“What are the steps for merging datasets in SAS?

Authored by

stats writer

June 24, 2024

The process of merging datasets in SAS involves combining two or more datasets into one, allowing for more comprehensive analysis and insights. This can be done through the following steps:
1. Identify the key variables that will be used to merge the datasets, ensuring they are of the same data type and have similar values.
2. Sort the datasets by the key variables to be used for merging.
3. Use the MERGE statement to combine the datasets, specifying the key variables to be used and the order in which the datasets should be merged.
4. Use the BY statement to indicate the common key variables between the datasets.
5. Check for any duplicate observations and remove them if necessary.
6. Perform any additional data cleaning or transformations as needed.
7. Save the merged dataset as a new dataset for future analysis.
By following these steps, users can effectively merge datasets in SAS and utilize the combined data for more accurate and thorough analysis.

Combining Datasets by Merging

In many practical situations, you may have relevant data in two different datasets, and in order to perform your analysis, you'll need to combine those datasets.

In general, combining datasets takes one of two forms:

Appending: Placing the second dataset below the first dataset (also called *stacking*)Match-merging: Joining the datasets in such a way that one or more cases from one dataset can be matched to one or more cases in a second dataset, based on a uniquely identifying ID variable in both datasets

Appending is useful when you have two or more datasets with similar or identical structures, but different cases. This can happen if you have datasets covering different time periods, and want to analyze trends over time: in order to do so, you'll need to put all of the time periods into a single dataset for analysis.

Merging is useful when you have relevant information stored in separate data sources. For example, you may have demographic information about customers in one dataset, and transaction information in a second dataset; both datasets will have a "customer ID" variable that uniquely identifies who made the purchase, but the variables in each dataset will be different.

Stacking or Appending Datasets

Suppose you have two or more datasets with the same structure (i.e., completely identical variables) but different cases (i.e., the rows in each dataset are unrelated to one another). This

may happen if you have to researchers collecting observations at different locations or times. You may want to combine these records into a single dataset by "appending" one dataset to the bottom of the other.

When you have two or more datasets with the same structure, then you can combine them using the `SET` statement within a data step:

```
DATA New-Dataset-Name (OPTIONS);
SET Dataset-Name-1 (OPTIONS) Dataset-Name-2 (OPTIONS);
RUN;
```

The code above is just an extension of the basic `SET` statement, but instead of having one dataset listed after the `SET` keyword, there are two or more datasets listed. The dataset names in the list are separated by a space.

Although this code is simple, there are a few things to keep in mind when combining datasets this way.

If the datasets have different variable names, the new dataset will include all variable names and assign missing values where appropriate.If the datasets contain the same variable names, but the formats, labels, and/or lengths are different for any given variable, the new dataset will use the definitions from the dataset listed first in the SET statement.If the datasets contain the same variables, but the variable types are different (i.e., a variable is numeric in one dataset but character in another dataset) then SAS will not execute the statements and no dataset will be created.

After you've created your new dataset with the SET statement, check your log to make sure the number of observations in your new dataset is the sum of the number of observations in the separate datasets.

Subject_ID	DOB	Gender
1	9/20/1980	Female
2	6/12/1954	Male
3	4/2/2001	Male
4	8/29/1978	Female
5	2/28/1986	Female

Subject_ID	DOB	Gender
6	3/9/1960	Male
7	5/21/1985	Female
8	4/13/1941	Male
9	8/11/2000	Male

A combined version of these datasets would simply stack one dataset on top of the other:

Subject_ID	DOB	Gender
1	9/20/1980	Female
2	6/12/1954	Male
3	4/2/2001	Male
4	8/29/1978	Female
5	2/28/1986	Female
6	3/9/1960	Male
7	5/21/1985	Female
8	4/13/1941	Male
9	8/11/2000	Male

Match-Merging Datasets

When you have two or more datasets that contain different information on the same subjects, you might want to combine them into one large dataset that has all the information on your subjects together. For example, you may initially store information about subjects' demographic information in a separate datafile than their survey responses.

To do this you use a MERGE statement and a BY statement within a data step, like this:

```
DATA New-Dataset-Name (OPTIONS);
MERGE Dataset-Name-1 (OPTIONS) Dataset-Name-2 (OPTIONS);
BY Variable(s);
RUN;
```

You must sort both datasets on your matching variable(s) before merging them!

In the code above, the datasets that you want to merge together are listed after the `MERGE`

keyword, each separated by a space. The `BY` statement contains the variable(s) that identifies the observation in the first dataset that represents the same subject as the observation in the second dataset. You must sort each of the datasets listed in the MERGE statement by the variable(s) listed in the BY statement before merging them together.

This same method will combine datasets regardless if you have a one-to-one match (i.e., each subject has only one record in all the datasets) or a one-to-many or many-to-many match.

Check the formats, informats, labels, and lengths of the newly created dataset to make sure the variables have the properties you want them to have.

