R Examples:

1. Patents come into the hospital since 9:00 am in the morning. Assume all the patent will come one after another. And assume if one person arrives at time t, the next person will arrive at t + z time. Here z is from exponential distribution with 1/mean equal 0.25 hours. Now assume a doctor will see any patient who arrives at the hospital between 9:00 am to 4pm. And the doctor will leave until he finishes seeing all the patients. Assume the time needed to see each patient is from Uniform(1/12 hour, 1/3 hour). Please simulate and answer the following questions:

a. Q1: Please simulate the arrival times for these patients.

b. Q2: On average, how many patients does the doctor need to see.

c. Q3: On average, how many patients will need to wait.

d. Q4: On average, when will the hospital close?

open = 9;close = 16;accrue = open

arrive = NULL

while(accrue < close){

arrive = c(arrive, rexp(1,4)) # time a new patient arrives

accrue = open + sum(arrive)

}

N = length(arrive) #number of patients

X = runif(N, min=5/60, max=20/60) # time seeing a doctor

W = rep(0,N) # wait time

count = 0

tick = open + arrive[1] + X[1]

for(i in 2:N){

if(arrive[i] > (W[i-1]+X[i-1])){

W[i] = 0 # no need to wait

tick = tick + arrive[i] + X[i]

}else{

count = count + 1

W[i] = W[i-1] + X[i-1] - arrive[i]

tick = tick + W[i] + X[i]

}

}

# mean wait time

mean(W)

2. Please Simulate pi.

MC1 <- function(n){ k <- 0; x <- runif(n); y <- runif(n) for (i in 1:n){ if (x[i]^2+y[i]^2 < 1) k <- k+1 } 4\*k/n }

3. Simulate two samples x, and y, both from normal(0, 1) distribution, and length(x) = 1000, and length(y) = 1000. Conduct a two sample t-test on x and y. Save the p-value from this two sample t-test. Repeat for 10000 times. Plot the distribution of the p-values.

N = 10000

p\_value = rep(0, N)

for(i in 1:N){

set.seed(i)

x = rnorm(1000)

y = rnorm(1000)

z = t.test(x,y)

p\_value[i] = z$p.value

}

hist(p\_value)

sum(p\_value<0.05)

4. What is the expected number of cards that need to be turned over in a regular 52-card deck in order to see the first ace?

| A | A | A | A |

| | | | | | | | |... |

| || A | | | A | | | A | || | A | | || |

Similar but not exactly a geometric distribution. 1 + 48/5 = 10.6. five places to hold the remaining 48 cards. For the first place, the average is 48/5. Use simulation/numerical method to get the estimate.

> pvec = rep(0, 48)

> npvec = rep(0, 48)

> pvec[1] = 4/52 # succeed at this step

> npvec[1] = 1-pvec[1] # not succeed at this step

> for(i in 2:48){

+ p = 4/(52-i+1) # stop at this step

+ pvec[i] = npvec[i-1]\*p

+ npvec[i] = 1-sum(pvec[1:i])

+ }

> sum(pvec\*c(1:48))

[1] 10.59982

# solution 2

N = 10000

first\_appear = rep(0, N)

for(i in 1:N){

card = c(rep(1, 4), rep(0, 48))

set.seed(i)

card\_permute = sample(card, length(card))

first\_appear[i] = min(which(card\_permute==1))

}

sum(first\_appear)/N

5. Two sport teams A and B, 7 games in total, whoever reaches 4 wins is the winner, if we assume Markov chain, P(A wins game k|A wins game k-1)=p11, P(A wins game k|A lost game k-1)=p01, How do we use simulation to get P(game 7 is played)?

# p11: A won game k given A won game k-1

# p01: A won game k given A lost game k-1

# p0: A won the first game

generateAsample = function(p11=0.6, p01=0.4, p0=0.5){

notgameover = T

outcomes = rep("none",7)

i = 1

outcomes[i] = sample(c("A","B"),1,p0)

while(notgameover){

i = i + 1 # game continues

if(outcomes[i-1] == "A")

outcomes[i] = sample(c("A","B"),1,p11)

else

outcomes[i] = sample(c("A","B"),1,p01)

if(sum(outcomes=="A")>=4 || sum(outcomes=="B")>=4)

notgameover = F

}

return(outcomes)

}

N = 10000

result = rep(0, N)

for(i in 1:N){

set.seed(i)

outcome = generateAsample(0.6, 0.4, 0.5)

# count how many rows with 7 plays

if(outcome[7]!='none') result[i] = 1

}

sum(result)/N