بسم الله الرحمن الرحيم

Assignment on,

Software Testing with PyTest

Department of Software Engineering (SE)

Assignment no. 2

Assignment To

Md. Mahmodul Hasan

Lecturer at SE ,

Bangladesh Digital University

Assignment By

Irshad Hossain (2303030)

Department of SE,

Bangladesh Digital University

Importance of Automation in Software Testing

- Faster Execution – Automated tests run much faster than manual tests.
- Increased Accuracy – Eliminates human errors in repetitive testing.
- CI/CD Integration – Supports continuous integration & deployment.
- Cost-Effective – Reduces long-term testing costs.
- Scalability – Can handle large applications efficiently.
- Better Bug Detection – Detects issues early in the development cycle.

.

Comparison of Unit, Integration, and End-to-End Testing

- **Unit Testing**
  - **Tests**: Individual functions/modules.
  - **Speed**: Very fast.
  - **Complexity**: Simple.
  - **Automation**: Easily automated.
  - **Tools**: Pytest.
  - **Purpose**: Ensures function correctness

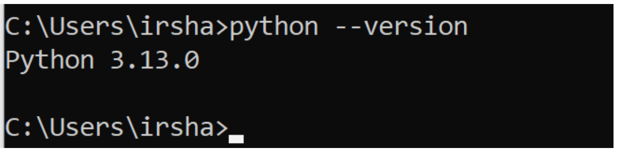
- **Integration Testing**
  - **Tests**: Interaction between modules.
  - **Speed**: Moderate.
  - **Complexity**: Moderate..
  - **Automation**: Can be automated..
  - **Tools**: Postman.
  - **Purpose**: Ensures proper module communication.

- **End-to-End (E2E) Testing**
  - **Tests**: The entire application flow.
  - **Speed**: Slow.
  - **Complexity**: High.
  - **Automation**: Automated but complex.
  - **Tools**: Selenium, Cypress.
  - **Purpose**: Simulates real-world user scenarios.

Software Test with **PyTest**

Prerequisite :

- Python 3.6 or above



```
C:\Users\irsha>python --version
Python 3.13.0

C:\Users\irsha>_
```

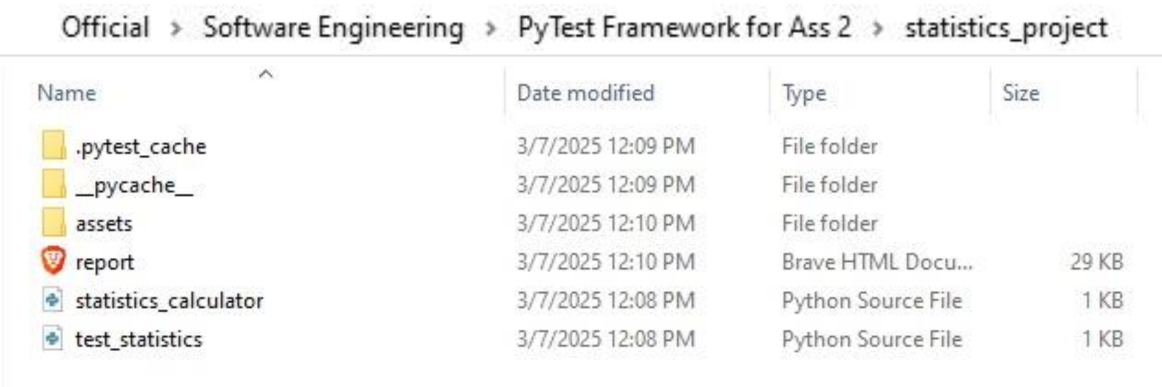
Use PyTest:

Step 1: Project Idea

We'll create a simple Statistics Calculator in Python that calculates:

- Mean (Average)
- Standard Deviation (Population Standard Deviation)

