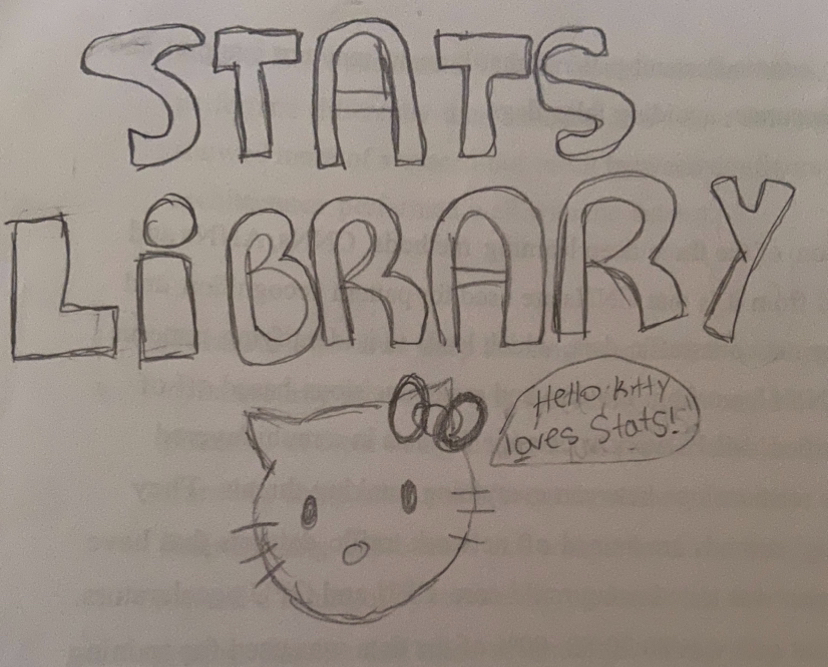
Welcome to my Stats library!



Please open if you haven’t: (https://byronprojectone.carrd.co/)

First open the folder “StatsLibrary” in the IDE of your choosing.

To use the Stats library, the user must insert values into the printlines in the “StatslibraryTester” class.

**Here you can see it computing the mean, median, mode, sdv, and variation of two Arraylists above.**

A screenshot of a computer screen

Description automatically generated

**The location of the coded methods for these is available in MeanMedMode.java.**

However, the code will be presented below. If you so choose, not to open the program.

MEAN METHOD

A computer screen shot of text

Description automatically generated

So basically, this method is taking the parameter (listOfNumbers) which is an arraylist with integers, we then use that array list to find the average. The loop counts all the numbers up (finding the sum) and then the return statement divides the sum of the numbers over the amount of numbers. So here, “listOfNumbers.size()” would give us the size of the array list… or in other words, how many number overall we have.

MEDIAN METHODA computer screen shot of a program

Description automatically generatedHOD

The median method takes the parameter numberList in which is an array list filled with integers. It makes sure to sort the list just in case using collections. The size variable holds the array list’s size. To find the two events, one being odd and one being even, we have a special if statement. The first part gives us if it’s odd, second if even. If the median is odd, the value is the middle number since it evens out. If it is even, there are two numbers left over. They are added together and divided by two. Then we return the median value.

MODE METHOD

A screenshot of a computer program

Description automatically generated

Basically, our parameter is the same – an array list. We are manipulating this array list to find values that are the same which would output us the mode. Within the for loops, this is being done as it understands the size of the array list and the values within it, counting each time there’s a same value. So, the return statement was cut off, but the mode is just returned that the end. The mode can be defined as the number which appears most. For example, in {5, 2, 3, 5}, 5 would be the mode.

VARIANCE METHOD

A computer screen shot of a program

Description automatically generated

Here is variance. The parameter is another array list! So, this could be within standard deviation (it still is but called by this method), but I thought what if someone wanted just the variance. Why? I mean, maybe they do. This method is just using the mean method to give us the mean, inputting that into the for-loop’s body, meaning the number minus mean to the power of 2. It’s the formula, but the for loop does the heavy work. Then we return the variance.

And lastly, from that class, we have the Standard Deviation method.

A computer screen shot of a program

Description automatically generated

Taking the famous array list in as a parameter, it basically calls the variance from the method… It takes the square root of that existing number over how many numbers in the list to give us the standard deviation. The variance method did all the hardcore work. This is the population version but in the sample version, it would be the amount of numbers minus one on the bottom.

