

How can I use IF-THEN-ELSE statements in SAS? Can you provide some examples?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use IF-THEN-ELSE statements in SAS? Can you provide some examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152826>

IF-THEN-ELSE statements in SAS are conditional statements used to control the flow of a program based on certain conditions. These statements are used to execute a specific set of actions if a given condition is met, and a different set of actions if the condition is not met.

To use IF-THEN-ELSE statements in SAS, the syntax is as follows:

```
IF condition THEN action1;  
ELSE action2;
```

The "condition" in the statement can be any logical expression that evaluates to either TRUE or FALSE. If the condition is TRUE, then the action1 will be executed, and if the condition is FALSE, then the action2 will be executed.

For example, if we want to print "Passed" if a student's grade is greater than or equal to 60, and "Failed" if the grade is less than 60, we can use the following IF-THEN-ELSE statement:

```
IF grade >= 60 THEN  
PRINT "Passed";  
ELSE  
PRINT "Failed";
```

In this case, if the grade is 70, the statement will evaluate to TRUE and "Passed" will be printed. If the grade is 50, the statement will evaluate to FALSE and "Failed" will be printed.

IF-THEN-ELSE statements can also be nested, meaning one IF-THEN-ELSE statement can be placed inside another. This allows for more complex conditions and actions to be executed.

In summary, IF-THEN-ELSE statements in SAS are powerful tools for controlling the flow of a program based on certain conditions, and can be used in various scenarios to make the program more efficient and effective.

Use IF-THEN-ELSE in SAS (With Examples)

You can use an IF-THEN-ELSE statement in SAS to return some value *if* some condition is true, *else* return another value if some condition is not true.

This statement uses the following basic syntax:

```
if var1 > 30 then var2 = 'good';  
else var2 = 'bad';
```

You can also chain together several ELSE IF statements to return more potential values based on more conditions:

```
if var1 > 35 then var2 = 'great';  
else if var1 > 30 then var2 = 'good';  
else var2 = 'bad';
```

The following examples show how to use each of these statements in practice with the following dataset in SAS:

```
/*create dataset*/  
data original_data;  
input team $ points;  
datalines;  
Cavs 12  
Cavs 14  
Warriors 15  
Hawks 18  
Mavs 31
```

Mavs 32

Mavs 35

Celtics 36

Celtics 40

;

run;

/*view dataset*/

proc printdata=original_data;

Obs	team	points
1	Cavs	12
2	Cavs	14
3	Warriors	15
4	Hawks	18
5	Mavs	31
6	Mavs	32
7	Mavs	35
8	Celtics	36
9	Celtics	40

Example 1: IF-THEN-ELSE in SAS

We can use the following IF-THEN-ELSE statement to create a new variable called rating that takes on a value of "good" if the value in the points column is greater than 30 or a value of "bad" otherwise:

```
/*create new dataset with new variable called rating*/  
data new_data;  
set original_data;  
if points > 30then rating = 'good';  
else rating = 'bad';  
run;  
  
/*view new dataset*/  
proc printdata=new_data;
```

Obs	team	points	rating
1	Cavs	12	bad
2	Cavs	14	bad
3	Warriors	15	bad
4	Hawks	18	bad
5	Mavs	31	good
6	Mavs	32	good
7	Mavs	35	good
8	Celtics	36	good
9	Celtics	40	good

Notice that the new column called rating takes on a value of "good" if the value in the points column is greater than 30 or a value of "bad" otherwise.

Example 2: IF-THEN-ELSE IF in SAS

We can use the following IF-THEN-ELSE IF statement to create a new variable called rating that takes on the following values:

"great" if points is greater than 35 else, "good" if points is greater than 30 else, "bad"

The following code shows how to do so:

```
/*create new dataset with new variable called rating*/  
data new_data;  
set original_data;  
if points > 35 then rating = 'great';  
else if points > 30 then rating = 'good';  
else rating = 'bad';  
run;  
  
/*view new dataset*/  
proc print data=new_data;
```

Obs	team	points	rating
1	Cavs	12	bad
2	Cavs	14	bad
3	Warriors	15	bad
4	Hawks	18	bad
5	Mavs	31	good
6	Mavs	32	good
7	Mavs	35	good
8	Celtics	36	great
9	Celtics	40	great

The new column called rating takes on a value of "great", "good", or "bad" based on the corresponding value in the points column.

Note: Feel free to use as many ELSE IF statements as you'd like to return as many different values as you'd like based on various conditions.

The following tutorials explain how to perform other common tasks in SAS: