

What could be causing the runtime warning “invalid value encountered in double_scalars” and how can it be fixed?

Authored by
stats writer

July 2, 2024

RECOMMENDED CITATION

stats writer (2024). *What could be causing the runtime warning “invalid value encountered in double_scalars” and how can it be fixed?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=165902>

The runtime warning "invalid value encountered in double_scalars" is caused by attempting to perform mathematical operations on values that are either undefined or too large/small to be represented by the computer. This can happen due to faulty input data, division by zero, or errors in the algorithm. To fix this warning, the code needs to be carefully checked and any potential errors in the input data or algorithm need to be addressed. Additionally, using data types with larger precision (such as floats instead of integers) can help prevent these errors from occurring. Proper error handling and debugging techniques can also be used to identify and resolve the issue causing the warning.

Fix: runtimewarning: invalid value encountered in double_scalars

One error you may encounter in Python is:

runtimewarning: invalid value encountered in double_scalars

This error occurs when you attempt to perform some mathematical operation that involves extremely small or extremely large numbers and Python simply outputs a NaN value as the result.

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

How to Reproduce the Error

Suppose we attempt to perform the following mathematical operation with two NumPy arrays:

```
import numpy as np
```

```
#define two NumPy arrays
```

```
array1 = np.array([])
```

```
array2 = np.array([])
```

```
#perform complex mathematical operation
```

```
np.exp(-3*array1).sum() / np.exp(-3*array2).sum()
```

RuntimeWarning: invalid value encountered in double_scalars

We receive a RuntimeWarning because the result in the denominator is extremely close to zero.

This means the answer to the division problem will be extremely large and Python is unable to handle this large of a value.

How to Fix the Error

Typically the way to fix this type of error is to use a special function from another library in Python that is capable of handling extremely small or extremely large values in calculations.

In this case, we can use the `logsumexp()` function from the SciPy library:

```
import numpy as np  
from scipy.special import logsumexp  
  
#define two NumPy arrays  
array1 = np.array()  
array2 = np.array()  
  
#perform complex mathematical operation  
np.exp(logsumexp(-3*array1) - logsumexp(-3*array2))  
  
2.7071782767869983e+195
```

Notice that the result is extremely large but we don't receive any error because we used a special mathematical function from the SciPy library that was designed to handle these types of numbers.

In many cases, it's worth looking up special functions from the that can handle extreme mathematical operations because these functions are designed specifically for scientific computing.

Note: You can find the complete online documentation

for the logsumexp() function .

Additional Resources

ARABPSYCHOLOGY.COM