#In this execrise you will perform a Monte Carlo study of a simple linear regression.

#Let β0 = 2 and β1 = 1 and assume the following relationship holds: y = β0 + β1x + ,

#where x ∈ [0, 10] and N(0, σ2) with σ2 = 3.

#For each combination of n and m where

#n ∈ {20, 100} and m ∈ {10, 500}, do the following:

#Create a vector of n evenly spaced values for x in the range [0,10], then generate n

#random observations of y according to the model above

Xvector <- function(n)

{

return (seq(0, 10, length.out = n))

}

Xvector(11)

Yvector <- function(x)

{

epsilon = rnorm(length(x), mean = 0, sd = sqrt(3))

y = 2 + 1\*x + epsilon

return (y)

}

Yvector(Xvector(200))

#Estimate the values of β0 and β1 using the least squares method (lm function).

CreateDataFrame <- function(n)

{

df <- data.frame(Xvector(n),Yvector(Xvector(n)))

names(df) <- c("X","Y")

return (df)

}

GetLm <- function(n)

{

return (model = lm(data = CreateDataFrame(n),formula =Y~X))

}

GetLm(10)$coefficients

#Repeat steps a) and b) m times.

GetCoefficientsDf <- function(m,n)

{

if(m<1){

return (NULL)

}

df <- data.frame(X= numeric(0), Y= numeric(0))

for(model in 1:m)

{

de <- GetLm(n)$coefficients

df <- rbind(df, de)

}

names(df)<-c("β0","β1")

return (df)

}

GetCoefficientsDf(2,12)

#Calculate the mean and variance of the estimates for β0 and β1 over the m simulations.

GetMeanSD <-function(m,n)

{

df = GetCoefficientsDf(m,n)

sd = apply(df, 2, sd)

mean = apply(df, 2, mean)

var = sd\*sd

return(rbind(mean,var))

}

GetMeanSD(100,100)

#Create a table summarising the results that includes the columns: n, m, beta0\_mean,

#beta1\_mean, beta0\_var and beta1\_var. Explain your findings regarding the effect of the

#size of n and the size of m on the results.

CreateSummarisingTable <- function(m,n)

{

exData = GetMeanSD(m,n)

df = data.frame(n,m,exData["mean","β0"],exData["mean","β1"],exData["var","β0"],exData["var","β1"])

names(df)<-c("n", "m", "beta0\_mean","beta1\_mean", "beta0\_var", "beta1\_var")

return(df)

}

CreateSummarisingTable(100,100)