R Assignment 2

Jonathan De Los Santos

Problem 1 (28 points)

Suppose you roll a die twice in succession, getting X_1 and X_2. Then divide them, getting $Y = \frac{X_1}{X_2}$. Thus, Y is discrete, ranging from a minimum of 1/6 to a maximum of 6. In R, we can generate all possible values for Y as follows.

```
# run this chunk of code:
X1 = c(1,2,3,4,5,6) # possible outcome of the first try
X2 = c(1,2,3,4,5,6) # possible outcome of the second try

Y = c() # defining an empty vector Y
# Then we divide each element of X1 by all value of X2 and record them in Y vector
for (i in X1) {
  for (j in X2) {
     Y = c(Y, i/j) # inserting X1/X2 into Y vector
  }
}
# Now Y is ready to be used in part a
```

a. Report the sorted list of Y values, then Report the mean of the Y. Present all values by 3 decimal places. (6 points)

Here are the sorted values of Y and the mean rounded to 3 decimals. The sort allows us to see the minimum and maximum values of Y

```
# Hint: you can use sort() and mean() functions in R.
round(sort(Y),3)
```

```
##  [1] 0.167 0.200 0.250 0.333 0.333 0.400 0.500 0.500 0.500 0.600 0.667 0.667
## [13] 0.750 0.800 0.833 1.000 1.000 1.000 1.000 1.000 1.000 1.200 1.250 1.333
## [25] 1.500 1.500 1.667 2.000 2.000 2.000 2.500 3.000 3.000 4.000 5.000 6.000
```

```
round(mean(Y),3)
```

```
## [1] 1.429
```

b. Simulate 10000 (or more) i.i.d observations $Ysim = (\frac{Xsim_1}{Xsim_2})$. Don't print the outcome of Ysim (you can only print the first 6 value). (10 points)

Simulating 10,000 instances of dice rolls twice. We use replace = TRUE to denote that these "rolls" are with replacement because dice rolls are independent activities. Afterwards we divide one simulation by the other to create a new IID variable Y.

```
# Hint: you can use sample() function in R to simulate 10000 instances of random dice
numbers. Do this in R: Xsim1 <- sample(1:6, size = 10000, replace = TRUE), then the s
ame for Xsim2. Now you have 10000 instances of rolling a die twice. Why replace = TRU
E?
Xsim1 <- sample(X1, size = 10000, replace = TRUE)
Xsim2 <- sample(X2, size = 10000, replace = TRUE)

# Then find Ysim=Xsim1/Xsim2 in R. Now you have 10000 instances of simulated Y values
, saved in Ysim. Ysim is a new independently and identically distributed (iid) random
variable. Ysim values are independent of each other because sampling is with replacem
ent, or the outcome of the first roll does not have any impact on the outcome of the
second roll.
Ysim <- Xsim1/Xsim2
head(Ysim)
```

```
## [1] 0.5000000 1.0000000 0.6000000 1.3333333 0.3333333 0.1666667
```

