How do you use the NOT EQUAL operator in SAS?

Authored by

stats writer

June 23, 2024

The NOT EQUAL operator in SAS, denoted by the symbol 'NE', is used to compare two values and determine if they are not equal. This operator is commonly used in SAS to create conditional statements, where certain actions are taken if the two values being compared are not equal. The 'NE' operator can be used with any data type, including numeric, character, and date values. To use the NOT EQUAL operator in SAS, simply place it between the two values being compared, and the result of the comparison will be either true or false. This operator is a powerful tool in data analysis and can help identify patterns and anomalies in datasets.

Use the NOT EQUAL Operator in SAS

There are two ways to use a NOT EQUAL operator in SAS:

ne^=

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

```
/*create dataset*/
data my_data;
input team $ position $ points assists;
datalines;
A Guard 14 4
A Guard 22 22
A Guard 24 9
A Forward 13 13
A Forward 13 9
```

```
A Forward 10 10
B Guard 24 4
B Guard 10 6
B Forward 34 2
B Forward 15 5
B Forward 23 23
B Forward 10 4
;
run;

/*view dataset*/
proc printdata=my_data;
```

Obs	team	position	points	assists
1	A	Guard	14	4
2	A	Guard	22	22
3	A	Guard	24	9
4	A	Forward	13	13
5	A	Forward	13	9
6	A	Forward	10	10
7	B	Guard	24	4
8	B	Guard	10	6
9	B	Forward	34	2
10	B	Forward	15	5
11	B	Forward	23	23
12	B	Forward	10	4

Example 1: Using ne as "NOT EQUAL" in SAS

The following code shows how to create a new dataset in SAS and use the ne operator to check if the values in the points and assists columns in each row are not equal:

```
/*create new dataset*/
data new_data;
set my_data;
if points ne assists then points_vs_assists = 'not equal';
else points_vs_assists = 'equal';
run;

/*view dataset*/
proc printdata=new_data;
```

Obs	team	position	points	assists	points_vs_assists
1	A	Guard	14	4	not equal
2	A	Guard	22	22	equal
3	A	Guard	24	9	not equal
4	A	Forward	13	13	equal
5	A	Forward	13	9	not equal
6	A	Forward	10	10	equal
7	B	Guard	24	4	not equal
8	B	Guard	10	6	not equal
9	B	Forward	34	2	not equal
10	B	Forward	15	5	not equal
11	B	Forward	23	23	equal
12	B	Forward	10	4	not equal

Notice that the new column called points_vs_assists has a value of "not equal" if the values in the points and assists columns are not equal or a value of "equal" if the values in the two columns are equal.

Example 2: Using ^= as "NOT EQUAL" in SAS

The following code shows how to create a new dataset in SAS and use the ^= operator to check if the values in the points and assists columns in each row are not equal:

```
/*create new dataset*/
data new_data;
set my_data;
```

```
if points ^= assists then points_vs_assists = 'not equal';
else points_vs_assists = 'equal';
run;

/*view dataset*/
proc printdata=new_data;
```

Obs	team	position	points	assists	points_vs_assists
1	A	Guard	14	4	not equal
2	A	Guard	22	22	equal
3	A	Guard	24	9	not equal
4	A	Forward	13	13	equal
5	A	Forward	13	9	not equal
6	A	Forward	10	10	equal
7	B	Guard	24	4	not equal
8	B	Guard	10	6	not equal
9	B	Forward	34	2	not equal
10	B	Forward	15	5	not equal
11	B	Forward	23	23	equal
12	B	Forward	10	4	not equal

Notice that the values in the points_vs_assists column match the values from the previous example.

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