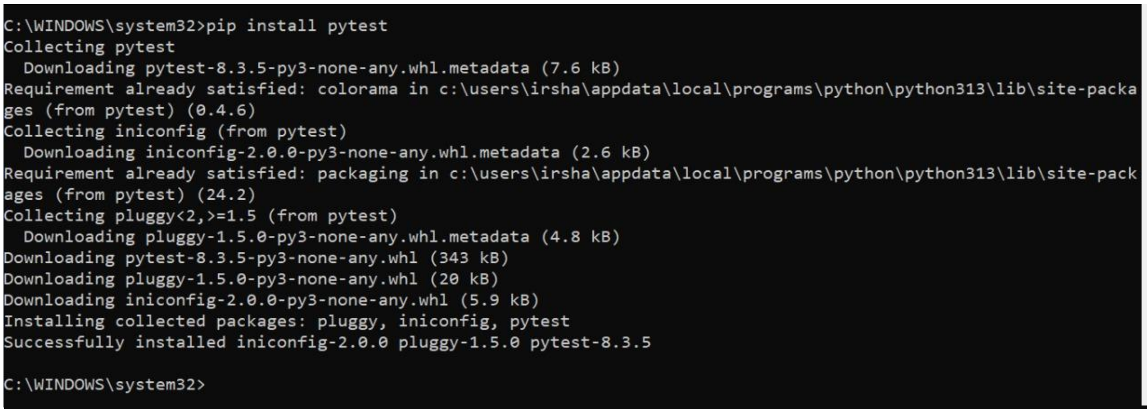
Step 2: Folder Structure



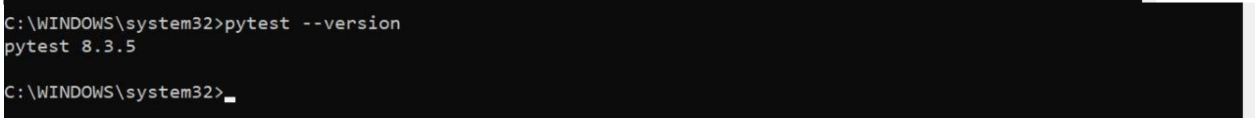
Official › Software Engineering › PyTest Framework for Ass 2 › statistics_project

Name	Date modified	Type	Size
.pytest_cache	3/7/2025 12:09 PM	File folder	
__pycache__	3/7/2025 12:09 PM	File folder	
assets	3/7/2025 12:10 PM	File folder	
report	3/7/2025 12:10 PM	Brave HTML Docu...	29 KB
statistics_calculator	3/7/2025 12:08 PM	Python Source File	1 KB
test_statistics	3/7/2025 12:08 PM	Python Source File	1 KB

