Statistical Methods for Data Science

MINI-PROJECT #1

Group-46

Anish Joshi (UTD ID-2021591978), Aneena Manoj (UTD ID-2021623362)

Contribution – We both discussed the mini-project and divided the work between the two of us. Anish completed half of the questions and the other half was completed by Aneena. After completing the R code and analyzing every question, we wrote the report together. Both the partners completed their parts efficiently.

<u>Question 1 A</u>

XA – Lifetime of Block A

XB – Lifetime of Block B

T—Lifetime of the satellite

Mean=1/(lambda)=10 years,

According to the question, the Probability Density Function (PDF) of T is

fT(t) = 0.2exp(-0.1t) – 0.2exp(-0.2t), $0<=t<\infty$,

fT(t) = 0, otherwise,

and E(T) is given as 15 years.

To Calculate – Probability that the lifetime of the satellite exceeds 15 years (>15 years).

Solution-:

We need to integrate the given PDF with the limits, 15 to ∞ .

$$P[\,15<T<\infty] = \int_{15}^{\infty} (0.2\cdot exp(-0.1t) - 0.2\cdot exp(-0.2t))\, dt$$

$$= 0.2\int_{15}^{\infty} (exp(-0.1t) - exp(-0.2t))\, dt$$

$$0.2\left[exp(-0.1t)/(-0.1) \;-\; exp(-0.2t)/(-0.2)\right]_{15}^{\infty}$$

$$0.2\left[0-\left[-\frac{exp(-0.1\cdot 15)}{(0.1)} + \frac{exp(-0.2\cdot 15)}{(0.2)}\right]\right]$$

$$0.2\left[0 + \frac{exp(-1.5)}{0.1} - \frac{exp(-3.0)}{0.2}\right]$$

$$0.2\left[0 + \frac{0.22}{0.1} - \frac{0.049}{0.2}\right]$$

$$0.2\left[0+\ 2.2 - 0.245\right]$$

$$0.2\left[0+1.945\right]$$

$$0.2\cdot\ 1.945$$

$$=0.391$$

$$=$$

Answer – The probability that the lifetime of the satellite exceeds is **0.391** .

Question 1 B

i & ii) Simulate a Draw of XA, XB, T and Repeat Step 10,000 times

```
> #Question1
> #This is the first test on the sample which will be run 10000 times.
> pdf.T<-function(x){return(0.2*exp(-0.1*x)-0.2*exp(-0.2*x))}
>
>
> xDraw10k<-replicate(10000,max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))
```

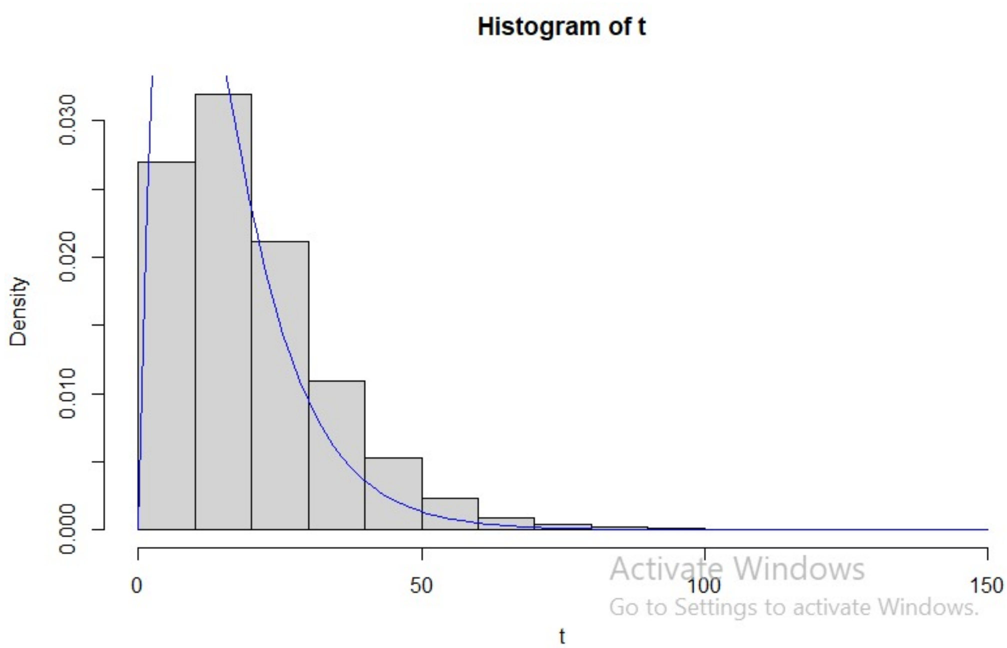
Result :-

Values	
xDraw10k	num [1:10000] 19.32 6.32 43.27 4.55 22.25…

iii) Make a histogram of the 10,000 draws using the 'hist' function

Plotted the histogram using the following command in R and draw the curve function of the probability of T with $\lambda = 0.1$.

```
> hist(xDraw10k)
> curve(pdf.T(x),add=TRUE)
```



iv) Estimate E(T) :-

Estimated the expected value of T using: -

```
> mean(xDraw10k)
[1] 14.87708
```

