NAME: SHUBHAM SHARMA

ROLL NO: 18i190002

MSC PHD (OR)

EX1:

part(a)[R]

We'll make a **function** simvirus(n)=k that will take input 'n' as number of students and will give output as the number of students whose computers are infected.

First we define an array of n elements.

class=[1,2,3,…,n]

Each element of the array 'class' represent a student.

WLOG, we take element '1' of class as Bindu.

Let cc=class(1), cc is a counter / parameter that will tell us the computer that will send the email to the next computer at that time in the for loop.

Initially , only Bindu's computer is infected , so, We define an array '**infected**' that initially has value 1, i.e, infected=[1]

As the maximum no. of computers that can be affected are n , so we use a for loop for i=1 to n and will stop the loop when the virus stops spreading any further, i.e.

for i=1 to n

let y=class

y.delete(cc)

#We are deleting only that student from the array whose computer will send the virus to the other computer at that stage

a=grand(1,1,'uin',1,n-1)

#We are randomly taking a number from 1 to n-1

#We will treat this number as index of the set y so that y[a] will tell us the student whom computer the virus select next

let z=y[a]

if z **not in** y:

infected.append(z)

#i.e, we will add that element to the set of infected

cc=z

#giving cc=z , so in the next iteration will give the correct result

else:

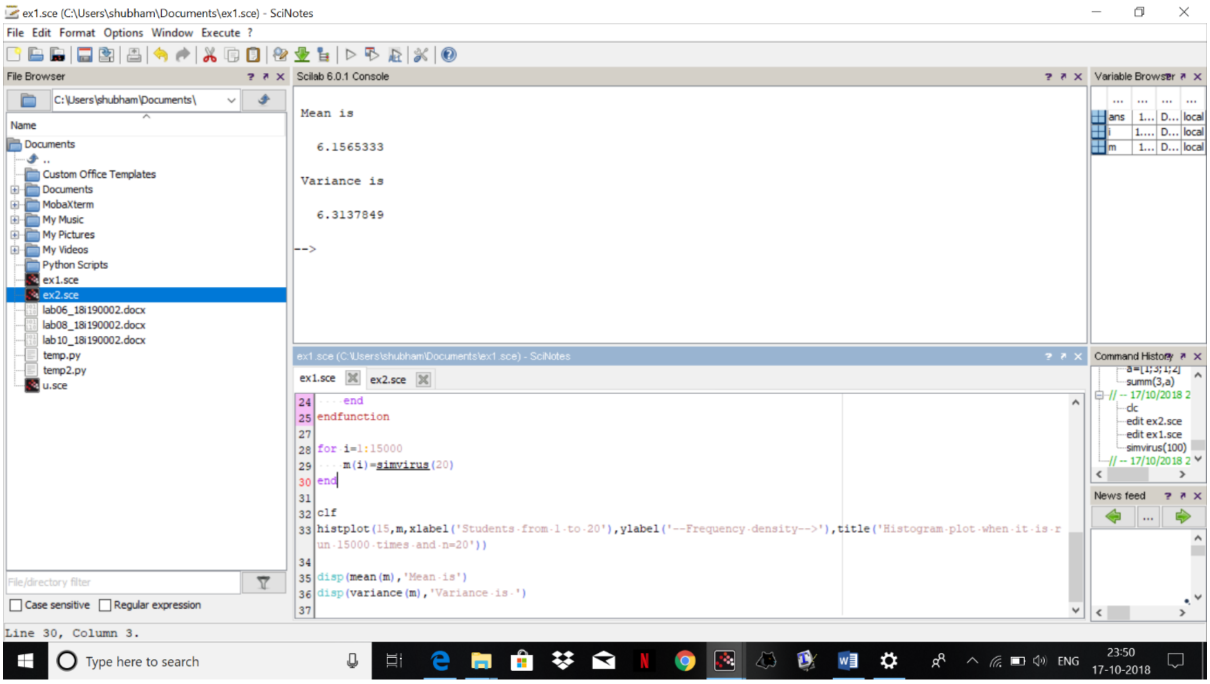
break

k=length(infected)

k is the value that we want. This value is correct as the loop will append the value to the set of infected people till it find an element/student that is already in the set infected and it will stop and will tell the number of students whose computer is infected.

---

part(c)[R]

