How can I view data in SAS Tutorials?

Authored by

stats writer

June 24, 2024

SAS Tutorials offer a comprehensive platform for learning and mastering the SAS software. To access data in SAS Tutorials, users can follow these steps:

1. Open the SAS software and navigate to the "Explorer" tab.
2. Under the "Explorer" tab, click on the "Libraries" option.
3. A new window will open, displaying all the available libraries.
4. Select the library that contains the desired data.
5. Once the library is selected, the data files will be displayed in the "Explorer" tab.
6. Users can then click on the desired data file to view its contents.

By following these steps, users can easily access and view data in SAS Tutorials, allowing for a more efficient and effective learning experience.

Viewing a Dataset Using the Viewtable Window

A SAS dataset can be viewed as a spreadsheet using the Viewtable window. To open a dataset in the Viewtable window:

In the Explorer window, double-click on the **Libraries** icon to display the libraries that are available in the current session. If the dataset was created without being stored to a user-created library, it will be located in the "Work" library. Double-click on the icon of the library where the dataset is located.Locate the dataset icon and double-click on it. This will open your dataset as a spreadsheet.

A typical Viewtable view of a dataset looks like this:

	ids	bday	Gender	Rank	State
1	23643	22NOV1990	0	.	
2	30953	23AUG1995	1	1	In state
3	20531	29DEC1994	0	1	In state
4	22416	.	0	1	In state
5	41227	19APR1994	1	2	In state
6	37301	06JUN1993	1	2	Out of state
7	39181	17MAY1992	0	3	In state
8	22652	04DEC1989	1	3	In state
9	35684	29JUN1991	0	4	In state
10	43344	26MAR1993	0	.	In state

Note that SAS is unable to execute any DATA or PROC steps on a dataset that is open in the Viewtable window. You will need to close your open Viewtable windows before you begin running your syntax.

Printing a Dataset to the Output Window with PROC PRINT

Alternatively, a SAS dataset can be viewed using the the PRINT procedure, which produces a print-out of your dataset in the Output window. The PRINT procedure offers the flexibility to specify what variables and/or observations to print. The general format of PROC PRINT is:

```
PROC PRINT DATA=dataset <options>;
BY variable(s);
ID variable;
VAR variable(s);
WHERE condition(s);
FORMAT formats;
RUN;
```

In the first line of the SAS code above, `PROC PRINT` tells SAS to execute the print procedure on the dataset specified by the `DATA=` argument. Immediately following PROC PRINT is where you put any procedure-level options you want to include. Let's review some of the more common options:

`LABEL`
Print variable labels instead of variable names if variable labels have been assigned. (Does not apply to value labels.)`NOOBS`
Do not print row numbers/row identifiers. (NOOBS is an abbreviation for "no observation numbers".)

As with all SAS procedures, the `DATA` command is optional, but recommended. If you do not specify a dataset, SAS will use the most recently created dataset by default. Note that if you want to begin printing at a specific record number or print a range of records (such as the first 20 records), you can supply the `FIRSTOBS` and `OBS` options immediately after the name of the dataset passed to the `DATA=` argument, enclosed by parentheses. For example, the code

```
PROC PRINT DATA=sample (FIRSTOBS=20 OBS=30);
RUN;
```

will print all of the observations in the sample dataset from row 20 through row 30.

The other statements (BY, ID, VAR, WHERE, and FORMAT) are **optional**, but are useful when you want to change what content is printed:

The `BY` statement is an optional statement that can be used to print the data in groups. Note that this action requires that the data is sorted with respect to the variable(s) included in the `BY` statement. This can be achieved by running PROC SORT before running PROC PRINT.The `ID`

statement allows you to specify a variable containing unique identifier labels for each observation. The variable you specify in the ID statement will print as the observation identifier in place of a row number.The `VAR` statement allows you to specify which variables to print, and what order to print them in. If this statement is omitted, SAS will print all variables in the dataset.The `WHERE` statement allows you to limit which rows are printed, based on a logical condition involving variables and values in your dataset. (If you want to print only the first *n* observations, or if you want to begin printing with the *n*th observation, use the FIRSTOBS and OBS options mentioned above.) We cover logical conditions in our Subsetting & Splitting Datasets tutorial.The `FORMAT` statement allows you to apply (or override) variable formats for printing. This can be useful, for example, if your numeric variables normally have 2-3 decimal places, but you to suppress decimal places in the PROC PRINT output. We cover variable formats and the FORMAT statement in our Informats and Formats tutorial.

There are other optional statements available in PROC PRINT; see the SAS Help and Documentation guide for their descriptions.

Example: printing observations by groups

Example. Using the sample dataset, let's print the height and weight of each student (rounded to the nearest whole number), grouping them by gender, and use the students' IDs in place of the observation numbers.

```
PROC SORT DATA=sample;
BY Gender;
RUN;
```

```
PROC PRINT DATA=sample LABEL;
BY Gender;
ID ids;
VAR Gender Height Weight;
FORMAT Height Weight 3.0;
RUN;
```

Because we want to print observations by gender, we must first sort the data using `PROC SORT`. The BY statement specifies that we want to group the printed output by the levels of variable Gender. The ID statement specifies that variable StudentID should be printed instead of the observation number. Because we are only interested in the height and weight of each student, these two variables are specified in the `VAR` statement. (Note, however, that the variable given in the ID statement will automatically print, regardless of whether or not it is listed in the VAR statement.) Finally, a `FORMAT` statement specifies that height and weight should print with no

decimal point. (Specifically, it says that the values should be no wider than three characters, and should have no decimal places.) This is what the beginning of the first page of output should look like:

The SAS System

Gender=.

ids	Gender	Height	Weight
23017	.	72	225
33153	.	65	200
20248	.	61	154
32415	.	68	195
42217	.	70	187
25714	.	61	237
30418	.	65	227
43678	.	67	187
49165	.	66	186

Gender=0

ids	Gender	Height	Weight
43783	0	72	.
20278	0	71	179
20389	0	71	199
28980	0	68	172

The variables ids, Gender, Height, and Weight are displayed with respect to the categories of variable Gender. By default, the groups are listed in ascending order. (Recall that Male was coded as 0, and Female was coded as 1. SAS treats missing values as the smallest possible value, so students with missing values for Gender appear first, then Male students appear next, then Female students.)