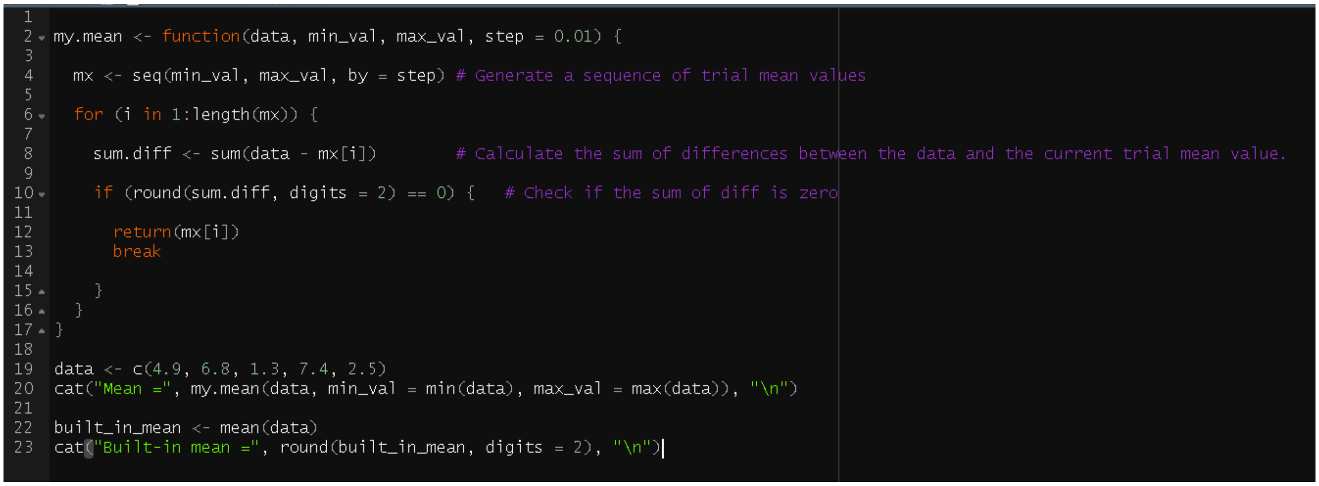
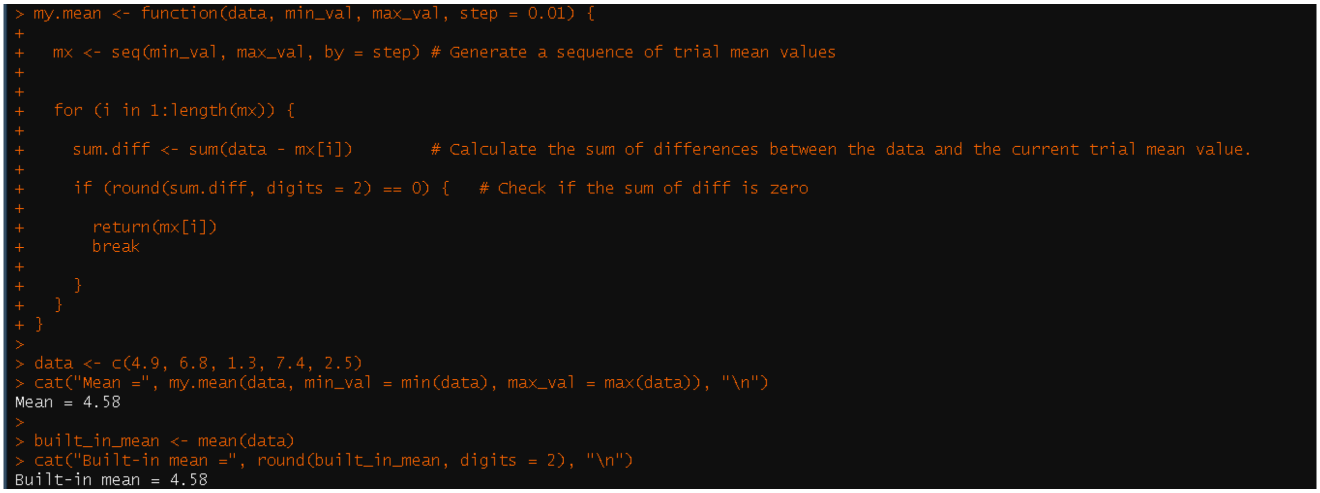
Coding :



```r

my.mean <- function(data, min_val, max_val, step = 0.01) {

  mx <- seq(min_val, max_val, by = step) # Generate a sequence of trial mean values

  for (i in 1:length(mx)) {

    sum.diff <- sum(data - mx[i])        # Calculate the sum of differences between the data and the current trial mean value.

    if (round(sum.diff, digits = 2) == 0) {    # Check if the sum of diff is zero

      return(mx[i])
      break

    }
  }
}

data <- c(4.9, 6.8, 1.3, 7.4, 2.5)
cat("Mean =", my.mean(data, min_val = min(data), max_val = max(data)), "\n")

built_in_mean <- mean(data)
cat("Built-in mean =", round(built_in_mean, digits = 2), "\n")
```

Result :



```
> my.mean <- function(data, min_val, max_val, step = 0.01) {
+ 
+   mx <- seq(min_val, max_val, by = step) # Generate a sequence of trial mean values
+ 
+ 
+   for (i in 1:length(mx)) {
+ 
+     sum.diff <- sum(data - mx[i])        # Calculate the sum of differences between the data and the current trial mean value.
+ 
+     if (round(sum.diff, digits = 2) == 0) {    # Check if the sum of diff is zero
+ 
+       return(mx[i])
+       break
+ 
+     }
+   }
+ }
> 
> data <- c(4.9, 6.8, 1.3, 7.4, 2.5)
> cat("Mean =", my.mean(data, min_val = min(data), max_val = max(data)), "\n")
Mean = 4.58 
> 
> built_in_mean <- mean(data)
> cat("Built-in mean =", round(built_in_mean, digits = 2), "\n")
Built-in mean = 4.58
```

Conclusion :

I made a function to calculate the mean of the dataset using the concept of balancing moments. First, I initialized mx which was a set consisting of all the values from the min value in the dataset to the max value with the step of 0.01. (1.31, 1.32, 1.33, …….7.39, 7.4) Then at each loop, I subtract each value in the data set with the current mx and sum the 5 values up, if the sum is 0 then mx[i] (current mx) will be returned. Then I check the mean from my function with the mean calculated with R’s built in mean. The results were the same.