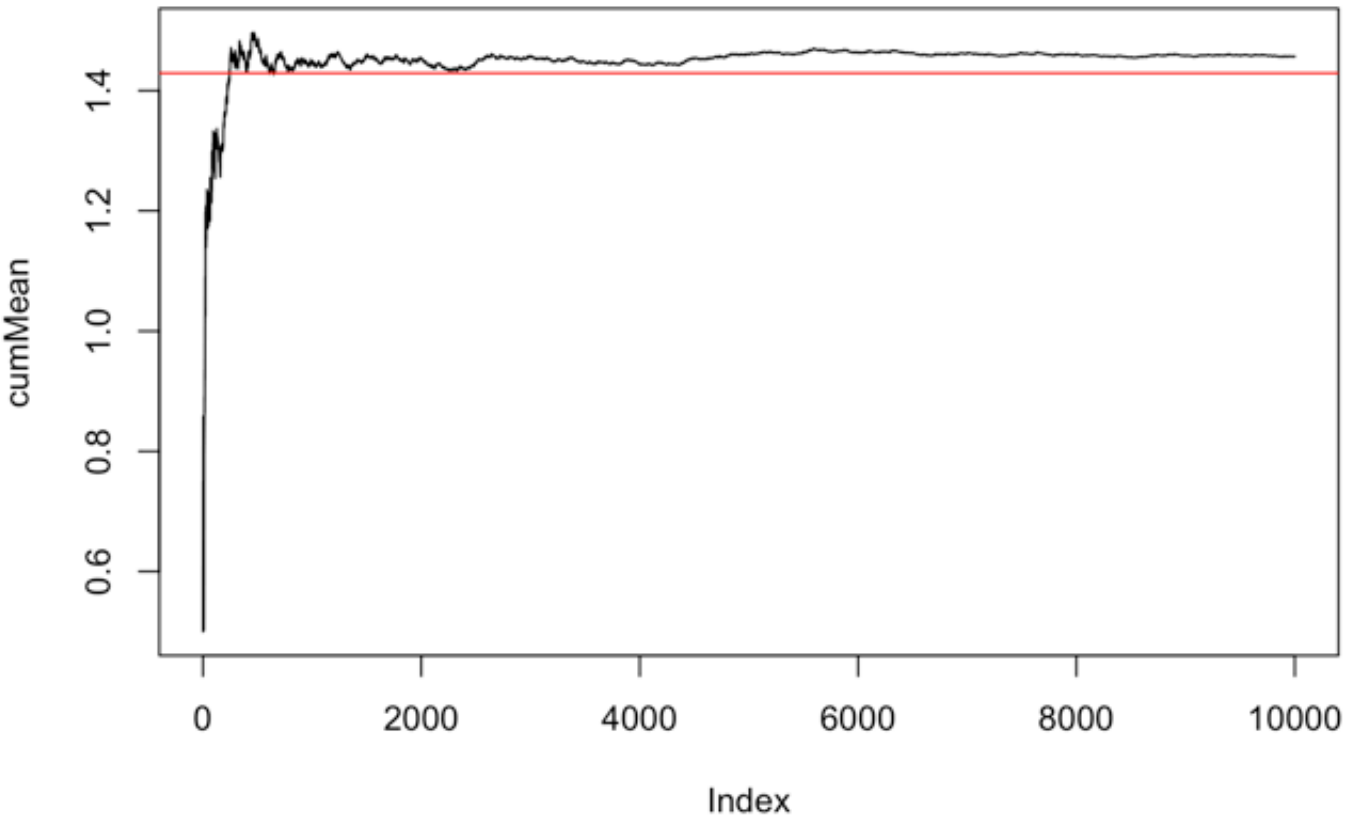
c. Draw the graph of successive average (cumulative mean) of $Ysim$. Discuss your observations based on what you see in the graph of successive average and the answer of part a for the mean(Y). (12 points)

To demonstrate the law of large numbers, we will calculate cumulative mean for the range of 0 to 10,000 and plot it. Notice that the plot starts to align to the mean as the n increases.

```
# For plotting the cumulative mean of Ysim values we use plot(cumMean, type = "l"), b
ut before that, you need to make a vector of cumulative means (cumMean) for Ysim valu
es. Please watch at 2:50 in my first video in Module 2.
n = 10000
cumSumY <- cumsum(Ysim)
cumMean <- c()

# Create loop that calculates cumMean
# Iterate over all values 1:n dividing the cumulative sum by each value and storing i
t in cumMean
for (i in 1:n) {
  cumMean[i] <- cumSumY[i]/i
}

# Type = (l)ine plot
plot(cumMean, type = "l")
# Use abline to add a straight line through the plot to see the mean
# (col)or = red
abline(h = 1.429167, col = "red")
```



Problem 2 (52 points)

A small hotel has 10 rooms. From experience they know that 20% of the time, people who make reservations do not show up, so as a result, they overbook by accepting 12 reservations for a given night. Let X be the number of no shows that night (people who don't show up).

a. What is the expected number of "no shows" that night? What is the standard deviation of that number? (8 points)

Expected no-shows can be calculated with expected value, we can also find the variance and then the standard deviation:

```
# Hint: X (the number of no shows) is a binomial random variable: X~binomial(p=0.2, n = 12). There is a given formula for finding the expected value and standard deviation of the binomial random variable. Watch Module 4, lecture part 3, when I talk about the properties of binomial random variables.
n = 12
p = .2
μX = n*p
varX = n*p*(1-p)
sdX = sqrt(varX)
print("Expected number of no-shows:")
```

```
## [1] "Expected number of no-shows:"
```

```
μX
```

```
## [1] 2.4
```

```
print("Standard deviation:")
```

```
## [1] "Standard deviation:"
```

```
sdX
```

```
## [1] 1.385641
```

b. Is X continuous or discrete. (2 points)

This is discrete, there is no range of variables only whether they showed up or not.

c. What is the range of X? (4 points)

The range of X is 0:12 or every possible reservation.

d. Find the probability of all possible value of X. Round by 3 decimal places. (10 points)

Here the probability of each possible value is computed and combined into a table

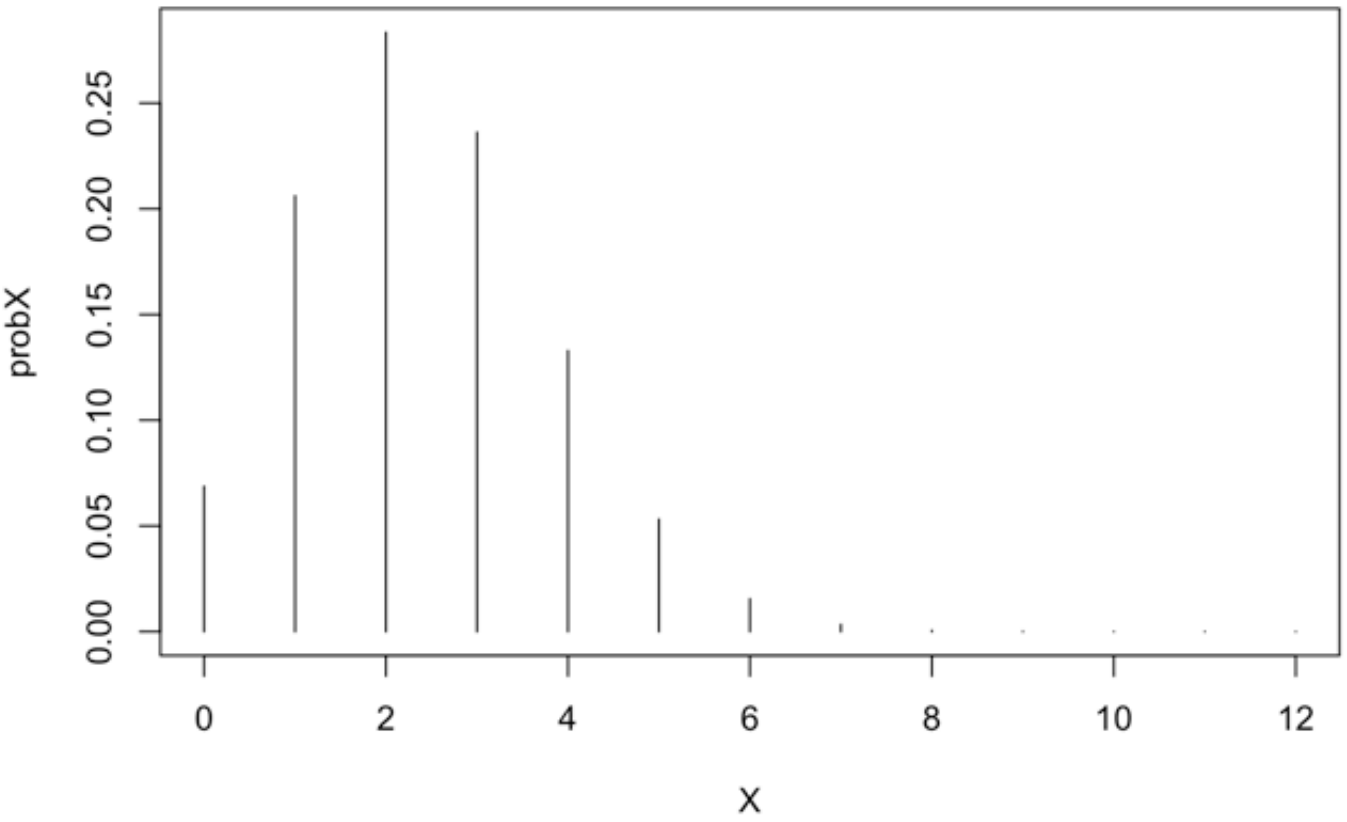
```
# You need to find P(X=x). You can answer it by using dbinom(X, size = 12, prob=.2) i
n R.
# Remember for probability of exact point you use dbinom, for multiple points use pbi
nom, and for simulations sbinom
# Watch Module 4, lecture part 3
X <- 0:12
probX <- dbinom(X, size = 12, prob=.2)
probX = round(probX, 3)
# Combine columns
cbind(X, probX)
```

```
##        X probX
##  [1,]  0 0.069
##  [2,]  1 0.206
##  [3,]  2 0.283
##  [4,]  3 0.236
##  [5,]  4 0.133
##  [6,]  5 0.053
##  [7,]  6 0.016
##  [8,]  7 0.003
##  [9,]  8 0.001
## [10,]  9 0.000
## [11,] 10 0.000
## [12,] 11 0.000
## [13,] 12 0.000
```

e. Create a needle plot for all probability values of part d. (10 points)

Plotting the probabilities of each value allows us to visualize the distribution:

```
# Watch Module 4, lecture part 3
X <- 0:12
probX <- dbinom(X, size = 12, prob=.2)
plot(X, probX, type = "h")
```



f. What is the probability that the hotel will end up with more customers than they can handle (that is, more people with reservations than available rooms will arrive)? (4 points)

The hotel will have too many customers if there are 0 or 1 no shows, the probability of 0 or 1 is:

```
# Hint: X is the number of no-shows and we know that the p=0.2, and n=12. If zero or one customer does not show up, then the hotel ends up with more customer than available rooms.
probFull <- pbinom(1, 12, .2)
probFull
```

```
## [1] 0.2748779
```

g. Simulate 10000 instances for the number of no-shows and call them $Xsim$; then estimate the expected value and standard deviation of the number of the simulated no-shows. What explains why the simulation results are different from the answers to a? (14 points)

The mean we calculate from our simulation is 2.394. This is different from the first “true” mean we calculated because it is a sample that can differ from the calculated mean. If we run even more simulations, we still start to arrive closer to the mean.

```
# Hint: You can generate 10000 (or more) binomial instances based on n=12, and p=0.2
as follows:
# Xsim <-  rbinom(10000, 12, 0.2)
# Now Xsim is a data vector, and we can find its means and stdev by mean() and sd() f
unctions in R.

Xsim <- rbinom(10000, 12, .2)
print('The mean of no-shows is:')
```

```
## [1] "The mean of no-shows is:"
```

```
mean(Xsim)
```

```
## [1] 2.3848
```

```
print("The standard deviation of no-shows:")
```

```
## [1] "The standard deviation of no-shows:"
```

```
sd(Xsim)
```

```
## [1] 1.372121
```