

## Lesson 04 Demo 16

### Debugging the Code Using Codeium

**Objective:** To identify and correct errors in JavaScript code using the Codeium debugging feature, which includes creating a sample file, running Codeium error detection, and applying the suggested fixes

**Tools required:** Visual Studio Code and Codeium extension

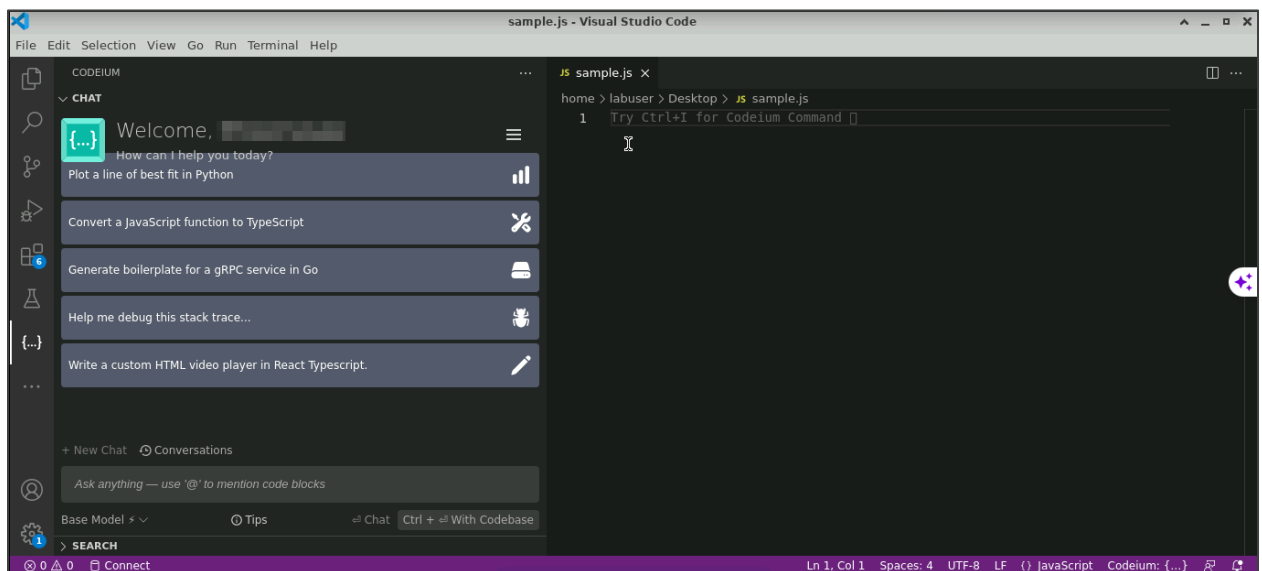
**Prerequisites:** Install the Codeium extension in VS Code (Refer to the Demo: Installing Codeium in VS Code)

Steps to be followed:

