#Estimador bootstrap para un estadistico univariado

bootstrap2=function(datos,B,fun,...)

{

n=length(datos)

estimac=rep(0,B)

for (i in 1:B)

{

estimac[i]=fun(sample(datos,n,T),...)

}

estboot=mean(estimac)

eeboot=sd(estimac)

sesgo=estboot-fun(datos,...)

ecor=2\*fun(datos,...)-estboot

hist(estimac)

return(list(estimador=estboot,eestandar=eeboot,sesgob=sesgo,ecor=ecor))

}

bootstrap2(rock[,1],100,quantile,0.4)

#Numero de muestras bootstrap para un valor de B

nmb1=function(datos,B,r,fun,...)

{

ee=rep(0,r)

for (i in 1:r)

{

ee[i]=bootstrap2(datos,B,fun,...)$eestandar

}

cv=sd(ee)/abs(mean(ee))

print(cv)

}

nmb1(rock[,1],10,100,quantile,0.4)

nmb1(rock[,1],50,100,quantile,0.4)

nmb1(rock[,1],100,100,quantile,0.4)

system.time(nmb1(rock[,1],100,100,quantile,0.4))[3]

#Numero de muestras bootstrap para un vector B

nmb2=function(datos,B,r,fun,...)

{

b=length(B)

ee=rep(0,r)

cv=rep(0,b)

for (j in 1:b)

{

for (i in 1:r)

{

ee[i]=bootstrap2(datos,B[j],fun,...)$eestandar

}

cv[j]=sd(ee)/abs(mean(ee))

}

print(cv)

}

B=c(10,50,100,500)

y=nmb2(rock[,1],B,30,quantile,0.4)

plot(B,y,type="b")

abline(h=0.1)

system.time(nmb2(rock[,1],B,30,quantile,0.4))[3]

#No se recomienda encontrar un valor exacto de muestras bootstrap

B=seq(50,60,1)

y=nmb2(rock[,1],B,30,quantile,0.4)

plot(B,y,type="b")

abline(h=0.1)

#Bootstrap parametrico univariado

#Con datos provenientes de una distribucion Normal

hist(rock[,1])

shapiro.test(rock[,1])

bootstrap3=function(datos,B,fun,...)

{

n=length(datos)

estimac=rep(0,B)

for (i in 1:B)

{

estimac[i]=fun(rnorm(n,mean(datos),sd(datos)),...)

}

estboot=mean(estimac)

eeboot=sd(estimac)

sesgo=estboot-fun(datos,...)

ecor=2\*fun(datos,...)-estboot

hist(estimac)

return(list(estimador=estboot,eestandar=eeboot,sesgo=sesgo,ecor=ecor))

}

bootstrap3(rock[,1],100,quantile,0.4)

#Bootstrap parametrico multivariado

#Con datos provenientes de una distribucion Normal

#Para el coeficiente de correlacion

library(mvnormtest)

iris1=as.matrix(iris[,1:2])

mshapiro.test(t(iris1))

library(MASS)

bootstrap4=function(datos,B,X,Y)

{

datos=datos[,c(X,Y)]

n=dim(datos)[1]

estimac=rep(0,B)

for (i in 1:B)

{

datosm=mvrnorm(n,colMeans(datos),cov(datos))

estimac[i]=cor(datosm)[1,2]

}

estboot=mean(estimac)

eeboot=sd(estimac)

hist(estimac)

return(list(estimador=estboot,eestandar=eeboot))

}

iris2=as.matrix(iris[,1:4])

bootstrap4(iris2,100,1,4)