Mean-Variance-Standard Deviation Calculator

You will be working on this project with our Replit starter code.

- Start by importing the project on Replit.
- Next, you will see a `.replit` window.
- Select `Use run command` and click the `Done` button.

We are still developing the interactive instructional part of the Python curriculum. For now, here are some videos on the freeCodeCamp.org YouTube channel that will teach you everything you need to know to complete this project:

- Python for Everybody Video Course (14 hours)
- How to Analyze Data with Python Pandas (10 hours)

Create a function named `calculate()` in `mean_var_std.py` that uses Numpy to output the mean, variance, standard deviation, max, min, and sum of the rows, columns, and elements in a 3 x 3 matrix.

The input of the function should be a list containing 9 digits. The function should convert the list into a 3 x 3 Numpy array, and then return a dictionary containing the mean, variance, standard deviation, max, min, and sum along both axes and for the flattened matrix.

The returned dictionary should follow this format:

```
{
  'mean': [axis1, axis2, flattened],
  'variance': [axis1, axis2, flattened],
  'standard deviation': [axis1, axis2, flattened],
  'max': [axis1, axis2, flattened],
  'min': [axis1, axis2, flattened],
  'sum': [axis1, axis2, flattened]
```

```
}
```

If a list containing less than 9 elements is passed into the function, it should raise a `ValueError` exception with the message: "List must contain nine numbers." The values in the returned dictionary should be lists and not Numpy arrays.

For example, `calculate([0,1,2,3,4,5,6,7,8])` should return:

```
{
  'mean': [[3.0, 4.0, 5.0], [1.0, 4.0, 7.0], 4.0],
  'variance': [[6.0, 6.0, 6.0], [0.6666666666666666, 0
  'standard deviation': [[2.449489742783178, 2.449489
  'max': [[6, 7, 8], [2, 5, 8], 8],
  'min': [[0, 1, 2], [0, 3, 6], 0],
  'sum': [[9, 12, 15], [3, 12, 21], 36]
}
```

The unit tests for this project are in `test_module.py`.

Development

For development, you can use `main.py` to test your `calculate()` function. Click the "run" button and `main.py` will run.

Testing

We imported the tests from `test_module.py` to `main.py` for your convenience. The tests will run automatically whenever you hit the "run" button.

Submitting

Copy your project's URL and submit it to freeCodeCamp.

Solution Link

https://replit.com/@WoshihsheiNigua/boilerplate-arithmetic-formatter#

I've completed this challenge

Get a Hint

Ask for Help