Analytical calculation gives us a value for E(T) of 15.

v) Estimate the probability that the satellite will last more than 15 years

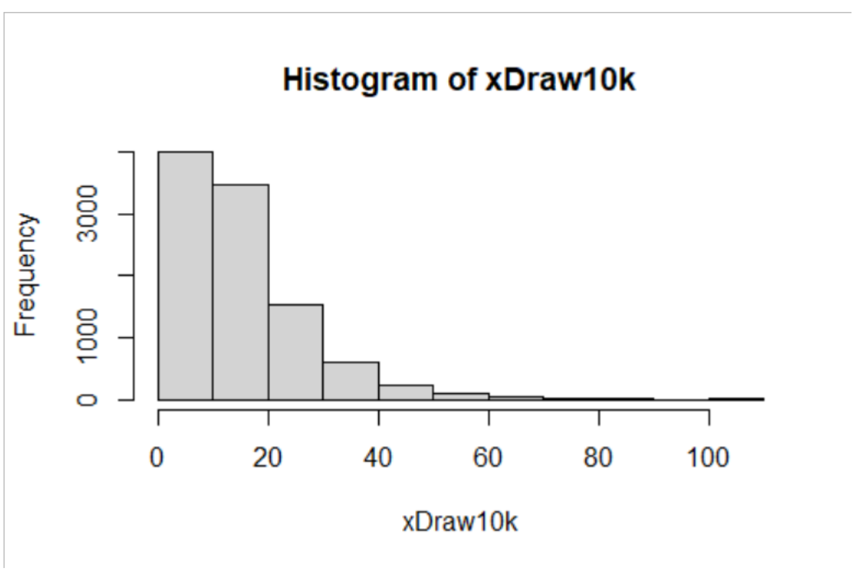
```
> 1-pexp(15,rate=1/mean(xDraw10k))
[1] 0.3648524
```

The probability which we computed in (a) is 0.39 and the probability which we got is 0.3648524. The reason for this slight difference is because the sample size is taken as 10000 random variables and differences in the mean.

vi) Taking 10,000 draws for 4 more times

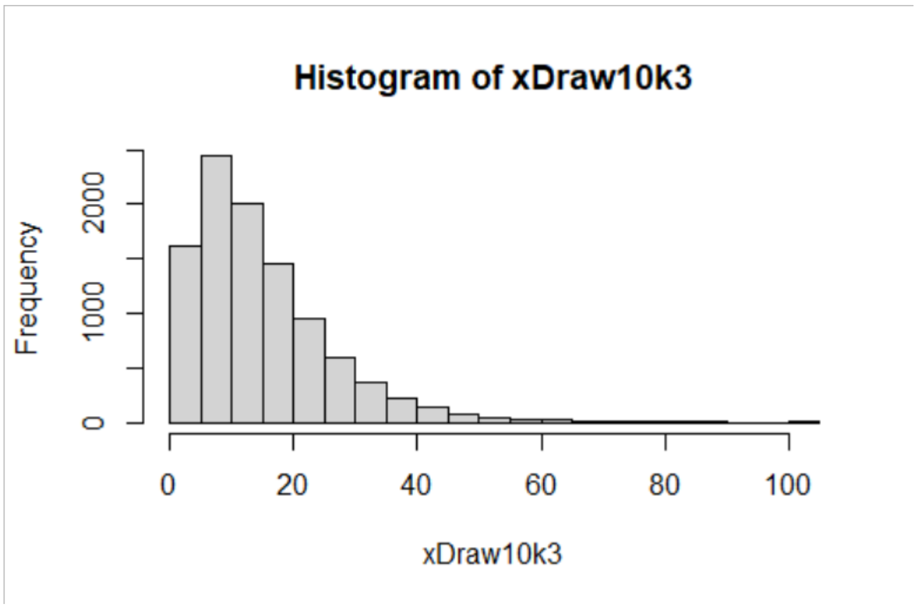
Test 2

```
>
> #This is the 2nd test with sample 10000
> xDraw10k=replicate(10000,max(rexp(n=1,rate=1/10)))
>
> xDraw10k=replicate(10000,max(rexp(n=1,rate=1/10), rexp(n=1,rate=1/10)))
> hist(xDraw10k)
> hist(x=xDraw10k)
> hist(xDraw10k)
>
>
> mean(xDraw10k)
[1] 15.04968
> 1-pexp(15,rate=1/mean(xDraw10k))
[1] 0.3690958
```



