DATA 252 / DATA 551: Homework 1 Solution

1. Let X be the number of heads in three coin tosses. The coins are assumed fair.

   a. What is the probability mass function of X?

      The probabilities can be calculated by listing all of the $2^3 = 8$ outcomes.

      $$P(X = 0) = 0.125, P(X = 1) = 0.375, P(X = 2) = 0.375, P(X = 3) = 0.125.$$

   b. Verify that this probability mass function satisfy the two conditions.

      It is easy to verify that i) all the probabilities are nonnegative and ii) they add up to 1.

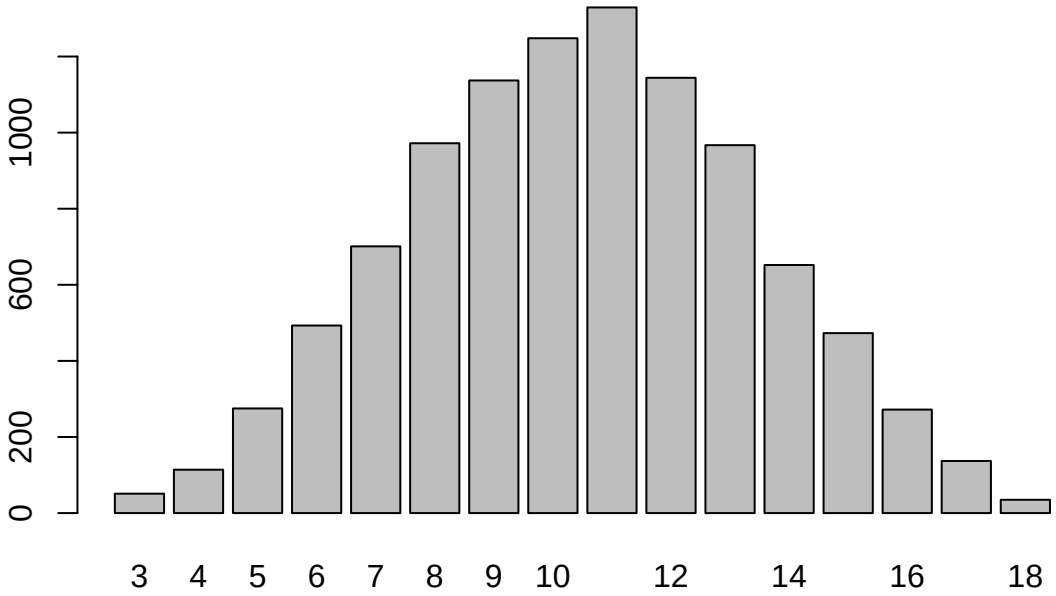
2. Suppose you roll three fair dice and calculate the sum of the three rolls. Is the sum more likely to be 9 or 10? Simulate the result of three dice rolls on https://www.random.org/dice/ for a few times and record your result. Describe how you would run a simulation to estimate the probability that the sum is 9 and the probability that the sum is 10.

The following *algorithm* performs this simulation:

- Step 1: Randomly draw three fair dice and calculate the sum M^1;
- Step 2: Repeat Step 1 for K times, recording $M^1, \ldots, M^K$, where K is a large number.

To estimate the probability that the sum is 9, we can calculate the proportion of 9's in $\{M^1, \ldots, M^K\}$. The following R code is given as example. It shows that the sum is more likely to be a 10.

```
nsim=10000  #number of simulated runs
result=rep(NA,nsim)  #empty vector for storing the simulated results
for (i in 1:nsim){
  dices=sample(c(1:6),size=3,replace=T) #sample 3 dices
  result[i]=sum(dices) #store the sum of dices
}
plot(as.factor(result),main='sum of three dice rolls')
```



3. Read Section 1.1 (Simulation of Discrete Probabilities) in *Introduction to Probability* (page 1 to page 12). Don't worry about the mentioned "programs." Some of the simulations are reproduced in the R sample code if you want to play with the simulations.

(Solution NA)

4. Consider question 10 on page 14 of *Introduction to Probability*. (Note: in roulette, probability of red is 18/38). Are you more likely to win 5 dollars or to lose 100 dollars? Design a simulation to find the answer. For now, you don't have to write the code for the simulation, but rather, you just need to describe how you plan to write the code.

Sample algorithm and R code given below. Here, we use X to denote the net winning, b to denote the current betting amount, and N to denote the number of games played. Based on this result, it is much more likely to win \$5 dollars (in about 94% of the simulated games).

