Homework 3

Q1. (25%) Two six-sided fair dices are thrown sequentially, and their face values are recorded. (a), (b), (c), & (d) do NOT need to be done in R.

(a) List the sample space.

```
(1,1)(1,2)(1,3)(1,4)(1,5)(1,6)
(2,1)(2,2)(2,3)(2,4)(2,5)(2,6)
(3,1)(3,2)(3,3)(3,4)(3,5)(3,6)
(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)
(5,1)(5,2)(5,3)(5,4)(5,5)(5,6)
(6,1)(6,2)(6,3)(6,4)(6,5)(6,6)
```

(b) List the elements that make up the following events:

(1) A=the sum of the two values is at least 5,

```
(1,4)(1,5)(1,6)
(2,3)(2,4)(2,5)(2,6)
(3,2)(3,3)(3,4)(3,5)(3,6)
(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)
(5,1)(5,2)(5,3)(5,4)(5,5)(5,6)
(6,1)(6,2)(6,3)(6,4)(6,5)(6,6)
```

(2) B=the value of the first die is higher than the value of the second,

```
(2,1)
(3,1)(3,2)
(4,1)(4,2)(4,3)
(5,1)(5,2)(5,3)(5,4)
(6,1)(6,2)(6,3)(6,4)(6,5)
```

(3) C=the first value is 4.

```
(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)
```

(c) List the elements of the following events:

(1) A and C

```
(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)
```

(2) B or C

```
(2,1)
(3,1)(3,2)
(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)
(5,1)(5,2)(5,3)(5,4)
(6,1)(6,2)(6,3)(6,4)(6,5)
```

(3) A and (B or C)

```
(3,2)
(4,1)(4,2)(4,3)(4,4)(4,5)(4,6)
(5,1)(5,2)(5,3)(5,4)
(6,1)(6,2)(6,3)(6,4)(6,5)
```

(d) Based on the classical approach, derive

P(A and C)

```
count=0
for(i in 1:6)
{
  if(i+4>=5){
    count=count+1
  }
}
prob1=count/6^2
print(prob1)
```

“

[1] 0.1666667

P(B or C)

```
count=0
for(i in 1:6)
{
  for(j in 1:6){
    if(i>j||i==4){
      count=count+1
    }
  }
}
prob2=count/6^2
print(prob2)
```

“

[1] 0.5

P(A and C)

```
count=0
for(i in 1:6)
{
  for(j in 1:6){
    if(i>j||i==4){
      if(i+j>=5){
        count=count+1
      }
    }
  }
```

```
}
prob3=count/6^2
print(prob3)
```

“

[1] 0.4444444

(e) Use *sample()* in *R* to simulate the throwing of two dice 1,000 times. Compute P(A and C), P(B or C), and P(A and (B or C)) from the 1,000 runs. How different are the results from (d)?

```
n=1000
d1=sample(6,n,replace=T)
d2=sample(6,n,replace=T)
result=cbind(d1,d2)
```

P(A and C)

```
s1=length(result[(d1==4&d1+d2>=5),1])/n
print(s1)
```

“

[1] 0.167

```
diff=prob1-s1
print(diff)
```

“

[1] -0.0003333333

P(B or C)

```
s2=length(result[(d1>d2|d1==4),1])/n
print(s2)
```

“

[1] 0.503

```
diff=prob2-s2
print(diff)
```

“

[1] -0.003

P(A and (B or C))

```
s3=length(result[d1+d2>=5&(d1>d2|d1==4),1])/n
print(s3)
```

“

[1] 0.447

```
diff=prob3-s3
print(diff)
```

“

[1] -0.002555556

Q2. (20%) A drawer of socks contains seven black socks, eight blue socks, and nine green socks. There is NO difference between left and right for those socks. Two socks are chosen in dark.

(a) What is the *exact probability* that they match (i.e., two socks have the same color)?

What is the *exact probability* that a black pair is chosen?

```
socks.total = 7 + 8 + 9
match.socks = (choose(7, 2) + choose(8, 2) + choose(9, 2)) / choose(socks.total, 2)
match.black.socks = choose(7, 2) / choose(socks.total, 2)

print(match.socks)
```

“

[1] 0.307971

```
print(match.black.socks)
```

“

[1] 0.07608696

(b) Design a simulation experiment (Hint: Use *sample()* in R), which repeats the random process of choosing two socks for 5,000 times.