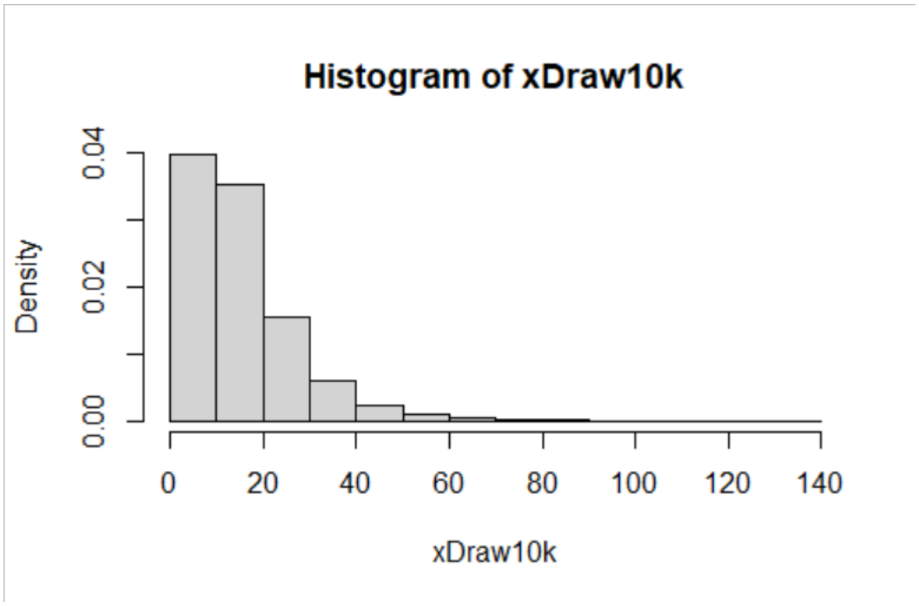
Test 3

```
>
> #This is the 3rd test with sample 10000
> xDraw10k<-replicate(10000,max(rexp(m=1,rate=1/10), rexp(m=1,rate=1/10)))
Error in rexp(m = 1, rate = 1/10) : unused argument (m = 1)
>
>
> xDraw10k3<-replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw10k3)
> mean(xDraw10k3)
[1] 14.85118
> 1-pexp(15,rate=1/mean(xDraw10k3))
[1] 0.3642114
>
```



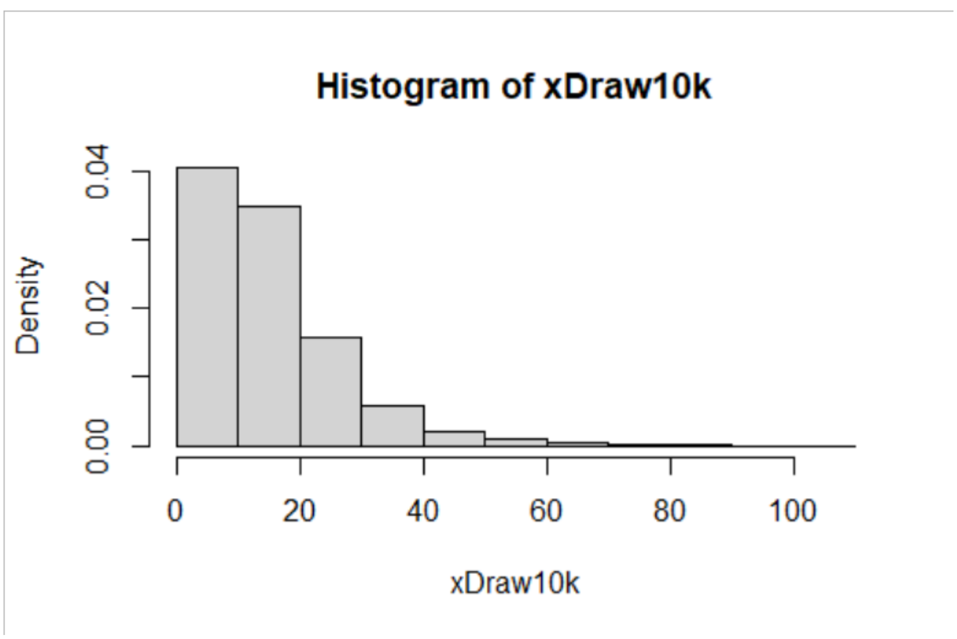
Test 4

```
>
> #This is the 4th test with sample 10000
> xDraw10k<-replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw10k,probability=TRUE)
> mean(xDraw10k)
[1] 15.00984
> 1-pexp(15,rate=1/mean(xDraw10k))
[1] 0.3681206
>
```



Test 5

```
>
> #Thiis is the 5th test with sample 10000
> xDraw10k<-replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw10k,probability=TRUE)
> mean(xDraw10k)
[1] 14.88608
> 1-pexp(15,rate=1/mean(xDraw10k))
[1] 0.3650748
>
```



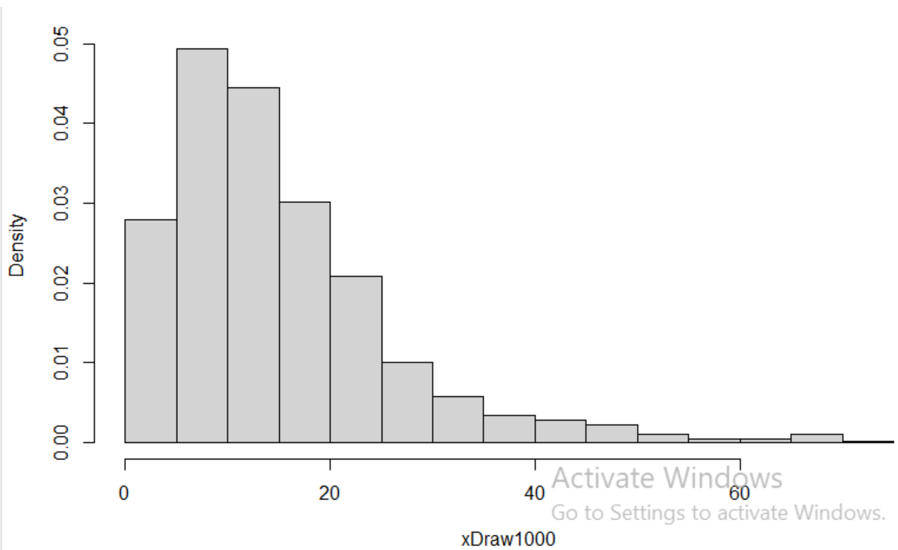
From all the tests which we conducted with the sample size 10000, we can conclude that for this particular sample size, the mean is nearer to 15 and the probability for the sample size is approximately equal to the value which we got analytically in Question 1 part A

.

Question 1 C Repeat part (e) 5 times with 1,000 and 100,000 draws

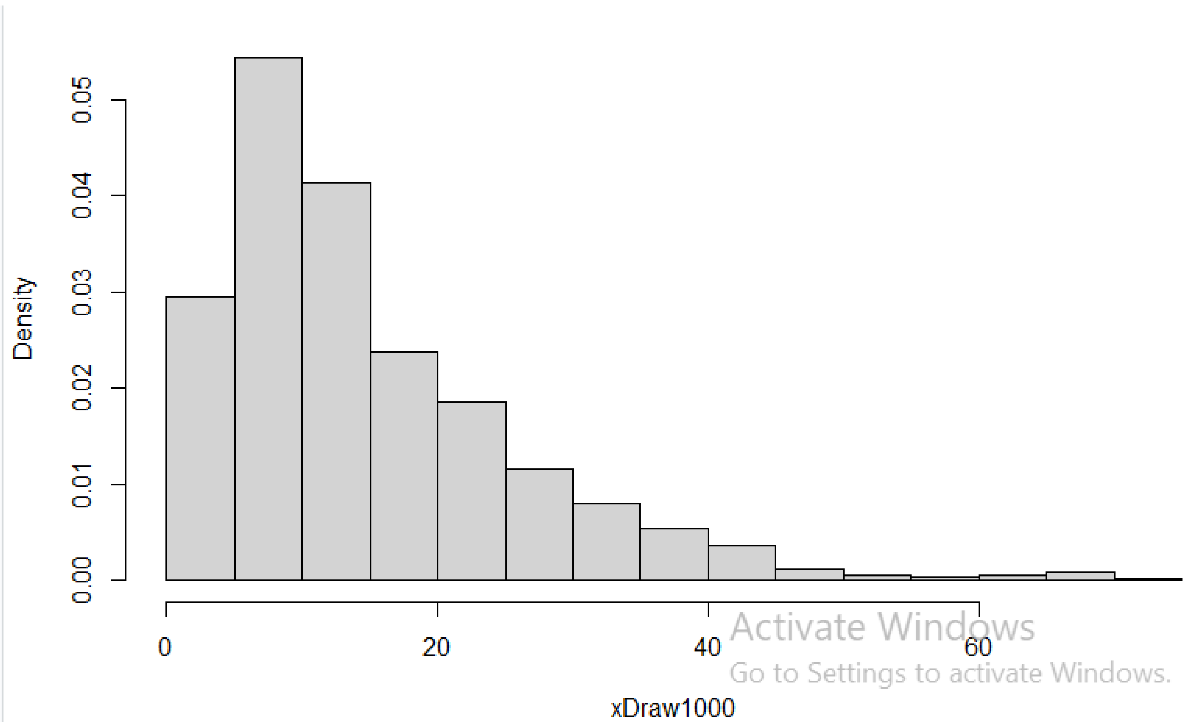
Test 1 with sample 1000 :-

```
> #Test1
> #Sample size -1000
>
> xDraw1000<-replicate(1000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw1000,probability=TRUE)
> mean(xDraw1000)
[1] 14.91871
> 1-pexp(15,rate=1/mean(xDraw1000))
[1] 0.3658804
```