1. Create a sample JavaScript file
2. Debug the code using Codeium

#### Step 1: Create a sample JavaScript file

##### 1.1 Create a new JavaScript file named **sample.js**



1.2 Enter the following JavaScript code to run the sample.js file:

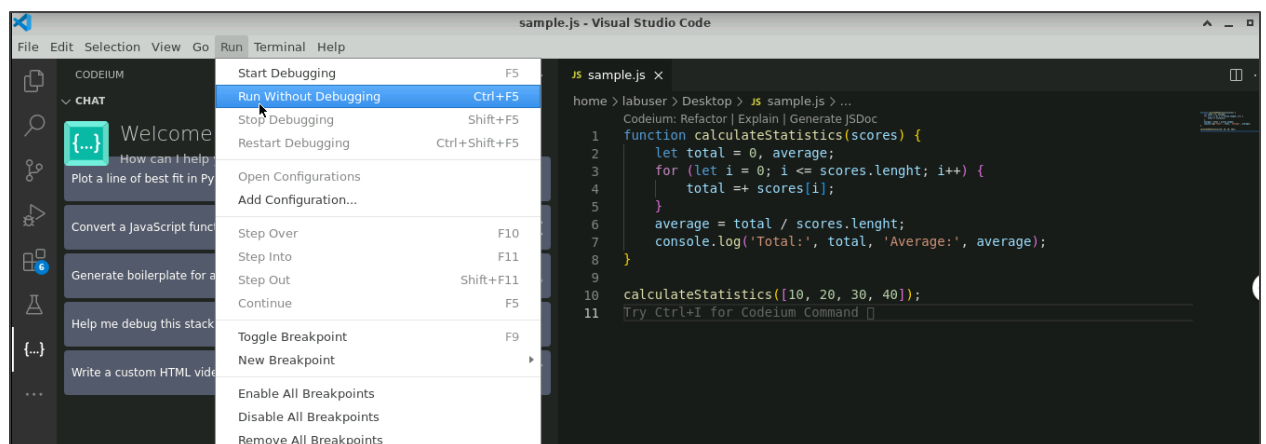
```
function calculateStatistics(scores) {
  let total = 0, average;
  for (let i = 0; i <= scores.length; i++) {
    total += scores[i];
  }
  average = total / scores.length;
  console.log('Total:', total, 'Average:', average);
}
```

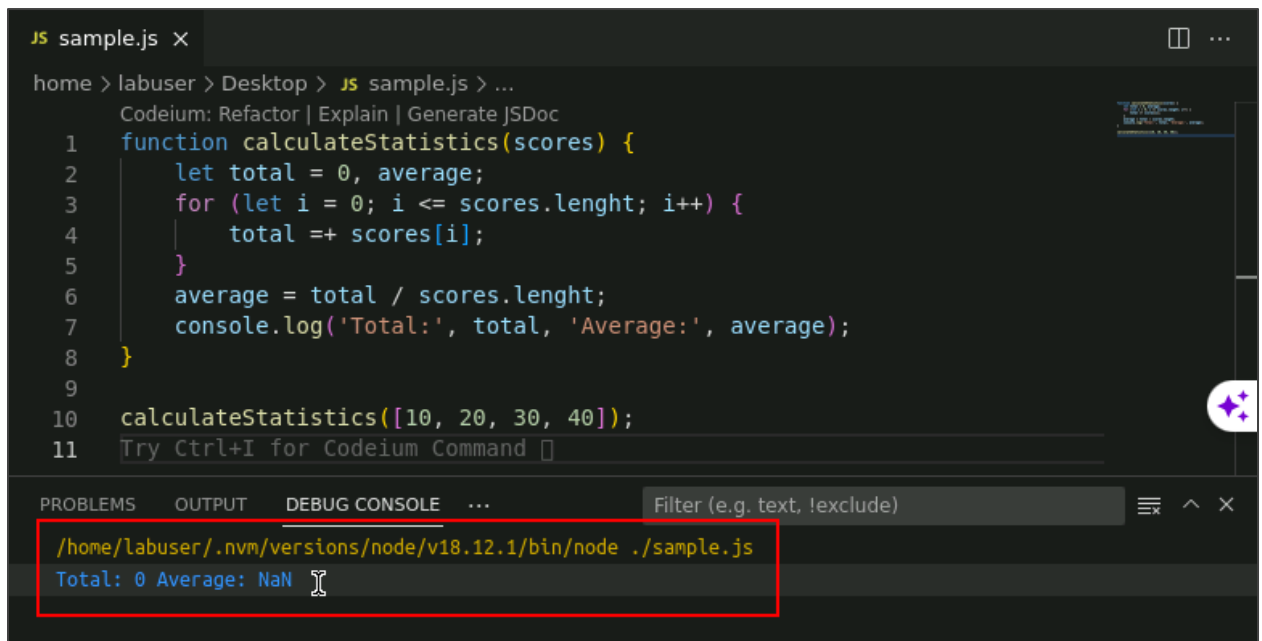
```
calculateStatistics([10, 20, 30, 40]);
```

The screenshot shows a code editor with a dark theme. The file is named 'sample.js'. The code is as follows:

```
home > labuser > Desktop > JS sample.js > ...
Codeium: Refactor | Explain | Generate JSDoc
1 function calculateStatistics(scores) {
2   let total = 0, average;
3   for (let i = 0; i <= scores.length; i++) {
4     total += scores[i];
5   }
6   average = total / scores.length;
7   console.log('Total:', total, 'Average:', average);
8 }
9
10 calculateStatistics([10, 20, 30, 40]);
11 Try Ctrl+I for Codeium Command
```

1.3 Save and run the code to check the output





```
JS sample.js x
home > labuser > Desktop > JS sample.js > ...
Codeium: Refactor | Explain | Generate JSDoc
1 function calculateStatistics(scores) {
2   let total = 0, average;
3   for (let i = 0; i <= scores.length; i++) {
4     total += scores[i];
5   }
6   average = total / scores.length;
7   console.log('Total:', total, 'Average:', average);
8 }
9
10 calculateStatistics([10, 20, 30, 40]);
11 Try Ctrl+I for Codeium Command
```

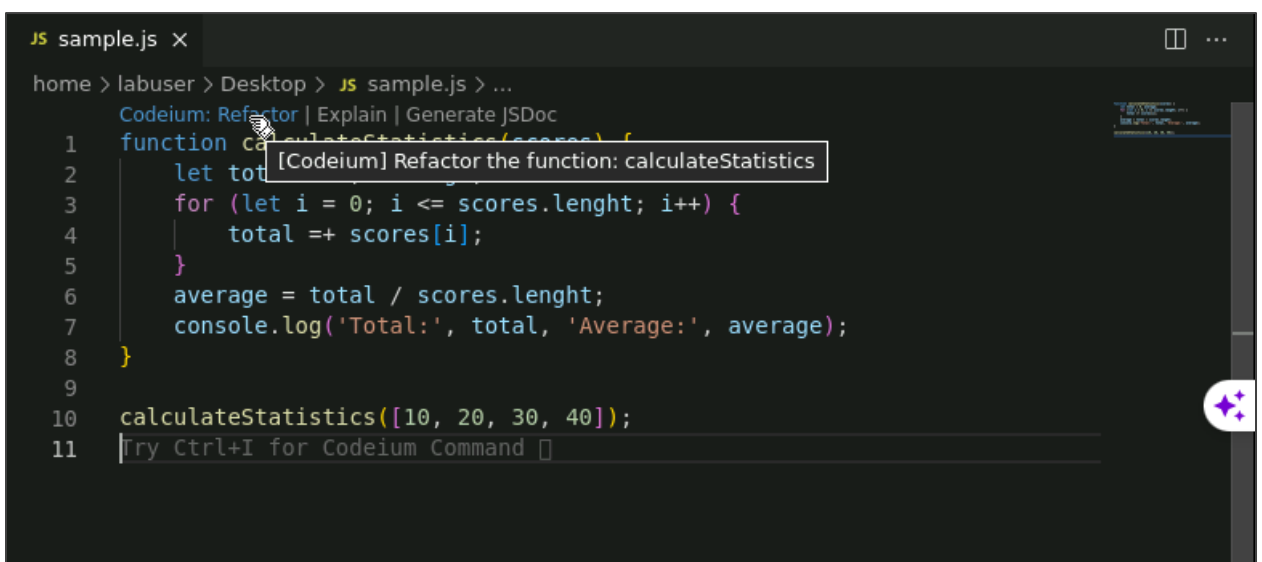
PROBLEMS OUTPUT DEBUG CONSOLE ... Filter (e.g. text, !exclude)

```
/home/labuser/.nvm/versions/node/v18.12.1/bin/node ./sample.js
Total: 0 Average: NaN
```

There is an error in the expected output.

## Step 2: Debug the code using Codeium

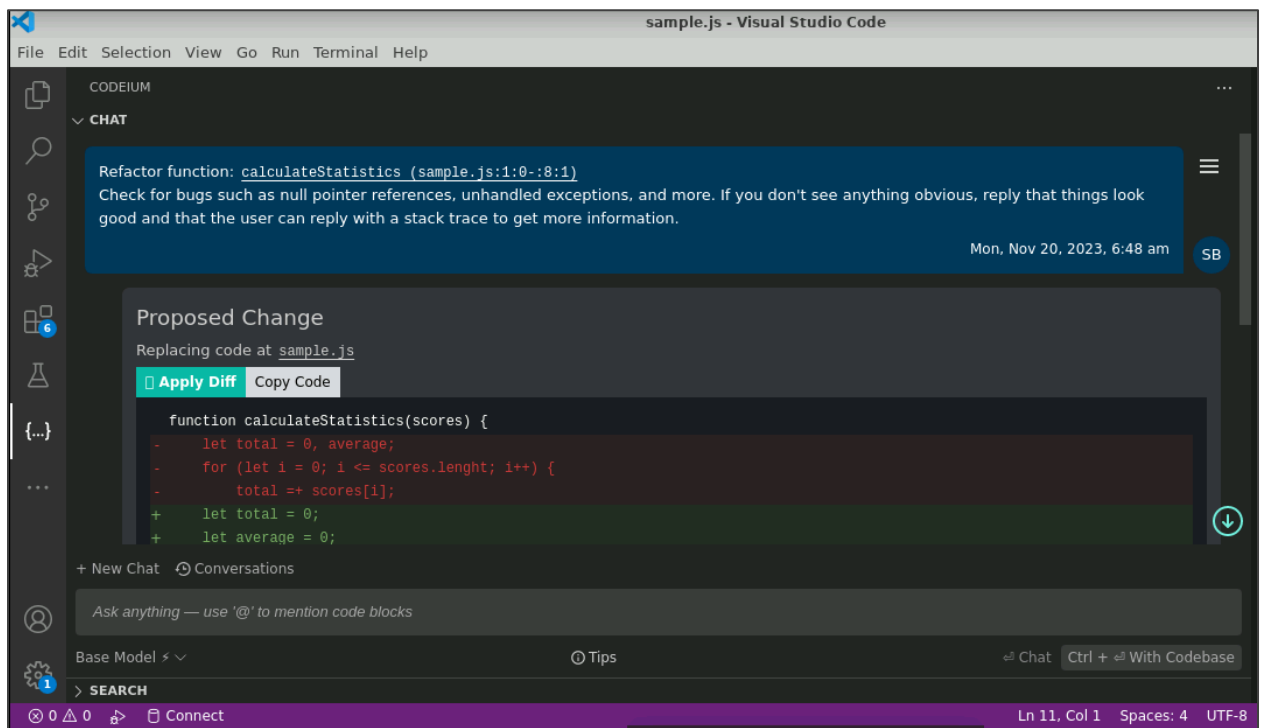
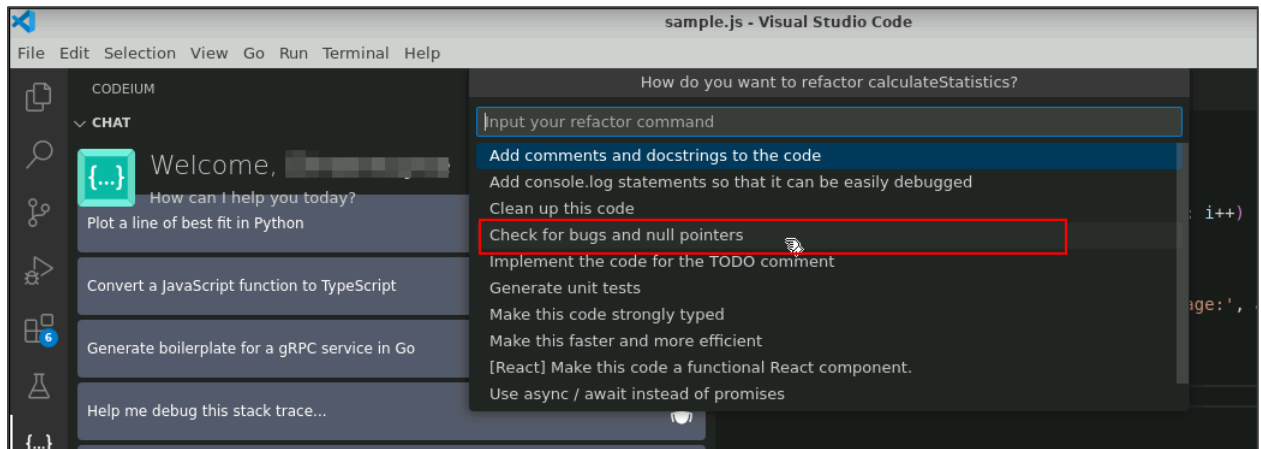
### 2.1 Click on the **Refactor** feature of Codeium

