Lab 2: Throwing dice

due date: Friday, Oct 4 in class

Introduction

This week you will complete a short R lab that involves rolling dice and computing sums, and tossing coins. Again and again and again. This is why we use R. Each question is worth 5 points.

You will be graded both on correctness of your answers and on the presentation of your answer. This means you need to *explain* your methodology well.

A list of useful R commands is included as the last page of this handout. You will need to modify the commands to your purposes. A copy of this file is available on the class webpage, if you want to use something like WORD or LaTeX for your write up, or to cut and paste commands.

Problems

1. In this problem, you will explore the sum of the roll of three dice. For sample sizes, try $n = 10, 25, 50, 100$ or other values n of your choice. Suppose you roll the three dice and sum the outcomes repeatedly. If you were to earn a \$1 for the sum, for instance, then a roll of 1,1,1 earns you \$3.

   How much do you expect to earn in an average game? Carefully, explain your answer and methodology below.

   Assuming the dice is fair, through R we can use the sumOfDice program to create a sample population with a uniform distribution.

   Consider a sample size of 10:

$$\begin{bmatrix} Min. & 1stQu. & Median & Mean & 3rdQu. & Max. \\ 6.0 & 8.0 & 10.5 & 10.4 & 12.0 & 15.0 \end{bmatrix} \tag{1}$$

   Consider a sample size of 25:

$$\begin{bmatrix} Min. & 1stQu. & Median & Mean & 3rdQu. & Max. \\ 3.00 & 10.00 & 11.00 & 11.12 & 13.00 & 16.00 \end{bmatrix} \tag{2}$$

   Consider a sample size of 50:

$$\begin{bmatrix} Min. & 1stQu. & Median & Mean & 3rdQu. & Max. \\ 3.00 & 9.00 & 11.00 & 10.86 & 14.00 & 17.00 \end{bmatrix} \tag{3}$$

   Consider a sample size of 50:

$$\begin{bmatrix} Min. & 1stQu. & Median & Mean & 3rdQu. & Max. \\ 3.00 & 8.00 & 11.00 & 10.44 & 13.00 & 17.00 \end{bmatrix} \tag{4}$$

   Just from these relatively small sample sizes we can see that the average winnings per roll is dancing around 10.5. With R we can take the sample population to 1,000,000 and we see that it converges to exactly 10.5.

   Consider a sample size of 1,000,000:

$$\begin{bmatrix} Min. & 1stQu. & Median & Mean & 3rdQu. & Max. \\ 3.0 & 8.0 & 11.0 & 10.5 & 13.0 & 18.0 \end{bmatrix} \tag{5}$$

2. You toss a *biased* coin repeatedly, with the probability of heads, $P(H) = p$, for some unknown value of p. Your goal is to give the best estimate of p that you can. To that end, suppose you flip this coin three times and that you earn \$3 for each H that comes up, and lose \$1 for each T that comes up.

   There are 3 data sets available for you. They are called *earningsN* where $N = 10, 100, 1000$. Again, your task is to give the best estimate for p you can give from these datasets and explain how you arrived at your answer.

   I estimate the value of p to be ___________ because

   Lets look at the summaries for each dataset,

   Consider earning10:

$$\begin{bmatrix} Min. & 1stQu. & Median & Mean & 3rdQu. & Max. \\ -3.0 & -3.0 & -1.0 & 0.6 & 4.0 & 9.0 \end{bmatrix} \tag{6}$$

   Consider earning100:

$$\begin{bmatrix} Min. & 1stQu. & Median & Mean & 3rdQu. & Max. \\ -3.00 & -3.00 & 1.00 & 0.08 & 1.00 & 9.00 \end{bmatrix} \tag{7}$$

   Consider earning1000:

$$\begin{bmatrix} Min. & 1stQu. & Median & Mean & 3rdQu. & Max. \\ 3.00 & 8.00 & 11.00 & 10.44 & 13.00 & 17.00 \end{bmatrix} \tag{8}$$

   Consider the following binomial distribution where the Y is the earnings in a game.

$$Y = \begin{bmatrix} -3 & ... & \binom{3}{0}(1-p)^3 \\ 1 & ... & \binom{3}{1}(p)^1(1-p)^2 \\ 5 & ... & \binom{3}{2}(p)^2(1-p)^1 \\ 9 & ... & \binom{3}{3}(p)^3 \end{bmatrix} \tag{9}$$

   We can take this distribution and generate a function for the Expected Value.

$$E(Y) = -3 * \binom{3}{0}(1-p)^3 + 1 * \binom{3}{1}(p)^1(1-p)^2 + 5 * \binom{3}{2}(p)^2(1-p)^1 + 9 * \binom{3}{3}(p)^3 \tag{10}$$

   Then we set our new function to equal the mean of the largest data set, earnings1000 (I would merge the data sets together but I couldn't figure it out in R). So our new equation is

$$-.14 = -3 * \binom{3}{0}(1-p)^3 + 1 * \binom{3}{1}(p)^1(1-p)^2 + 5 * \binom{3}{2}(p)^2(1-p)^1 + 9 * \binom{3}{3}(p)^3 \tag{11}$$

   From here we can simply solve for p.

$$p \approx .238 \tag{12}$$

Helpful commands from R:

```
# generate 10 random rolls of a di with values between 5 and 9
        d10=floor(runif(10,min=5,max=10))

# more or less the contents of my R file named dice.R
        sampleSize<-10
        numberDice=7

        sumOfDice=rep(0,sampleSize)

        str=sprintf("Generating %d samples of the experiment   'Find the sum of %d dice'  ",
             sampleSize,numberDice)
        print(str)

        for (ii in 1:sampleSize){
           sumOfDice[ii]=sum(floor(runif(numberDice,5,11)))
           #print(sumOfDice[ii])
        }
        if (sampleSize<1000){
          print(sumOfDice)
        }
# end of batch file dice.R

# how to execute the commands in dice.R
        source("dice.R")

# reminder for making histograms
        hist(d10,seq(.5,6.5,1))
        hist(d10,seq(.5,6.5,1),freq=T)
        hist(d10,seq(.5,6.5,1),freq=F)


# how to load the datasets with earnings, or load and save to a variable
        load(file="earnings10")
        ....

# count number of entries in a vector matching some value c
# In this example, count the number of times vec has entries = 0
       vec = c(0, 0, 1, 1, 1, 2, 0, 2, 0, 0, 5)
       sum(vec==0)
```