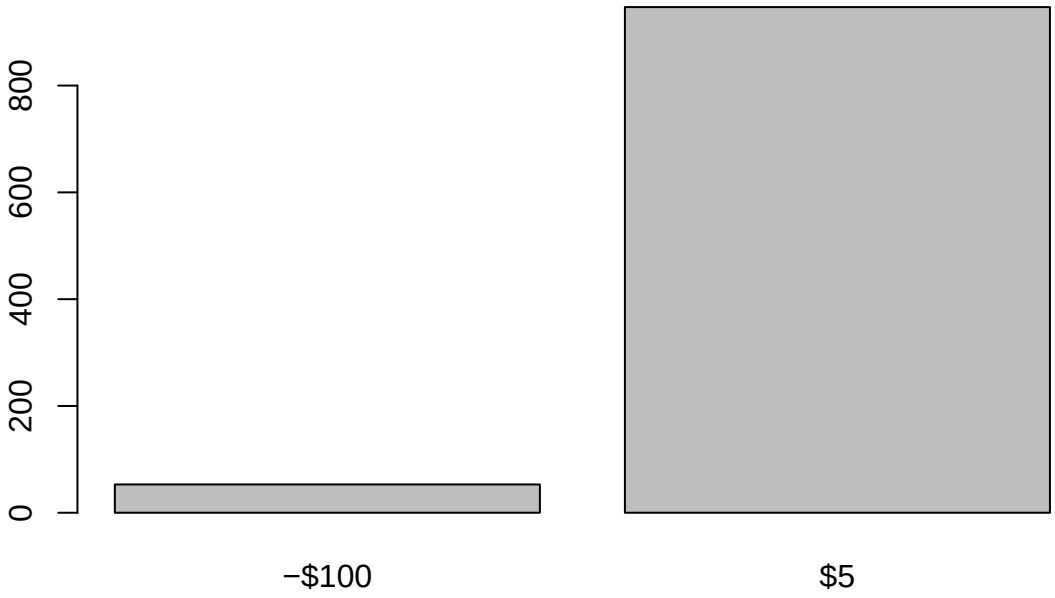
For $k = 1, \ldots$ to the desired number of simulated runs, do the following.

- Step 1: Initialize $X^0 = 0$, $b^0 = 1$, $N^0 = 0$;
- Step 2: Update $\{X^i, b^i, N^i\}$ to $\{X^{i+1}, b^{i+1}, N^{i+1}\}$:
  - Randomly sample $R = 1$ w.p. (with probability) 18/38 and $R = 0$ w.p. 20/38;
  - If $R = 1$, set $X^{i+1} = X^i + b^i$ and $b^{i+1} = 1$;
  - If $R = 0$, set $X^{i+1} = X^i - b^i$ and $b^{i+1} = 2b^i$;
  - Set $N^{i+1} = N^i + 1$;