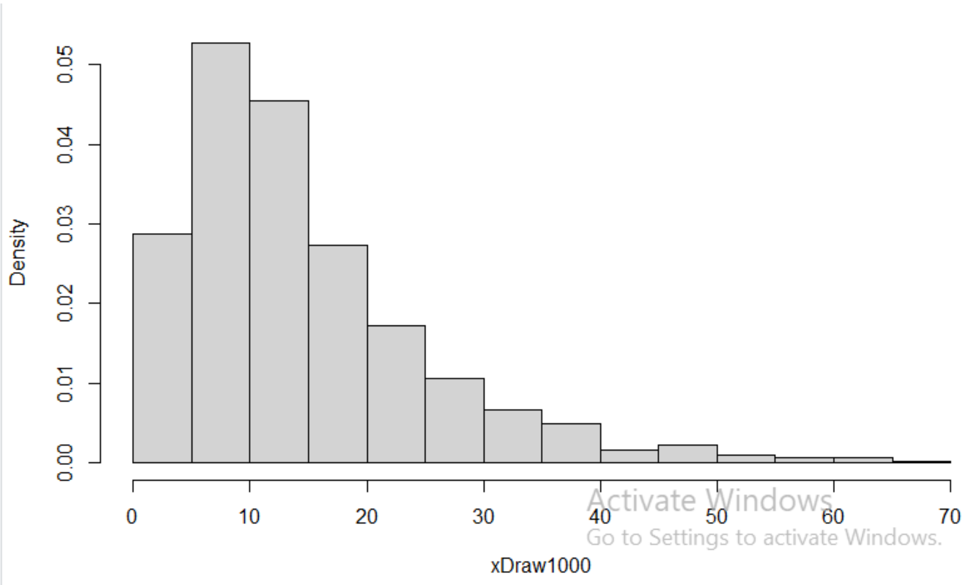
Test 2 with sample 1000 :-

```
> #Test2
> #Sample Size -1000
>
> xDraw1000<-replicate(1000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw1000,probability=TRUE)
> mean(xDraw1000)
[1] 14.88083
> 1-pexp(15,rate=1/mean(xDraw1000))
[1] 0.3649451
>
```



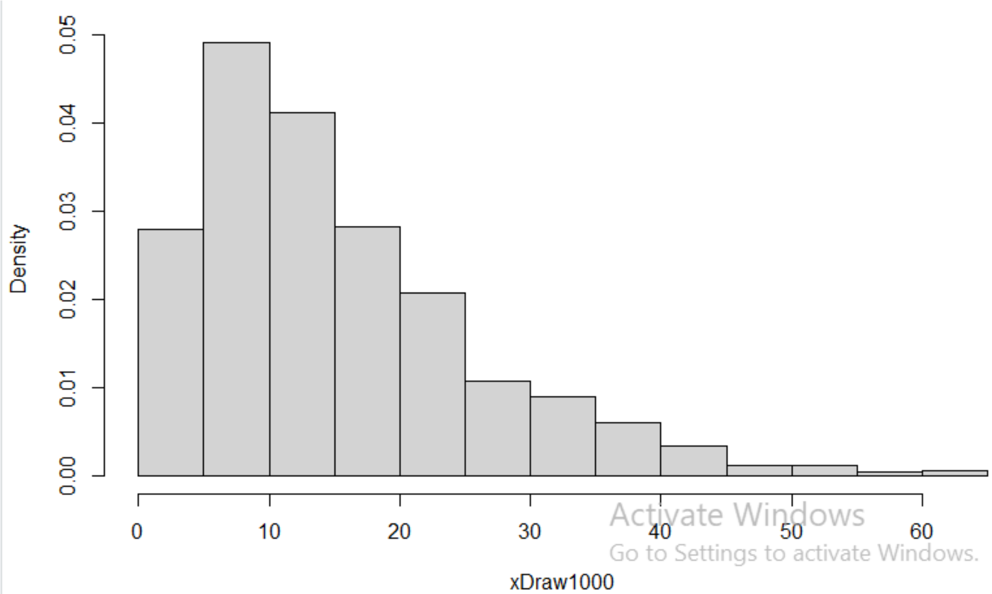
Test 3 with sample 1000 :-

```
> #Test3
> #Sample Size -1000
>
> xDraw1000<-replicate(1000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw1000,probability=TRUE)
> mean(xDraw1000)
[1] 14.46122
> 1-pexp(15,rate=1/mean(xDraw1000))
[1] 0.3544255
```



