**Desarrolle la simulación según sea el caso.**

**SIMULACIÓN DE MONEDAS**

#Crea un vector de valores cara y sello

moneda=c("cara","sello")

moneda[1]

moneda[2]

#Lanzar moneda una vez

sample(moneda,1,F)

#Lanzar moneda dos veces

sample(moneda,2,T)

#Lanzar moneda tres veces

sample(moneda,3,T)

**SIMULACIÓN DADOS**

#Tres formas diferentes de crear dados.

dado1=c(1,2,3,4,5,6)

dado2=seq(1,6)

dado3=c(1:6)

#Lanzar el dado

sample(dado1,2,T)

a=sample(dado1,1200,T)

table(a)

table(a)/1200

#Lanzar dos dados n veces y guardar el resultado en una matriz

#gana el jugador que mas veces saca el numero mayor.

dosdados=function(n)

{

dado1=c(1,2,3,4,5,6)

dado2=seq(1,6)

a=cbind(sample(dado1,n,T),sample(dado2,n,T))

R=rep(0,n)

for (i in 1:n) {

if(a[i,1]>a[i,2]) { R[i]=1}

}

G= table(R)

if(G[1]>G[2]) {

print("gana El primer jugador")} else {if(G[1]<G[2]) {

print("gana El segundo jugador") } else {print("empate") }

}

}

dosdados(100)

**SIMULACIÓN NAIPES**

#Crea naipe, se crean los vectores de los numeros y de los simbolos.

num=rep(1:13,4)

palo1=c(rep("corazon",13),rep("trebol",13),rep("espada",13),rep("diamante",13))

palo2=rep(c("corazon","trebol","espada","diamante"),each=13)

naipe=cbind(num,palo2)

#Barajar las cartas

naipe[sample(1:52,52,F),]

g=sample(1:52,4,F)

Jugador1=naipe[g,]

naipe2= naipe[-g,]

Jugador2=naipe2[sample(1:48,4,F),]

#MONEDA

#Se lanza la moneda n veces y se calcula la probabilidad de seleccionar cara,

#esta funcion grafica la funcion de probabilidad de seleccionar cara luego de n lanzamientos.

simulacion2=function(n)

{

numlan=seq(1,n)

lanza=sample(0:1,n,T)

proba=cumsum(lanza)/numlan

print(proba)

plot(numlan,proba,type="l")

abline(h=0.5)

}

simulacion2(150)

#ANALIZA LA FUNCION

simulacionM=function(n)

{

numlan=seq(1,n)

resul=rep(0,n)

for (i in 1:n)

{

resul[i]=sum(sample(0:1,numlan[i],T))

}

proba=resul/numlan

plot(numlan,proba,type="l")

abline(h=0.5)

}

simulacionM(500)

**SIMULACIÓN 3DADOS**

#se lanza tres dados y se calcula el valor de la probabilidad del numero maximo del lanzamiento.

simulacion3dados=function(n)

{

dado=seq(1,6)

resul=matrix(0,n,3)

for (i in 1:n)

{

resul[i,]=sample(dado,3,T)

}

maxi=apply(resul,1,max)

proba=table(maxi)/n

print(proba)

}

simulacion3dados(10000)

Box Muller

bm<-function(n,mu,sigma){

num<-rep(0,n)

for (i in 1:n){

U1<-runif(1)

U2<-runif(1)

num[i]<-sqrt(-2\*log(U1))\*cos(2\*pi\*U2)

}

z<-num\*sigma+mu

return(z)

}

set.seed(15)

dat2<-bm(30,6,2)

shapiro.test(dat2)