Logo and Measures of Central Tendency

W. George Cathcart
University of Alberta

Measures of central tendency such as the arithmetic mean, median, and mode are commonly taught in upper elementary and junior high school grades. Each of these statistics provide a different view of central tendency. The **arithmetic mean** is commonly referred to as the "average" by most students. The calculation of the mean is relatively straightforward.

The **median** is the midpoint of a set of scores; the point above and below which one-half of the scores fall. The calculation of the median is a little more involved than the calculation of the mean. Given the data, 18, 14, 12, 14, 27, 20, 30, 21, 14, it is not immediately apparent what the median is. When the data is sorted, 12, 14, 14, 14, 18, 20, 21, 27, 30, then it is obvious that 18 is the midpoint with four scores on each side. If, however, the 20 was eliminated from the list there would no longer be a specific score (data piece) that would serve as the midpoint.

12 14 14 14 ↑ 18 21 27 30

The point of balance is now midway between 14 and 18. That is, (14 + 18)/2.

The **mode** is the most frequently occurring value. It is probably the easiest of the three measures of central tendency to teach to children. Frequently children are asked to draw a graph (histogram) of the data. When this is done the mode is the "highest bar." The fact that some sets of data may have more than one mode adds a little difficulty to children understanding it.

Logo Procedures for Central Tendency

The activities and Logo procedures described in the remainder of this article assume that students have a reasonably good understanding of Logo, including experience with the basic list processing commands and operations. All sample procedures are written in Apple Logo (LCSI).

Writing Logo procedures to calculate the mean, median, and mode requires extensive use of some of the list processing features of Logo. Such an exercise will be a good demonstration for your students of some of the non-graphic uses of Logo.

Mean

A Logo procedure to calculate the mean is reasonably easy to write. Two steps are required: finding the sum and determining the number of scores in a set of data. Most versions of Logo have a primitive, SUM, which will calculate the sum of only two numbers. The first assignment you might give your students would be to write a procedure to calculate the sum of a list of numbers with any number of entries.

ACTIVITY 1 **LOGO SUM**

Write a Logo procedure which will output (use the OUTPUT command) the sum of a list of numbers.

Elementary students or older students with minimal experience using Logo could be provided with a procedure that they could use as a primitive. You should write your own to give your students--but if you prefer you can use the following procedure.

```
TO ADDUP :ALIST
IF :ALIST = [] [OUTPUT 0]
OUTPUT SUM FIRST :ALIST ADDUP BUTFIRST :ALIST
END
```

Apple Logo has a COUNT primitive which returns the number of elements in a list. (If your version does not have COUNT you will have to write a short procedure to count the items in a list.) The COUNT primitive gives us the number to divide the sum by in order to obtain the mean. Assign activity 2 at this point.

ACTIVITY 2 **LOGO MEAN**

Write a Logo procedure which will output (use OUTPUT) the mean of a set of numbers.

a) Use the procedure you wrote for activity 1 (or the procedure given you by your teacher) to find the sum.
b) Use the COUNT primitive to obtain the divisor.
c) Use the following form for your procedure:

```
TO MEAN :ALIST
OUTPUT _____
END
```

d) Test your procedure using 12, 15, 20, 8, 19.

Median

To calculate the median the data should be sorted first. Only students with a good understanding of Logo will be able to write a procedure to sort data. You will likely want to give your students a sorting procedure which they can use as a primitive. Here is a procedure that sorts data into ascending order.

```
TO SORT :DATA
IF EMPTYP :DATA [OUTPUT []]
OUTPUT NEWLIST FIRST :DATA SORT BUTFIRST :DATA
END

TO NEWLIST :ELEMENT :DATA
IF EMPTYP :DATA [OUTPUT (LIST :ELEMENT)]
IF :ELEMENT < FIRST :DATA [OUTPUT FPUT :ELEMENT :DATA]
    [OUTPUT FPUT FIRST :DATA NEWLIST :ELEMENT BUTFIRST :DATA]
END
```

Next, we need to determine whether the number of entries in the data list is odd or even. This can easily be done in Logo with the REMAINDER and COUNT operations.

ACTIVITY 3 **ODD/EVEN**

Write a Logo statement which will determine whether the number of entries in a list is odd or even.
HINT: Experiment with REMAINDER and COUNT.

If the number of scores is odd, we have to pick out the middle score from the sorted list. If the number of scores is even, we have to find the two middle scores and compute the mean of these two scores. The Logo operations, COUNT, ITEM, and QUOTIENT, enable us to do these tasks relatively easily.

