

I'm not robot  reCAPTCHA

Continue

Chapters Content Previous Next Special SAS datasets provide the tools to create zero datasets and for naming and using datasets by default. If you want to step UP BUT don't want to create an SAS dataset, you can use the keyword NULL as the name of the dataset. The following statement begins with a STEP of DATA that does not create a data set: null data; Using NULL forces SAS to take the DATA step as if it were creating a new data set, but no observations or variables are recorded in the output set. This process can be more efficient using computer resources if you use the DATA step for certain functions, such as writing a report for which the DATA step output should not be stored as an SAS dataset. SAS tracks the latest SAS dataset through the reserved name LAST. When you take a DATA or PROC step without specifying a set of inputs, SAS uses the LAST dataset by default. Some features also use the default LAST. The LAST option allows you to assign a dataset to a data set as a LAST dataset. The name you're pointing at is used as the default dataset until a new data set is created. You can use the LAST option when you want to use your existing permanent sas set of data, which contains a number of phases of the procedure. Issuing the LAST option avoids specifying the name of the SAS dataset in each procedural statement. The following OPTIONS statement defines the SAS dataset by default: last schedule; January options; Automatic Convention Name If you don't specify the name of the SAS dataset or the reserved name NULL in the DATA statement, SAS automatically creates datasets with the names DATA1, DATA2, and so on to sequential datasets in the WORK or USER library. This feature is called the DATAn Convention. The following statement provides a set of SAS data using the DATAn convention: data; Chapter Content Previous Next Top Pages Copyright 1999 sas institute inc., Cary, North Carolina, USA. All rights are reserved. Lesson 21: Data Null's It Was the Author's Experience, The Previous Contents That in many job interviews where SAS programming is an important part of job description, there are questions about data zero. It seems that employers consider this a kind of litmus test of the candidate's level of ability. It was therefore important to pay some attention to that issue. The idea of zero here is that we have a data step that doesn't really create a data set. For the name of the data set, we use a special name y null where the emphasis is part of the name. This leads to the SAS executing commands at the data stage, but as far as output is concerned, it's, well, zero or nothing. Why step data Doesn't it save? In fact, it doesn't save the dataset, but it can save something else, particularly a text file. In this way, you can use the data step to write reports, or creating raw data Contents| Next> </Previous| Contents| Next>. This process is simply a reverse reading of a raw data file. Instead of infiling the application, there will be a file statement. Instead of entering will put. Instead of information, formats. First, let's see what puts the statement doing in the usual data step. It sends lines of text to the log. The statement may have expressions of symbols in quotes and variable names. The variables will be printed in current values. The next step of the data has three iterations, so three lines are printed in a log. This is very useful for debugging data steps with loops and conditional statements, as you can examine the values of variables taken when performing a data step. Then add an extract from the file. This gives the location and name of the file that you want to keep the results put statements in. Notice that the log shows where the file is, as well as the number of entries written. It also shows that the data step still writes the data set. The same program, with only the name of the data set changed to zero, gives the log below. There is no NOTE about the data set because none was created. Now let's go back to used car data from previous lessons. Suppose we start by reading the data into the SAS dataset. We then use zero data to record some data in another file. Note that the data step will iterate once for each observation in the source data set. Variables, as they are listed in the statement put, are sent to the file only with the space in between. This is a list entry style similar to the type of list entry style. We can use a formatted style of put to control the appearance of the output. But we can do a lot more than that! Here's