Math 324 Spring 2018 Name

Kafai A2 (Due 14/5/2018) Yuehongxiao Ma

Use at least 10000 simulations to answer the following questions.

Suppose Xi for i=1, 2, 3… has Uniform(0,1) distribution.

Let N = min (n+1: Xn > Xn+1). Find E (N) by simulation.

2. Toss a pair of fair dice. If you get any double stop and lose. Otherwise keep tossing. If any sum gets repeated before getting any doubles stop and win.

1. Find the probability of winning
2. Find the expected number of tosses per game.

**Note:**

* + **Use this cover page with your name(s) typed – At most two students**
  + **Use comments in your programs**
  + **Each source and output must be clearly marked with the question number**
  + **Circle the answers**

**Code:**

package assignment2;

import java.util.Random;

public class Assignment2 {

public static void main(String[] args) {

// at least 10000 simulations

double SampleSize = 10000;

/\*Question 1\*/

double q1SampleSum = 0.0;

for (int i = 0; i < SampleSize; i++) {

double N = 2.0;

// Xi for i=1, 2, 3… has Uniform(0,1) distribution

double x1 = Math.random();

double x2 = Math.random();

//N = min (n+1: Xn > Xn+1)

while (x1 < x2) {

x1 = x2;

x2 = Math.random();

N++;

}

N++;

q1SampleSum += N;

}

//Find E (N)

double EN = q1SampleSum / SampleSize;

System.out.println("Question 1: E(N)=" + EN);

/\*Question 2\*/

Random rand = new Random();

int q2SumTossTime = 0;

int q2SumWinTime = 0;

for (int n = 0; n < SampleSize; n++) {

int[] sums = new int[15];

boolean toss = true;

int i = 0;

int q2TossTime = 0;

int win = 0;

int q2DicesSum = 0;

while (toss) {

q2TossTime++;

//Toss a pair of fair dice

int dice1 = rand.nextInt(6) + 1;

int dice2 = rand.nextInt(6) + 1;

q2DicesSum = dice1 + dice2;

if (dice1 == dice2) {

//get any double stop and lose

toss = false;

break;

} else {

//any sum gets repeated before getting any doubles stop and win

int j = 0;

while (sums[j] != 0) {

if (q2DicesSum == sums[j]) {

win = 1;

toss = false;

break;

}

j++;

}

}

sums[i] = q2DicesSum;

i++;

}

q2SumTossTime += q2TossTime;

q2SumWinTime += win;

}

//Find the probability of winning

//Find the expected number of tosses per game

double q2ProbWin = (double) q2SumWinTime / SampleSize;

double q2ExpectedValue = (double) q2SumTossTime / SampleSize;

System.out.println("Question 2: A. The probability of win: " + q2ProbWin);

System.out.println("Question 2: B. Expected value: " + q2ExpectedValue);

}

}

**Output:**

Question 1: E(N)=3.7177

Question 2: A. The probability of win: 0.4826

Question 2: B. Expected value: 3.1567