Programming Exercise 12
Statistical Functions

C# Step by Step

1 Zeroth part, set up

Create some data structure (possibly an integer array or a generic List) containing integers. Integers will be randomized from 0 to 1000, inclusive. The size of the list will be from 10 to 1000, as returned by a call to the Random method. Each time you run the program, it should produce a different list of integers. Print the list as a comma-separated list.

2 First part, mean of list, 50 points

Write a function that takes the list as a parameter, calculates the mean of the list, and returns the mean. You have already done this in a prior exercise. The *mean* is simply the average of the numbers in the list. Add all the integers together and divide the sum by the number of items in the list. The formula is as follows:

$$\text{mean}(X) = \frac{\sum_{i=0}^{n}(x_i)}{n} \tag{1}$$

where X is the list of numbers, x_i is each individual integer, and n is the count of integers in the list.

3 Second part, median of list, 60 points

Write a function that takes a list as a parameter, calculates the median of the list, and returns the median. Please note that you must use a *sorted* list to calculate the median. You can use the built-in `Sort` methods — I don't expect you to write a sorting method. The *median* is simply the midpoint of the numbers in the list. For example, the median of an odd numbered list is the middle value, for (1, 2, 3, 4, 5) the median is 3. The median of an even numbered list is the average of the two middle values, for (1, 2, 3, 4, 5, 6) the median is 3.5. The formula is as follows:

$$\text{median}(X) = \frac{x_{[1+m]} + x_{[L-m]}}{2} \tag{2}$$

where X is the list of numbers, x_i is the *ith* element of the list, n is the length of the list, and m is the integer portion of L divided by 2.[1] Please note that you must use a sorted list for this calculation.

DO NOT MAKE A **FENCE POST** ERROR! Remember that Lists and arrays are zero based. Make sure you test your method with lists that you can calculate by hand and that your method returns the correct value.

[1] In C#, use integer division.

4 Third part, standard deviation, 70 points

Write a function that takes the list as a parameter, calculates the standard deviation of the list, and returns the standard deviation. The *standard deviation* is the average distance of each value from the mean. To calculate the standard deviation, subtract each value from the mean, square the result (which makes all the results positive), and sum the results. Then, divide the summation by the length of the list, and take the square root. The formula is:

$$\text{sd}(X) = \sqrt{\frac{\sum_{i=0}^{n}(x_i - \text{mean}(X))^2}{n}} \tag{3}$$

where X is the list of numbers, x_i is each value and n is the length of the list.

5 Fourth part, median absolute deviation, 80 points

The *median absolute deviation* is the median of the absolute values of the difference between each individual value and the median value. To calculate, first determine the median value, then determine the absolute difference between each value and the median, obtaining a list of differences. Sort the resulting list, and then determine the median of the list of medians. As always, when calculating the median, sort the list first. The formula is:

$$\text{mad}(X) = \text{median}(|x_i - \text{median}(X)|) \tag{4}$$

where X is the list of numbers and x_i is each value. Note two tricky points. First, the term $x_i - \text{median}(X)$ returns the *absolute value*, so that the result is always positive. Second, both calculations of the median take sorted lists. In other words, you need an original sorted list to calculate the inner median. Then, you need to sort the lists of medians to calculate the outer median.

6 Fifth part, skewness, 90 points

Skewness is a measure of the "lopsidedness" of the list. A list that has approximately the same values on either side of the median has a low skewness. A list that does not have approximately the same values on either side of the median has a large skewness. A positive skewness indicates that the distribution is skewed to the right. A negative skewness indicates that the distribution is skewed to the left. A skewness of zero indictes that the distribution is not skewed. Skewness is important because it determines the effectiveness of mean and standard deviation as measures of center. For distributions with low skewness, mean and standard deviation are valid measures. For distributions with high skewness, median and median absolute deviation are valid measures. The formula is as follows:

$$\text{kurtosis}(X) = \frac{\frac{\sum_{1=0}^{n}(x_i - \text{mean}(X))^3}{\text{sd}(X)^3}}{n} \tag{5}$$

with mean, median, and standard deviation calculated as described above.

7 Sixth part, kurtosis, 100 points

Kurtosis is a measure of the "flatness" of the distribution. A kurtosis of zero means that the distribution has a normal shape. A positive kurtosis indicates a flatter distribution. A negative skewness indicates a "pointier" distribution. The formula is as follows:

$$\text{kurtosis}(X) = \frac{\frac{\sum_{1=0}^{n}(x_i - \text{mean}(X))^4}{\text{sd}(X)^4}}{n-3} \tag{6}$$

where is n is the length of the list, X is the list, and x_i represents each individual item.

8 Helps

My main program is below. You may use it if you want, or improve it. Lines 6 and 7 repressent my test arrays; I had more but I deleted them as I worked on my program. Figure 1 shows a sample run of my program.

```
1  class Program
2  {
3      static void Main(string[] args)
4      {
5          int[] testarr = Util.CreateTest();
6          //int[] testarr = {  1, 1, 1, 1, 1, 1, 1, 1, 1, 1};
7          //int[] testarr = { 1, 2, 3, 4, 5, 4, 3, 2, 1 };
8          Util.PrintArray(testarr);
9
10         Console.WriteLine("Calculate the mean:");
11         double mean = Util.CalcMean(testarr);
12         Console.WriteLine($"The mean is {mean}.");
13
14         Console.WriteLine("Calculate the median:");
15         double median = Util.CalcMedian(testarr);
16         Console.WriteLine($"The median is {median}.");
17
18         Console.WriteLine("Calculate the standard deviation:");
19         double sd = Util.CalcSD(testarr, mean);
20         Console.WriteLine($"The standard deviation is {sd}.");
21
22         Console.WriteLine("Calculate the median absolute deviation:");
23         double mad = Util.CalcMAD(testarr, median);
24         Console.WriteLine($"The median absolute deviation is {mad}.");
25
26         Console.WriteLine("Calculate the skewness:");
27         double skew = Util.CalcSkew(testarr, mean, sd);
28         Console.WriteLine($"The skew is {skew}.");
29
30         Console.WriteLine("Calculate the kurtosis:");
31         double kurt = Util.CalcKurt(testarr, mean, sd);
32         Console.WriteLine($"The kurtosis is {kurt}.");
33     }
34 }‘
```

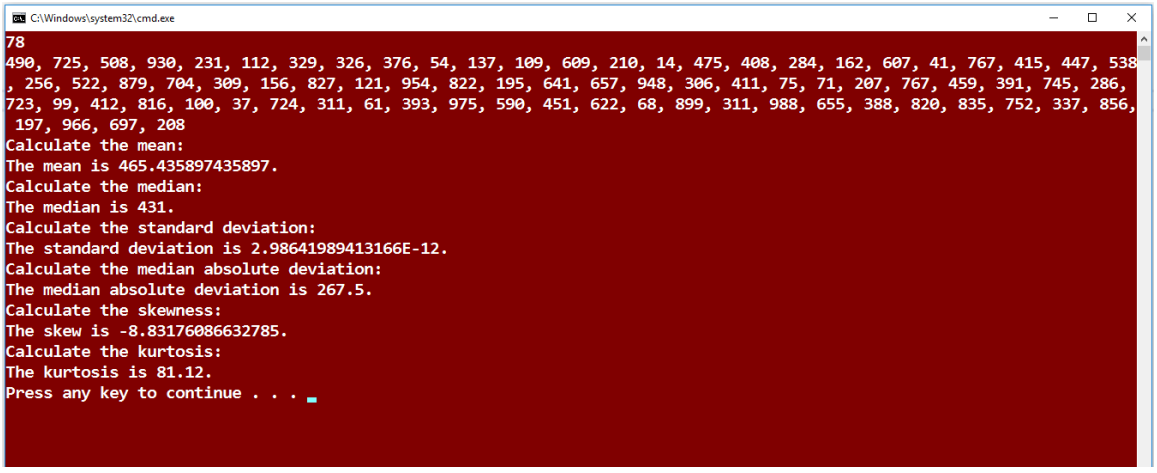

Figure 1: Example output for the stats program