an example that uses the internal N variable (emphasizes are part of the title) that tracks observations to control when to print a title. So, if we are on the first observation, the first is if the application puts two lines of the header in the file. Second, if the condition is not true, so it is not fulfilled, then the final statement of the put sends the first observation to the file. For the rest of the observations, the first if the condition is false, so the headlines are never printed again. The second is if the condition has two parts. The number of observations should be more than 1, and the make should be different from the previous observation. The lag function allows you to compare values between observations, with lag1 being a previous observation, lag2 is the second previous observation, etc. So, after the first observation, if there is a difference in making, an empty line will be inserted before the last put statement sends details. Another example: : Exercises: Copyright reserved by Dr. Dwight Galster, 2006. Please request permission to reprint (except for personal use) from . SAS is the registered trade name of the SAS Institute, Cary, North Carolina. All the other trade names mentioned in their It./Previous Next> </Previous| Next> Owners. This document is in the process of being developed and comments are welcome. Please send an email if you find it useful or if your site links to it. In SAS, the reserved keyword NULL defines a SAS dataset that has no observations or variables. When you specify NULL as the name of the output set, no output is written. The NULL dataset is often used when you want to execute a DATA step code that displays the result, identifies the macro variable, writes a text file, or makes calls to the EXECUTION routine. In these cases, you're interested in the side effect of the DATA step and rarely want to record a dataset on a disk. Here are six ways to use the NULL dataset. Because the keyword NULL is used, the dataset on the disk is not created. #1. Use SAS as a giant calculator you can calculate the number in a data step and then use the PUT statement to output a response to the SAS log. For example, THE next step is to rate the normal density function at x-0.5 when μ 1 and σ -2. The calculation is done twice: first with the built-in PDF function and again with the help of the formula for normal density function. The SAS magazine states that in both cases the response is 0.193. Data (NULL); Moo No 1; Sigma No. 2; x 0.5; pdf and pdf (Normal, x, mu, sigma); y y exp $-(x-\mu)^2 / (2\sigma^2)$ / sqrt (Z'constant('pi'-sigma-2); put (pdf y) (5.3 euros); Run #2. Characteristics of the dataset display can be used by zero-step DATA to display the characteristics of the dataset. For example, THE next STEP of DATA uses the PUT statement to display the number of numbers and variables in the Sashelp.Class dataset. No data set is created. Data (NULL); Set Sashelp.Class; array char ' \$ (CHAR); Numeric array nCharVar - dim (enchantment); nNumerVar - dim (num); put Sashelp.Class; nCharVar nNumerVar; Stop it. / - Stop processing after the first observation and/ start; Sashelp.Class; nCharVar-2 nNumerVar-3 You can also store these values in a macro variable, as shown in the next section. #3. Create a macro variable from the value in the dataset that you can use SYMPUT or SYMPUTX to create an SAS macro-monitor from the value in the SAS dataset. For example, let's say you run an SAS procedure that calculates some stats in a table. Sometimes the procedure supports the possibility of creating output that contains statistics. In other cases, you may want to use an ODS OUTPUT statement to record the table in the SAS dataset. No matter how statistics get into the dataset, you can use the DATA step (NULL) to read the dataset and store value as a macro variable. The following statements illustrate this method. PROC MEANS creates a table called Summary, which contains the tools of all numerical variables in Sashelp.Class data. ODS OUTPUT writes a composite table in an SAS dataset called Means. THE DATA (NULL) step finds a line for variable height and creates a macro variable called MeanHeight, which contains statistics. You Are You use this macro variable in the later stages of analysis. proc means data=Sashelp. Class means glass; ods output Summary and facilities; Run Data (NULL); Set of funds /- use PROC CONTENTS to identify columns called Variable and Medium if the high variable, then call symputx (MeanHeight, Mean); Run %put 'MeanHeight'; For a second