Simulación de la Probabilidad

Simulación Estadística

Diana Carolina Arias Sinisterra Cod. 1528008
Cesar Andres Saavedra Vanegas Cod. 1628466

Universidad Del Valle
Facultad De Ingenieria
Estadistica
Febrero
2018

1. Situación 1

- **Parte A:** En la situación 1 se plantea un escenario para el cual se tiene una población de 100 chips de memoria, de los cuales 90 se encuentran en buen estado y 10 dañados. Debemos escoger 5 entre los 100 para su uso, por medio del software estadístico R usamos la función "sample"para dar una aproximación del valor de la probabilidad de que los 5 chips sean buenos.
  Mediante la simulación se logra obtener como resultado aproximado una probabilidad de 0.61 de que los 5 chips escogidos de manera aleatoria para reparar la computadora sean buenos.

- **Parte B:** Tomando como base los resultados obtenidos en la Parte A. se propone utilizar una forma alterna para seleccionar los chips y de esta manera calcular la probabilidad y que permita obtener un resultado mucho mas fiel a la realidad y a lo mostrado en el punto anterior. Para los cual decidimos usar la distribución Hipergeometrica para realizar dicha selección con un mejor ajuste obteniendo de esta forma una probabilidad de 0.5837.

2. Situación 2

- **Parte A:** Para la situación 2, contamos con un problema matemático denominado como "La paradoja del cumpleaños."en el cual se plantea encontrar la probabilidad de que dos o mas personas tengan la misma fecha de cumpleaños. Para verificar dicha hipótesis en la cual a mayor numero de personas en un cuarto mayor es la probabilidad de coincidencia se realizo una simulación y los resultados se presentan mediante las siguiente gráfica.

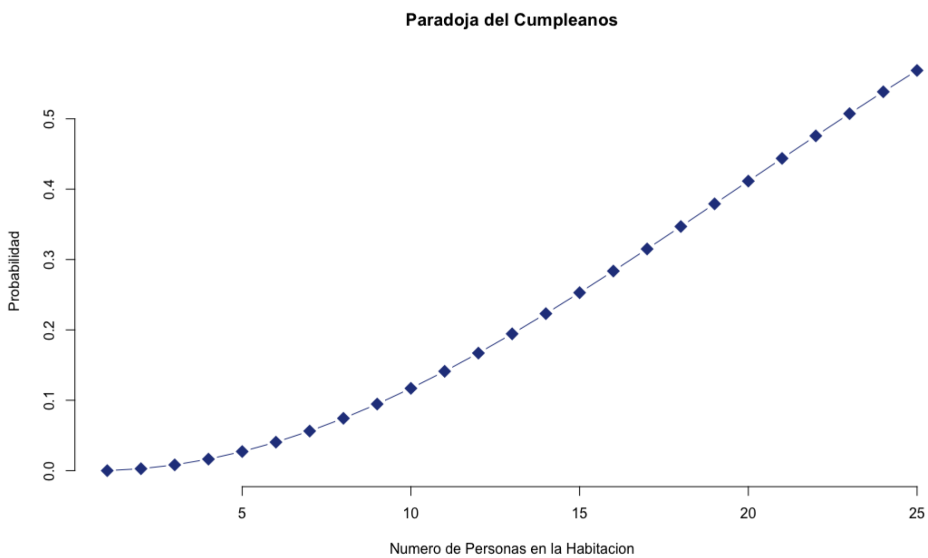

Figura 1: Probabilidad de coincidencia para 25 personas

Destacar que a medida que el numero de personas en la habitación se incrementa la probabilidad de encontrar coincidencia es mayor, algunas de estas probabilidades se muestra mediante la tabla:

Individuos	Probabilidad
0	0.000
5	0.02713
15	0.2529
25	0.5686

- **Parte B:**

3. Situación 3

- **Parte A:** La relación que existe entre la función "samplez çhoose.es que la primera toma una muestra del tamaño especificado de los elementos del vector x con o sin reemplazo, y la segunda realiza la combinatoria entre dos números. Para este caso particular: Sample toma 5 chips de computadora al azar de los 100 de la muestra. Choose Realiza la combinatoria de las distintas formas en las que se puede seleccionar los 5 chips de computadora.
- **Parte B:** Al calcular la probabilidad de $P(x = 2)$ usando las funciones "dhyper y choose"se obtiene como resultado que ambos metodos entregan como probabilidad 0.00638352.
- **Parte C:** Al simular la media de chips buenos se obtiene como resultado un valor de para la media de 4.41
- **Parte D:** Al ejecutar la linea de código sum((0:5)*dhyper(0:5,90,10,5)) se obtiene como resultado 4.5 que es la media de los chips en buen estado, valor próximo al obtenido anteriormente por medio de la simulación.

4. Situación 4

- **Parte A:**

* a = c(5, 6, 7, 6, 8, 7): Crea un vector con los componentes.
* length establece la longitud de los vectores (incluyendo listas)y los factores, de cualquier objeto en el que se haya definido un metodo, por lo tanto al ser aplicado en el vector a este nos aroja como resultado un 6 lo que nos indica los elementos de este.
* unique es una función que nos devuelve el vector solo con los elementos no duplicados, es decir, si se observa el vector a entre sus elementos el número 6 y 7 estan duplicados, por tanto en el momento de usar unique este me devuelve un vector sin elementos repetidos, por tanto, el vector queda a=5,6,7,8
* Ahora utilizamos las funciones length y unique en conjunto de manera que una queda en función de la otra, es decir length(unique(a)), como length se encarga de contar el número de elementos y unique de mostrarme los elementos sin su duplicado entonces como resultado de este caso obtenemos el conteo de los elementos del vector sin sus valores repetidos, por tanto el resultado es 4, como se pudo observar en el punto anterior el vector quedo compuesto por 4 elementos.
* Como la función length nos arroja es el número de elementos del vector y la función compuesta de length(unique(a)) tambien nos da como resultado el número de elementos pero sin duplicados se puede operar entre estos resultados, es decir, length(a) - length(unique(a)), como resultado nos arroja un 2que es el resultado de restar los resultados que nos arrrojo el primer y tercer punto ya que como se observo en el primer punto se contaron en total 6 elementos y en el tercer punto el conteo dio 4 elementos
* duplicated es una función que determina que elementos de un vector o marco de datos son duplicados, por tanto este me devuelve un vector lógico que indica que elementos(filas)estan repetidos, por tanto el resultado de aplicar esta función al vector a obtenemos (FALSE, FALSE, FALSE,TRUE,FALSE,TRUE),lo que indica que Duplicated va leyendo el vector y va informando que el elemento no esta repetido a lo que describe con un false, cuando ya encuentra el elemento duplicado lo cataloga como un True.
* Ahora colocamos length(duplicated(a)), como length es la encargada de contar los elementos en un vector lo que hace es contar cuantos elemetos arroja duplicated, por tanto el resulado obtenido es 6 valores como se observo en el punto 4, el cual fueron 2 valores True y 4 valores False.
* La función sum es la encargada de retornar todos los valores presentes en el vector, como duplicated me muestra dos valores que corresponde a False y True entonces al poner una en funcion de la otra así sum(duplicated(a)) como resultado obtenemos 2.

- **Parte B:** Mediante la simulación usando las funciones mencionadas en el punto anterior se busco calcular la probabilidad de emparejar a las 25 personas, lo que dio como resultado una probabilidad del 0.3.

