23847-Lab-5

23847

2024-05-02

Gen AI has not been used within the creation of this document

# Problem 1

## a)

We have

Setting

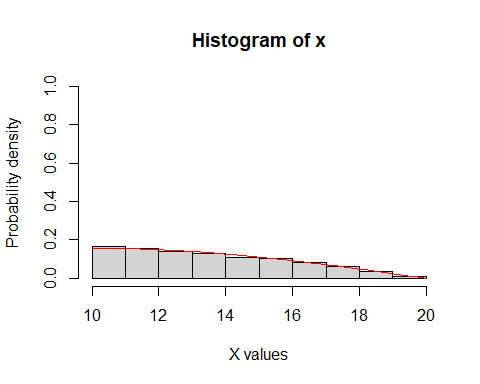
we may then generate Unif(0, 1) realizations.

pr1 = function(n){  
 return(20 / pi \* asin(runif(n, 0, 1)) + 10)  
}

## b)

We plot a histogram of the realizations generated from the pr1 function. Then the graph of the PDF is overlaid, to which we can see the sample matches the PDF.

pr1\_check = function(n){  
 x = pr1(n)  
 hist(x , prob = TRUE, xlab = "X values", ylab = "Probability density", xlim = c(10, 20), ylim = c(0, 1))  
 curve((pi / 20 \* cos((x-10) / 20 \* pi)), add = TRUE, col = "red")  
}  
  
pr1\_check(5000)



## c)

To have a standard deviation below 0.002 we must first repeat the **pr1\_prob** function to find a standard deviation. To calculate it we may use the **sd** function, where the number of repetitions is chosen to be 1000. Then we use a while loop to increment the number of repetitions **pr1\_prob** takes until the desired deviation, here approximately 52000, is reached.

pr1\_prob = function(n){  
 x1 = pr1(n)  
 x2 = pr1(n)  
 count = sum((x1 + x2) > 25)  
 return (count / n)  
}  
  
# Checking sdev   
nreps = 10000  
std\_dev = 1  
while(std\_dev > 0.002){  
 std\_dev = sd(replicate(1000, pr1\_prob(nreps)))  
 nreps = nreps + 1000  
}  
print(nreps)

## [1] 48000

pr1\_prob(52000) # Estimate

## [1] 0.7212692

# Problem 2

## a)

We calculate the times

by comparing the values of

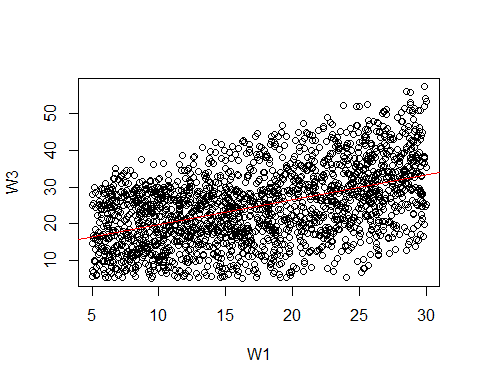
, if it is greater then the appointment overran and the following appointments will be delayed. Then we iterate over the values in Y := X + T, subtracting their arrival times to calculate W, which is then directly inputted into the array. To check the function was working as intended a test with T = (16, 15, 30) was inputted, resulting in W = (16, 16, 32) to be returned, as desired.

gentimes = function(n){   
 z = array(0, c(n, 3))  
 for(i in 1:n){  
 T = runif(3, 5, 30)   
 X = c(0, max(T[1], 15), max(T[1] + T[2], 30) )  
 z[i, ] = (X + T)[1:3] - 15 \* (0:2)   
 }  
 return (z)  
}

## b)

The plot has a linear correlation, which can be seen by adding a line of best fit. This implies the variables are not independent.

times = gentimes(2000)  
plot(times[1:2000, 1], times[1:2000, 3], xlab = "W1", ylab = "W3")  
abline(lm(times[1:2000, 3] ~ times[1:2000, 1]), col = "red")



## c)

Firstly we calculate the expectaions as given in the problem sheet. Then we calculate the variance using

and then the covariance,

. Using these two values we may then calculate the correlation coefficient,

This gives us a correlation coefficient of approximately 0.48.

n = 20000  
wPairs = gentimes(n)[, -2]  
E\_W1 = 1/n \* sum(wPairs[ , 1])  
E\_W1\_2 = 1/n \* sum(wPairs[ , 1]^2)  
  
E\_W3 = 1/n \* sum(wPairs[ , 2])  
E\_W3\_2 = 1/n \* sum(wPairs[ , 2]^2)  
  
E\_W1\_W3 = 1/n \* sum(wPairs[ , 1] \* wPairs[ , 2])  
  
var\_W1 = E\_W1\_2 - E\_W1^2  
var\_W3 = E\_W3\_2 - E\_W3^2  
covar\_W1\_W3 = E\_W1\_W3 - (E\_W1 \* E\_W3)  
  
corr = covar\_W1\_W3 / sqrt(var\_W1 \* var\_W3)  
print(corr)

## [1] 0.4802548