

How can I use IF-THEN-DO statements in SAS with examples?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use IF-THEN-DO statements in SAS with examples?*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=164485>

IF-THEN-DO statements in SAS are a conditional programming tool used to execute a specific set of actions when a certain condition is met. This allows for more efficient and precise data processing and analysis. The syntax for an IF-THEN-DO statement in SAS is as follows:

```
IF THEN DO;
```

```
END;
```

The can be any logical or arithmetic expression, and the can range from simple assignments to complex data manipulation.

For example, if we want to create a new variable "age_group" based on the age of a person, we can use the following IF-THEN-DO statement in SAS:

```
IF age = 18 AND age
```

```
END;
```

Overall, IF-THEN-DO statements in SAS provide a powerful and flexible way to control the flow of data processing and analysis. They can be used in various scenarios to make the code more efficient and accurate.

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

You can use an IF-THEN-DO statement in SAS to *do* a block of statements *if* some condition is true.

This statement uses the following basic syntax:

```
if var1 = "value" then do;
```

```
new_var2 = 10;
```

```
new_var3 = 5;
```

```
end;
```

Note: An IF-THEN statement is used when you only

want to do *one* statement. An IF-THEN-DO statement is used when you want to do several statements.

The following example shows how to use an IF-THEN-DO statement in practice.

Example: IF-THEN-DO in SAS

Suppose we have the following dataset in SAS that shows the total sales made by two stores during consecutive days:

```
/*create dataset*/  
data original_data;  
input store $ sales;  
datalines;
```

A 14

A 19

A 22

A 20

A 16

A 26

B 40

B 43

B 29

B 30

B 35

B 33

;

run;

/*view dataset*/

proc printdata=original_data;

Obs	store	sales
1	A	14
2	A	19
3	A	22
4	A	20
5	A	16
6	A	26
7	B	40
8	B	43
9	B	29
10	B	30
11	B	35
12	B	33

We can use the following IF-THEN-DO statement to create two new variables that take on certain values if the store is equal to "A" in the original dataset:

/*create new dataset*/

```
data new_data;  
set original_data;  
if store = "A" then do;  
region="East";  
country="Canada";  
end;  
run;  
  
/*view new dataset*/  
proc printdata=new_data;
```

Obs	store	sales	region	country
1	A	14	East	Canada
2	A	19	East	Canada
3	A	22	East	Canada
4	A	20	East	Canada
5	A	16	East	Canada
6	A	26	East	Canada
7	B	40		
8	B	43		
9	B	29		
10	B	30		
11	B	35		
12	B	33		

Here's how this chunk of code worked:

If the store was equal to "A" then a new variable called

region was created with a value of "East" *and* a new variable called country was created with a value of "Canada."

Note that we can use multiple IF-THEN-DO statements as well:

```
/*create new dataset*/  
data new_data;  
set original_data;  
  
if store = "A" then do;  
region="East";  
country="Canada";  
end;  
  
if store = "B" then do;  
region="West";  
country="USA";  
end;  
run;  
  
/*view new dataset*/  
proc printdata=new_data;
```

Obs	store	sales	region	country
1	A	14	East	Canada
2	A	19	East	Canada
3	A	22	East	Canada
4	A	20	East	Canada
5	A	16	East	Canada
6	A	26	East	Canada
7	B	40	West	USA
8	B	43	West	USA
9	B	29	West	USA
10	B	30	West	USA
11	B	35	West	USA
12	B	33	West	USA

If the store was equal to "A" then a new variable called region was created with a value of "East" and a new variable called country was created with a value of "Canada." If the store was equal to "B" then the value for region was "West" and the value for country was "USA."

Additional Resources

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