Katie Bogan and Jamie Chang 4/5/17

CSC 2300 Random Means

Random Means Project

We began this project by hypothesizing what the histograms would look like for each part of the assignment, and we concluded and that histograms for the averages in Parts One and Two would have normal distributions, but Part 3 would probably be very top heavy. We did not have too much trouble in the beginning, as we achieved the desired histograms for both Parts One and Two before running into our biggest problem in Part 3. We were unsure of how to go about splitting a probability between two separate ends of a set of numbers, but Jamie had the idea of creating our own CDF to use within the program that generates a random number between 1 and 10 and if the number falls between 1 and 5 inclusive then it works with numbers 1-5, and it is between 6 and 10 inclusive, then it works with numbers 11-20. Part Three concludes our thoughts on its histogram pattern, as each number in 1-5 has a 10% chance of being selected, but the numbers in 11-20 each have a 5% chance of being selected, so the values are more likely to be lower. Overall, this assignment took us about twenty minutes to finish.

cdfProbability.java

**package** support;

**import** java.util.Random;

**public** **class** cdfProbability {

Random generator=**new** Random();

**private** **int** num;

**public** **int** k;

**public** cdfProbability(){

}

**public** **int** getRandom()

{

num=generator.nextInt(10)+1;

**if** (num<=5){

k=generator.nextInt(5)+1;

**return** num;

}

**else**{

k=generator.nextInt(9)+11;

**return** num;

}

}

}

RandomMeans.java

**import** java.util.Random;

**import** support.\*;

**public** **class** RandomMeans {

**public** **static** **void** main(String[]args)

{

Random generator=**new** Random();

Histogram hist=**new** Histogram(1, 20);

Histogram hist2=**new** Histogram(1, 20);

Histogram hist3=**new** Histogram(1, 20);

cdfProbability cdf=**new** cdfProbability();

**final** **int** LOOP=10000;

**final** **int** L1=3, L2=10;

**int** sum=0;

//Part 1

**for** (**int** i=0; i< LOOP; i++)

{

**for** (**int** j=0; j<L1; j++)

{

sum+=(generator.nextInt(20)+1);

}

hist.submit(sum/L1);

sum=0;

}

System.***out***.println("Part 1");

System.***out***.println(hist);

//Part 2

sum=0;

**for** (**int** i=0; i< LOOP; i++)

{

**for** (**int** j=0; j<L2; j++)

{

sum+=(generator.nextInt(20)+1);

}

hist2.submit(sum/L2);

sum=0;

}

System.***out***.println("Part 2");

System.***out***.println(hist2);

//Part 3

sum=0;

**for** (**int** i=0; i< LOOP; i++)

{

**for** (**int** j=0; j<L2; j++)

{

sum+=cdf.getRandom();

}

hist3.submit(sum/L2);

sum=0;

}

System.***out***.println("Part 3");

System.***out***.println(hist3);

}

}

Output

Part 1

\* = approximately 38 occurrences

1: \*

2: \*\*

3: \*\*\*\*

4: \*\*\*\*\*\*

5: \*\*\*\*\*\*\*\*\*\*

6: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

8: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

12: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

13: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

14: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

15: \*\*\*\*\*\*\*\*\*\*\*\*

16: \*\*\*\*\*\*\*\*

17: \*\*\*\*\*

18: \*\*\*

19: \*

20: \*

Part 2

\* = approximately 70 occurrences

1:

2:

3:

4: \*

5: \*

6: \*\*\*\*

7: \*\*\*\*\*\*\*\*

8: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

9: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

10: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

11: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

12: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

13: \*\*\*\*\*\*\*\*\*\*

14: \*\*\*\*

15: \*

16: \*

17:

18:

19:

20:

Part 3

\* = approximately 137 occurrences

1:

2: \*

3: \*\*\*\*

4: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

5: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

6: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

7: \*\*\*\*

8: \*

9:

10:

11:

12:

13:

14:

15: