Assignment 4

Objective: This assignment will test your ability creating a basic Python package and using a virtual environment to manage dependencies.

Instructions:

1. **Create a Package Directory:** Start by creating a directory for your Python package. You can name it something like `math_operations`.

2. **Package Structure:** Inside the package directory, create the following files:

```
math_operations/
├── mymath/
│   ├── __init__.py
│   ├── arithmetic.py
│   └── statistics.py
├── tests/
│   ├── __init__.py
│   └── test_mymath.py
├── setup.py
└── README.md
```

   - `math_operations/`: This is the root directory of your package.
   - `mymath/`: This directory contains the main package module.
   - `arithmetic.py`: Python module for basic arithmetic operations (e.g., addition, subtraction).
   - `statistics.py`: Python module with statistical functions (e.g., mean, median).
   - `tests/`: This directory contains unit tests for your package.
   - `test_mymath.py`: Python script with unit tests for the arithmetic and statistics functions.
   - `setup.py`: The script for packaging and distributing your custom package.
   - `README.md`: A documentation file explaining how to use your package and the purpose of the exercise.

3. **Package Implementation:**

   - In `arithmetic.py`, define functions for basic arithmetic operations.
   - In `statistics.py`, provide functions for statistical calculations.

4. **Unit Tests:**

   - In `test_mymath.py`, write simple unit tests to verify the correctness of your arithmetic and statistics functions. For example:

```
import unittest

 # Function to test
 def add(a, b):
     return a + b

 class TestAddition(unittest.TestCase):
     def test_add_positive_numbers(self):
         result = add(2, 3)
         self.assertEqual(result, 5)  # Assertion to check if the result is 5

     def test_add_negative_numbers(self):
         result = add(-2, -3)
         self.assertEqual(result, -5)
```

```
if __name__ == '__main__':
    unittest.main()
```

5. **Virtual Environment:**

    - Create a Python virtual environment named `math_env` for your project using `venv`.
    - Activate the virtual environment and use it for the rest of the exercise.

6. **Package Installation:**

    - Install your `math_operations` package in your virtual environment using `python setup.py install`.

7. **Testing:**

    - Run the unit tests within the virtual environment to verify that your functions are working correctly.

8. **Documentation:**

    - Update the `README.md` file to include instructions on how to use your package.
    - Include examples of how to perform basic arithmetic operations and use statistical functions.

9. **Submission:**

    - Submit your package directory (including all files and the directory structure) as a .zip.