ACTIVITY 4 **MIDDLE NUMBER**

Write a Logo statement which will pick out the middle number from a sorted list that has an odd number of elements.
Example: Pick out the middle number, 17, in 6, 12, 17, 22, 29.
HINT: ITEM 3 :ALIST will pick out the third item in the list called ALIST.
HINT: Experiment with QUOTIENT.

ACTIVITY 5 **MIDDLE NUMBER**

Write a Logo statement which will calculate the middle value for a set of data that has an even number of elements.
Example: Find the middle value, 11 in the list 5, 9, 13, 21.

ACTIVITY 6 **MEDIAN**

Use the statements you wrote for activities 3, 4, and 5 to write a procedure which will output (use OUTPUT) the median of any set of scores.

For activity 6, students may end up with a procedure something like this:

```
TO MEDIAN :ALIST
MAKE "SORTED.LIST SORT :ALIST
TEST REMAINDER COUNT :ALIST 2 = 0
IFFALSE [OUTPUT ITEM 1 + QUOTIENT COUNT :ALIST 2 :SORTED.LIST]
IFTRUE [OUTPUT MEAN LIST (ITEM (COUNT :ALIST)/2 :SORTED.LIST)
     ITEM 1 + (COUNT :ALIST)/2 :SORTED.LIST]
END
```

Mode

While the mode may be the easiest of the three measures of central tendency to teach to children, it is probably the most difficult to write a Logo procedure for. It is recommended that you assign the mode as a challenge for your "better" programmers.

There may be a number of equally good strategies to use to get the mode. Here is one strategy that works.

1. Arrange the data into ascending order.
2. Create a list of all the unique data pieces. For example, if the sorted data was 9, 14, 14, 14, 17, 17, 20, 20, 20, then the unique list would contain 9, 14, 17, 20.
3. Count the frequency in the data list of each value in the unique list. We now have two lists in one-to-one correspondence.
   Unique list: [9 14 17 20]
   Frequency list: [1 3 2 3]
4. Find the maximum value in the frequency list.
5. Print all the values from the unique list which correspond to the maximum frequency.

ACTIVITY 7 **UNIQUE ENTRIES**

Write a Logo procedure which will create a list with all the unique elements from a set of data.

Example: 12, 18, 12, 14, 18, 21, 18
Your unique list will contain only 12, 14, 18, 21.

ACTIVITY 8 **FREQUENCY COUNT**

Write a Logo procedure which will count the frequency of occurrence of each value in a set of data.

Example:
Data: 12, 12, 14, 18, 18, 18, 21
Frequencies: 2, 1, 3, 1

ACTIVITY 9 **MAXIMUM VALUE**

Write a Logo procedure which will find the largest value in a list.

ACTIVITY 10 **MODE**

Use the procedures you developed for activities 7, 8, and 9 as subprocedures in a superprocedure which will print all the modes in a set of data.

You may need to modify your procedures slightly so that they will work together.

The following superprocedure for the mode is an example of what students may end up with.

```
TO MODE :ALIST
MAKE "SORTED.LIST SORT :ALIST
MAKE "U.LIST []   (To contain the unique elements.)
MAKE "F.LIST []   (For the frequencies of each unique element.)
PRINT []
TYPE [THE MODE(S) IS (ARE):]
TYPE CHAR 32
MAKE "MAX FIND.MAX COUNT.FREQ :SORTED.LIST GET.UNIQUE :SORTED.LIST
GET.ALL.MODES :U.LIST :F.LIST
PRINT []
END
```

GET. UNIQUE takes a sorted list (ascending order) as input and outputs a list, U.LIST, of the unique elements in the list. COUNT.FREQ. then takes this U.LIST and the sorted list as inputs and outputs a list, F.LIST, of the frequencies of each data value. This list is then used as input to FIND.MAX which determines the largest frequency. Finally, GET.ALL.MODES compares each entry in the list of frequencies with the maximum. When a frequency is encountered which equals the maximum, the corresponding entry from the unique list, U.LIST, is printed.

Writing the subprocedures GET.UNIQUE, COUNT.FREQ., FIND.MAX, and GET.ALL. MODES is left as a challenge for you, the teacher, as well as for your students. Better yet, see if you can design a more elegant strategy for finding the mode using Logo.

Dr. Cathcart is a Professor in the Department of Elementary Education, University of Alberta. George has served as Delta-K editor, president of MCATA, and as a member of the executive. George invites you to compare procedures with him.