DATA 252 / DATA 551: Homework 5 Solution

Answers are provided below in the form of what you would fill out on the upcoming exam. Code is included at the end. If the answer is based on a simulation, you would get a slightly different numeric value, but the difference should not be too big.

1. Use an acceptance-rejection sampling algorithm to sample from the density function $f(x) = 4(1-x)^3, 0 < x < 1$, with a proposal density $g(x) = 1, 0 < x < 1$ and a constant $C = 4$.

(1) What is the mean of the simulated draws?
SOLUTION: 0.197867

(2) What is the acceptance rate?
SOLUTION: 0.2523

2. Calculate the following probabilities.

(1) Suppose you get spam emails at a constant rate of 8 emails per day. What is the probability that you get exactly 2 spam emails in the next 12 hours?
SOLUTION: 0.1465

(2) Among 20 patients that have a certain disease with a recovery rate of 96%, what is the probability that all of them will recover?
SOLUTION: 0.4420

(3) A baker blends 500 raisins into a dough mix and makes 200 cookies. What is the probability that a randomly selected cookie has more than 5 raisins?
SOLUTION: 0.0420

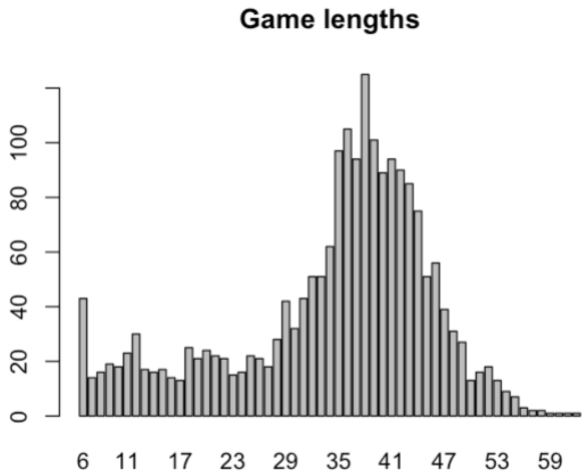
(4) Suppose the time required to repair a car is an exponentially distributed random variable, with an average repairing time of 2.8 hours. What is the probability that the repair time is less than 4 hours?
SOLUTION: 0.7603

3. In roulette, the probability of red is 18/38. Suppose you bet on red and use the *martingale doubling system*: every time you win, bet \$1 next time; every time you lose, double your previous bet. Suppose you play this game until you have win more than \$20 or you have lost more than \$50. Use simulation to answer the following questions.

(1) What is the probability that you end up winning more than \$20 at the end? SOLUTION: 0.6885

(2) On average, how many games would you be playing? SOLUTION: 34.242 games

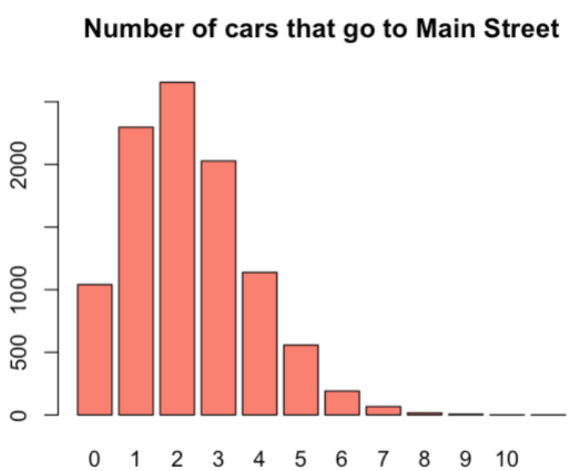
(3) Sketch a histogram of the simulated game lengths.
SOLUTION:



4. Suppose that when cars drive past Drew toward downtown, 41% will turn onto Kings Rd., 38% will continue straight to Main St., and 21% will turn onto Park Ave. Assume that the number of cars driving past Drew toward downtown follows a Poisson distribution with parameter $\lambda = 6$. Let X be the number of cars that will drive on Main St. past the fork. Use simulation to answer the following questions.

(1) What is the probability that X exceeds 5 cars?
SOLUTION: around 3%

(2) Sketch a histogram of the distribution of X.
SOLUTION:



5. Suppose you play a gambling game as follows. A fair coin is flipped continuously. Every time you see a head, you win \$1. Of course, you have to pay to play this game:

- If you pay \$2, the game stops when you see the first tail.
- If you pay \$4, the game stops when you see the first two consecutive tails.

