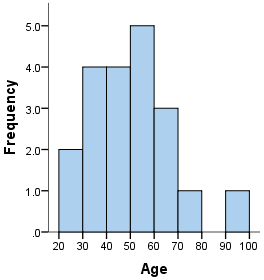
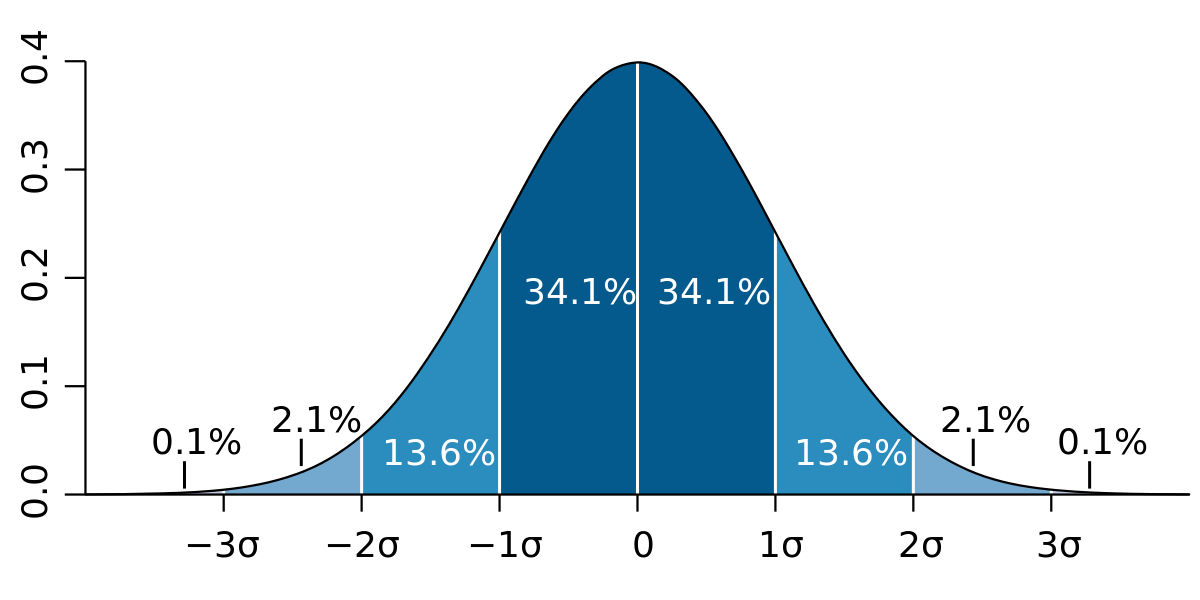
Ethan Seow Planning

Layout of the Roll Dice Project



The histogram is the basis of the frequency table and this project



Standard Deviation after numerous dice rolls, the dice roll values in the middle have the greatest frequency

Functionality:

The user has to input a number of dice and a number of rolls

* Here I can use a form to input data into the backend variables
* I can store the form information into a variable called diceRolls and numOfDice
* After the user inputs the data, the form is cleared

Simulating the rolls

* There will be one for loop to simulate the number of rolls
  + For(var rolls = 0 ; rolls < number of rolls; rolls++)
* Then there will be another for loop inside that loop to simulate the number of dice to be rolled
  + For(var rolls = 0 ; rolls < number of rolls; rolls++)
    - For( var dice = 0; dice < numOfDice; dice++)
* Then inside that loop there will be a variable that stores a random number from 1 – 6 to simulate the roll of a dice

To retrieve the sum and frequency of data

* After every dice roll, add the roll sum to an array called rangeArray that keeps track of it. This can be achieved by getting the number of dice times 6 for the maximum roll amount, and dice times 1 for the minimum roll amount. The index of the array will represent the lower limit and upper limit, and the value at the index will represent the how many times this value has been rolled.
* Also after every dice roll, add the value of the sum of the dice rolls into an array called numbersArray This will later be used to find the median, mean, and mode of the data set

To retrieve the doubles and triples

* After a single dice roll in the for loop dice that represents it, store that single dice roll value into an array. Then after every single dice roll, check the previous elements in the array to see if it matches. If it matches, increase a counter. After 2 increments, the counter should be at 2, representing a triple. After 1 increment, the counter should be at 1, representing a double. The counter and array will reset after ever for dice loop has ended

To get the mean, median, mode

* Sort the numbersArray from least to greatest first
* Then to get the mean, add up all the numbers in the array then divide by the number of elements
* To get the median, store a variable called medianNumber. This median number will be the number of elements in the numbersArray divided by 2 and rounded down. It will be rounded down because when the number of elements are even, for example, 2,4,5,6, we need to get the 2 middle elements(4 and 5), starting with the element that appears first(4). Then, after storing the variable, check if the number of elements in the numbersArray are even or odd. If they are odd, simply return the medianNumber index of the numbersArray. However, if it is even, for example 2,4,5,6, get the element at the medianNumber index of the numbersArray and add that to the next element, in this case 4 + 5. Then divide that number by 2, 4.5.
* To get the mode, iterate through the rangeArray, aka the frequency table and find the largest element. Then return the element corresponding to the actual value

Converting the JavaScript arrays to html table cells

* Create a display function that gets the element of the statistics table in the rangeArray and for each element in the array, create a cell that has its innerHTML equal to the element value of the rangeArray.
* Then after that create a new row in that table and insert cells to represent the number of doubles, triples, mean, median, and mode.
* Create another display function that gets the elements of the numbersArray, and iterate through it, inserting cells into a row of another table that contain the element values of the numbersArray