALINES.

```
*Input datalines for Boy and Girls*;
data boys;
 input name$ sex$ age height weight;
 datalines;
 Jeffrey M 13 62.5 84
 Alfred  M 14 69 112.5
 Ronald  M 15 67 133
 Philip  M 16 72 150
 ;
run;

data girls;
 input name$ age sex$ height;
 datalines;
 Alice   13 F 56.5
 Barbara 13 F 65.3
 Carol   14 F 62.8
 Judy    14 F 64.3
 ;
run;
```

Second, we use SET in a DATA step to merge these two datasets, and then print it.

```
*Merge Boys and Girls using SET*;
data allkids;
  set boys girls;
run;

proc print data=allkids;
run;
```

Results:

Obs	name	sex	age	height	weight
1	Jeffrey	M	13	62.5	84.0
2	Alfred	M	14	69.0	112.5
3	Ronald	M	15	67.0	133.0
4	Philip	M	16	72.0	150.0
5	Alice	F	13	56.5	.
6	Barbara	F	13	65.3	.
7	Carol	F	14	62.8	.
8	Judy	F	14	64.3	.

Note: With the use of SET statement, SAS must process all data responses to create a new one.

b. Use PROC APPEND to merge cases.

PROC APPEND adds observations from one dataset to the end of another dataset. Therefore, it does not process the original dataset (simply adds new observations), which is more efficient. Using PROC APPEND, we need to specify the role of each dataset.

- BASE = original dataset (it is usually larger)
- DATA = data set to be added to the original dataset

Note:

- Again, it is critical that if the same variable appears in both datasets, it should be of the same type (character or numeric).
- If DATA= dataset is not specified, SAS uses the most recently created dataset in SAS.

Example Code:

```
*PROC APPEND to merge cases*;
proc append base=boys data=girls;
run;

proc print data=boys;
run;
```

No new dataset is created using PROC APPEND. The BASE dataset is changed. The output table is the same as above.

4.2 Merge Datasets Horizontally (add more variables)

For example, we may store different variables for the same subjects in different datasets during data collection. Later on, we need to combine them in order to conduct data analysis.

We will introduce the most common approach – use MERGE option in a DATA step.

- Must first sort the datasets that are being merged by the key variable(s), and then merge by the same key variable(s).

a. By default, SAS combine the observations one by one in the order that they appear in the datasets.

- If a subject appears only in one dataset, this subject will have missing values of all the variables in the other dataset.

Step 1: Input data sets

```
*Horizontal merging: Input data responses*;
data height;
  input id sex$ age height;
  datalines;
  1 F 25 72
  2 F 33 68
  3 F 47 65
  4 F 29 69
  5 M 37 62
  6 M 42 64
  ;


data weight;
  input id weight;
  datalines;
  1 156
  4 190
  3 182
  6 156
  9 129
  ;
run;
```

Step 2: Identify key variable and sort the data by the key variable.
In this example, the key variable to link both datasets is ID. We use PROC SORT to sort.

```
*Sort both datasets by ID*;
proc sort data=height;
  by ID;
run;

proc sort data=weight;
  by ID;
run;
```

Step 3: Use MERGE in a DATA step to combine two datasets horizontally and print it out.

- MERGE: specify names of datasets to be merged.
- BY: identify key variable(s) for linking two datasets.

```
*Horizontally merge two datasets*;
data fulldata;
  merge height weight;
  by id;
run;

proc print data=fulldata;
run;
```

Results:

Obs	id	sex	age	height	weight
1	1	F	25	72	156
2	2	F	33	68	.
3	3	F	47	65	182
4	4	F	29	69	190
5	5	M	37	62	.
6	6	M	42	64	156
7	9		.	.	129

b. If we want to only merge the cases/subjects that appear in both datasets (i.e., 1, 3, 4, and 6), we use an IF statement.

```
*Horizontally merge two datasets -- selected cases*;
data completedata;
  merge height(in=a) weight(in=b);
  by id;
  if a and b;
run;

proc print data=completedata;
run;
```

Results:

Obs	id	sex	age	height	weight
1	1	F	25	72	156
2	3	F	47	65	182
3	4	F	29	69	190
4	6	M	42	64	156

c. We can also use the IF statement to include the cases that are in one of the datasets, e.g., Height (i.e., ID 1-6).

```
*Horizontally merge two datasets -- cases in dataset Height*;
data partialheight;
  merge height(in=a) weight(in=b);
  by id;
  if a;
run;

proc print data=partialheight;
run;
```

Results:

Obs	id	sex	age	height	weight
1	1	F	25	72	156
2	2	F	33	68	.
3	3	F	47	65	182
4	4	F	29	69	190
5	5	M	37	62	.
6	6	M	42	64	156

Note: "If b" will generate a dataset only for cases 1, 3, 4, 6, and 9.

4.3 Use MERGE to combine more complex datasets

a. Example I – two datasets have common variable names but carry different information.

Step 1: input data

```
*Example I -- Merge in both directions (add cases and variables)*;
data oldsalary;
  input name$ ID sex$ age salary jobcat year;
  datalines;
  Roger  518 M 45 7677 2 1989
  Martha 321 F 28 5000 1 1989
  Zeke   444 M 33 6075 1 1989
  Barb  1728 F 40 9023 2 1989
  Bill   993 M 36 7739 3 1989
  Sandy 1002 F 29 6161 3 1989
  ;

data newsalary;
  input name$ ID salary jobcat year;
  datalines;
  Hank   108 11138 1 1995
  Fred   519 10035 2 1995
  Zeke   444  9697 1 1995
  Martha 321  7987 2 1995
  Sandy 1002  6995 2 1995
  Bill   993 12400 3 1995
  Roxy   773 10119 2 1995
  ;
run;
```

Step 2: sort the datasets by ID (key variable)

```
proc sort data=oldsalary;
  by ID;
run;

proc sort data=newsalary;
  by ID;
run;
```

Step 3: merge two datasets using RENAME option and print it out.

```
*Merge with RENAME option*;
data combine1;
  merge oldsalary (rename=(salary=salary89 jobcat=jobcat89))
        newsalary (rename=(salary=salary95 jobcat=jobcat95));
  by ID;
  drop year;
run;

proc print data=combine1;
run;
```

Note: DROP statement removes unwanted variables from the combined dataset.

Results:

Obs	name	ID	sex	age	salary89	jobcat89	salary95	jobcat95
1	Hank	108		.	.	.	11138	1
2	Martha	321	F	28	5000	1	7987	2
3	Zeke	444	M	33	6075	1	9697	1
4	Roger	518	M	45	7677	2	.	.
5	Fred	519		.	.	.	10035	2
6	Roxy	773		.	.	.	10119	2
7	Bill	993	M	36	7739	3	12400	3
8	Sandy	1002	F	29	6161	3	6995	2
9	Barb	1728	F	40	9023	2	.	.

b. Example II – one dataset contains multiple lines for one subject, and the other dataset includes one line per subject.

Step 1: input data

```
*Example II -- one-to-many merging*;
data time_varying;
  input ID date:mmddyy10. weight height;
  format date mmddyy10.;
  datalines;
  4 01/09/2007 117 82
  2 03/15/2007 111 74
  2 04/25/2007 108 65
  1 05/17/2007 145 94
  1 11/22/2007 130 90
  1 01/12/2008 120 80
  3 01/22/2008 128 83
  ;

data one_per_line;
  input ID sex$ DOB:mmddyy10.;
  format DOB mmddyy10.;
  datalines;
  3 M 12/01/1980
  1 F 01/10/1978
  2 F 05/15/1976
  4 M 04/11/1981
  5 F 07/17/1980
  ;
run;
```

Step 2: sort the datasets by ID (key variable)

```
*Sort by ID*;
proc sort data=time_varying;
  by ID;
run;

proc sort data=one_per_line;
  by ID;
run;
```

Step 3: merge two datasets using RENAME option and print it out.

```
*Merge two datasets using default options*;
data combine2;
  merge time_varying one_per_line;
  by ID;
run;

proc print data=combine2;
run;
```

Results:

Obs	ID	date	weight	height	sex	DOB
1	1	05/17/2007	145	94	F	01/10/1978
2	1	11/22/2007	130	90	F	01/10/1978
3	1	01/12/2008	120	80	F	01/10/1978
4	2	03/15/2007	111	74	F	05/15/1976
5	2	04/25/2007	108	65	F	05/15/1976
6	3	01/22/2008	128	83	M	12/01/1980
7	4	01/09/2007	117	82	M	04/11/1981
8	5	.	.	.	F	07/17/1980