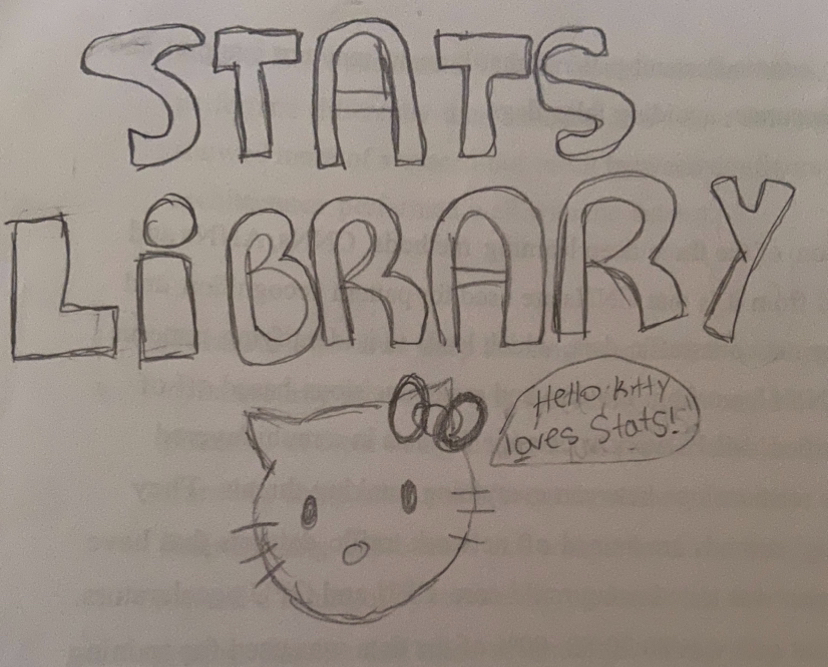
Stats Library Documentation



Please open if you haven’t:

https://byronproject2.carrd.co/

Contents

[NEGATIVE BINOMINAL DISTRIBUTION 3](#_Toc184688506)

[EXPECTED NEGATIVE BINOMINAL DISTRIBUTION 4](#_Toc184688507)

[VARIANCE NEGATIVE BINOMINAL DISTRIBUTION 4](#_Toc184688508)

[HYPERGEOMETRIC DISTRIBUTION 5](#_Toc184688509)

[EXPECTED HYPERGEOMETRIC DISTRIBUTION 6](#_Toc184688510)

[VARIANCE HYPERGEOMETRIC DISTRIBUTION 6](#_Toc184688511)

[POISSON DISTRIBUTION 7](#_Toc184688512)

[EXPECTED & VARIANCE POISSON DISTRIBUTION 8](#_Toc184688513)

First the outputs:

A screenshot of a computer program

Description automatically generated

There is a StatsLibrary class which holds all of the methods and a StatsLibraryTester class which runs the test() method out of the StatsLibraryTester class to produce all the results.

# NEGATIVE BINOMINAL DISTRIBUTION

A computer screen shot of a computer code

Description automatically generated

This function returns a double value. It takes in the parameters p (the probability), r (the number of required successes), and y (the total number of trials needed). After that, the combination method from the ComboPerm class is taken to use the combination method. It was split up for organization, but the distributions still need to use it. The “combination” is put into a double variable. That makes the combination . That value is multiplied by p to the power of r times 1 – p to the power of y – r. Math.pow is used again, and it comes in handy. It is so nice to have a built-in function.

# EXPECTED NEGATIVE BINOMINAL DISTRIBUTION

A screenshot of a computer program

Description automatically generated

The method returns a double value. The parameters are p (the probability) and ( r ) the number of successes. It is a simple method that returns the negative binominal expected formula: r / p. It returns the expected number of trials to achieve r successes.

# VARIANCE NEGATIVE BINOMINAL DISTRIBUTION

A computer screen shot of a computer code

Description automatically generated

The method returns a double value. It takes in the parameters p (probability) and r which is the number of required successes. It returns the formula which is r (the number of required successes) multiplied by ( 1 – the probability) divided by the p to the power of 2. This gives us the variance of negative binominal distribution.

# HYPERGEOMETRIC DISTRIBUTION

A computer screen shot of a program

Description automatically generated

The function’s return type is a double. The parameters are n (the number of draws from the population, r (the number of successes in the entire population), N (the total size) and y (the number of successes in the sample). Three variables are declared: combination, combination2 and combination3 for clarity. These create the hypergeometric formula. The formula is as follows: . That is then returned. The computeCombiation method from the ComboPerm class is used to achieve this.

# EXPECTED HYPERGEOMETRIC DISTRIBUTION

A screenshot of a computer

Description automatically generated

The method returns a double value. The parameters are n (the number of draws), r (the total number of successes), and N (the total size). The purpose is to return to us the hypergeometric expected value. It returns the formula n \* r divided by N.

# VARIANCE HYPERGEOMETRIC DISTRIBUTION

A screenshot of a computer program

Description automatically generated

This is the hypergeometric variance method. The return type is a double. The parameters are n (the number of draws), r (the total number of successes), and N (the total size). The probability p was created using r / N. q is 1 – p. Then the variance is computed by multiplying n by the p and q value as well as (N – n) divided by (N - 1).

# POISSON DISTRIBUTION

A computer screen shot of a code

Description automatically generated

This is the poisson distribution method. The return type is a double. It takes in Lamda and y as the parameters. Lamda serves as the word which is used as the symbol. The formula is then returned by using Math.pow to put Lamda to the power of y. This is multiplied by the e to the -lamda. Then it is divided by the factorial of y. The ComboPerm class is used to take the factorial. This was the first place I ran into an error. I did not have BigInteger set up, so I was not facing true values from the output. After adding a BigInteger factorial method, then it worked out. The bottom value is converted into a double.

# EXPECTED & VARIANCE POISSON DISTRIBUTION

A screenshot of a computer program

Description automatically generated

Both methods are grouped together here, because they both return the same value. They both have a double return type. They both take in Lamda as a parameter. After doing so, they return lamda which serves as both the variance and mean of Poisson distribution. (yay).

# Tchebysheff’s method

A computer screen shot of a code

Description automatically generated

The parameters are the mean, standard deviation and upper bound. Usually, all three are given to you in the problem. The absolute value is taken of the upper bound subtracted by the mean. The value must be positive. K is defined as the within number divided by the standard deviation. The last part is the result which is 1 – (1 divided by k to the power of 2). This displays the method we did in class but in coding form. It will then return a double value.