23603-lab-5.R

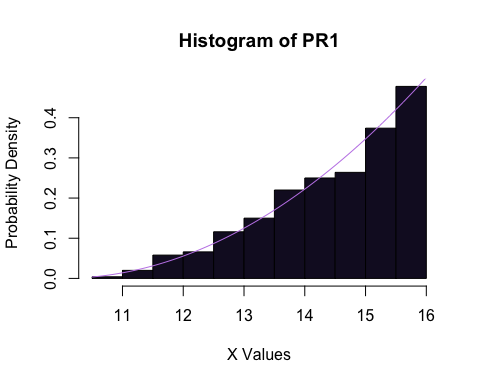
2023-05-02

#1  
#a  
#We first derive the cdf from the given pdf  
#Integrating gives F(x) = (x-10)^3 / 216  
pr1 <- function(n)  
{  
 usample = runif(n, 0, 1)  
 rsample = (216\*usample)^(1/3)+10  
 rsample  
}  
pr1(1)

## [1] 13.83913

#This seems reasonable as you want a value between  
#10 and 16

#b  
pr1\_check <- function(n)  
{   
 sample = pr1(n)  
 hist(sample, prob = TRUE, col ='#160F29', xlab = 'X Values', ylab = 'Probability Density', main = 'Histogram of PR1')  
 curve((x-10)^2/72, add = TRUE, col = '#C287E8')  
}  
pr1\_check(1000)



#The curve seems to fit the histogram well so  
#our function seems to work well  
  
#c  
pr1\_prob <- function(n)  
{  
 x1 = pr1(n)  
 x2 = pr1(n)  
 p = sum(x2 > x1 + 1) / n  
 p  
}  
#First, find Standard deviation with 10000 trials  
#to find about how many trials should be used  
  
testtrials = 10000  
probtest = pr1\_prob(testtrials)  
probtest

## [1] 0.2519

sdprobtest = sqrt(probtest \* (1-probtest) / testtrials)  
sdprobtest

## [1] 0.004341041

#We need to increase the number of trials by a factor of  
#(sdprobtest / 0.0005) ^2 to obtain sd = 0.0005  
trials = ceiling(testtrials \* (sdprobtest / 0.0005)^2)  
trials

## [1] 753786

prob = pr1\_prob(trials)  
prob

## [1] 0.257658

sd = sqrt(prob \* (1-prob) / trials)  
sd

## [1] 0.0005037323

#2  
#a  
n=1  
#Generate values for time  
times=array(0,c(n,3))  
t1 = runif(n, 20, 50)  
t2 = runif(n, 20, 50)  
t3 = runif(n, 20, 50)  
times[,1] = t1  
times[,2] = t2  
times[,3] = t3  
#Initialize Starts array  
starts = array(0,c(n,3))

#Assign the value of X1 to be 0  
starts[,1] = rep(0, n)

#Initialize Ends array  
ends = array(0,c(n,3))

#Assign values of Y1 to time spent cutting Crystal's hair  
ends[,1] = times[,1]

#Initialize array  
waitandcut =array(0,c(n,3))

#Assign W1 to be time spent cutting Crystal's hair  
waitandcut[,1] = times[,1]  
thirts = rep(30, n)

#Set X2 to be the maximum of 30 and time spent cutting Crystal's hair  
starts[,2] = pmax(ends[,1], thirts)

#Set Y2 to be starting time + time spent  
ends[,2] = starts[,2] + t2  
sixts = rep(60, n)  
starts[,3] = pmax(ends[,2], sixts)  
ends[,3] = starts[,3] + t3

#Set W2 and W3 to be time where individual ended - time when they arrived  
waitandcut[,2] = ends[,2] - 30  
waitandcut[,3] = ends[,3] - 60  
times

## [,1] [,2] [,3]  
## [1,] 29.16702 34.6151 49.97033

starts

## [,1] [,2] [,3]  
## [1,] 0 30 64.6151

ends

## [,1] [,2] [,3]  
## [1,] 29.16702 64.6151 114.5854

waitandcut

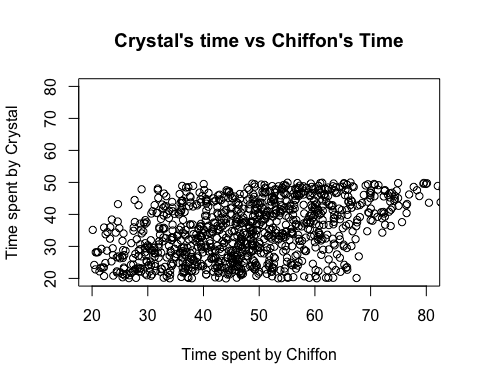
## [,1] [,2] [,3]  
## [1,] 29.16702 34.6151 54.58542

These values all make sense.

#b  
gentimes <- function(n)  
{  
 times=array(0,c(n,3))  
 t1 = runif(n, 20, 50)  
 t2 = runif(n, 20, 50)  
 t3 = runif(n, 20, 50)  
 times[,1] = t1  
 times[,2] = t2  
 times[,3] = t3  
 starts = array(0,c(n,3))  
 starts[,1] = rep(0, n)  
 ends = array(0,c(n,3))  
 ends[,1] = t1  
 waitandcut =array(0,c(n,3))  
 waitandcut[,1] = times[,1]  
 thirts = rep(30, n)  
 starts[,2] = pmax(ends[,1], thirts)  
 ends[,2] = starts[,2] + t2  
 sixts = rep(60, n)  
 starts[,3] = pmax(ends[,2], sixts)  
 ends[,3] = starts[,3] + t3  
 waitandcut[,2] = ends[,2] - 30  
 waitandcut[,3] = ends[,3] - 60  
 waitandcut  
}  
gentimes(1)

## [,1] [,2] [,3]  
## [1,] 29.09104 29.63760 34.31574

#c  
sample = gentimes(1000)  
plot(sample[,3], sample[,1], xlim = c(20,80), ylim = c(20,80), main = "Crystal's time vs Chiffon's Time", xlab = 'Time spent by Chiffon', ylab = 'Time spent by Crystal')



#The plot suggests that W1 and W3 are not independent as the plot  
#shows that smaller values of W1 tend to correspond  
#to smaller values of W3 and large values of W1 tend to correspond to larger  
#values of W3. For example, when W1 is around 20, we see only a few values above  
#60 for W3, but when W1 is around 50, we see a much larger amount of occurences of  
#W3 greater than 60. As such, knowing the value of W1 could give us insight  
#as to how large we expect W3 to be.  
  
#d  
pairer <- function(n)  
{  
 pairs = array(0,c(n,2))  
 sample = gentimes(n)  
 pairs[,1] = sample[,1]  
 pairs[,2] = sample[,3]  
 pairs  
}  
pairs = pairer(10000)  
Ew1 = sum(pairs[,1]) / length(pairs[,1])  
Ew3 = sum(pairs[,2]) / length(pairs[,2])  
Ew1sq = sum(pairs[,1]^2) / length(pairs[,1])  
Ew3sq = sum(pairs[,2]^2) / length(pairs[,2])  
Ew1w3 = sum(pairs[,2] \* pairs[,1]) / length(pairs[,1])  
cov = Ew1w3 - Ew1 \* Ew3  
varw1 = Ew1sq - Ew1^2  
varw3 = Ew3sq - Ew3^2  
corr = cov / sqrt(varw1 \* varw3)  
Ew1

## [1] 35.12391

Ew3

## [1] 47.5327

Ew1sq

## [1] 1310.139

Ew3sq

## [1] 2427.866

Ew1w3

## [1] 1719.152

corr

## [1] 0.437155