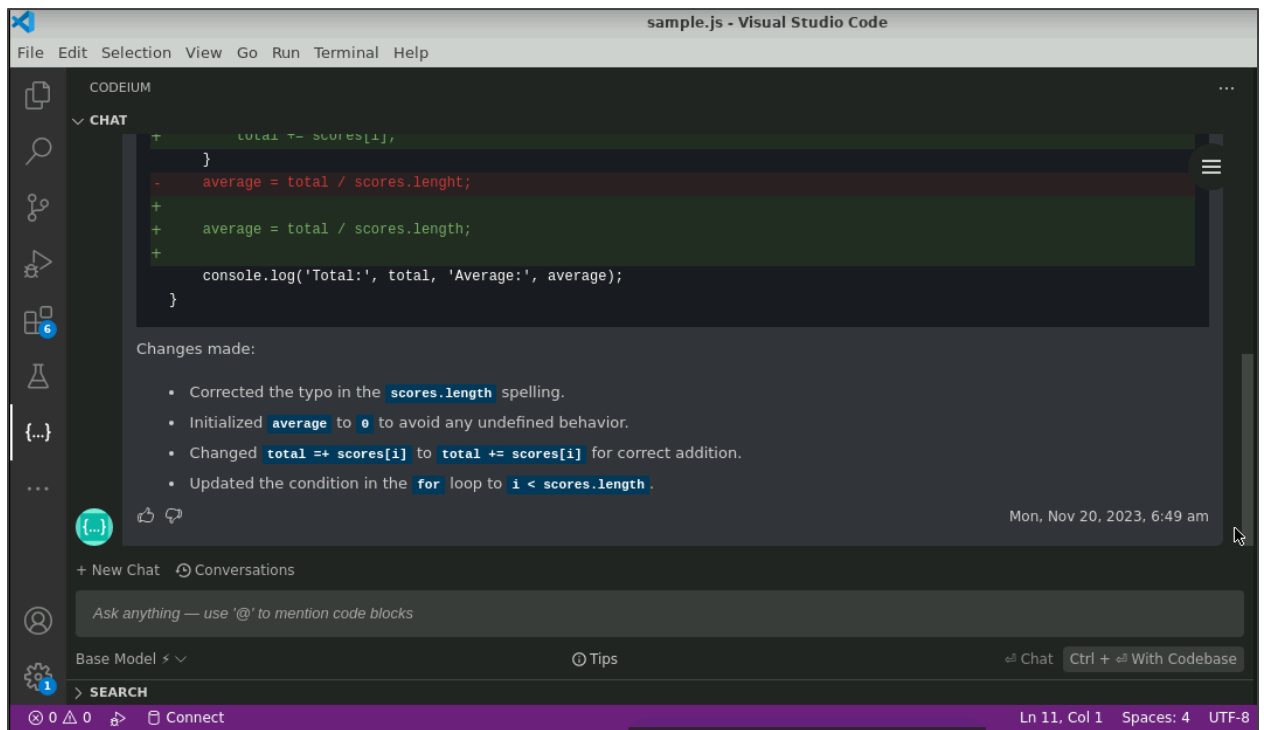


```
JS sample.js x
home > labuser > Desktop > JS sample.js > ...
Codeium: Refactor | Explain | Generate JSDoc
1 function calculateStatistics(scores) {
2   let total
3   for (let i = 0; i <= scores.length; i++) {
4     total += scores[i];
5   }
6   average = total / scores.length;
7   console.log('Total:', total, 'Average:', average);
8 }
9
10 calculateStatistics([10, 20, 30, 40]);
11 Try Ctrl+I for Codeium Command
```

