Mini Project # 1

Names of group members: (Group-18)

Deepika Mamidipelly

Preethi Pasunuri

Contribution of each group member

Both group members contributed equally to the inputs for both questions and best is chosen among them to solve these problems. Collaboratively learned R, ran the scripts, and assessed the results. Some of the scripts written by Deepika were analysed and finalized by Preethi and similarly scripts written by Preethi were assessed and finalized by Deepika. Both group members distribute equal amounts of report documentation, which is then integrated into a single final document. Members of the group worked diligently to meet all the project criteria.

Question 1:

1) **Solution:**
Given:

We inferred from given question that

X_A → Lifetime of the block A

X_B → Lifetime of the block B

T → Lifetime of the satellite

Block B is backup of block A

Also, X_A and X_B are independent exponential distributions

Expected value of lifetime of A and B is 10 years:

$E(X_A) = E(X_B) = 10$ years

Also expected value of lifetime of satellite E(T) =15 years

PDF of lifetime of satellite(T) is

$$f_T(t) = \begin{cases} 0.2\exp(-0.1)\,t - 0.2exp(-0.2t) & ,0 \le t < \infty \\ 0 & ,\ Otherwise \end{cases}$$

1.a) As question states to calculate the probability that the satellite's lifetime will be greater than 15 years i.e., P(T>15).

As distribution of random variable T (lifetime of satellite) is continuous and exponential, we have

$P(T>15) = 1-P(T\le 15)$

Here, to find P(T>15) we add up all the probabilities from 0 till 15 (i.e., F (15)) and subtract it from 1(Total Probability).

$P(T>15) = 1-P(T\le 15)$

So first we have to find $P(T\le 15) = F(15)$

Given that $f_T(t) = \begin{cases} 0.2\exp(-0.1)\,t - 0.2exp(-0.2t) & ,0 \le t < \infty \\ 0 & ,\ Otherwise \end{cases}$

We know that $F(15) = \int_0^{15} f_T(t)\,dt$

$$= \int_0^{15} (0.2e^{-0.1t} - 0.2e^{-0.2t})dt \qquad \text{(\#integrating the PDF to get F (15))}$$

$$= \left[0.2\frac{e^{-0.1t}}{-0.1} - 0.2\frac{e^{-0.2t}}{-0.2}\right]_0^{15}$$

$$= [-2e^{-0.1t} + e^{-0.2t}]_0^{15}$$

$$= \left[-2\left(e^{-0.1(15)} - e^{-0.1(0)}\right) + \left(e^{-0.2(15)} - e^{-0.2(0)}\right)\right]$$

$$=[-2(0.223 - 1) + (0.049 - 1)]$$

$$= [-2 * (-0.777) - 0.951]$$

$$= [1.554 - 0.951]$$

F (15) = 0.603

Now that we have F (15), we can find P(T>15)

P(T>15) = 1- P(T≤15)

= 1- F (15)

= 1- 0.603

= 0.397

P(T>15) = 0.397

1.b)

i) We can draw any number of observations from an exponential distribution using the built-in function in R i.e.

rexp (n, rate) where *n=no of observations*;

rate=1/mean

As we move on to the second part of the question, since block B is a backup of block A, satellite will work even if one of the blocks (A or B) is functional. As a result, the satellite's lifetime is determined by the block with longer lifetime.

So, to draw the lifetime of a satellite, we use the **maximum of both satellite's lifetimes**.

RCODE:

```

#pdf function of satellite T= 0.2*exp(-0.1*x)-0.2*exp(-0.2*x)

#rate=1/mean; mean=10 for xA, xB;
#one draw of the block lifetimes
xA=rexp(n=1,rate=1/10);
xB=rexp(n=1,rate=1/10);

#1draw_-satellite_lifetime is one draw of lifetime of satellit
x1draw_satellite_lifetime=max(xA,xB)

```

OUTPUT:

Environment | History | Connections | Tutorial

Global Environment

Values

x1draw_satellite_lifetime	10.3168517905743
xA	6.08357158489525
xB	10.3168517905743

ii) As stated in the question, to repeat earlier steps (b-(i)), we use the built-in r function "**replicate**"

RCODE:

```
#Repeat previous step 10000 times using replicate
x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)));
```

OUTPUT:

Global Environment

Values

x10kdraws_satellite_lifet...	num [1:10000] 6.84 23.11 11 9.44 13.64 ...

iii) We used the "**hist**" function as suggested in the question to represent distribution of data. and then used "**curve**" to superimpose the density on hist.

