

Can I use an IF statement in PROC SQL to implement the SAS technique of “SAS: Use IF Statement in PROC SQL”?

Authored by
stats writer

June 25, 2024

RECOMMENDED CITATION

stats writer (2024). *Can I use an IF statement in PROC SQL to implement the SAS technique of “SAS: Use IF Statement in PROC SQL”?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=152820>

The "SAS: Use IF Statement in PROC SQL" technique allows for the usage of conditional logic within PROC SQL code, similar to the IF statement in traditional SAS programming. This can be achieved by using the IF statement within the SELECT clause of the PROC SQL statement, allowing for more efficient and streamlined data manipulation.

SAS: Use IF Statement in PROC SQL

While it's not possible to use an IF statement in PROC SQL in SAS, you can use the CASE operator to define the values that a variable should take on based on certain conditions.

The following examples show how to use the CASE operator in practice with the following dataset in SAS that contains information about various basketball players:

```
/*create dataset*/  
data my_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=my_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: Use CASE Operator with Only Two Outcomes

We can use the CASE operator in PROC SQL to generate a new column in the dataset called points_flag that takes a value of 0 if the value in the points column is less than 20 or a value of 1 otherwise:

```
/*create new column called points_flag using case operator*/  
proc sql;  
select *,  
casewhen points < 20then0else1endas points_flag  
from my_data;  
quit;
```

team	points	points_flag
Cavs	12	0
Cavs	14	0
Warriors	15	0
Hawks	18	0
Mavs	31	1
Mavs	32	1
Mavs	35	1
Celtics	36	1
Celtics	40	1

Notice that the points_flag column takes on a value of 0 if the value in the points column is less than 20 or a value of 1 otherwise.

Example 2: Use CASE Operator with More Than Two Outcomes

We can also use the CASE operator in PROC SQL to generate a new column in the dataset called points_flag

that takes a value of 0 if the value in the points column is less than 20, a value of 1 if points is less than 35, or a value of 2 otherwise:

```
/*create new column called points_flag using case operator*/  
proc sql;  
select *,  
casewhen points < 20then0  
when points < 35 then 1else 2endas points_flag  
from my_data;  
quit;
```

team	points	points_flag
Cavs	12	0
Cavs	14	0
Warriors	15	0
Hawks	18	0
Mavs	31	1
Mavs	32	1
Mavs	35	2
Celtics	36	2
Celtics	40	2

Notice that the points_flag column takes on a value of 0, 1, or 2 depending on the corresponding value in the

points column.

Note: Feel free to use as many when statements as you'd like to generate as many different values as you'd like in a new column.

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

ARABPSYCHOLOGY.COM