5. Situación 5

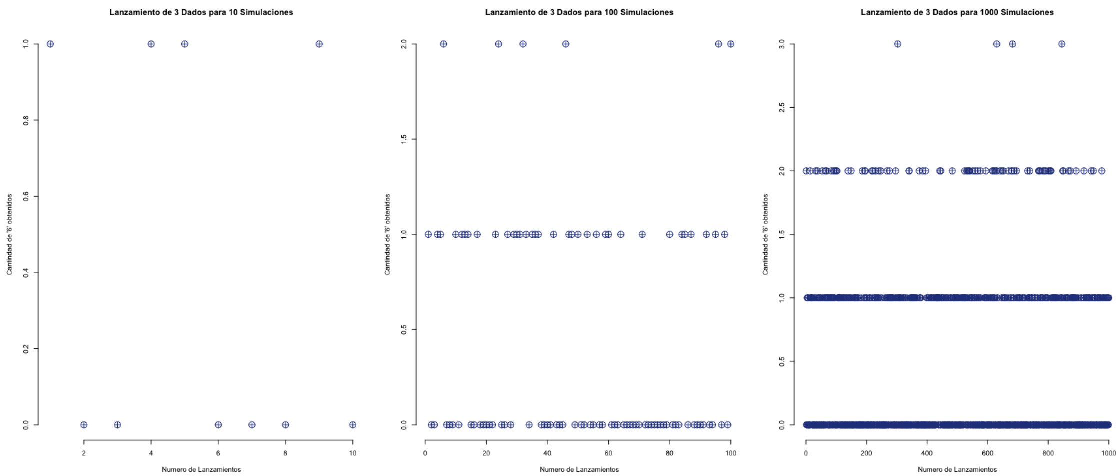

Figura 2: Simulación para el lanzamientos de 3 Dados

Para la simulación del lanzamiento de los dados se puede observar mediante la gráfica, como la probabilidad de obtener los 3 seises es muy pequeña pues es de 0.02 con lo que es puede asegurar que las ganancias al ser proporcionales al numero de 6 que obtenga el jugador no serán las mejores.

6. Situación 7

- **Parte A:** Mediante la función choose se puede calcular por combinatoria la probabilidad de que un juego de póquer no tenga Aces, dicho resultado obtenido por medio de tecnicas de probabilidad fue de 0.658842.
- **Parte B:**

7. Situación 8

Se Utiliza el método de integración aproximado Monte Carlo para estimar

$$\int_0^{\pi/4} log(1 + tan^2(x)) \cdot dx \tag{1}$$

Para ello se crearon variables aleatoria con la distribución uniforme ,ya que este método de integración solo es valido para muestras aleatorias, pues este método son algoritmos para encontrar una evaluación aproximada de una integral definida. La precisión del método se mejora utilizando una gran cantidad de simulaciones para este caso se uso un n=1000000 de manera que como resultado se obtuvo un valor de 0.1729043
Mientras que el realizar la integral sin variables aleatorias me arroja el siguiente resultado 0.172827 por lo anterior se puede decir que el método de integración de Montecarlo me arrojo un resultado más exacto.

8. Situación 6

Al realizar la simulación para cada una de las distribuciones mencionadas, se observa como al cambiar el valor del parametro de referencia la grafica se va moviendo, haciéndose mas ancha y decreciendo.