One-to-one match

One-to-one matching assumes that each subject appears exactly once in each of the datasets being merged.

Subject_ID	DOB	Gender
1	9/20/1980	Female
2	6/12/1954	Male
3	4/2/2001	Male
4	8/29/1978	Female
5	2/28/1986	Female

plus

Subject_ID	Visit_Date	Doctor
1	1/31/2012	Walker
2	2/2/2012	Jones
3	1/15/2012	Jones
5	1/29/2012	Smith

Merging dataset A with dataset B yields

Subject_ID	DOB	Gender	Visit_Date	Doctor
1	9/20/1980	Female	1/31/2012	Walker
2	6/12/1954	Male	2/2/2012	Jones

Subject_ID	DOB	Gender	Visit_Date	Doctor
3	4/2/2001	Male	1/15/2012	Jones
4	8/29/1978	Female		
5	2/28/1986	Female	1/29/2012	Smith

```
DATA patients;
INPUT Subject_ID DOB Gender $;
INFORMAT DOB MMDDYY10.;
FORMAT DOB MMDDYY10.;
DATALINES;
1 9/20/1980 Female
2 6/12/1954 Male
3 4/2/2001 Male
4 8/29/1978 Female
5 2/28/1986 Female
;
RUN;
```

```
DATA initial_appointments;
INPUT Subject_ID Visit_Date Doctor $;
INFORMAT Visit_Date MMDDYY10.;
FORMAT Visit_Date MMDDYY10.;
DATALINES;
1 1/31/2012 Walker
2 2/2/2012 Jones
3 1/15/2012 Jones
5 1/29/2012 Smith
;

PROC SORT DATA=patients;
BY Subject_ID;
RUN;

PROC SORT DATA=initial_appointments;
BY Subject_ID;
RUN;

DATA one_to_one_match;
MERGE patients initial_appointments;
BY Subject_ID;
```

RUN;

One-to-many match

One-to-many matching assumes that each subject appears exactly once in one dataset, but can have multiple matching records in another dataset. Thus, when the datasets are merged, information from one dataset may be repeated on multiple rows.

In the below example:

Dataset A represents patient demographic information. There is exactly one row per patient (uniquely identified by the variable Subject_ID).Dataset B represents appointment information. There may be one or more rows corresponding to a particular patient (who is again identified using the variable Subject_ID).

The dataset we want to create by match-merging should still have one row per appointment, but should have the patient demographic information added to the table. Therefore, we will match the tables on the Subject_ID variable. However, when writing the `MERGE` statement for a one-to-many match, the table containing the "ones" (in this case, the patients table) must be listed first, and the table containing the "many" (in this case, the appointments table) must be listed second.

Subject_ID	DOB	Gender
1	9/20/1980	Female
2	6/12/1954	Male
3	4/2/2001	Male
4	8/29/1978	Female
5	2/28/1986	Female

plus

Subject_ID	Visit_Date	Doctor
1	1/31/2012	Walker
1	5/29/2012	Walker
2	2/2/2012	Jones
3	1/15/2012	Jones
5	1/29/2012	Smith

Subject_ID	Visit_Date	Doctor
5	2/6/2012	Smith

Merging dataset A with dataset B yields

Subject_ID	DOB	Gender	Visit_Date	Doctor
1	9/20/1980	Female	1/31/2012	Walker
1	9/20/1980	Female	5/29/2012	Walker
2	6/12/1954	Male	2/2/2012	Jones
3	4/2/2001	Male	1/15/2012	Jones
4	8/29/1978	Female		
5	2/28/1986	Female	1/29/2012	Smith
5	2/28/1986	Female	2/6/2012	Smith

```
DATA patients;
INPUT Subject_ID DOB Gender $;
INFORMAT DOB MMDDYY10.;
FORMAT DOB MMDDYY10.;
DATALINES;
1 9/20/1980 Female
2 6/12/1954 Male
3 4/2/2001 Male
4 8/29/1978 Female
5 2/28/1986 Female
;
RUN;
```

DATA appointment_log;
INPUT Subject_ID Visit_Date Doctor $;
INFORMAT Visit_Date MMDDYY10.;
FORMAT Visit_Date MMDDYY10.;
DATALINES;
1 1/31/2012 Walker
1 5/29/2012 Walker
2 2/2/2012 Jones
3 1/15/2012 Jones
5 1/29/2012 Smith
5 2/6/2012 Smith

```
;
RUN;

PROC SORT DATA=patients;
BY Subject_ID;
RUN;

PROC SORT DATA=appointment_log;
BY Subject_ID;
RUN;

DATA one_to_many_match;
MERGE patients appointment_log;
BY Subject_ID;
RUN;
```

For More Information

While we have given simple examples here, merging data can present highly complex problems, especially for datasets with many observations or variables. We recommend the following books for more information.

Combining and Modifying SAS Data Sets by Michelle M. BurlewISBN: 9781590479209Publication Date: 2009-11-01