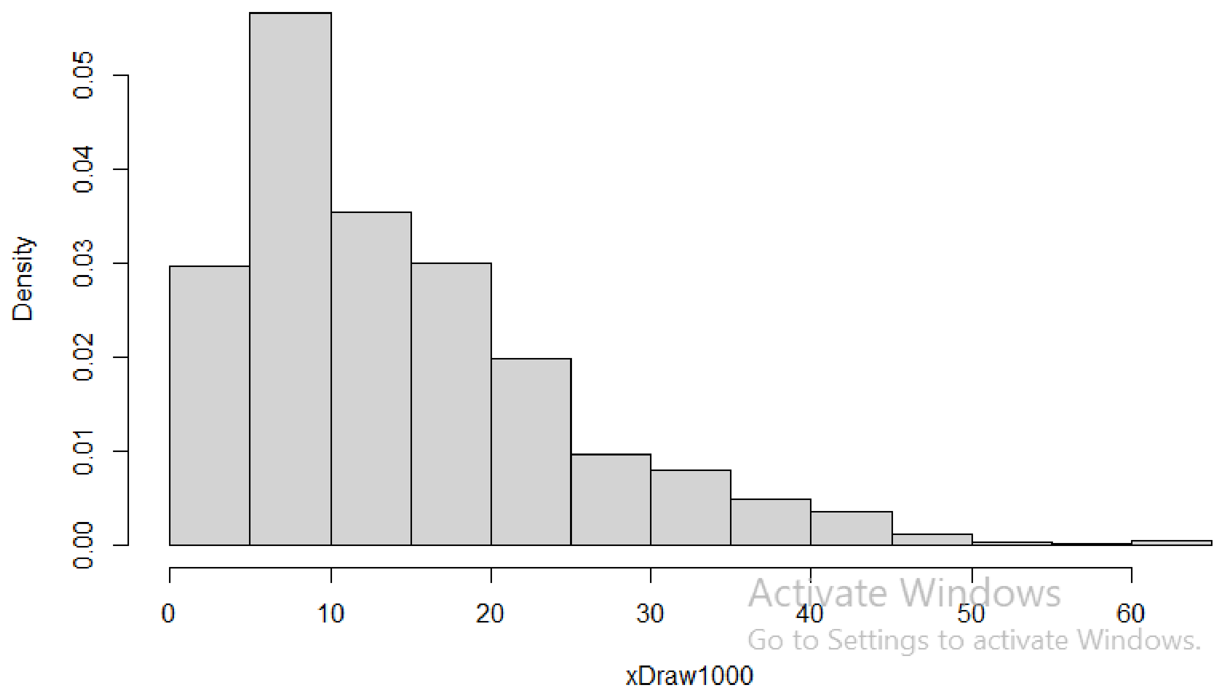
Test 4 with sample 1000:-

```
> #Test4
> #Sample Size -1000
>
> xDraw1000<-replicate(1000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw1000,probability=TRUE)
> mean(xDraw1000)
[1] 15.24465
> 1-pexp(15,rate=1/mean(xDraw1000))
[1] 0.3738308
```



Test 5 with sample 1000 :-

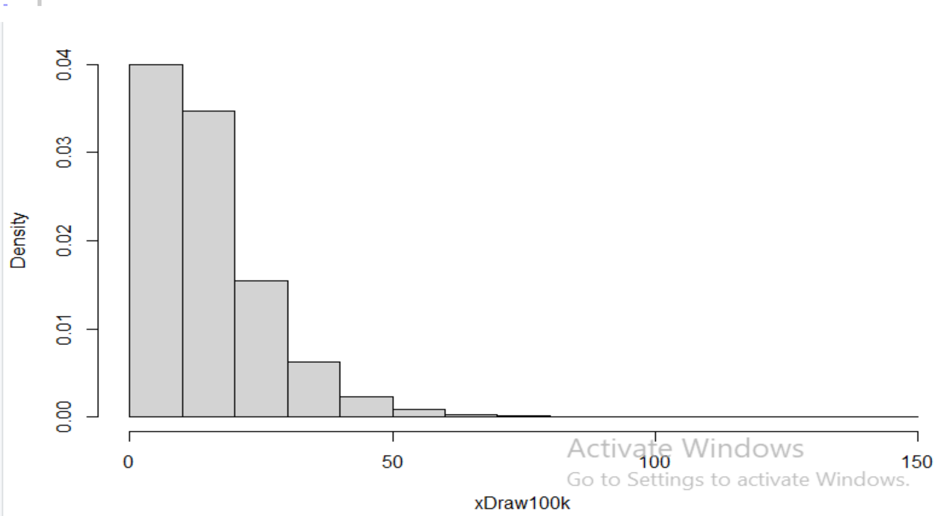
```
> #Test5
> #Sample Size -1000
>
> xDraw1000<-replicate(1000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw1000,probability=TRUE)
> mean(xDraw1000)
[1] 14.43894
> 1-pexp(15,rate=1/mean(xDraw1000))
[1] 0.3538588
```



Sample 100000

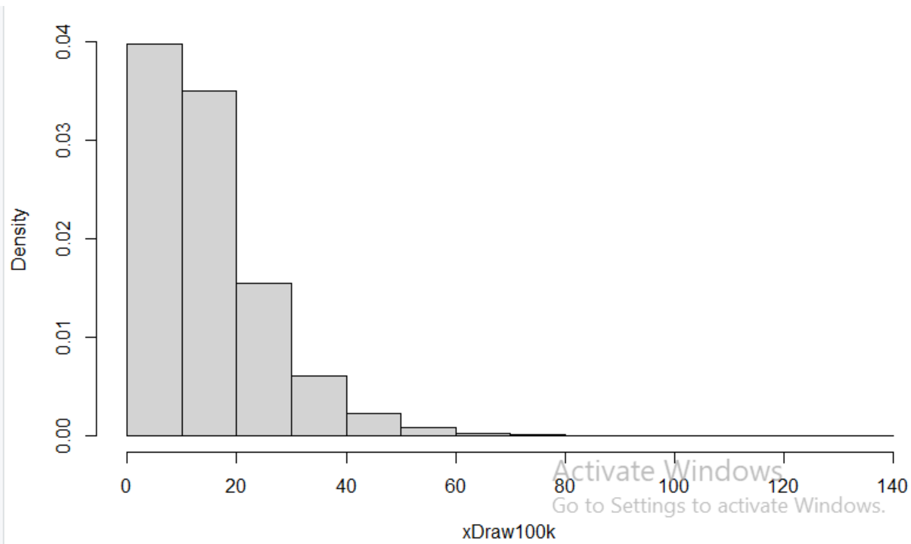
Test 1 with sample 100000 :-

```
> #Test1
> #Sample Size -100000
>
> xDraw100k<-replicate(100000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw100k,probability=TRUE)
> mean(xDraw100k)
[1] 15.00935
> 1-pexp(15,rate=1/mean(xDraw100k))
[1] 0.3681086
```