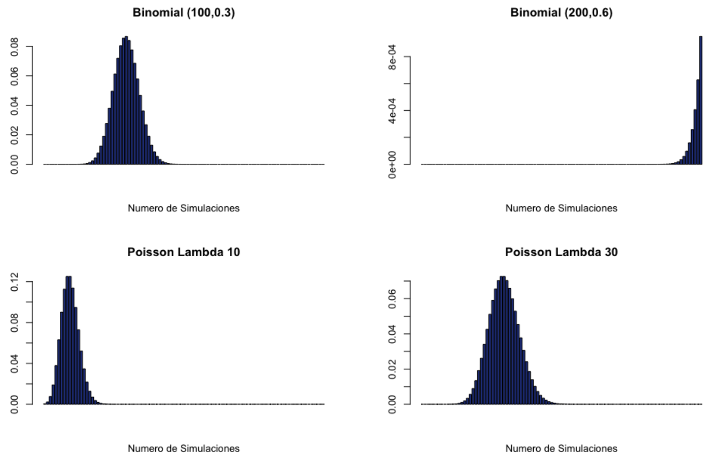

Figura 3: Simulación para las variables aleatorias Binomial y Poisson

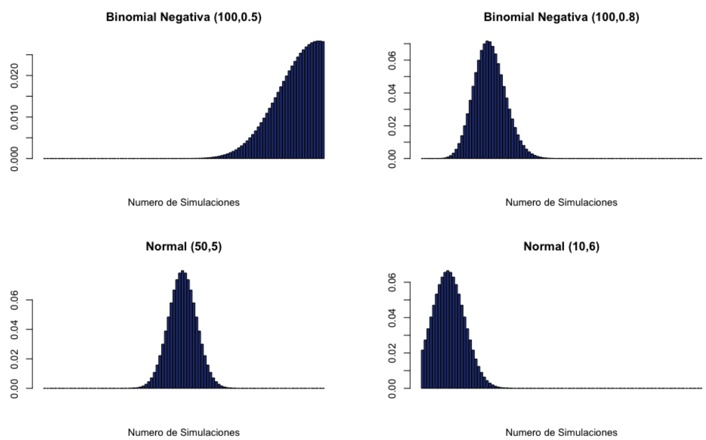

Figura 4: Simulación para las variables aleatorias Binomial Negativa y Normal

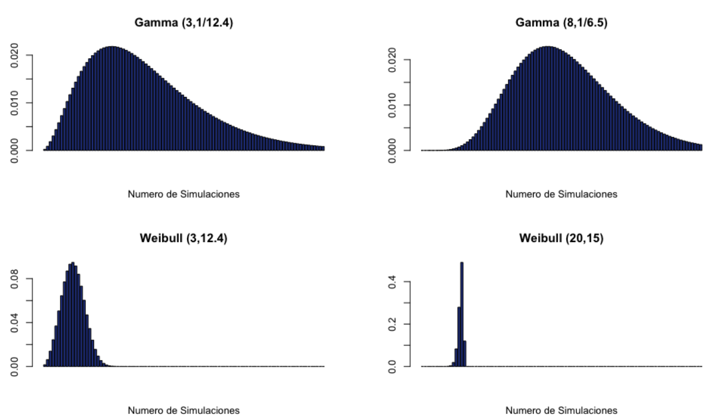

Figura 5: Simulación para las variables aleatorias Gamma y Weibull

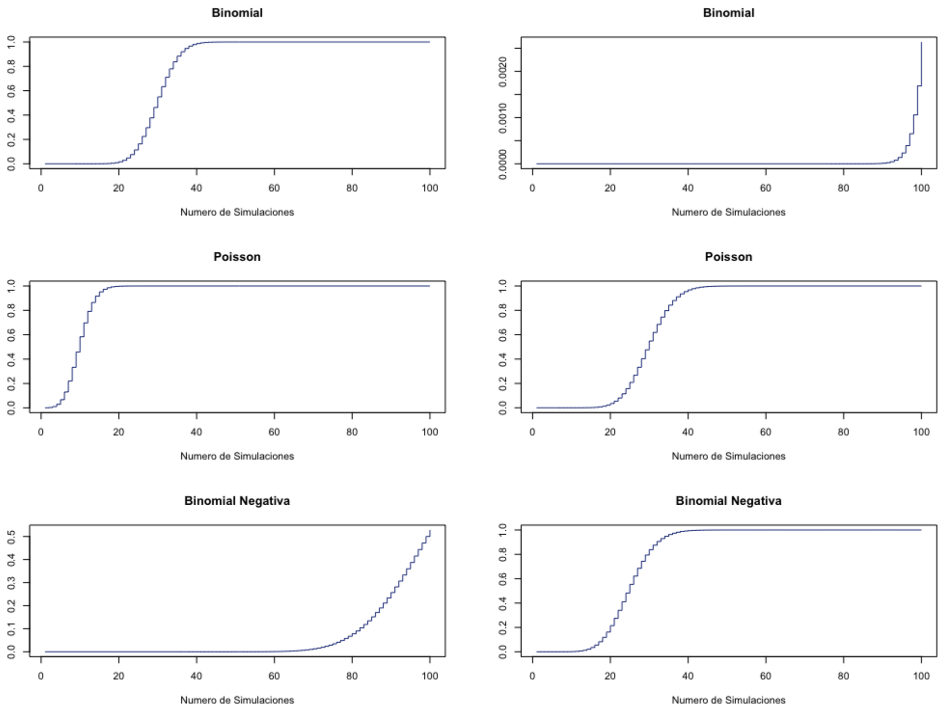

Figura 6: Simulación para las variables aleatorias Binomial, Poisson y Binomial Negativa