Step 3: Install and Verify *PyTest* using *pip*

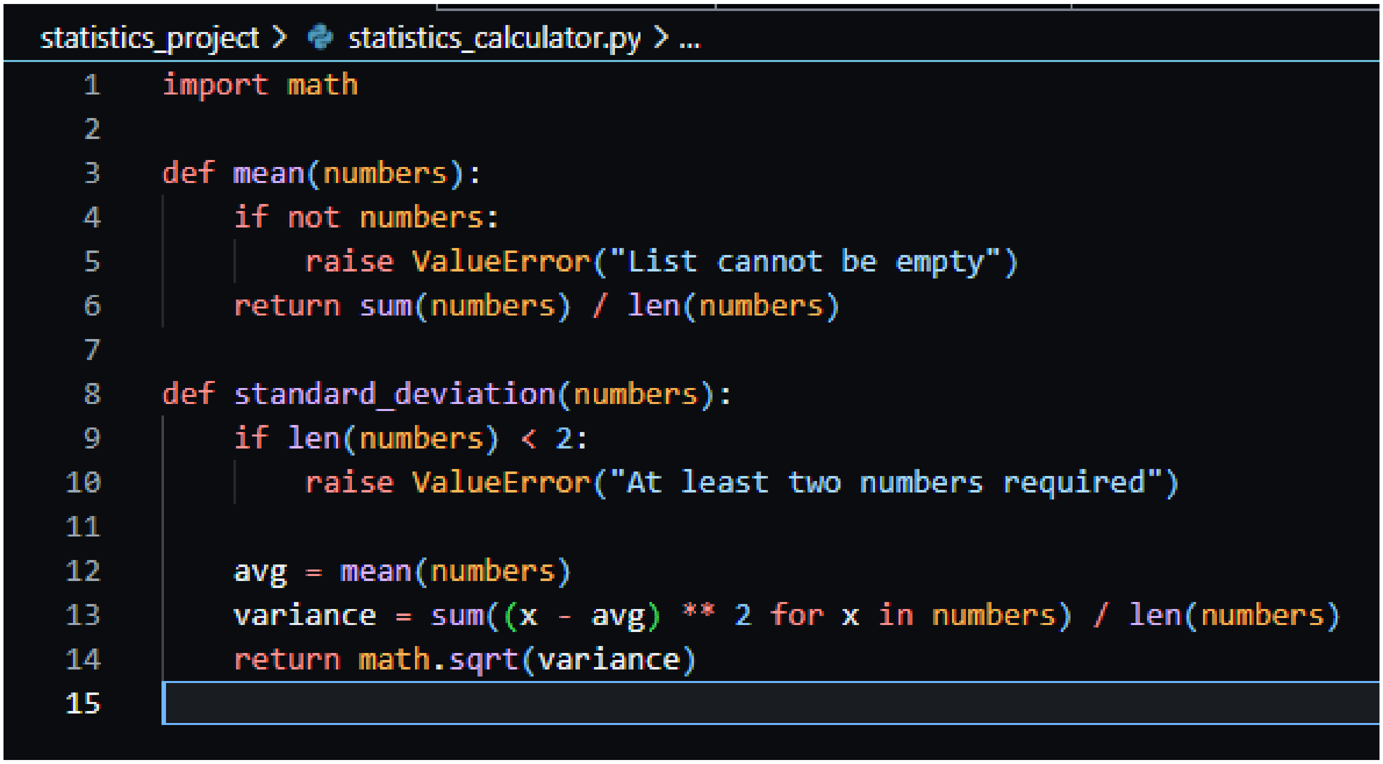


```
C:\WINDOWS\system32>pip install pytest
Collecting pytest
  Downloading pytest-8.3.5-py3-none-any.whl.metadata (7.6 kB)
Requirement already satisfied: colorama in c:\users\irsha\appdata\local\programs\python\python313\lib\site-packa
ges (from pytest) (0.4.6)
Collecting iniconfig (from pytest)
  Downloading iniconfig-2.0.0-py3-none-any.whl.metadata (2.6 kB)
Requirement already satisfied: packaging in c:\users\irsha\appdata\local\programs\python\python313\lib\site-pack
ages (from pytest) (24.2)
Collecting pluggy<2,>=1.5 (from pytest)
  Downloading pluggy-1.5.0-py3-none-any.whl.metadata (4.8 kB)
Downloading pytest-8.3.5-py3-none-any.whl (343 kB)
Downloading pluggy-1.5.0-py3-none-any.whl (20 kB)
Downloading iniconfig-2.0.0-py3-none-any.whl (5.9 kB)
Installing collected packages: pluggy, iniconfig, pytest
Successfully installed iniconfig-2.0.0 pluggy-1.5.0 pytest-8.3.5

C:\WINDOWS\system32>
```