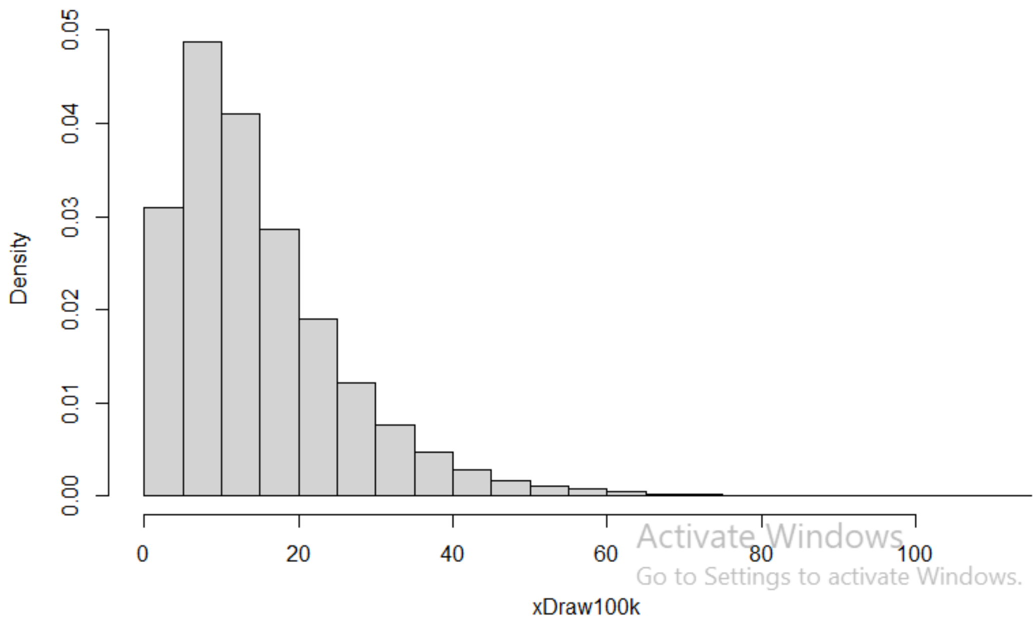
Test 2 with sample 100000 :-

```
> #Test2
> #Sample Size -100000
>
> xDraw100k<-replicate(100000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw100k,probability=TRUE)
> mean(xDraw100k)
[1] 15.00912
> 1-pexp(15,rate=1/mean(xDraw100k))
[1] 0.3681029
```



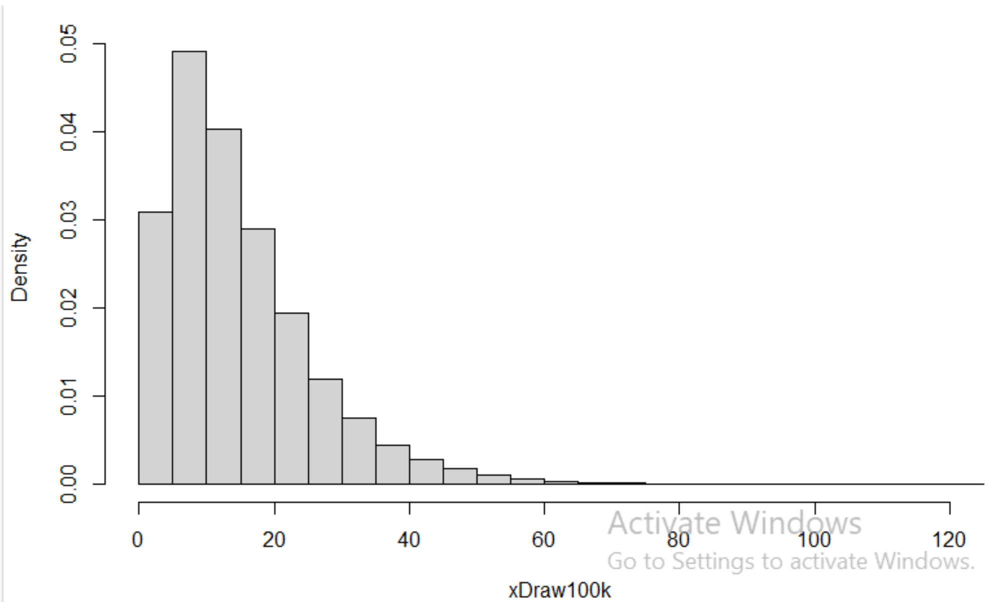
Test 3 with sample 100000 :-

```
> #Test3
> #Sample Size -100000
>
> xDraw100k<-replicate(100000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw100k,probability=TRUE)
> mean(xDraw100k)
[1] 15.01153
> 1-pexp(15,rate=1/mean(xDraw100k))
[1] 0.368162
```