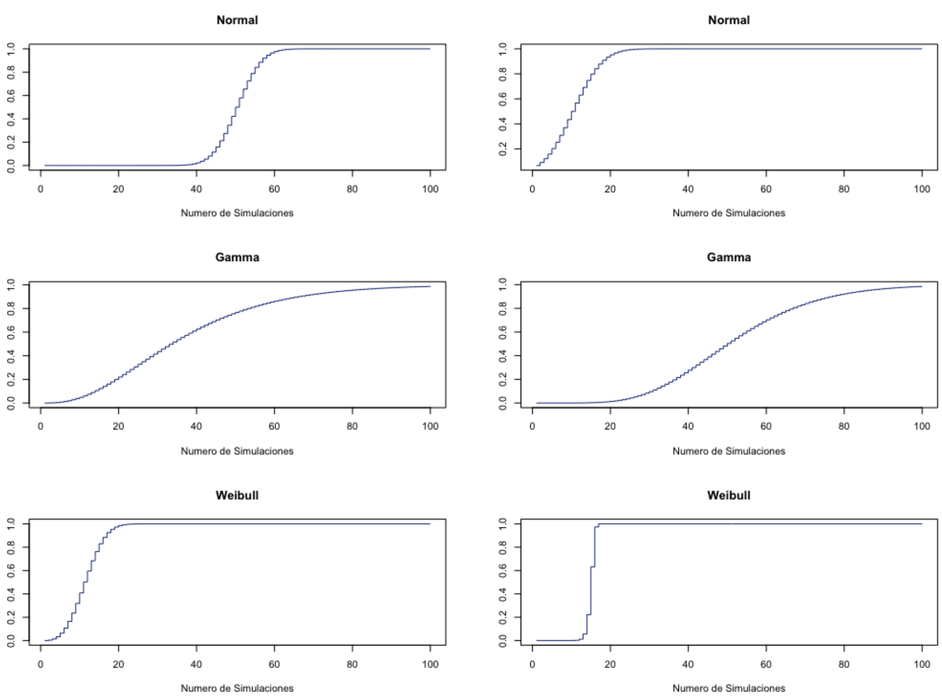

Figura 7: Simulación para las variables aleatorias Normal, Gamma y Weibull

9. Bibliografia

- **B. P. Demidowitsch. I. A. Maron, E. S. Schuwalowa. Métodos numéricos de análisis. Editorial Paraninfo (1980)**
- **http://www.bdigital.unal.edu.co/4748/2/nelcyyazminninoalfonso.2011.parte2.pdf**
- **https://www.uam.es/personalpdi/ciencias/carlosp/html/pid/montecarlo.html**
- **http://www.estadisticaparatodos.es/taller/cumpleanos/cumpleanos.html**