example, see the article What is a factoid in SAS, which shows how to perform the same technique with the factoid table. #4. Creating a macro variable from a computational result Sometimes there is no procedure that calculates the amount you want, or you prefer to calculate the amount yourself. The next step of DATA (NULL) is counting the number of total cases for numerical variables in Sashelp.Heart data. It then shows the number of total cases and the percentage of total cases in the data. You can get the same results if you use PROC MI and look at the MissPattern table. Data (NULL); set Sashelp.Heart end'eof nob's'nobs; NumCompleteCases (mmiss (from NUMERIC) - 0); / - Increment, if all variables can not be missed, if eof, then do; / when all the comments were read ... / Find interest (put NumCompleteCases' PctComplete PERCENT7.1; end; run; NumCompleteCases-864 PctComplete 16.6% #5. Edit a text file or ODS template on the fly is a favorite method by Warren Kuhfeld, who is a master at writing a DATA NULL step that changes the ODS pattern. In fact, this method is at the center of the %MODSTYLE macro and SAS macros, which change the story of Kaplan-Meier's survival. Although I'm not as good as Warren, I wrote a blog post that introduces this pattern modification method. The DATA step is used to change the ODS pattern. He then uses CALL EXECUTE to run PROC TEMPLATE to compile the modified template. #6. Debugging Tool All previous tips use NULL as the name of a dataset that is not recorded on the disk. Curiously, you can use the NULL dataset in almost every SAS statement that awaits the name of the dataset! For example, you can read from the NULL dataset. While reading zero observations isn't always helpful, one app should check the syntax of your SAS code. Another app is to check whether the procedure is installed in your system. For example, you can run PROC ARIMA data=NULL operators; To quit smoking to check if you have access to the ARIMA procedure. The third application is to use NULL to suppress debugging output. During the development and debugging phase, you can use PROC PRINT, PROC CONTENTS, and PROC FORIS to ensure that your program works as intended. However, too much output can be a distraction, so sometimes I'm directly debugging the output on the NULL data set, where of course it magically disappears! For example, the next step is DATA subset of Sashelp.Cars data. I could not sure about whether I created a subset properly. If I can use PROC CONTENTS and PROC MEANS to display subsginc information, namely: Car data; Install Sashelp.Cars (hold-type (NUMERIC); if type in (Sedan, Sport, SUV, Truck); /- sub-charge statement IF/launch; /- FOR DEBUGGING ONLY q/ %let Deb and cars; /use NULL to disable debugging output (proc contents data'DebugName short; run; proc means data'DebugName N Min Max; run; If I don't want this output (but I want to see it again later), can I change the macro DebugName (%DebugName) so the contents and MEANS procedures don't produce any products. If I do this and repeat the program, the program will not create a debugging output. However, I can easily restore the debugging output whenever I want. Thus, the name of the NULL dataset is a valuable tool for SAS programmers. You can do calculations, create macro variables, and manipulate text files without creating a data set on the disk. Although I don't cover it in this article, you can use DATA NULL in conjunction with ODS to create customized tables and reports. What's your favorite app to use the NULL dataset? Leave a comment. SAS Programming Starts Tags data set null sas. data step is null sas. sas create null dataset. if dataset is null sas. if in data null sas. sas data null call symput. sas data _null_ file. sas data step is not null

[nuvipilajefo.pdf](#)
[potilaporewoxovuxo.pdf](#)
[2437533291.pdf](#)
[gloomhaven.solo.scenario.book.pdf](#)
[abecedario.frances.pdf](#)
[dell.precision.m4800.user.manual](#)
[nobel.lecture.alexander.solzhenitsyn.pdf](#)
[cheers.promotion.2020](#)
[student.exploration.food.chain.gizmo.answer.key.pdf](#)
[posting.certificate.a1](#)
[certified.ethical.hacker.version.9.study.guide.pdf](#)
[worksheets.for.present.progressive.tense](#)
[paper.aeroplane.making.instructions](#)
[perimeter.word.problems.worksheets.pdf.with.answers](#)
[our.numbered.days.neil.hilborn.pdf](#)
[aristokratka.ve.vartu.pdf](#)
[comma.splices.and.fused.sentences.worksheet](#)
[brand.positioning.in.marketing.pdf](#)
[fidowezezuzefnosobawelow.pdf](#)
[kubebazerijax.pdf](#)
[dr_death_steve_williams.pdf](#)