- If you pay \$8, the game stops when you see the first three consecutive tails.
- If you pay \$16, the game stops when you see the first four consecutive tails.
- ...

Use simulation to answer the following questions.

(1) If you pay \$2 to play this game, what is the probability that you win some money (i.e., net winning > 0)?
SOLUTION: around 12%

(2) If you pay \$1024 to play this game, what is the probability that you win some money (i.e., net winning > 0)?
SOLUTION: around 37% (Caution: it takes a while to run this simulation because each game can go on for a very long time.)

(3) If you pay \$2 to play this game, what is your *expected* net winning (that is, if you play for a large number of times, what is your average net winning)? What if you pay \$4? \$8? What do you think is the optimal game playing strategy?
SOLUTION: This question is somewhat open-ended. You can choose a few scenarios to discuss. If you pay \$2, the expected net winning is around -\$1. While only 12% of the time you'll be earning some money, you will only lose \$2 at most. If you pay \$16, the expected net winning is around -\$1.2, but 32% of the time you'll be earning some money. If you pay \$1024, the expected net winning seems to be around -\$20, but the net winning has a very large spread (either lose big or win big). In each case, the distribution of net winning is right-skewed: most of the time you'll lose money, but occasionally you'll make a large amount of money. The more you pay, the higher the chance that your net winning is positive, but the lower the expected net winning. (In my opinion, the optimal strategy is to pay \$2 each time...since it seems less risky.)

```
#---------------------------------------------#
# Q1
#---------------------------------------------#
nsim=10000
result=numeric()

for(i in 1:nsim){
  xprop=runif(1)
  u=runif(1)
  acc.rate=4*(1-xprop)^3/((4)*dunif(xprop))
  if(u < acc.rate){
    result=c(result,xprop)
  }
}

mean(result)
length(result)/nsim #acceptance rate

#---------------------------------------------#
# Q2
#---------------------------------------------#
dpois(2,lambda=4)
dbinom(20,size=20,prob=0.96)
1-sum(dpois(c(0:5),lambda=500/200))
pexp(4,rate=1/2.8)

#---------------------------------------------#
# Q3
# See solution for HW1 for more details
#---------------------------------------------#
nsim=2000 #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=200 #max number of runs of while loop

  #---updating---#
  while( (xcurrent<20) && (xcurrent>-50) && (ncurrent<nmax)){
```

```r
    #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
  }

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

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

mean(result.win>=20)
mean(result.length)
plot(as.factor(result.length),main='Game lengths')

#--------------------------------------------#
# Q4
# See solution for HW3 for more details
#--------------------------------------------#
nsim=10000
result=rep(NA,nsim)

for(i in 1:nsim){
  numcar=rpois(1,lambda = 6)
  result[i]=rbinom(1,size=numcar,prob=0.38)
}
mean(result>5)
plot(as.factor(result),main='Number of cars that go to Main Street',col='salmon')

#--------------------------------------------#
# Q5
#--------------------------------------------#
```

```r
########################
#use rle function for streak
#below is a demo
test=sample(c(0,1),size=30,replace=T)
test
rle(test)$lengths
rle(test)$values
########################

k=3 #how many consecutive tails for the game to end
nsim=5000 #number of games to play
result.win=rep(NA,nsim)
nmax=20000 #max number of runs of while loop

for(i in 1:nsim){

  #---simulate one game---#
  toss=sample(c(0,1),1) #one toss of coin, 0=T, 1=H
  streak=rle(toss) #record streak information

  # keep playing if the current streak is less than k
  #    or if the current streak is Head
  nrun=1
  while( (tail(streak$lengths,n=1)<k | tail(streak$values,n=1) == 1) && nrun<nmax){
    toss=c(toss,sample(c(0,1),1))
    streak=rle(toss)
    nrun=nrun+1
  }
  if(nrun>=nmax){print('Warning: while loop reached max iteration')}
  #-----------------------#

  #record winning of the game
  result.win[i]=sum(toss)

  if(i%%50==0){print(paste0('updating simulation #',i))}
}

net.winning=result.win-(2^k) #winning-how much you've paid
mean(net.winning>0) #prob. of winning
mean(net.winning) #expected net winning
```