How do you calculate a trimmed mean in Python, and what are some examples of its usage?

Authored by

stats writer

May 6, 2024

The trimmed mean is a statistical measure used to calculate the average of a set of data after removing a certain percentage of outliers from both ends of the data. In Python, the trimmed mean can be calculated using the "trim_mean" function from the "scipy.stats" library. This function takes in two parameters - the data set and the trimming percentage (usually between 5-25%). It then removes the specified percentage of data from both the top and bottom of the data set and calculates the mean of the remaining values.

For example, if we have a data set of 100 numbers and we specify a trimming percentage of 10%, the "trim_mean" function will remove the top and bottom 10 numbers (10% of 100) and calculate the mean of the remaining 80 numbers. This helps to eliminate the impact of extreme values on the overall average and provides a more accurate representation of the data.

The trimmed mean is commonly used in situations where the data set contains outliers or extreme values that can skew the results, such as in financial data or survey data. It is also useful when dealing with data sets that have a large range of values. Overall, the trimmed mean is a valuable tool for calculating a more reliable average in situations where the traditional mean may not accurately represent the data.

Calculate a Trimmed Mean in Python (With Examples)

A trimmed mean is the mean of a dataset that has been calculated after removing a specific percentage of the smallest and largest values from the dataset.

The easiest way to calculate a trimmed mean in Python is to use the trim_mean() function from the SciPy library.

This function uses the following basic syntax:

from scipy import stats

#calculate 10% trimmed mean
stats.trim_mean(data, 0.1)

The following examples show how to use this function to calculate a trimmed mean in practice.

Example 1: Calculate Trimmed Mean of Array

The following code shows how to calculate a 10% trimmed mean for an array of data:

from scipy import stats

#define data
data =

#calculate 10% trimmed mean
stats.trim_mean(data, 0.1)

12.375

The 10% trimmed mean is 12.375.

This is the mean of the dataset after the smallest 10% and largest 10% of values have been removed from the dataset.

Example 2: Calculate Trimmed Mean of Column in Pandas

The following code shows how to calculate a 5% trimmed mean for a specific column in a pandas DataFrame:

```
from scipy import stats
import pandas as pd

#define DataFrame
df = pd.DataFrame({'points': ,
'assists': ,
'rebounds': })

#calculate 5% trimmed mean of points
stats.trim_mean(df.points, 0.05)
```

20.25

The 5% trimmed mean of the values in the 'points' column is 20.25.

This is the mean of the 'points' column after the smallest 5% and largest 5% of values have been removed.

Example 3: Calculate Trimmed Mean of Multiple Columns

The following code shows how to calculate a 5% trimmed mean for multiple columns in a pandas DataFrame:

```
from scipy import stats
import pandas as pd

#define DataFrame
df = pd.DataFrame({'points': ,
'assists': ,
'rebounds': })

#calculate 5% trimmed mean of 'points' and 'assists' columns
stats.trim_mean(df], 0.05)

array()
```

From the output we can see:

The 5% trimmed mean of the 'points' column is 20.25.The 5% trimmed mean of the 'assists' column is 7.75.

Note: You can find the complete documentation for the trim_mean() function .