**Then, we have set operations such as union, intersection and the complement.**

**These can be found in SetOperations.java.**

**Next, we have the combination and permutation class, which features factorials! (in its own method).**

A black background with white text

Description automatically generated

**The location of said methods is in ComboPerm.java.**

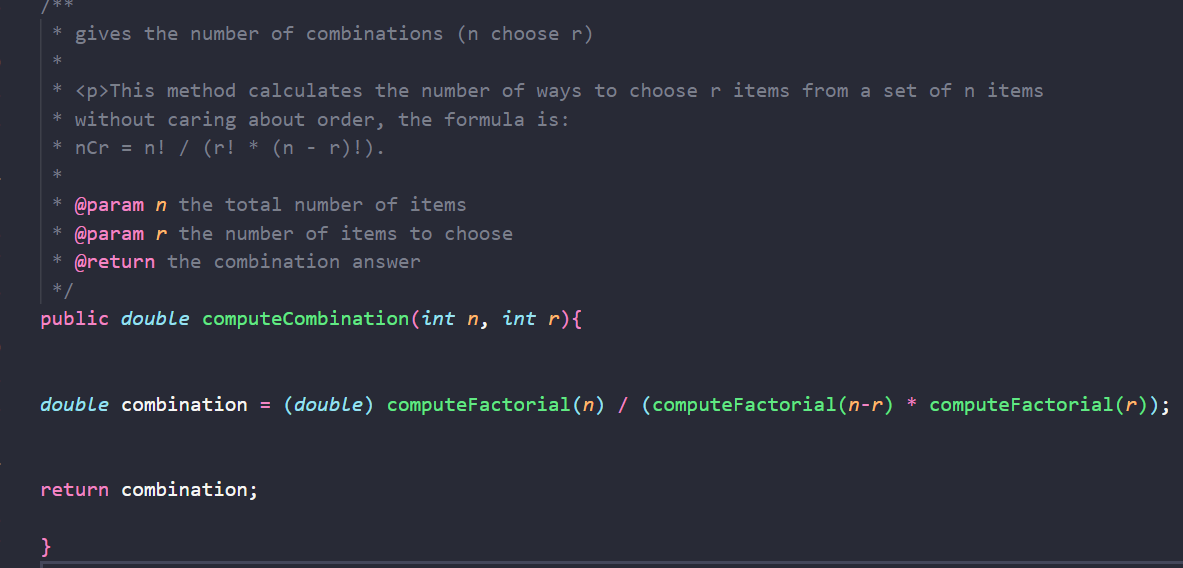
FACTORIAL METHOD

A screenshot of a computer program

Description automatically generated

In the factorial method, it takes in an integer as the parameter. After that, it uses a loop to make sure the inputted number is dealt with properly. The number will be multiplied by everything less than it until hitting one, then stops after. It then returns that number

COMBINATION METHOD



The combination method takes n and r as parameters. It then computes the formula which uses the factorial method within it. It computes: the factorial of n divided by the factorial of (n-r) times the factorial of r. Then, it returns the final value as a double.

PERMUTATION METHOD

A computer screen shot of a computer code

Description automatically generated

It’s very similar to the combination one except it’s the permutation formula being expressed. It takes n and r in as well and uses the factorial method within the formula. It is the factorial of n over the factorial of (n-r). That is then returned.

**Next, we have conditional, bayes and independence check.**

**A computer screen shot of a math problem

Description automatically generated**

**You can find them in Conditional.java.**

**A computer screen shot of text

Description automatically generated**

This method takes in the probability of A and B as a parameter. It returns Baye’s theorem which is PA \* BA divided by PB. BA can be found from the probability of a and b divided by the probability of b.

**A computer screen with text

Description automatically generated**

Conditional probability takes in probability of A and B as parameters. It returns the formula for conditional probability by multiplying the values to get PANB and returns it as a double value (the percent).

**A computer screen shot of text

Description automatically generated**

This is just a series of if statements that check the independence rules. If it doesn’t meet them, then it’s dependent. The parameters are just the probability of A, B and A and B. Then it returns true or false based on if the parameters included end up independent.

**Next, we have Binominal and geometric distributions. Those being the only distributions we have done so far. (Thankfully, they are easy to mix up.)**

**A screenshot of a computer

Description automatically generated**

**You can find the code for these in distributionTypes.java.**

BINOMINAL DISTRIBUTION METHOD

A screenshot of a computer code

Description automatically generated

The binominal distribution method takes in the probability (p), the number of trials and the value x into it as parameters. It uses the combination method (specified before in another class) to take the combination of trials and x. Then, it returns the formula to binominal distribution using Math.pow which helps square the values to each other, like the formula.

GEOMETRIC DISTRIBUTION

A screenshot of a computer program

Description automatically generated

So, this method takes in the probability ‘p’, ‘n’ number of trials and X which is the trial where the first success is talked about as the parameters. Then there’s an if-statement which figures out if X equals to n or if less than or equal to n, and that determines which formula is used, returning 0 if no condition is met. It returns the geometric distribution formula.