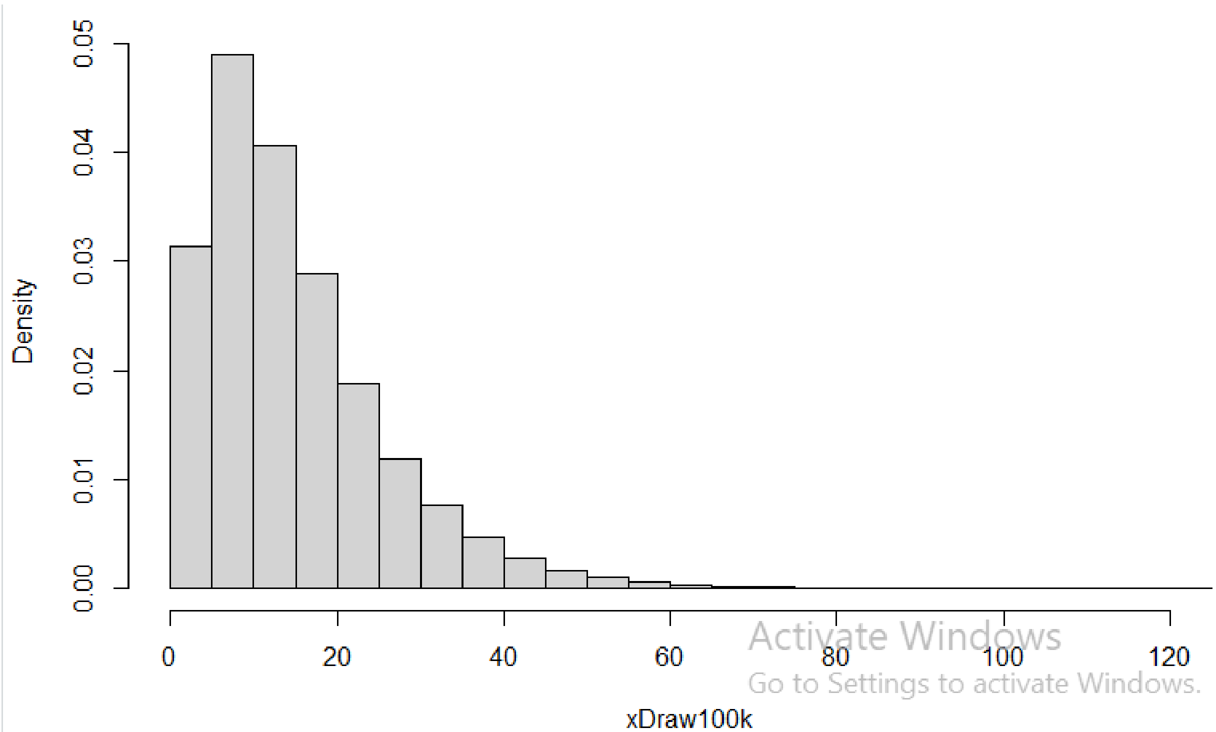
Test 4 with sample 100000 :-

```
> #Test4
> #Sample Size -100000
>
> xDraw100k<-replicate(100000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw100k,probability=TRUE)
> mean(xDraw100k)
[1] 15.00708
> 1-pexp(15,rate=1/mean(xDraw100k))
[1] 0.3680531
```



Test 5 with sample 100000 :-

```
> #Test5
> #Sample Size -100000
>
> xDraw100k<-replicate(100000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)))
> hist(xDraw100k,probability=TRUE)
> mean(xDraw100k)
[1] 14.94257
> 1-pexp(15,rate=1/mean(xDraw100k))
[1] 0.3664682
```



<u>Sample Size: - 1000</u>

Test Case	E(T)	P(T>15)
Test case 1	14.91871	0.3658804
Test case 2	14.88083	0.3649451
Test case 3	14.46122	0.3544255
Test case 4	15.24465	0.3738308
Test case 5	14.43894	0.3538588

Sample Size : - 100000

Test Case	E(T)	P(T>15)
Test case 1	15.00935	0.3681086
Test case 2	15.00912	0.3681029
Test case 3	15.01153	0.368162
Test case 4	15.00708	0.3680531
Test case 5	14.94257	0.3664682

After comparing the five tests which we conducted on sample sizes 1000,10000 and 100000, we can conclude that as the sample size increases, the variation in the values of E(T) and P(T>15) decreases.

Question 2

Computing the approximate value of pi using Monte Carlo Simulations . We write a function piMonto(), which uses the concept of a circle of radius 0.5 with center at (0.5,0.5) circumscribed in a square of unit length with vertices at (0,0), (0,1), (1,0), and (1,1).

Getting 10,000 draws for x-coordinate and y-coordinate of points inside the unit square. Then we find the distance of points from the centre and finally find the number of points that fall inside the circle.

```
> piMonto <- function(){
+
+   x = runif(10000)
+
+   y = runif(10000)
+
+   x = sqrt((x-0.5)^2 + (y-0.5)^2)
+
+   sum(x<=0.5)/10000*4
+
+ }
>
> piMonto()
[1] 3.1404
```