[Codeium] Refactor the function: calculateStatistics

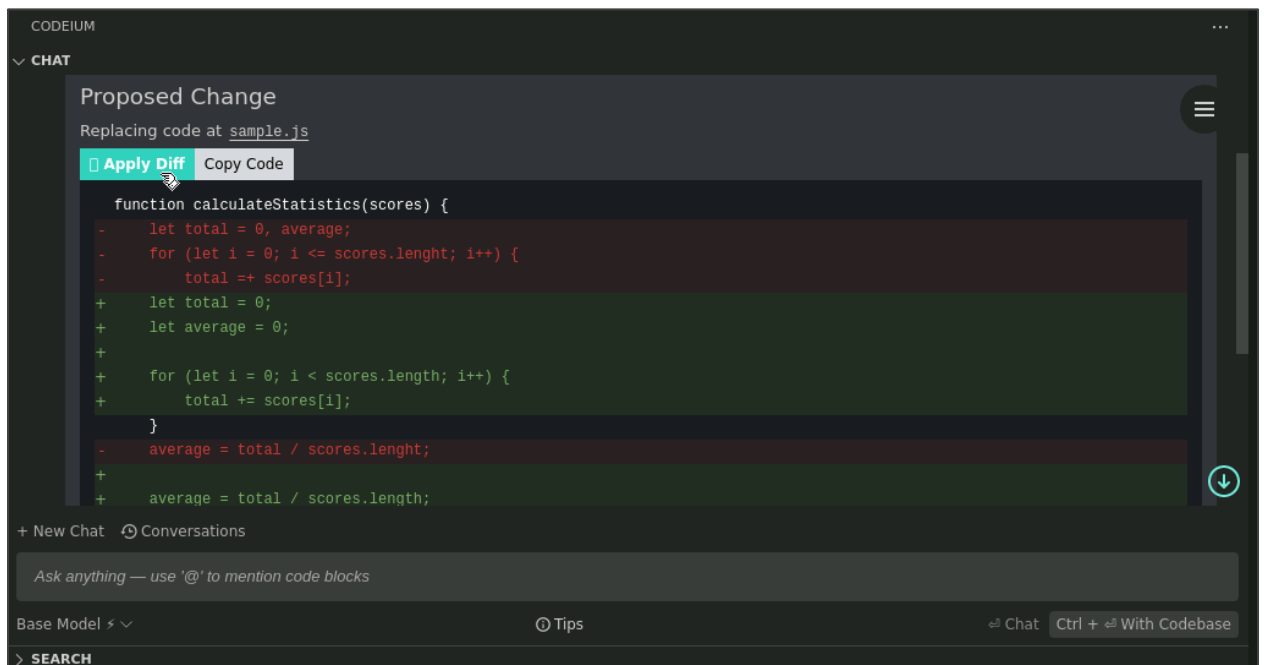
## 2.2 Select Check for bugs and null pointers





Codeium will display the necessary changes to the code.

