Francis Kumitis Applied Statistics and Probability Project 2

Project 2: Class descriptions

Card Class

-Very simple class using enum of Strings for suits and integers of 1-13 to represent any of the 52 cards in a deck A,2,…,10,J,Q,K.

Deck Class

-Started to get into using array lists extensively, only method of note is shuffleDeck which creates a temporary array list to store the original 52 cards in by removing a random number between 0 and the remaining decks size until the original array list variable is empty, and then drawing cards from the temporary list until it is full again.

Hand Class

-This class uses an array list of cards to store the state of the hand, drawing cards from a parameter deck when the constructor is called. This class also has a ton of public helper methods to help any external class determine what poker combinations the current hand could have. Reuses some code from project1 to check for pairs with the modified version of the getMode method. The isPair method makes sure that it is not also two pairs, and the full house and two pairs methods also make sure they would not both be equal to true.

DistributionLibrary

-Uses BigInteger to avoid data overflow problems with large factorials that had occurred in the previous project, BigInteger numbers are also fed into the getCombination method.

getBinomial

getGeometric

getHypergeometric

getPoisson

I tried to make these all very small with nearly the exact formula in the return statement.

CSV Graph Classes –

-These classes manipulate CSV files for meaningful data analysis functionality, utilizing scanner, buffered writer and file writer.

GraphPlotter

-Generates values for CSV based upon the variables for the formula, y=mx+b,

GraphSalter

-Salts a CSV file by reading it into a locally scoped arrayList of the values and performing an addition or subtraction of a random number (determined by the only variable of note in the class).

GraphSmoother

-I had originally decided to smooth values with indexes attempting to go out of bounds being prevented with a circular data structure, but quickly realized that (especially with a large a range for smoothing) numbers would quickly become incredibly similar through the smoothing process.

Statistics Formulas:

Probability =

Permutation:

Combination:

Conditional Probability: Given P(B) > 0:

De Morgan’s Laws:

Inverse Probability:

Binomial:

Geometric:

Hypergeometric: