Dana DiVincenzo
BANA.680.01

Assignment 1

Function for Calculating Z-scores for data in a Numpy Array

Description:

In this assignment I was tasked to create a function that took in a numpy array and a scalar, and output a transformed numpy array. The array should be transformed, where the column of the array indicated by the scalar changes into z scores, and if there is no scalar given all columns are changed into z scores. The purpose of this assignment was to improve my numpy and python proficiency, as well as increase my familiarity with Jupyter Notebook.

My Code:

In [32]:

```
import numpy as np

# This function takes in a numpy array, arr, and a scalar, y. It outputs a
# transformed numpy array, where the column of the array indicated by the
# scalar is transformed into z scores. If no scalar is inputted, the entire
# array is transformed into z scores. If inapplicable inputs are given, such
# as the wrong type of inputs or a scalar that is out of bounds relative to
# the length of the array, an appropriate error is returned as well as an
# empty float numpy array.

def myz(arr, y=-1):
    if type(arr) != np.ndarray:
        print("ERROR: Expected numpy.ndarray as first argument")
        return np.array([], dtype=float)
    elif type(y) != int:
        print("ERROR: Expected int as second argument")
        return np.array([], dtype=float)
    elif len(arr) == 0 and y!= -1:
        print("ERROR: Got an empty array, but scalar was specified")
        return np.array([], dtype=float)
    elif y < -1 or y==0 or y> len(arr[0]):
        print("ERROR: Scalar is out of bounds")
        return np.array([], dtype=float)
    else:
        arr = arr.astype(float)
        if y == -1:
            for j in range(len(arr[0])):
                processcol(arr, j)
        elif y > 0 and y <= len(arr[0]):
            processcol(arr, y-1)
        return(arr)

# This function takes in a numpy array, arr, and a scalar, y. It then
# isolates the yth column in the array, and replaces every value in
# that column with its z score using the function calculatezscore.
def processcol(arr, y):
    ythcol = arr[:, y]
    u = np.mean(ythcol)
    std = np.std(ythcol)
    for i in range(len(ythcol)):
        x = ythcol[i]
        z = calculatezscore(x, u, std)
        arr[i, y] = z

# This function takes in a value x, a mean, u, and a standard deviation,
# std, and calculates the z score for that value.
def calculatezscore(x, u, std):
    return (x-u)/std
```

Testing My Code

In [27]:

```
# Given inputs
x1 = np.array([[4,3,12],[1,5,20],[1,2,3],[10,20,40],[7,2,44]])
x2 = 3
x3 = 6
x4 = 'two'

# Test 1
# This will return an output where every value in the numpy array is
# transformed into z scores, as no scalar is given.
myz(x1)
```

Out[27]:

```
array([[-0.17149859, -0.49363572, -0.74427584],
       [-1.02899151, -0.20326177, -0.23968205],
       [-1.02899151, -0.63882269, -1.31194386],
       [ 1.54348727,  1.97454287,  1.02180243],
       [ 0.68599434, -0.63882269,  1.27409932]])
```

In [28]:

```
# Test 2
# This will return an output where the third column of the numpy array
# is transformed into z scores.
myz(x1,x2)
```

Out[28]:

```
array([[ 4.        ,  3.        , -0.74427584],
       [ 1.        ,  5.        , -0.23968205],
       [ 1.        ,  2.        , -1.31194386],
       [10.        , 20.        ,  1.02180243],
       [ 7.        ,  2.        ,  1.27409932]])
```

In [29]:

```
# Test 3
# This will print an error message letting the user know that their
# second input is out of bounds relative to the length of the array,
# as well as return an empty float numpy array.
myz(x1,x3)
```

```
ERROR: Scalar is out of bounds
```

Out[29]:

```
array([], dtype=float64)
```

In [30]:

```
# Test 4
# This will print an error message letting the user know that their
# first input needs to be a numpy array, as well as return an empty
# float numpy array.
myz(x2,x3)
```

```
ERROR: Expected numpy.ndarray as first argument
```

Out[30]:

```
array([], dtype=float64)
```

```
In [31]: # Test 5
         # This will print an error message letting the user know that they
         # need to enter an integer as the second argument, as well as return
         # an empty float numpy array.
         myz(x1,x4)

         ERROR: Expected int as second argument

Out[31]: array([], dtype=float64)
```

Conclusion

I have not had much experience with Python, so this assignment helped me improve on some of the core fundamentals of the language. It was great practice in making functions, for loops, and if/else statements. I also had to utilize the fact that Python is 0 indexed in my code, which is important to know. It also helped me improve on my general Python practices such as making my code clear and readable and writing accurate comments.

I gained more experience and knowledge of numpy and numpy functions such as np.mean() and np.std(). I also believe I am more comfortable with Jupyter Notebook now, and I really enjoy how easy it is for me to easily create and test code within the Kernal. I had never written a report in Jupyter Notebook before, but it was easy to make a clean looking report once I found some information online.

Overall, I believe the assignment was successful, as I improved my coding skills and was able to produce a clean code with the correct output.