What is the *simulated probability* that they match? What is the *simulated probability* that a black pair is chosen?

```
f.socks.sim = function (times, token) {
  s1 = 0
  s2 = 0
  s3 = 0

  for (i in 1:times) {
    result = sample(24, 2)

    has.black = all(result >= 1  & result <= 7)
    has.blue  = all(result >= 8  & result <= 15)
    has.green = all(result >= 16 & result <= 24)

    if (has.black) {
      s1 = s1 + 1
    } else if (has.blue) {
      s2 = s2 + 1
    } else if (has.green){
      s3 = s3 + 1
    }
  }

  switch(token,
         'black' = {
           return(s1 / times)
         },
         'blue' = {
           return(s2 / times)
         },
         'green' = {
           return(s3 / times)
         },
         'all' = {
           return((s1 + s2 + s3) / times)
         })
}
f.socks.sim(5000, 'all')
```

“

[1] 0.2994

```
f.socks.sim(5000, 'black')
```

“

[1] 0.0726

Q3. (20%) A committee consists of five Chicanos, two Asians, three African Americans, and two Caucasians. A subcommittee of five is chosen at random.

(a) What is the *exact probability* that all the ethnic groups are represented on the subcommittee (please mathematically derive the probability)?

```
committee=5+2+3+2
prob=choose(5,1)*choose(2,1)*choose(3,1)*choose(2,1)*choose(committee-4,1)/factorial(2)/choose(12,5)
print(prob)
```

“

[1] 0.3030303

(b) Design a simulation experiment (Hint: Use *sample()* in R), which repeats the random process of choosing five guys for 5,000 times. What is the *simulated probability* of all the ethnic groups are represented on the subcommittee?

```
f.committee.sim=function(times){
  count=0

  for (i in 1:times) {
    result = sample(12,5)

    has.Chicanos=any(result>=1 & result<=5)
    has.Asians=any(result>=6 & result<=7)
    has.AfricanAmericans=any(result>=8 & result<=10)
    has.Caucasians=any(result>=11 & result<=12)

    if (has.Chicanos&has.Asians&has.AfricanAmericans&has.Caucasians) {
      count=count+1
    }
  }
  return(count / times)

}
f.committee.sim(5000)
```

“

[1] 0.3052

Is it close to the probability in part (a)?

```
diff=prob-f.committee.sim(5000)
print(diff)#ans:yes
```

“

[1] -0.008769697

Q4. (20%) Simulating Blackjack

(a) In the poker game Blackjack, let each of the 4 aces denote 11 points and each of the 16 cards >= 10 (i.e., 10, J, Q, K) denote 10 points. Suppose someone picks 2 cards randomly out of a deck of cards (52 cards total), what is the probability of getting 21 points (i.e., Blackjack)?

Use the *choose* function to obtain the answer.

```
denominator = choose(52, 2)
classical = choose(4, 1) * choose(16, 1) / denominator
print(classical)
```

“

[1] 0.04826546

(b) Now, let the numbers 1-52 represent a deck of cards. Assign the numbers 1-4 to the four aces and the numbers 37-52 to the 16 cards > 10.

Use the *sample* function and the *for* loop to simulate the random draw of 2 cards 50,000 times (hint: with or without replacement?). Create a variable *success* that represents the number of times you hit 21 points (i.e., Blackjack). Divide success by 50,000 to obtain the relative probability. Is the relative probability close to the classical one from part (a)?

```
f.bj.sim = function(times) {
  success = 0

  for (i in 1:times) {
    result  = sample(52, 2)
    has.ace = any(result <= 4)
    has.ten = any(result >= 37 & result <= 52)

    if (has.ace & has.ten) {
      success = success + 1
    }
  }
  return(success / times)
}

relative = f.bj.sim(50000)

print(relative)
```

“

”

[1] 0.04766

```
print(paste0("Classical Probability minus Relative Probability is :: " , classical - relative))
```

“

[1] "Classical Probability minus Relative Probability is :: 0.000605460030165914"

What will happen to the relative probability if you only simulate the game for 50 times?

```
f.bj.sim(50)
```

“

[1] 0.06

Q5. (15%) Assume that the last 4 digits of one's ID in NCCU could be any number between 0000-9999. It is POSSIBLE for two guys to have identical last 4 digits of their ID. User *R* to

(a) Find that the actual probability that at least two students in a class of 100 share the same last 4 digits of their ID.

```
1-prod(c(9901:10000)/10000)
```

“

[1] 0.391434

(b) Simulate the last 4 digits for 100 students 5,000 times. How many times do you find at least two students have the same ID? Divide the number by 5,000, what is the fraction?

```
sameID=function(){
  S=sample(c(0:9999),100,replace=TRUE)
  n=length(unique(S))
  return(n)
}
F=replicate(5000,sameID())
sum(F<100)
```

“

[1] 1924

```
sum(F<100)/5000
```

“

[1] 0.3848

(c) What is the smallest class enrollment (i.e., number of students) for which the probability that at least two students have the same ID numbers is at least 0.5?

```
findN=function(){
  for(n in 1:10000){
    p=1-prod((10000-n+1):10000/10000)
    if(p>=0.5){
      return(n)
    }
  }
}
findN()
```

“

[1] 119
