Statistics 110 Strategic Practice & Homework 1: Dice Problem

Goutham Swaminathan

Problem

Decide whether the blank should be filled in with =, <, or >, and give a short but clear explanation. * (probability that the total after rolling 4 fair dice is 21) ____ (probability that the total after rolling 4 fair dice is 22)

Explanation

This problem involves determining whether the probability that the total summed values after rolling 4 fair dice is 21 is greater than, less than, or equal the probability that the total summed values after rolling 4 fair dice is 22. In order to solve this problem using simulation, we will need to do the following: 1. Simulate n iterations of 4 dice rolls and sum their values (n = 10,000) 2. Count how many sums are equal to 21 and 22 to determine the probabilities of each outcome 3. Compare the calculated probabilities to answer the original questions

In order to accomplish the first step, we will need to initialize variables for the two sums we will be comparing and the number of iterations as shown above.

```
sum1 = 21
sum2 = 22
iterations = 25000
```

The **iterations** variable will be initialized to 25,000 because we want to simulate as many iterations as possible in order to obtain more accurate probabilities.

In addition, we will be creating 4 dice vectors to store the value of the die rolled for every iteration. For each **die** vector, we invoke the *sample* function to generate n values between 1 and 6 (n being the # of iterations). Afterwards, we create a **sumDies** vector that stores the sum of **die1**, **die2**, **die3**, and **die4** for each round.

```
set.seed(2)
die1 = sample(6, iterations, replace = TRUE)
die2 = sample(6, iterations, replace = TRUE)
die3 = sample(6, iterations, replace = TRUE)
die4 = sample(6, iterations, replace = TRUE)
sumDies = die1 + die2 + die3 + die4
```

Afterwards, the *mean()* function will be used to calculate the probability for each of the two sums. Inside of the *mean()* function, we will insert a logic condition, which will be evaluated for every index of the vector. The output will be a vector of TRUE and FALSE, which can also be represented as 1 and 0, respectively. As a result, the 1s will represent when the sum of the dies equal the desired value (**sum1** and **sum2**) and 0s will represent when they do not. Note that only larger numbers of iterations will yield accurate results. This is because in order to obtain more accurate probability values, we will need to run many simulations.

```
prob1 = mean(sumDies == sum1)
prob2 = mean(sumDies == sum2)
c(prob1, prob2)
```

```
## [1] 0.01508 0.00812
```

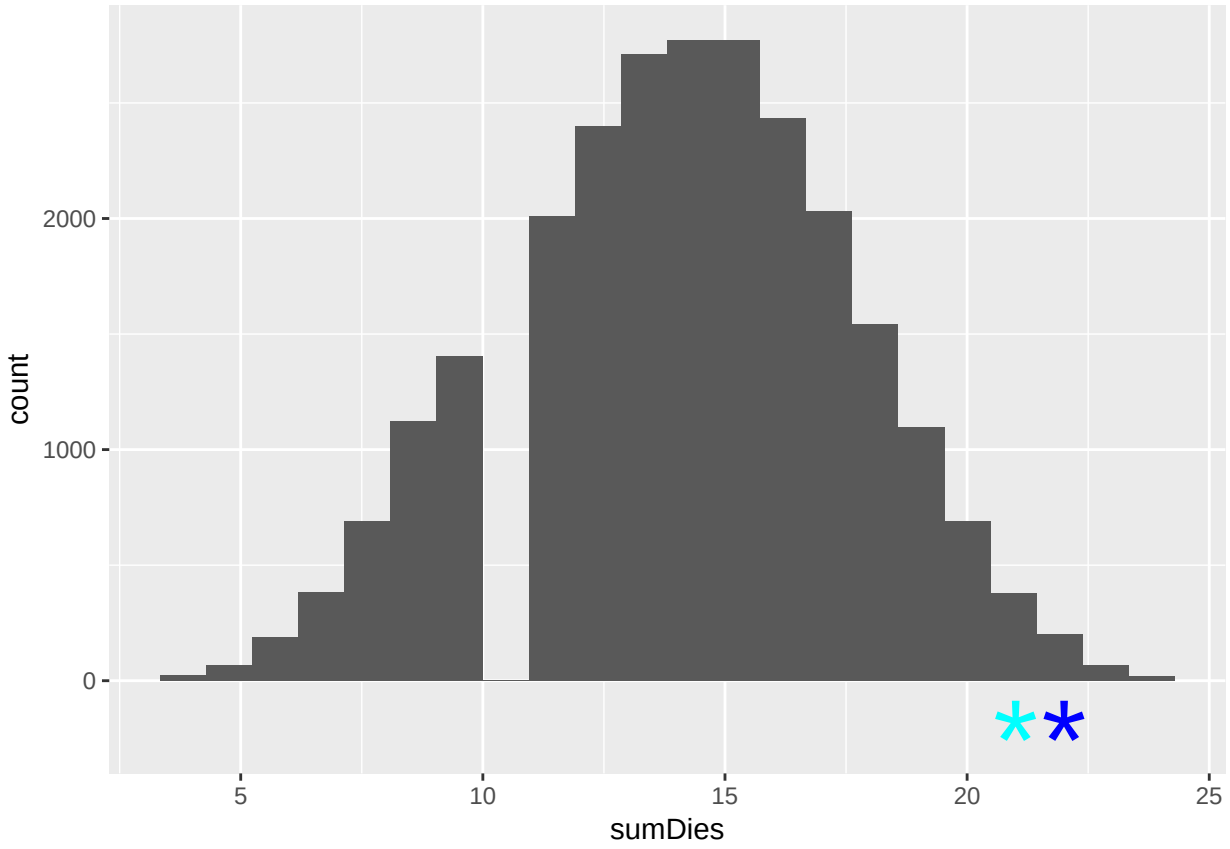
We can see that the probability of the sum of 4 fair dice rolls being 21 is greater than the probability of the sum of 4 fair dice rolls being 22. Although the scope of the problem ends here, we will now track the results after every iteration and plot them.

First, we will plot a histogram showing the frequency for the sum of 4 fair dice rolls between 1 and 25 using the *qplot()* function. The teal asterisk is used to denote when the sum is equal to 21 and the blue asterisk is used to denote when the sum is equal to 22.

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.3
```

```
qplot(sumDies, geom = "histogram", bins = 22) +
  annotate("text", x = sum1, y = -iterations/100, label = "*", size = 15, color = 5) +
  annotate("text", x = sum2, y = -iterations/100, label = "*", size = 15, color = 4)
```



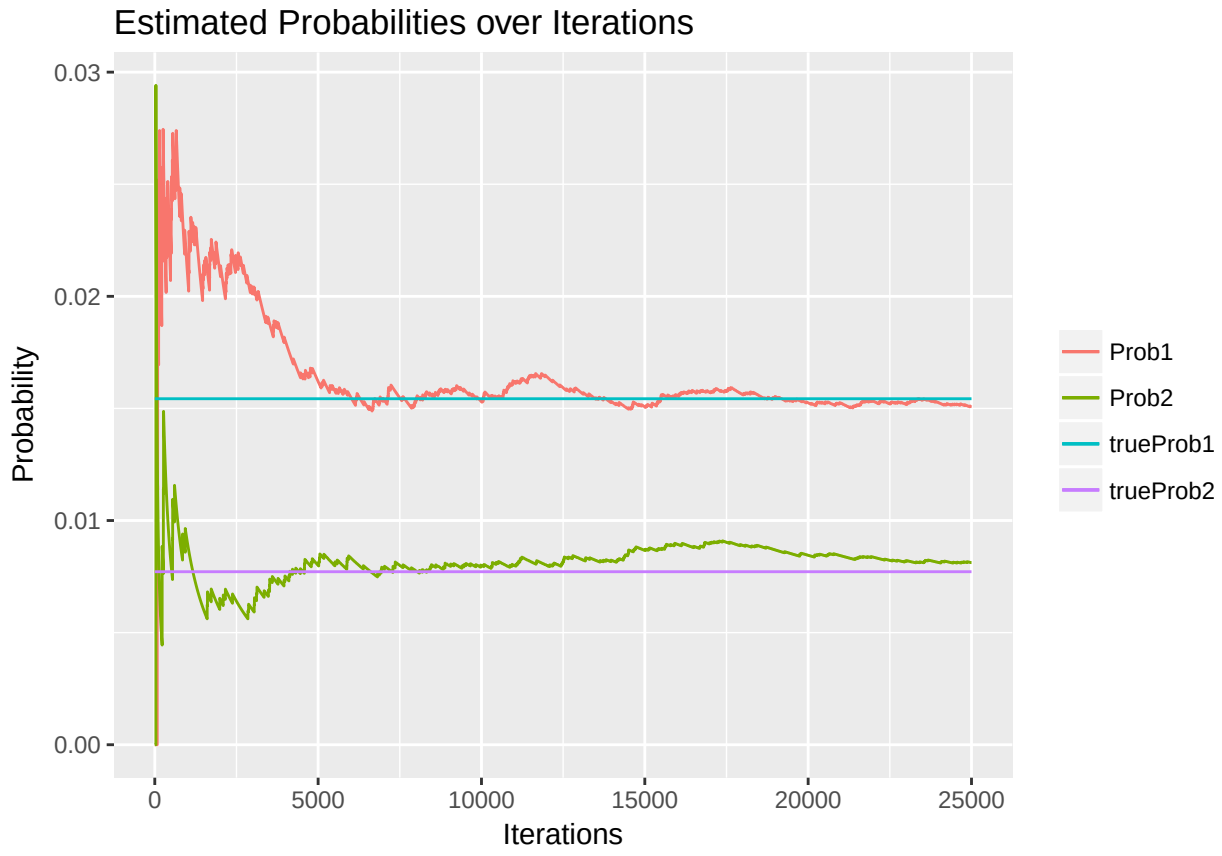
The results of this histogram confirm our answer by showing that the frequency of the sum being 21 is, in fact, higher than the frequency of the sum being 22.

We will now create a line graph to track our calculated probabilities over every iteration and compare it with the true probability values. First, we initialize the **subsetProb1** and **subsetProb2** vectors using the *vector()* function. These vectors will hold the calculated probabilities for every iteration. Note that every iteration's probability is calculated using all iterations before it. This is done using a *for* loop. For the _i_th index of **subsetProb1** and **subsetProb2**, we will calculate the mean for every value of sumDies from 1 to i that meets each of our logic conditions using the *mean()* function. Afterwards, we will use the *ggplot()* function and specify the x axis as being from 1 to **iterations**. Then, we will use the *geom_line()* function to plot 4 lines: the values of **subsetProb1** and **subsetProb1**, as well as the true probability values, which are calculated using the expressions $\frac{20}{6^4}$ (for Prob1) and $\frac{10}{6^4}$ (for Prob2). See the source link for this problem to understand how to calculate the true theoretical probabilities.

```r
subsetProb1 = vector()
subsetProb2 = vector()

for(i in 1:iterations){
  subsetProb1[i] = mean(sumDies[1:i] == sum1)
  subsetProb2[i] = mean(sumDies[1:i] == sum2)
}

ggplot(data = NULL, aes(x = 1:iterations)) +
  geom_line(aes(y=subsetProb1, color = "Prob1")) +
  geom_line(aes(y=subsetProb2, color = "Prob2")) +
  geom_line(aes(y=(20/6^4), color = "trueProb1")) +
  geom_line(aes(y=(10/6^4), color = "trueProb2")) +
  ylab("Probability") + ggtitle("Estimated Probabilities over Iterations") +
  theme(legend.title = element_blank()) +  xlab("Iterations")
```



This line graph allows us to see how simulation can be used to estimate true probabilities. We can see that with very few iterations, random variation dominates the result and leaves us with inaccurate probabilities. However, as the number of iterations significantly increases, the calculated probability values converge to the true probabilities.

Conclusion

Thank you for taking the time to read this explanation for the dice problem from "Statistics 110: Strategic Practice & Homework 1", courtesy of Harvard University. Link: https://projects.iq.harvard.edu/files/stat110/files/strategic_practice_and_homework_1.pdf. This problem was taken from the public resources to the course and **is not my orginal work**. The explanation, however, is 100% original. Please feel free to visit my GitHub page at http://github.com/goutham1220 where I will be posting more explanations as well as other statistics and data science-related resources. In addition, please feel free to visit my YouTube channels "GSDataScience" (http://bit.ly/gsdatascience), where I will be posting more data science and statistics-related videos, and "Gooth" (http://youtube.com/gooth), where I post more cinematic-style, slice-of-life videos.