Print

Summary of Lesson 10: Combining SAS Data Sets

This summary contains topic summaries, syntax, and sample programs.

Topic Summaries

To go to the movie where you learned a task or concept, select a link.

Concatenating Data Sets

You can concatenate two or more data sets by combining them vertically to create a new data set. It is important to know the structure and contents of the input data sets.

You use a DATA step to concatenate multiple data sets into a single, new data set. In the SET statement, you can specify any number of input data sets to concatenate. During compilation, SAS uses the descriptor portion of the first data set to create variables in the PDV, and then continues with each subsequent data set, creating additional variables in the PDV as needed. During execution, SAS processes the data sets in the order in which they are listed in the SET statement.

```
DATA SAS-data-set;
     SET SAS-data-set1 SAS-data-set2 ...;
RUN;
```

If the data sets have differently named variables, every variable is created in the new data set, and some observations have missing values for the differently named variables. You can use the RENAME= data set option to change variable names in one or more data sets. After they are renamed, they are treated as the same variable during compilation and execution, and in the new data set.

```
SAS-data-set (RENAME=(old-name-1=new-name-1;
                       old-name-2=new-name-2
                       ...
                        old-name-n=new-name-n))
```

Merging SAS Data Sets One-to-One

Merging combines observations from two or more SAS data sets into a single observation in a new data set. A simple merge combines observations based on their positions in the original data sets. A match-merge combines them based on the values of one or more common variables. The result of a match-merge is dependant on the relationship between observations in the input data sets.

You use a DATA step with a MERGE statement to merge multiple data sets into a single data set. The BY statement indicates a match-merge and specifies the common variable or variables to match. The common variables are referred to as BY variables. The BY variables must exist in every data set, and each data set must be sorted by the value of the BY variables.

```
DATA SAS-data-set;
     MERGE SAS-data-set1 SAS-data-set2 ...;
     BY <DESCENDING> BY-variable(s);
     <additional SAS statements>
RUN;
```

Merging SAS Data Sets One-to-Many

In a one-to-many merge, a single observation in one data set matches more than one observation in another data set. The DATA step is the same, regardless of the relationship between the data sets being merged. SAS processes each BY group before reinitializing the PDV.

Merging SAS Data Sets That Have Non-Matches

When you merge data sets, observations in one data set might not have a matching observation in another data set. These are called non-matches. By default, both matches and non-matches are included in a merged data set. The observations that are matches contain data from every input data set. The non-matching observations do not contain data from every input data set.

You can use the IN= data set option in a MERGE statement to create a temporary variable that indicates whether a data set contributed information to the observation in the PDV. The IN= variables have two possible values: *0* and *1*. You can test the value of this variable using subsetting IF statements to output only the matches or only the non-matches to the merged data set.

MERGE *SAS-data-set1* <**(IN=*variable*)**>...

Sample Programs

Concatenating Data Sets with Different Variables

```
********** Create Data **********;
data empscn;
   input First $ Gender $ Country $;
   datalines;
Chang    M    China
Li       M    China
Ming     F    China
;
run;

data empsjp;
   input First $ Gender $ Region $;
   datalines;
Cho      F    Japan
Tomi     M    Japan
;
run;

********** Unlike-Structured Data Sets **********;
data empsall2;
   set empscn empsjp;
run;

proc print data=empsall2;
run;
```

Merging Data Sets One-to-One

```
********** Create Data **********;
data empsau;
   input First $ Gender $ EmpID;
   datalines;
Togar    M    121150
Kylie    F    121151
Birin    M    121152
;
run;

data phoneh;
   input EmpID Phone $15.;
   datalines;
121150 +61(2)5555-1793
121151 +61(2)5555-1849
121152 +61(2)5555-1665
;
run;

********** Match-Merge One-to-One**********;
data empsauh;
   merge empsau phoneh;
   by EmpID;
run;
```

```
proc print data=empsauh;
run;
```

Match-Merging Data Sets with Non-Matches

```
********** Create Data **********;
data empsau;
   input First $ Gender $ EmpID;
   datalines;
Togar   M   121150
Kylie   F   121151
Birin   M   121152
;
run;

data phonec;
   input EmpID Phone $15.;
   datalines;
121150 +61(2)5555-1795
121152 +61(2)5555-1667
121153 +61(2)5555-1348
;
run;

********** Match-Merge with Non-Matches**********;
data empsauc;
   merge empsau phonec;
   by EmpID;
run;

proc print data=empsauc;
run;
```

Selecting Non-Matches

```
********** Create Data **********;
data empsau;
   input First $ Gender $ EmpID;
   datalines;
Togar   M   121150
Kylie   F   121151
Birin   M   121152
;
run;

data phonec;
   input EmpID Phone $15.;
   datalines;
121150 +61(2)5555-1795
121152 +61(2)5555-1667
121153 +61(2)5555-1348
;
run;

********** Non-Matches from empsau Only **********;
data empsauc2;
   merge empsau(in=Emps)
         phonec(in=Cell);
   by EmpID;
   if Emps=1 and Cell=0;
run;

proc print data=empsauc2;
run;
```

SAS Programming 1: Essentials

Close