Fix: “cannot compare a dtyped [float64] array with a scalar of type [bool]”

Authored by

stats writer

November 29, 2025

RECOMMENDED CITATION

stats writer (2025). *Fix: “cannot compare a dtyped [float64] array with a scalar of type [bool]”*. PSYCHOLOGICAL SCALES. Retrieved from https://scales.arabpsychology.com/?p=101919

This error occurs when trying to compare a data type array of float numbers with a boolean scalar value. This operation is not allowed in Python because the data types are incompatible. To fix the error, you need to convert the data type array of float numbers to a boolean array so that it can be compared with the boolean scalar value.

One error you may encounter when using pandas is:

TypeError: cannot compare a dtyped array with a scalar of type

This error usually occurs when you attempt to subset a DataFrame based on multiple conditions and fail to place parenthesis around each individual condition.

The following example shows how to fix this error in practice.

How to Reproduce the Error

Suppose we create the following pandas DataFrame:

import pandas as pd

#create DataFrame
df = pd.DataFrame({'team': ,
'position': ,
'points': })

#view DataFrame
print(df)

team position points
0 A G 21
1 A G 30
2 A F 26
3 A C 29
4 B G 14
5 B F 29
6 B F 22
7 B C 16

Now suppose we attempt to use the **.loc** function to only display the rows where the team is equal to 'A' and the position is equal to 'G':

#attempt to only show rows where team='A' and position='G'
df.loc

TypeError: cannot compare a dtyped array with a scalar of type

We receive a **ValueError** because we didn't place parenthesis around the individual conditions.

Since the **&** operator takes precedence over the **==** operator, pandas fails to interpret this statement in the correct order.

How to Fix the Error

The easiest way to fix this error is to simply add parenthesis around the individual conditions as follows:

#only show rows where team='A' and position='G'
df.loc

team position points
0 A G 21
1 A G 30

Notice that we don't receive any **ValueError** and we are able to successfully subset the DataFrame.