## 2.3 Click on **Apply Diff** to apply the proposed changes to the code

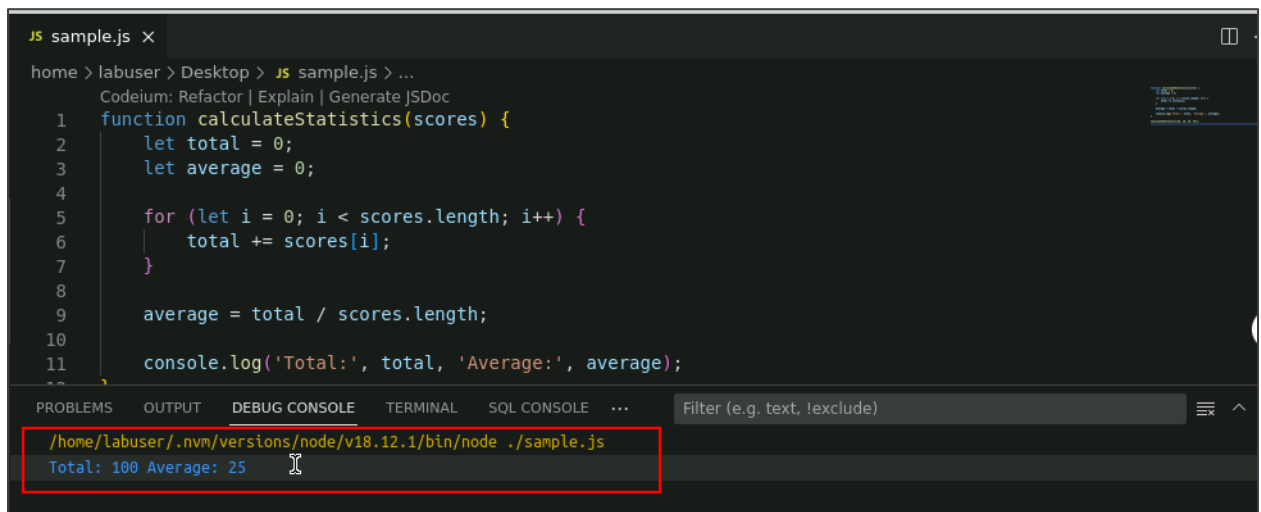


## 2.4 Press **Alt+A** to accept the changes to the code

```
JS sample.js 9+
home > labuser > Desktop > JS sample.js > ...
Accept [ ] (Alt+A) | Reject [ ] (Alt+N)
1 <<<<<<<<<< [ ] Codeium AI Suggestion >>>>>>>>>>>>
2 function calculateStatistics(scores) {
3   - let total = 0, average;
4   - for (let i = 0; i <= scores.length; i++) {
5   -   total += scores[i];
6   + let total = 0;
7   + let average = 0;
8   +
9   + for (let i = 0; i < scores.length; i++) {
10  +   total += scores[i];
11  }
12  - average = total / scores.length;
13  +
14  + average = total / scores.length;
15  +
16  console.log('Total:', total, 'Average:', average);
17  }
18  <<<<< bot-1908f989-b36a-47ad-9053-31f55428b865 >>>>>
19
20 calculateStatistics([10, 20, 30, 40]);
21 Try Ctrl+I for Codeium Command [ ]
```

```
JS sample.js
home > labuser > Desktop > JS sample.js > ...
Codeium: Refactor | Explain | Generate JSDoc
1 function calculateStatistics(scores) {
2   let total = 0;
3   let average = 0;
4
5   for (let i = 0; i < scores.length; i++) {
6     total += scores[i];
7   }
8
9   average = total / scores.length;
10
11  console.log('Total:', total, 'Average:', average);
12 }
13
14 calculateStatistics([10, 20, 30, 40]);
15 Try Ctrl+I for Codeium Command [ ]
```

## 2.5 Save and run the code again to check the output



The screenshot shows the VS Code editor interface. The editor window displays a JavaScript file named `sample.js` with the following code:

```
1 function calculateStatistics(scores) {  
2   let total = 0;  
3   let average = 0;  
4  
5   for (let i = 0; i < scores.length; i++) {  
6     total += scores[i];  
7   }  
8  
9   average = total / scores.length;  
10  
11   console.log('Total:', total, 'Average:', average);  
12 }
```

The bottom panel of the editor shows the **DEBUG CONSOLE** tab. The command `/home/labuser/.nvm/versions/node/v18.12.1/bin/node ./sample.js` has been executed, and the output is displayed as `Total: 100 Average: 25`. The output line is highlighted with a red rectangle.

Now, you get the correct output for the code.

By following these steps, you have successfully identified and debugged the errors in code using the debugging feature of Codeium.