DISTRIBUCION BERNOULLI

#Funcion para generar números pseudoaleatorios de una distribución de Bernoulli con parámetro theta

bernoulli=function(n,theta)

{

uniformes=runif(n, min=0, max=1)

resultados=rep(0,n)

for (i in 1:n)

# {

if (uniformes[i]>=1-theta)

{

resultados[i]=1

}

else

{

resultados[i]=0

}

}

return(resultados)

}

muestra=bernoulli(10,0.4)

table(muestra)

En R tambien se usar un muestreo con probabilidades determinadas.

bernoullimuestra=sample(c(1,0), size=10, replace=TRUE, prob=c(.4, .6))

bernoullimuestra

table(bernoullimuestra)

DISTRIBUCION BINOMIAL

#Funcion para generar numeros pseudoaleatorios de una distribucion Binomial con parametros r y theta (primera forma)

binomial1=function(n,r,theta)

{

U=matrix(runif(n\*r),n,r)

resultados=matrix(0,n,r)

for (i in 1:n)

{

for (j in 1:r)

{

if (U[i,j]<=theta)

{

resultados[i,j]=1

}

else

{

resultados[i,j]=0

} } }

#print(resultados)

vector=apply(resultados,2,sum)

prob=table(vector)/r

#print(vector)

print(prob)

}

binomial1(6,50,0.85)

#Funcion para generar numeros pseudoaleatorios de una distribucion Binomial con parametros r y theta (segunda forma)

binomial2=function(n,r,theta)

{

x=rep(0,n)

for (i in 1:r)

{

x[i]=sum(bernoulli(n,theta))

}

print(x)

prob=table(x)/r

print(prob)

}

binomial2(6,50,0.85)

En R también se puede usar la función rbinom(n, size, prob)

Ejemplo

rbinom(5,4,0.5)

DISTRIBUCION GEOMETRICA

#Funcion para generar numeros pseudoaleatorios de una distribucion geometrica con parametro theta

geometrica=function(n,theta)

{

x=rep(0,n)

for (i in 1:n)

{

U=runif(1)

x[i]=floor(log(U)/log(1-theta))

}

print(x)

}

geometrica(12,0.4)

En R también se puede usar la función rgeom(n, prob)

Ejemplo

rgeom(12,0.4)

**DISTRIBUCION EXPONENCIAL**

#Funcion para generar numeros pseudoaleatorios de una distribucion exponencial con parametro theta

exponencial=function(n,theta)

{

uniform=runif(n)

x=-log(uniform)/theta

return(x)

}

Ejemplo :

exponencial(12,0.4)

hist(exponencial(12,0.4))

En R también se puede usar la función rexp(n, prob)

Ejemplo

rexp(12,0.4)

hist(rexp(12000,0.4))

DISTRIBUCION ERLANG

#Funcion para generar numeros pseudoaleatorios de una distribucion Erlang con parámetro alfa, theta

erlang=function(n,alfa,theta)

{

y=rep(0,n)

for (i in 1:n)

{

y[i]=sum(exponencial(alfa,theta))

}

return(y)

}

Ejemplo

erlang(10,12,0.4)

hist(erlang(100,122,0.4))

DISTRIBUCION BETA

#Funcion para generar numeros pseudoaleatorios de una distribucion beta con parametro alfa, beta

beta1=function(n,alfa,beta)

{

x=rep(0,n)

for(i in 1:n)

{

x1=rgamma(1,alfa,rate=1)

x2=rgamma(1,beta,rate=1)

x[i]=x1/(x1+x2)

}

print(x)

}

Ejemplo

beta1(12,0.4,0.5)

hist(beta1(1200,0.4,0.5))

beta2=function(n,alfa,beta)

{

x=rep(0,n)

for(i in 1:n)

{

x1=erlang(1,alfa,1)

x2=erlang(1,beta,1)

x[i]=x1/(x1+x2)

}

print(x)

}

Hist(beta2(1200,3,3))

DISTRIBUCION NORMAL

#Funcion para generar numeros pseudoaleatorios de una distribucion Normal con parametros mu y sigma

**normal1=function(n,mu, sigma)**

{

x=rep(0,n)

for(i in 1:n)

{

u1=runif(12, min=0,max=1)

x[i]=(sum(u1)-6)

}

y=mu+sigma\*x

return(y)

}

hist(normal1(120,54, 10))

**normal2=function(n,mu,sigma)**

{

x=rep(0,n)

y=rep(0,n)

for(i in 1:n)

{

u1=runif(1, min=0,max=1)

u2= runif(1, min=0,max=1)

R=sqrt(-2\*log(u1))

Theta=2\*pi\*u2

x[i]=R\*cos(Theta)

y[i]=R\*sin(Theta)

A=mu+sigma\*x

B=mu+sigma\*y

}

return(list(A=A,B=B))

}

Ejemplo

normalmuestra=normal2(1000,54,10)

hist(normalmuestra$B)

hist(normalmuestra$A)

**En R se tiene la función rnorm**

**Ejemplo:**

rnorm(10,54,30)

hist(rnorm(10,54,30))

**PRUEBAS DE BONDAD DE AJUSTE**

#Pruebas de Bondad de ajuste y Normalidad

**Pueba ks para comparar si dos variables provienen de la misma** **distribución.**

Sean las v.a. x e y

x <- rnorm(50)

y <- runif(30)

ks.test(x, y)

**Prueba ks para comparar dos va con distribución normal**

x <- rnorm(50)

x2 <- rnorm(50, -1)

plot(ecdf(x), xlim = range(c(x, x2)))

plot(ecdf(x2), add = TRUE, lty = "dashed")

ks.test(x, x2, alternative = "l")

**Prueba Wilcoxon**

wilcox.test(x, x2, alternative = "g")

#Prueba de Shapiro Wilk

a=rnorm(50,54,32)

shapiro.test(a)

**Propuesto, revisar la bibliografía para estas otras pruebas en R**

#Prueba de Anderson-Darling

library(nortest)

ad.test(a)

#Prueba de Cramer-von Mises

cvm.test(a)

#Prueba de Lilliefors

Lillie.test(a)

#Prueba Chi Cuadrado de Pearson

pearson.test(a)

#Prueba de Shapiro-Francia

sf.test(b)