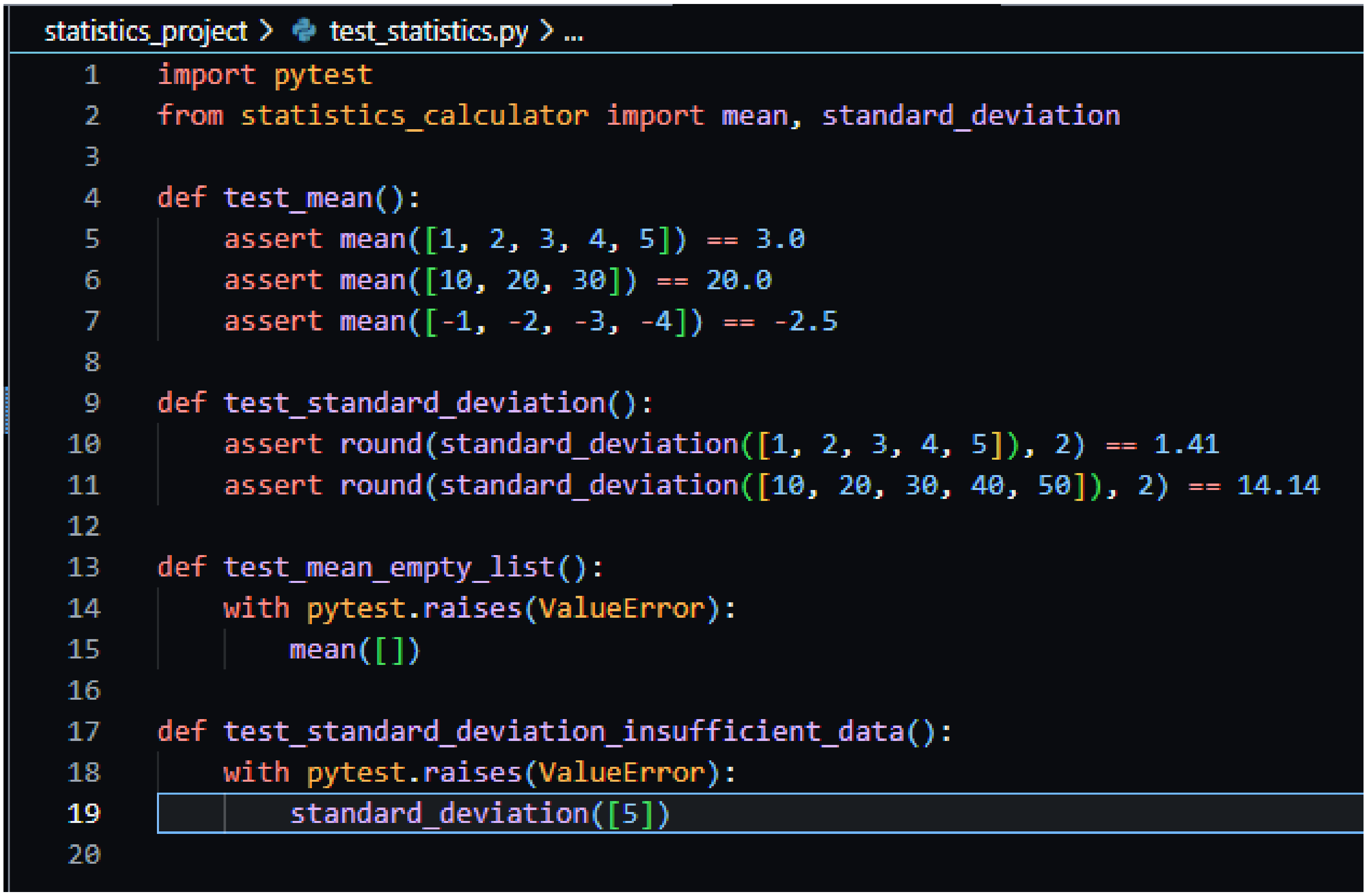
```
C:\WINDOWS\system32>pytest --version
pytest 8.3.5

C:\WINDOWS\system32>_
```

