How do you center data in Python, and can you provide examples?

Authored by

stats writer

July 1, 2024

Centering data in Python refers to the process of shifting the data values around a central point, usually the mean, to make it easier to analyze and interpret. This can be achieved by using the NumPy library's "center" function, which subtracts the mean from each data point. Alternatively, the Pandas library's "center" method can also be used for centering data in a DataFrame. An example of centering data in Python using the NumPy library would be:

```
import numpy as np
data = np.array()
centered_data = np.center(data) # centered_data = array()
```

Similarly, using the Pandas library, the following code can be used to center data in a DataFrame:

```
import pandas as pd
df = pd.DataFrame({'A': , 'B': })
centered_df = df.center() # centered_df = A B 0 -2 -20 1 -1 -10 2 0 0 3 1 10 4 2 20
```

Center Data in Python (With Examples)

To center a dataset means to subtract the from each individual in the dataset.

Once you've centered a dataset, the mean value of the dataset becomes zero.

The following examples show how to center data in Python.

Example 1: Center the Values of a NumPy Array

Suppose we have the following NumPy array:

import numpy as np

```
#create NumPy array
data = np.array()

#display mean of arrayprint(data.mean())
```

14.0

We can define a to subtract the mean value of the array from each individual observation:

```
#create function to center data
center_function = lambda x: x - x.mean()

#apply function to original NumPy array
data_centered = center_function(data)

#view updated Array
print(data_centered)
```

array()

The resulting values are the centered values of the dataset.

Since the mean of the original array was 14, this function simply subtracted 14 from each individual

value in the original array.

For example:

1st value in centered array = 4 - 14 = -102nd value in centered array = 6 - 14 = -83rd value in centered array = 9 - 14 = -5

And so on.

We can also verify that the mean of the centered array is zero:

```
#display mean of centered arrayprint(data_centered.mean())
```

0.0

Example 2: Center the Columns of a Pandas DataFrame

Suppose we have the following pandas DataFrame:

```
import pandas as pd

#create DataFrame
df = pd.DataFrame({'x': ,
'y': ,
```

'z': })

#view DataFrame
print(df)

x y z
0 1 7 3
1 4 7 3
2 5 8 4
3 6 8 4
4 6 8 6
5 8 9 7
6 9 12 7

We can use the pandas apply() function to center the values of each column in the DataFrame:

#center the values in each column of the DataFrame
df_centered = df.apply(lambda x: x-x.mean())

#view centered DataFrameprint(df_centered)

x y z
0 -4.571429 -1.428571 -1.857143
1 -1.571429 -1.428571 -1.857143

2 -0.571429 -0.428571 -0.857143

3 0.428571 -0.428571 -0.857143

4 0.428571 -0.428571 1.142857

5 2.428571 0.571429 2.142857

6 3.428571 3.571429 2.142857

We can then verify that the mean value of each column is zero:

#display mean of each column in the DataFrame
df_centered.mean()

x 2.537653e-16
y -2.537653e-16
z 3.806479e-16
dtype: float64

The column means are shown in scientific notation, but each value is essentially equal to zero.

Additional Resources

The following tutorials explain how to perform other common operations in Python: