Assignment 2 Report

Name: Stephen Martin

NetID: svm89

Github: smartin2003

Function Explanation

```python
def calcBMI(height, weight):
    heightConv = height * 0.025
    weightConv = weight * 0.45
    bmi = round(weightConv / (heightConv**2), 1)

    if bmi < 18.5:
        category = 'Underweight'
    elif bmi <= 24.9:
        category = 'Normal Weight'
    elif bmi <= 29.9:
        category = 'Overweight'
    else:
        category = 'Obese'

    return bmi, category
```

This function uses the BMI formula to calculate the BMI of the user based on their inputs.

BMI Formula: https://extoxnet.orst.edu/faqs/dietcancer/web2/twohowto.html

Height and weight are inputted using the imperial system (inches and pounds).

The BMI formula calls for the metric system so we convert the inputs by:
    Height(inches) * 0.025 = Height(meters)
    Weight(pounds) * 0.45 = Weight(kilograms)

The BMI is calculated using these values and rounds its result to the tenths place.

The function then categorizes the BMI into four different categories.
    Underweight: BMI < 18.5
    Normal Weight: 18.5 ≤ BMI ≤ 24.9
    Overweight: 25.0 ≤ BMI ≤ 29.9
    Obese: BMI ≥ 30.0

Function Test Cases

The function is tested using the pytest framework. By running pytest bmiTest.py categories, boundaries, and the boundary shift are tested.

Category Tests:

Test	Input (Height, Weight)	Expected Output
testUnderweight	(68, 110)	(17.1, 'Underweight')
testNormalWeight	(68, 140)	(21.8, 'Normal Weight')
testOverweight	(68, 180)	(28.0, 'Overweight')
testObese	(68, 230)	(35.8, 'Obese')

Boundary Tests:

Test	Input (Height, Weight)	Expected Output
testBoundaries – Lower Normal	(68, 119)	(18.5, 'Normal Weight')
testBoundaries – Upper Normal	(68, 160)	(24.9, 'Normal Weight')
testBoundaries – Lower Overweight	(68, 160.5)	(25.0, 'Overweight')
testBoundaries – Upper Overweight	(68, 192)	(29.9, 'Overweight')
testBoundaries – Obese Lower	(68, 192.5)	(30.0, 'Obese')

Boundary Shift Tests:

Test	Input (Height, Weight)	Expected Output
testBoundaryShift - Lower Normal Shift	(68, 118)	(18.4, 'Underweight')
testBoundaryShift - Normal Lower	(68, 119)	(18.5, 'Normal Weight')

Boundary Testing Technique

For this project, boundary value analysis (BVA) was used to identify edge cases where classification changes. This technique helps ensure the accuracy of classifications by testing the smallest allowable variations (0.1) at decision points.

Boundary Shift

To simulate a boundary shift error, the lower boundary of Normal Weight (18.5 BMI) was tested at 18.4 BMI (118 lbs) instead of 18.5 BMI (119 lbs).

Test Before Shift:

```
smartin2003@StephensLaptop:/mnt/c/Users/Stephen/Desktop/bmiCalc/bmiCalc$ pytest bmiTest.py
================================ test session starts ================================
platform linux -- Python 3.10.12, pytest-8.3.5, pluggy-1.5.0
rootdir: /mnt/c/Users/Stephen/Desktop/bmiCalc/bmiCalc
collected 6 items

bmiTest.py ......

================================ 6 passed in 0.05s ================================
smartin2003@StephensLaptop:/mnt/c/Users/Stephen/Desktop/bmiCalc/bmiCalc$
```

Boundary Shift:

```python
def calcBMI(height, weight):
        heightConv = height * 0.025
        weightConv = weight * 0.45
        bmi = round(weightConv / (heightConv**2), 1)

        if bmi < 18.4:
                category = 'Underweight'
        elif bmi <= 24.9:
                category = 'Normal Weight'
        elif bmi <= 29.9:
                category = 'Overweight'
        else:
                category = 'Obese'

        return round(bmi, 1), category

if __name__ == "__main__":
    print('Welcome to the BMI Calculator')
    height = float(input('What is your height in inches? '))
    weight = float(input('What is your weight in pounds? '))
    bmi, category = calcBMI(height, weight)
    print(f'Your current BMI is {bmi:.1f} which would be categorized as {category}.')
```