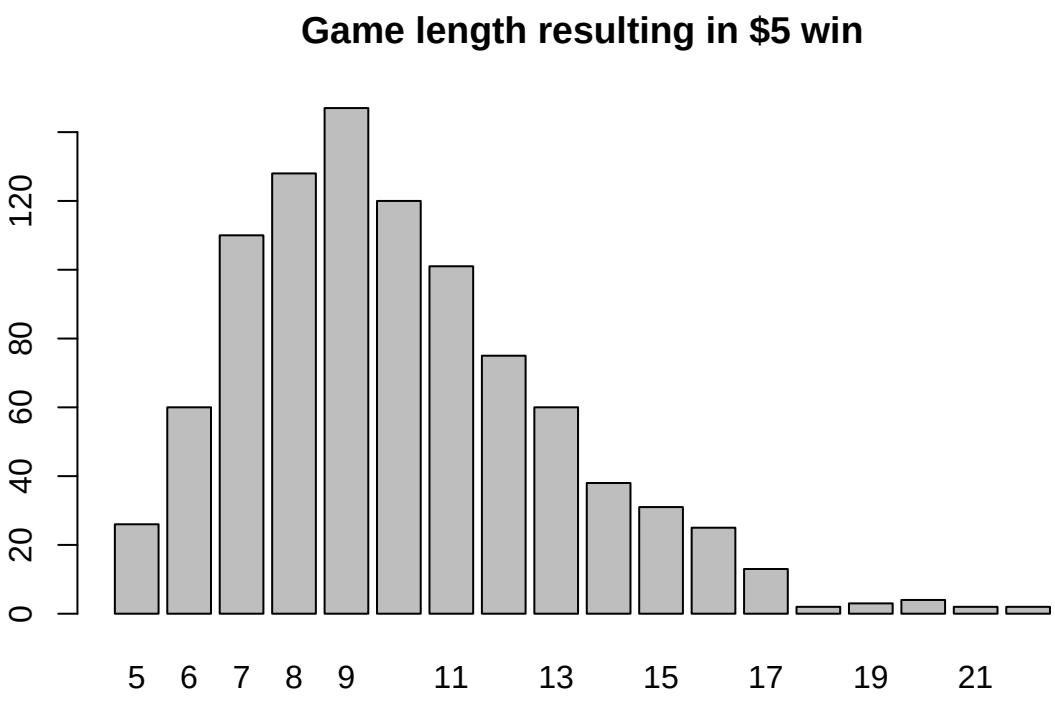
- Step 3: Repeat Step 2 until $X^i \geq 5$ or $X^i \leq 100$. Record the last N^i and X^i.

```
nsim=1000 #number of games to simulate
result.length=rep(NA,nsim) #store the length of each game
result.win=rep(NA,nsim) #store the outcome of each game

for(k in 1:nsim){  ###start of for loop

#---initialization---#
xcurrent=0
bcurrent=1
ncurrent=0
nmax=100 #max number of runs of while loop

#---updating---#
while( (xcurrent<5) && (xcurrent>-100) && (ncurrent<nmax)){

  #simulate one game
  R = sample(c(1,0),size=1,prob=c(18/38,20/38))
  if(R==1){
    xnext=xcurrent+bcurrent
    bnext=1
  }else{
      xnext=xcurrent-bcurrent
      bnext=2*bcurrent
  }
  nnext=ncurrent+1

  #update current values
  xcurrent=xnext
  bcurrent=bnext
  ncurrent=nnext

  #print results for one iteration
  #print(paste0("Game", ncurrent, ' - result=', R,
  #             '  net winning=$', xcurrent,
  #             '  next betting=$', bcurrent))
}

if(ncurrent==nmax){print('Warning: while loop reached max iteration')}

result.length[k]=ncurrent
result.win[k]=xcurrent
} ###end of for loop
```

```r
# visualize result
plot(factor(result.win==5,labels=c('-$100','$5')))
```




```r
plot(as.factor(result.length[result.win==5]),
     main='Game length resulting in $5 win')
```



5. Design a simulation to answer question 13 on page 14 of *Introduction to Probability*. For now, you don't have to write the code for the simulation, but rather, you just need to describe how you plan to write the code.

Sample algorithm and R code given below. Based on the simulation results, the *small* hospital is more likely to have the greater number of days on which more than 60% of the babies born were boys. Comment: this result is related to the *Central Limit Theorem*. With a larger sample size (i.e., large hospital), the observed proportion of boys is expected to be closer to the truth, which is 50%. Therefore, it is harder for the large hospital to have an observed proportion that is as high as 60% or even higher.

For $k = 1, \ldots$ to the desired number of simulated years, do the following.

- Step 1: Let $large^d$ be a randomly drawn vector of length 45, where each entry can be 0 or 1 w.p. 0.5;
- Step 2: Calculate $year.large^d$, the percentage 1's in $large^d$ (this represents the percentage of boys in the given day d);
- Step 3: Let $small^d$ be a randomly drawn vector of length 15, where each entry can be 0 or 1 w.p. 0.5;
- Step 4: Calculate $year.small^d$, the percentage 1's in $small^d$.
- Step 5: Repeat Steps 1 to 4 for $d = 1, \ldots, 365$.
- Step 6: For $\{year.large^d\}_{d=1,\ldots,365}$ and $\{year.small^d\}_{d=1,\ldots,365}$ respectively, calculate and record the number of entries that are greater than 0.6.

```
nsim=200  #number of years to simulate

#number of days on which more than 60% babies were boys in each hospital
result.large=rep(NA,nsim)
result.small=rep(NA,nsim)

for(k in 1:nsim){

  #---simulate one year---#
  year.large=rep(NA,365) #percentage of boys in large hospital
  year.small=rep(NA,365) #percentage of boys in small hospital

  for(d in 1:365){
    #simulate one day
    large=sample(c(0,1),size=45,replace=T)
    year.large[d]=mean(large)
    small=sample(c(0,1),size=15,replace=T)
    year.small[d]=mean(small)
    }

  # store the end-of-year results for the simulated year
  result.large[k]=sum(year.large>=0.6)
  result.small[k]=sum(year.small>=0.6)
}
```

```r
# visualize results
par(mfrow=c(1,2))
hist(result.large, xlab='Num days with >60% boys',
     col='salmon',
     main='large hostipal')
hist(result.small, xlab='Num days with >60% boys',
     col='lightblue',
     main='small hostipal')
```