Step 4: Implement the code with functions



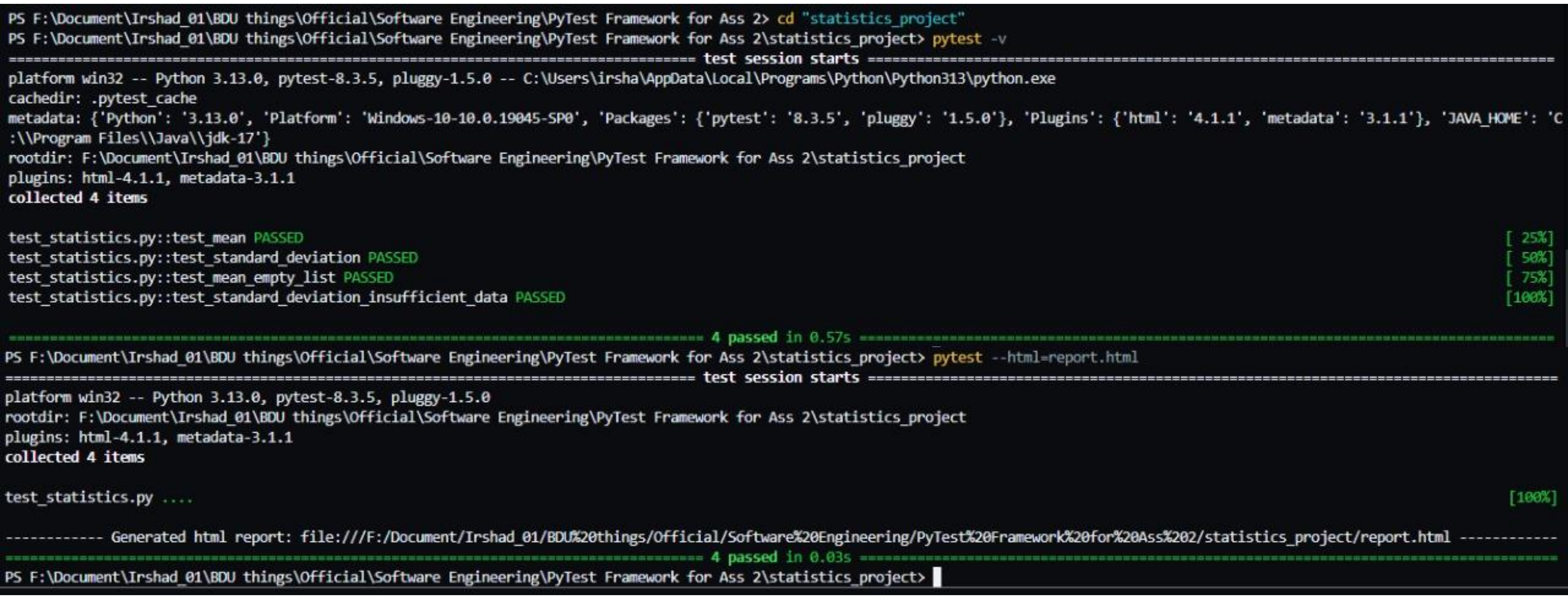
```python
import math

def mean(numbers):
    if not numbers:
        raise ValueError("List cannot be empty")
    return sum(numbers) / len(numbers)

def standard_deviation(numbers):
    if len(numbers) < 2:
        raise ValueError("At least two numbers required")

    avg = mean(numbers)
    variance = sum((x - avg) ** 2 for x in numbers) / len(numbers)
    return math.sqrt(variance)

```

Step 5: Create test cases

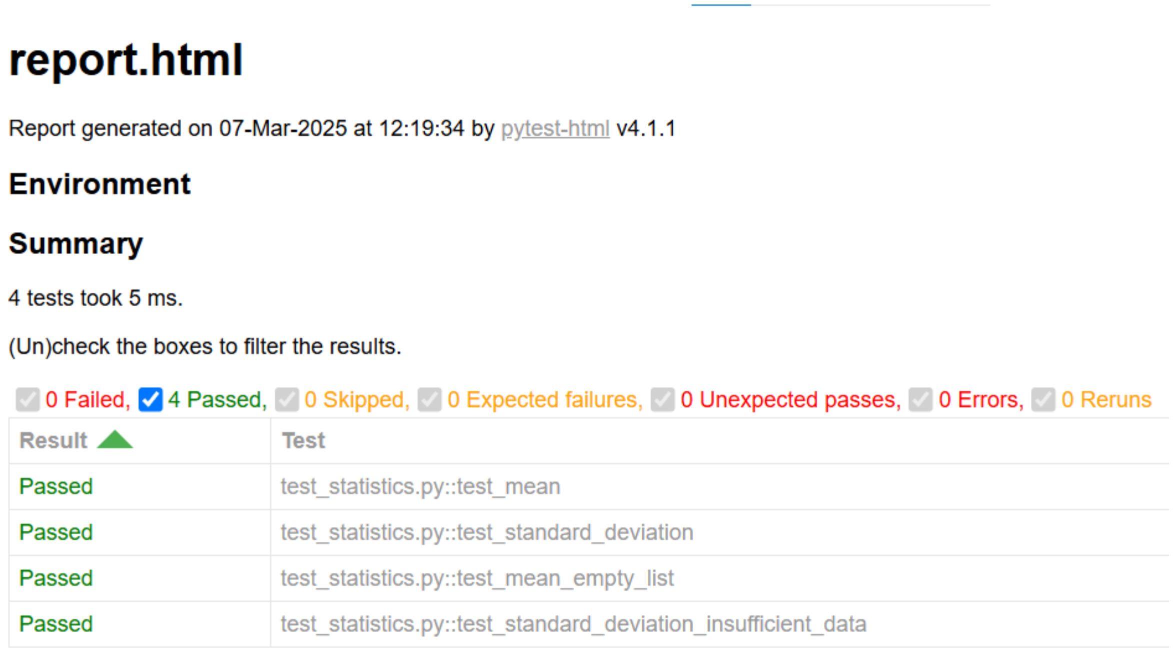


```python
import pytest
from statistics_calculator import mean, standard_deviation

def test_mean():
    assert mean([1, 2, 3, 4, 5]) == 3.0
    assert mean([10, 20, 30]) == 20.0
    assert mean([-1, -2, -3, -4]) == -2.5

def test_standard_deviation():
    assert round(standard_deviation([1, 2, 3, 4, 5]), 2) == 1.41
    assert round(standard_deviation([10, 20, 30, 40, 50]), 2) == 14.14

def test_mean_empty_list():
    with pytest.raises(ValueError):
        mean([])

def test_standard_deviation_insufficient_data():
    with pytest.raises(ValueError):
        standard_deviation([5])