Test After Shift:

```
smartin2003@StephensLaptop:/mnt/c/Users/Stephen/Desktop/bmiCalc/bmiCalc$ pytest bmiTest.py
===================================== test session starts =====================================
platform linux -- Python 3.10.12, pytest-8.3.5, pluggy-1.5.0
rootdir: /mnt/c/Users/Stephen/Desktop/bmiCalc/bmiCalc
collected 6 items

bmiTest.py .....F                                                                      [100%]

========================================== FAILURES ===========================================
_____ testBoundaryShift _____

    def testBoundaryShift():
>       assert calcBMI(68, 118) == (18.4, 'Underweight')
E       AssertionError: assert (18.4, 'Normal Weight') == (18.4, 'Underweight')
E
E         At index 1 diff: 'Normal Weight' != 'Underweight'
E         Use -v to get more diff

bmiTest.py:24: AssertionError
================================== short test summary info ==================================
FAILED bmiTest.py::testBoundaryShift - AssertionError: assert (18.4, 'Normal Weight') == (18.4, 'Underweight')
================================== 1 failed, 5 passed in 0.10s ==================================
smartin2003@StephensLaptop:/mnt/c/Users/Stephen/Desktop/bmiCalc/bmiCalc$
```

"Did your test cases catch this boundary shift problem? Why or why not?"

My test cases did catch the boundary shift problem. This is because the test case's expected output did not match what was actually outputted. The test at 118 lbs (18.4 BMI) was caught being categorized as "Normal Weight" instead of "Underweight". The test at 119 lbs (18.5 BMI) passed. This confirms that the function adheres to the BMI classification rules correctly.

Setup and Execution Instructions

Requirements:
- Windows 10 OS
- Python 3.x (latest version recommended)
- pytest (for testing)

Step 1: Install Python
1. Download Python from python.org.
2. Run the installer and check the box for "Add Python to PATH".
3. Verify installation in the terminal by running python --version.

Step 2: Download Application
1. Go to the Github repository: https://github.com/smartin2003/bmiCalc
2. Click "Code" → "Download ZIP".
3. Extract the ZIP file to a folder.

Step 3: Install Pytest
1. Navigate to the extracted folder.
2. Run pip install pytest in the terminal.

Step 4: Run Application
   BMI Calculator
1. Run using:
      python bmiCalc.py or python3 bmiCalc.py
2. Enter height (inches) and weight (pounds).
3. The program will output the BMI and category.

Testing
1. Run using:
   pytest bmiTest.py
2. If successful, all test cases should pass.

Application Example Screenshot

```
smartin2003@StephensLaptop:/mnt/c/Users/Stephen/Desktop/bmiCalc/bmiCalc$ python3 bmiCalc
.py
Welcome to the BMI Calculator
What is your height in inches? 68
What is your weight in pounds? 100
Your current BMI is 15.6 which would be categorized as Underweight.
smartin2003@StephensLaptop:/mnt/c/Users/Stephen/Desktop/bmiCalc/bmiCalc$ python3 bmiCalc
.py
Welcome to the BMI Calculator
What is your height in inches? 68
What is your weight in pounds? 150
Your current BMI is 23.4 which would be categorized as Normal Weight.
smartin2003@StephensLaptop:/mnt/c/Users/Stephen/Desktop/bmiCalc/bmiCalc$ python3 bmiCalc
.py
Welcome to the BMI Calculator
What is your height in inches? 68
What is your weight in pounds? 190
Your current BMI is 29.6 which would be categorized as Overweight.
smartin2003@StephensLaptop:/mnt/c/Users/Stephen/Desktop/bmiCalc/bmiCalc$ python3 bmiCalc
.py
Welcome to the BMI Calculator
What is your height in inches? 68
What is your weight in pounds? 250
Your current BMI is 38.9 which would be categorized as Obese.
smartin2003@StephensLaptop:/mnt/c/Users/Stephen/Desktop/bmiCalc/bmiCalc$ |
```