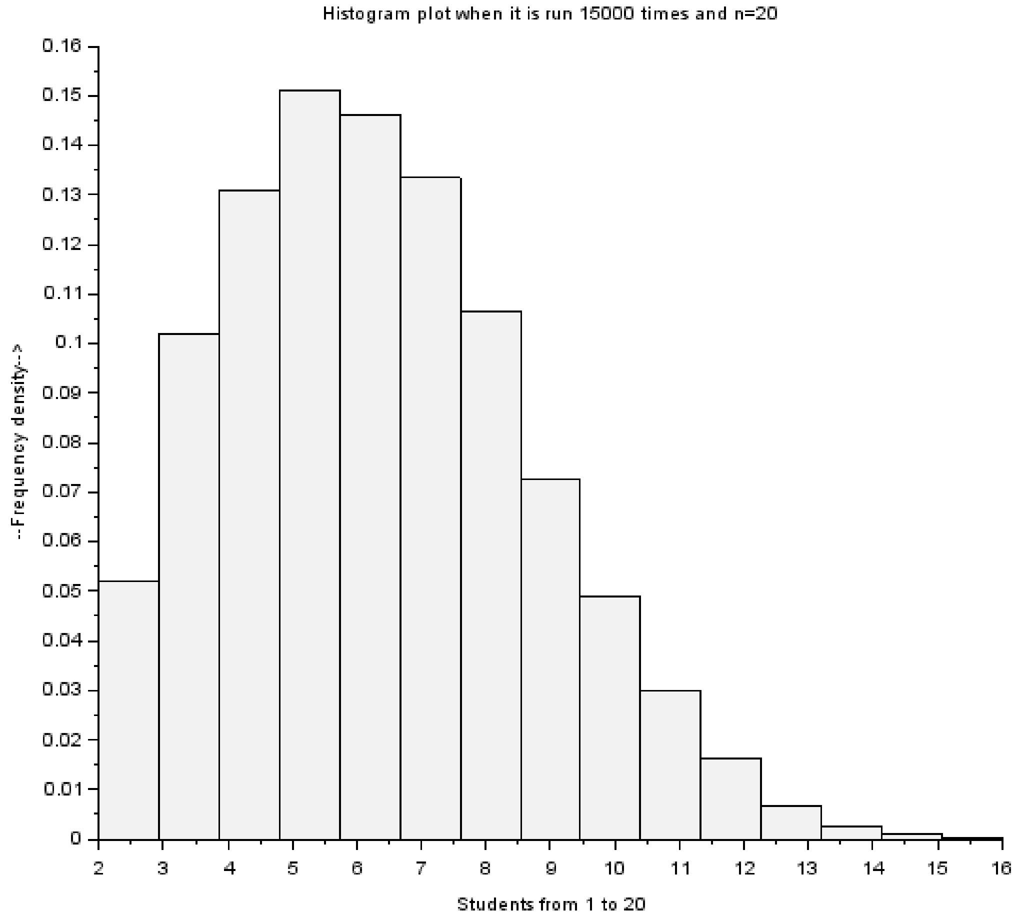


We have run the simulation for **n=20** for **15000** number of times.

Mean is **6.1565333**

Variance is **6.3137849**

part(d)[R]



part(e)[R]

The distribution as we can see resembles the positively skewed poisson distribution and we can also see it as the mean and variance are approximately equal.

EX2:

part(2)[R]

We have to write the algorithm such that we have to create a function that takes input as (T,L,n), First it will generate a random variable Z1 as below:

Z1=inf{i:$\sum_{s=1}^{i} Xi \geq T$}

where {Xi} are exponential random variables with parameter L.

and then we have to repeat this for n times and generate a random sample $\{Z_k\}:\{Z_1, Z_2, Z_3, \ldots, Z_n\}$

ALGORITHM:

function x=myfunction(T,L,n)

#input:

#We are defining a function inside this function in order to create Z1 as per asked in the question

function Z1=myfunction2(T,L)

s=0

flag=1

i=0 #the iteration of number of times while loop is running

while flag==1:

X=grand(1,1,"exp",L)

s=s+X

i=i+1

if s>=T:

flag=0 #i.e., it will stop as $\sum_{s=1}^{i} Xi \geq T.$

Z1=i

return Z1

x=[ ]

for i=1 to n:

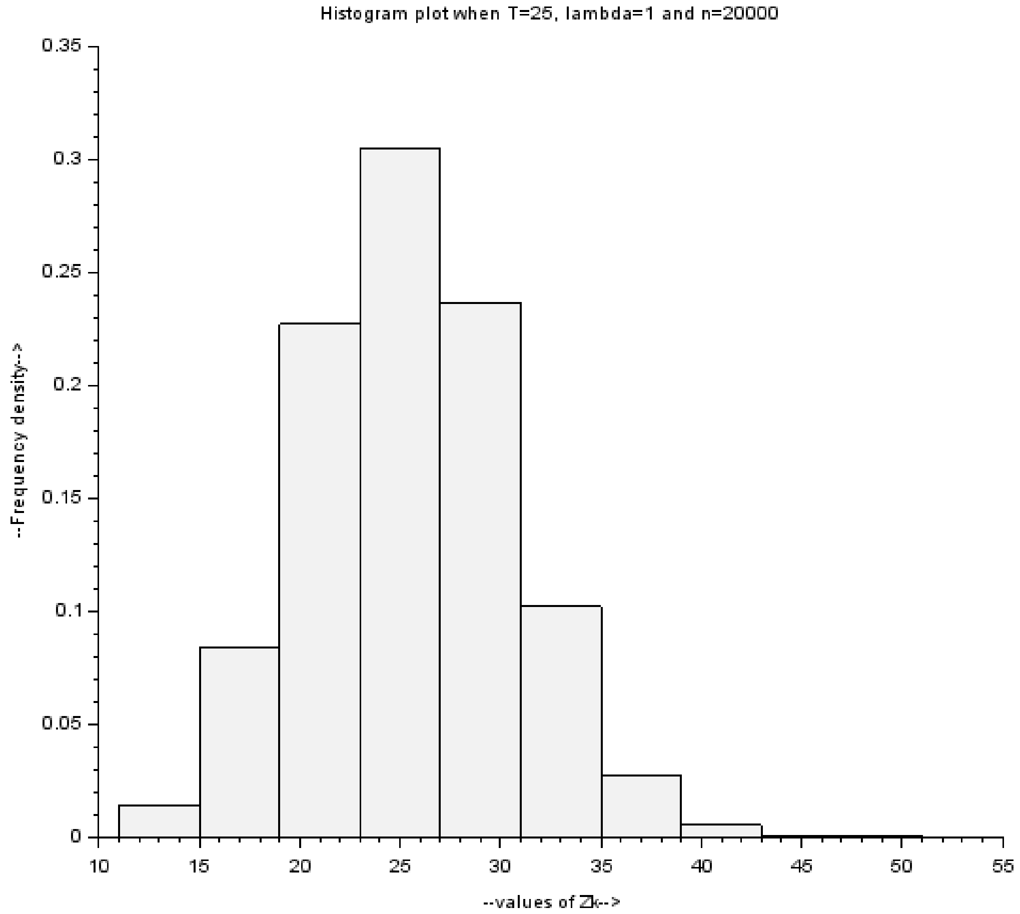
x.append(myfunction(T,L))

#the above loop will make a random sample $\{Z_k\}:\{Z_1, Z_2, Z_3, \ldots, Z_n\}$, as per asked in the question

return x

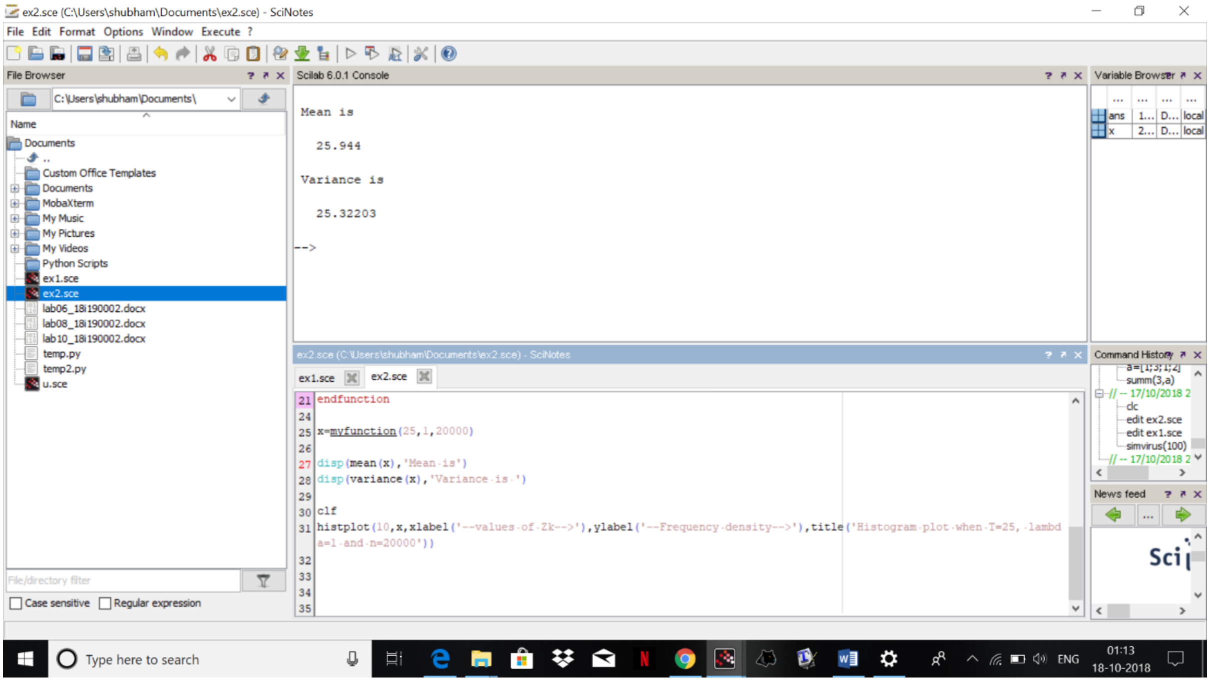
i.e. the above function is returning the array **x i.e ,** basically Z as per asked in the question

part(3)(a)



We have plot the histogram plot the random sample $\{Z_k\}$, when T=25, parameter lambda=1, and n=20000.

part(3)(b)



Mean is **25.944**

Variance is **25.32203**

part(3)(c)

As we can clearly see , the histogram is coming to be symmetric , so it follows normal distribution.