**SCRIPT: Subsetting Observations**

Recorded in: Camtasia 2.8.1 for Mac

Edit in: Camtasia 8.4.1 for Windows

Record: Full Screen

System audio: 15%

|  |  |
| --- | --- |
| **Action on Screen** | **Narration** |
| S1 | You’ve already seen how to conditionally process your data using IF-THEN-ELSE statements and DO Groups. The final technique we’ll discuss for telling SAS to perform some action, given some condition, is the WHERE statement. |
| S2 | The general form of a WHERE statement is shown here. The where-expression specifies conditions for selecting observations, and can be any valid SAS expression.  Where-expressions are just a set of conditions related to your data that SAS evaluates. You will see some examples soon. |
| S3 | In addition to the general categories of operators you’ve just seen, there are several special operators that can only be used in a WHERE statement. Let’s look more closely at these special WHERE operators. |
| S4 | The between-and operator selects observations in which the value of a variable falls within an inclusive range of values. For example, between 1000 and 2000 mosquitos. The second WHERE statement shown here is equivalent to the first, just written without using the between-and operator. |
| S5 | The is null and is missing operators select observations in which the value of a variable is missing. These operators can be used for both character and numeric variables. |
| S6 | The contains operator selects observations that include the specified substring. The position of the substring within the variables value does not matter. |
| S7 | For example, the WHERE statements shown here select observations in which the value for job\_title is infection control manager, information systems manager, or special initiatives manager. However, the contains operator is case sensitive, so you must specify the substring in the exact case you want it to match. |
| S8 | The like operator selects observations by comparing character values to specified patterns. There are two special characters available for specifying a pattern. The percent sign replaces any number of characters. The underscore replaces one character, and you may use consecutive underscores to replace multiple consecutive characters. Additionally, a percent sign and an underscore can be specified in the same pattern.  For example, the WHERE statement shown here selects observations in which the value of first name ends with a capital m, and is preceded by any number of characters. The second WHERE statement selects observations in which the value of first name begins with a capital t and ends with a capital m, and has any one character between the t and the m. This would include TIM or TOM. |
| S9 | In this demonstration you will learn how to subset observations in a DATA step using the WHERE statement. |
| Open on list report from m5.3\_lecture\_demo\_Subsetting Observatons | In this program you will create a new temporary SAS data set called subset1. You will use a WHERE statement to specify that only employees whose job title contains Lab, or whose employee id is less than 120110 be included in the new data set.  Then you will create another list report and view the data portion of subset1. |
| Show list report | In observations 1 through 7, job title does not include the character string Lab, but the employee ID is less than 120110. Conversely, observations 10 through 25 have an employee id greater than or equal to 120110, but Lab is included in all of the job titles.  Here’s a question: If you create a new list report that uses the AND logical operator in the WHERE statement instead of the OR logical operator, how many observations will be in that report?  In this case, only observations 8 and 9 from this list report will be included in the new report. |
| S11 | You’ve seen how to modify a data set so that the output data set contains a subset of the OBSERVATIONS contained in the input data set. |
| S12 | Now suppose you need to create a new SAS data set that only includes a subset of the VARIABLES contained in the input data set.  You can control which variables are included in the output data set by using a DROP or KEEP statement in your DATA step. |
| S13 | You can see the general form of the drop and keep statements here. The DROP statement lists the set of variables that you want to omit from the output data set. When you use a DROP statement, all variables not listed are included in the output data set.  The KEEP statement lists the set of variables that you wan to include in the output data set. When you use a KEEP statement, ONLY the variable list will be included in the output data set. All other variables will be omitted. You should not use a DROP and a KEEP statement in the same DATA step.  Additionally, you may want to use the KEEP statement instead of the DROP statement if the number of variables you want to keep is significantly smaller than the number of variables that you want to drop.  Another benefit of using the keep statement is that you are able to see the variables included in the data set just by looking at the DATA step. |
| S14 | Here is an example of the KEEP statement in a data step. In this example, regardless of how many variables were in the data set ntrhd.original, only the variables id, height, and weight will be written to the new data set ntrhd.subset.  Importantly, however, all the variables from ntrhd.original are initialized in the PDV, and available for SAS expressions. They just aren’t written out to the new data set. |
| S15 | You’ve already seen how to create a new data set that contains only a subset of the full data set. But, let’s make things slightly more complicated. Building on the previous example, let’s say that in addition to only keeping the variables id and BMI, you only want to keep a subset of observations whose BMI is 30 or above.  Let’s think about what you need SAS to do. You don’t want to select observations based on variables in the input data set. You want to select observations based on variables that you’re creating. You could consider using a WHERE statement, but the WHERE statement selects observations as they are read from the input data set to the Program Data Vector. In this situation you want to subset observations based on the variable BMI, which is not a part of the ntrhd.original data set. So, you can’t use the WHERE statement. |
| S16 | To subset observations based on the value of a variable you create, you can use the subsetting IF statement. The syntax for a subsetting IF statement is just the keyword IF followed by an expression that you want to evaluate. |
| S17 | Now you can use the subsetting IF statement to select observations where BMI is greater than or equal to 30. Just remember that although the IF expression functions much like a WHERE expression, you cannot use the special where operators in IF expressions. |
| S18 | Now it’s your turn to put your new SAS skills to work |
| S19 | There is a lot to unpack in this module. Conditional processing is probably the most important part of SAS programming. If you do it incorrectly, your results will likely be wrong, and SAS won’t be able to tell you. So, we will try to give you a lot of practice in this course. And of course, if it isn’t making sense, please come ask us for help. |