```

Step 6: Run PyTest & Generate Test Report



```
PS F:\Document\Irshad_01\BDU things\Official\Software Engineering\PyTest Framework for Ass 2> cd "statistics_project"
PS F:\Document\Irshad_01\BDU things\Official\Software Engineering\PyTest Framework for Ass 2\statistics_project> pytest -v
======================================================================== test session starts ========================================================================
platform win32 -- Python 3.13.0, pytest-8.3.5, pluggy-1.5.0 -- C:\Users\irsha\AppData\Local\Programs\Python\Python313\python.exe
cachedir: .pytest_cache
metadata: {'Python': '3.13.0', 'Platform': 'Windows-10-10.0.19045-SP0', 'Packages': {'pytest': '8.3.5', 'pluggy': '1.5.0'}, 'Plugins': {'html': '4.1.1', 'metadata': '3.1.1'}, 'JAVA_HOME': 'C
:\\Program Files\\Java\\jdk-17'}
rootdir: F:\Document\Irshad_01\BDU things\Official\Software Engineering\PyTest Framework for Ass 2\statistics_project
plugins: html-4.1.1, metadata-3.1.1
collected 4 items

test_statistics.py::test_mean PASSED                                                                                                                          [ 25%]
test_statistics.py::test_standard_deviation PASSED                                                                                                            [ 50%]
test_statistics.py::test_mean_empty_list PASSED                                                                                                               [ 75%]
test_statistics.py::test_standard_deviation_insufficient_data PASSED                                                                                          [100%]

========================================================================= 4 passed in 0.57s =========================================================================
PS F:\Document\Irshad_01\BDU things\Official\Software Engineering\PyTest Framework for Ass 2\statistics_project> pytest --html=report.html
======================================================================== test session starts ========================================================================
platform win32 -- Python 3.13.0, pytest-8.3.5, pluggy-1.5.0
rootdir: F:\Document\Irshad_01\BDU things\Official\Software Engineering\PyTest Framework for Ass 2\statistics_project
plugins: html-4.1.1, metadata-3.1.1
collected 4 items

test_statistics.py ....                                                                                                                                       [100%]

------------ Generated html report: file:///F:/Document/Irshad_01/BDU%20things/Official/Software%20Engineering/PyTest%20Framework%20for%20Ass%202/statistics_project/report.html ------------
========================================================================= 4 passed in 0.03s =========================================================================
PS F:\Document\Irshad_01\BDU things\Official\Software Engineering\PyTest Framework for Ass 2\statistics_project>
```



report.html

Report generated on 07-Mar-2025 at 12:19:34 by pytest-html v4.1.1

Environment

Summary

4 tests took 5 ms.

(Un)check the boxes to filter the results.

☑ 0 Failed, ☑ 4 Passed, ☑ 0 Skipped, ☑ 0 Expected failures, ☑ 0 Unexpected passes, ☑ 0 Errors, ☑ 0 Reruns

Result ▲	Test
Passed	test_statistics.py::test_mean
Passed	test_statistics.py::test_standard_deviation
Passed	test_statistics.py::test_mean_empty_list
Passed	test_statistics.py::test_standard_deviation_insufficient_data

Ways Automated Testing Helps Improve Software Quality

- Early Bug Detection
- Faster Development Cycle
- Improved Accuracy
- Better Code Coverage
- Consistent Results
- Scalability
- Enhanced Security & Performance

Assignment Publishing Date: March 7, 2025

Tools Used: PyTest

Assigned By: Irshad Hossain (ID: 2303030)