10. Scripts

```
#------------------------------------------------------------#
# Diana Carolina Arias Cod. 1528008
# Cesar Andres Saavedra Cod. 1628466
# Simulacion Estadistica
# Tarea No. 2
#------------------------------------------------------------#
########################################################
########################################################
#------------------------Punto1------------------------#
#---Situacion A---#
count_bueno<-function (data,n){
  y<-0
  x<-sample(data,n,replace=F)
  for(i in 1:n){
    if(x[i]=="Buen_Estado"){
      y[i]<-1
    }else{
      y[i]<-0
    }
  }
  all_sum<-sum(y,na.rm=T)
  if(all_sum==n){
    return(1)
  }else{return(0)}
}

aprox_x<-function(n_simul){
  x<-0
  for(i in 1:n_simul){
    x[i]<-count_bueno(chips_memoria,5)
  }
  y<-sum(x)/length(x)
  return(y)
}

chips_memoria<-c(rep('Buen_Estado',90),rep('Danados',10))
aprox_x(100)


#---Situacion B---#
dhyper(5,90,10,5,log=F)


#------------------------Punto2------------------------#
#---Situacion A---#
prob <- matrix(200)
nsim=25
total=1
for (i in 1:nsim){
  total=total*((366-i)/365)
  prob[i] <- 1-total
  plot(prob, xlab="Numero de Personas en la Habitacion", ylab="Probabilidad",
       main="Paradoja del Cumpleanos", pch=18, type="b", col="royalblue4", bty="n", cex=2)
```

```
}
print(prob)

#---Situacion B---#
cuartos = 100000
personas = 1
asfd <- matrix(NA)

for(i in 1:nsim){
  personas  = personas*((100001-i)/cuartos)
  asfd[i]<- personas
}
cuartos<-100
x<-0
matrix<- matrix(NA)
for(i in 1:25){
  matrix[i] <- sample(cuartos,1)
}
matrix
print(asfd)

#------------------------Punto3------------------------#
#---Situacion A---#
sample(1:100,5)
choose(100,5)

#---Situacion B---#
dhyper(2,90,10,5)
(choose(90,2)*choose(10,3))/choose(100,5)

#---Situacion C---#
chips.memoria<-rhyper(100,90,10,5)
mean(chips.memoria)
var(chips.memoria)

#---Situacion D---#
sum((0:5)*dhyper(0:5,90,10,5))

#------------------------Punto4------------------------#
#---Situacion A---#
a=c(5,6,7,6,8,7)
length(a)
unique(a)
length(unique(a))
length(a)-length(unique(a))
duplicated(a)
length(duplicated(a))
sum(duplicated(a))

#---Situacion B---#
Habitaciones<-10
Parejas<-0
igual<-matrix(0,ncol=2, nrow=Habitaciones)

for(i in 1:Habitaciones){
  cump<-sample(1:365,25,replace=TRUE)
```

```
  igual[i,1]<-i

  igual[i,2]<- sum(duplicated.matrix(as.matrix(cump)))
  if(igual[i,2]>=1){Parejas<-Parejas+1}
}
pr<-Parejas/Habitaciones
pr

#------------------------Punto5------------------------#
#---Situacion A---#
dado <- c(1:6)
sample(dado,3,replace=T)

sim1 = 10
sim2 = 100
sim3 = 1000
conteo1 <- 0
conteo2 <- 0
conteo3 <- 0

# Para 10 simulaciones
for (j in 1:sim1){
  lanzamiento1 <- sample(dado,3,replace=T)
  total = 0
  for(i in 1:3){
    if(lanzamiento1[i]== "6"){
      total=total+1}
    else{total=total}
  }
  conteo1[j]=total
}

# Para 100 simulaciones
for (j in 1:sim2){
  lanzamiento2 <- sample(dado,3,replace=T)
  total = 0
  for(i in 1:3){
    if(lanzamiento2[i]== "6"){
      total=total+1}
    else{total=total}
  }
  conteo2[j]=total
}

# Para 1000 simulaciones
for (j in 1:sim3){
  lanzamiento3 <- sample(dado,300,replace=T)
  total = 0
  for(i in 1:3){
    if(lanzamiento3[i]== "6"){
      total=total+1}
    else{total=total}
  }
  conteo3[j]=total
}
par(mfrow=c(1,3))
```

```
plot(conteo1, pch=10, col="royalblue4" ,main="Lanzamiento de 3 Dados para 10 Simulaciones",
     xlab="Numero de Lanzamientos", ylab="Cantindad de '6' obtenidos", bty="n", cex=2)
plot(conteo2, pch=10, col="royalblue4", main="Lanzamiento de 3 Dados para 100 Simulaciones",
     xlab="Numero de Lanzamientos", ylab="Cantindad de '6' obtenidos", bty="n", cex=2)
plot(conteo3, pch=10, col="royalblue4", main="Lanzamiento de 3 Dados para 1000 Simulaciones",
     xlab="Numero de Lanzamientos", ylab="Cantindad de '6' obtenidos", bty="n", cex=2)

#------------------------Punto6------------------------#
#---Binomial---#
Nsim1=100
matriz1=matrix(200)
for(i in 1:Nsim1){
  dbinom=dbinom(i,100,0.3)
  matriz1[i]=dbinom
}
matriz2=matrix(200)
for(i in 1:Nsim1){
  dbinom=dbinom(i,200,0.6)
  matriz2[i]=dbinom
}
#---Poisson---#
Nsim2=100
matriz3=matrix(200)
for(j in 1:Nsim2){
  dpois1=dpois(j,lambda=10)
  matriz3[j]=dpois1
}
matriz4=matrix(200)
for(j in 1:Nsim2){
  dpois3=dpois(j,lambda=30)
  matriz4[j]=dpois3
}
par(mfrow=c(2,2))
barplot(matriz1, main="Binomial (100,0.3)", xlab="Numero de Simulaciones", ylab="", col="royalblue4")
barplot(matriz2, main="Binomial (200,0.6)", xlab="Numero de Simulaciones", ylab="", col="royalblue4")
barplot(matriz3, main="Poisson Lambda 10", xlab="Numero de Simulaciones", ylab="", col="royalblue4")
barplot(matriz4, main="Poisson Lambda 30", xlab="Numero de Simulaciones", ylab="", col="royalblue4")

#---Binomial Negativa---#
Nsim4=100
matriz6=matrix(200)
for(i in 1:Nsim4){
  dnbinom=dnbinom(i,100,0.5)
  matriz6[i]=dnbinom
}
matriz7=matrix(200)
for(i in 1:Nsim4){
  dnbinom=dnbinom(i,100,0.8)
  matriz7[i]=dnbinom
}
par(mfrow=c(1,2))
#---Normal---#
Nsim5=100
matriz8=matrix(200)
for(i in 1:Nsim5){
  dnorm=dnorm(i,50,5)
```

```
  matriz8[i]=dnorm
}
matriz9=matrix(200)
for(i in 1:Nsim5){
  dnorm=dnorm(i,10,6)
  matriz9[i]=dnorm
}
par(mfrow=c(2,2))
barplot(matriz6, main="Binomial Negativa (100,0.5)", xlab="Numero de Simulaciones", ylab="",
col="royalblue4")
barplot(matriz7, main="Binomial Negativa (100,0.8)", xlab="Numero de Simulaciones", ylab="",
col="royalblue4")
barplot(matriz8, main="Normal (50,5)", xlab="Numero de Simulaciones", ylab="", col="royalblue4")
barplot(matriz9, main="Normal (10,6)", xlab="Numero de Simulaciones", ylab="", col="royalblue4")
#---Gamma---#
Nsim6=100
matriz10=matrix(200)
for(i in 1:Nsim6){
  dgamma=dgamma(i, 3,1/12.4)
  matriz10[i]=dgamma
}
matriz11=matrix(200)
for(i in 1:Nsim6){
  dgamma=dgamma(i, 8,1/6.5)
  matriz11[i]=dgamma
}
par(mfrow=c(1,2))
#---Weibull---#
Nsim9=100
matriz12=matrix(200)
for(i in 1:Nsim9){
  dweibull=dweibull(i, 3, 12.4)
  matriz12[i]=dweibull
}
matriz13=matrix(200)
for(i in 1:Nsim9){
  dweibull=dweibull(i, 20, 15)
  matriz13[i]=dweibull
}
par(mfrow=c(2,2))
barplot(matriz10, main="Gamma (3,1/12.4)", xlab="Numero de Simulaciones", ylab="", col="royalblue4")
barplot(matriz11, main="Gamma (8,1/6.5)", xlab="Numero de Simulaciones", ylab="", col="royalblue4")
barplot(matriz12, main="Weibull (3,12.4)", xlab="Numero de Simulaciones", ylab="", col="royalblue4")
barplot(matriz13, main="Weibull (20,15)", xlab="Numero de Simulaciones", ylab="", col="royalblue4")
#---------------------------------------------------------------------------------------------#
#---Binomial---#
Nsim1=100
matriz1=matrix(200)
for(i in 1:Nsim1){
  pbinom=pbinom(i,100,0.3)
  matriz1[i]=pbinom
}
matriz2=matrix(200)
for(i in 1:Nsim1){
  pbinom=pbinom(i,200,0.6)
  matriz2[i]=pbinom
```

```
}
#---Poisson---#
Nsim2=100
matriz3=matrix(200)
for(j in 1:Nsim2){
  ppois1=ppois(j,lambda=10)
  matriz3[j]=ppois1
}
matriz4=matrix(200)
for(j in 1:Nsim2){
  ppois3=ppois(j,lambda=30)
  matriz4[j]=ppois3
}
#---Binomial Negativa---#
Nsim4=100
matriz6=matrix(200)
for(i in 1:Nsim4){
  pnbinom=pnbinom(i,100,0.5)
  matriz6[i]=pnbinom
}
matriz7=matrix(200)
for(i in 1:Nsim4){
  pnbinom=pnbinom(i,100,0.8)
  matriz7[i]=pnbinom
}
par(mfrow=c(3,2))
plot(matriz1, main="Binomial", xlab="Numero de Simulaciones", ylab="", col="royalblue4", type="s", pch=9)
plot(matriz2, main="Binomial", xlab="Numero de Simulaciones", ylab="", col="royalblue4", type="s", pch=9)
plot(matriz3, main="Poisson", xlab="Numero de Simulaciones", ylab="", col="royalblue4", type="s", pch=9)
plot(matriz4, main="Poisson", xlab="Numero de Simulaciones", ylab="", col="royalblue4", type="s", pch=9)
plot(matriz6, main="Binomial Negativa", xlab="Numero de Simulaciones", ylab="",
col="royalblue4", type="s", pch=9)
plot(matriz7, main="Binomial Negativa", xlab="Numero de Simulaciones", ylab="",
 col="royalblue4", type="s", pch=9)
#---Normal---#
Nsim5=100
matriz8=matrix(200)
for(i in 1:Nsim5){
  pnorm=pnorm(i,50,5)
  matriz8[i]=pnorm
}
matriz9=matrix(200)
for(i in 1:Nsim5){
  pnorm=pnorm(i,10,6)
  matriz9[i]=pnorm
}
#---Gamma---#
Nsim6=100
matriz10=matrix(200)
for(i in 1:Nsim6){
  pgamma=pgamma(i, 3,1/12.4)
  matriz10[i]=pgamma
}
matriz11=matrix(200)
for(i in 1:Nsim6){
  pgamma=pgamma(i, 8,1/6.5)
```

```
  matriz11[i]=pgamma
}

#---Weibull---#
Nsim9=100
matriz12=matrix(200)
for(i in 1:Nsim9){
  pweibull=pweibull(i, 3, 12.4)
  matriz12[i]=pweibull
}
matriz13=matrix(200)
for(i in 1:Nsim9){
  pweibull=pweibull(i, 20, 15)
  matriz13[i]=pweibull
}
par(mfrow=c(3,2))
plot(matriz8, main="Normal", xlab="Numero de Simulaciones", ylab="", col="royalblue4", type="s", pch=9)
plot(matriz9, main="Normal", xlab="Numero de Simulaciones", ylab="", col="royalblue4", type="s", pch=9)
plot(matriz10, main="Gamma", xlab="Numero de Simulaciones", ylab="", col="royalblue4", type="s", pch=9)
plot(matriz11, main="Gamma", xlab="Numero de Simulaciones", ylab="", col="royalblue4", type="s", pch=9)
plot(matriz12, main="Weibull", xlab="Numero de Simulaciones", ylab="", col="royalblue4", type="s", pch=9)
plot(matriz13, main="Weibull", xlab="Numero de Simulaciones", ylab="", col="royalblue4", type="s", pch=9)

#------------------------Punto7------------------------#
#---Situacion A---#
Combinaciones <- choose(52,5)
SinAs <- choose(48,5)
pro <- SinAs/Combinaciones

#---Situacion B---#
baraja <- c('AC','AD','AP','AT','2C','2D','2P','2T','3C','3D','3P','3T','4C','4D','4P','4T',
            '5C','5D','5P','5T','6C','6D','6P','6T','7C','7D','7P','7T','8C','8D','8P','8T',
            '9C','9D','9P','9T','10C','10D','10P','10T','JC','JD','JP','JT',
            'KC','KD','KP','KT','QC','QD','QP','QT')
sample(baraja,5,F)



#------------------------Punto8------------------------#
b = pi/4
a = 0
integral = (b-a)*mean(log(1+(tan(runif(100000, min=a, max=b)))^2))
integral
#-----------------------------------------------------------#
```