RCODE:

```
#pdf function of satellite T= 0.2*exp(-0.1*x)-0.2*exp(-0.2*x)

#To get 10000 draws from distribution of T using replicate
x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)));

#histogram for previous x10kdraws_satellite_lifetime
hist(x10kdraws_satellite_lifetime,xlim=c(0,100),ylim=c(0,0.05),
        xlab="Data from distribution",ylab="Frequency", prob= TRUE);

#curve function for drawing the density function and superimposing it on histogram
curve(0.2*exp(-0.1*x)-0.2*exp(-0.2*x), 0, 100,add= TRUE)
```

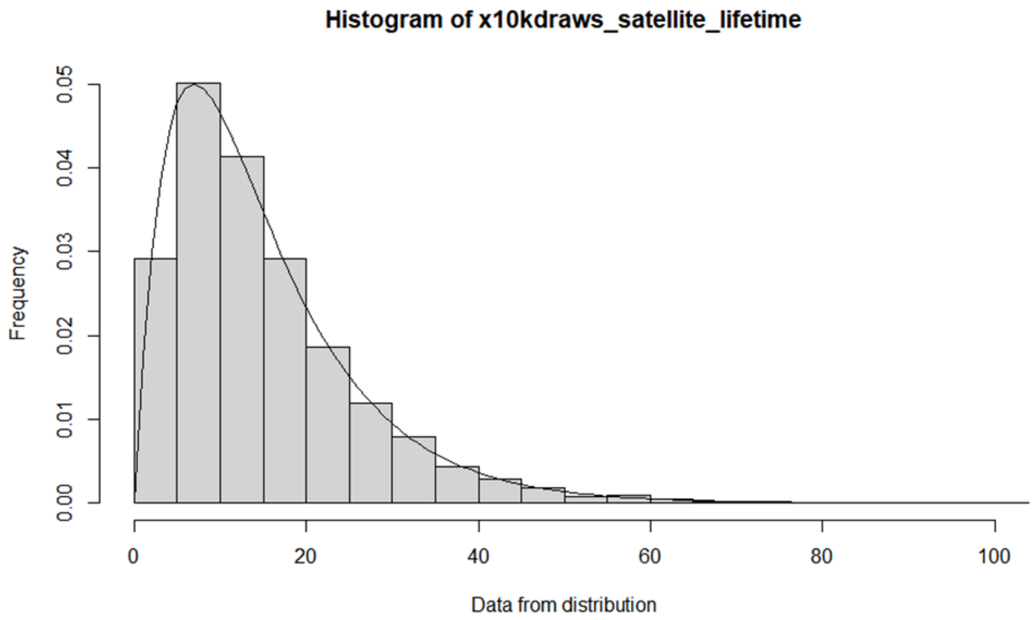
OUTPUT:

Environment | History | Connections | Tutorial

Global Environment

Values

x10kdraws_satellite_lifet...	num [1:10000] 22.553 12.343 6.907 45.821 0.953 ...



iv) We implemented the built-in function "**mean**" since we know that the satellite's estimate

E(T)= mean or weighted average.

RCODE:

```
#To get 10000 draws from distribution of T using replicate
x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)));

#saved draws to calculate mean
mean(x10kdraws_satellite_lifetime);
```

OUTPUT:

```
>
>
> #To get 10000 draws from distribution of T using replicate
> x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)));
>
> #saved draws to calculate mean
> mean(x10kdraws_satellite_lifetime);
[1] 14.90013
>
```

Note: The mean of the lifetime of the satellite given is 15 and the mean generated by Monte Carlo simulation which is 14.90013 are comparable.

v) The R function that calculates the probability, that a random variable(T) will take values less than

15 years is

$\boxed{\textbf{pexp (n, rate)}}$ where *n=15, rate=1/mean*

- To calculate P(T>15), we subtract **pexp** function from 1;

$\boxed{\textbf{1-pexp (15, rate=1/mean)}}$

RCODE:

```
#To get 10000 draws from distribution of T using replicate
x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)));

#the satellite lasts more than 15 years
#pexp to calculate the cumulate probabilty less than years
1-pexp(15, rate=1/mean(x10kdraws_satellite_lifetime))

```

OUTPUT:

```
>
> #To get 10000 draws from distribution of T using replicate
> x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)));
>
> #the satellite lasts more than 15 years
> #pexp to calculate the cumulate probabilty less than years
> 1-pexp(15, rate=1/mean(x10kdraws_satellite_lifetime))
[1] 0.3670199
>
```

Note: The simulated probability with 10000 replications (Sample size 10000) which is 0.367 differs slightly from the analytically calculated probability from part (a) which is 0.397.

vi) To repeat the above procedure of generating an estimate of E(T) and a probability estimate four more times, we implemented "for loop" instead of executing all previous steps four more times.

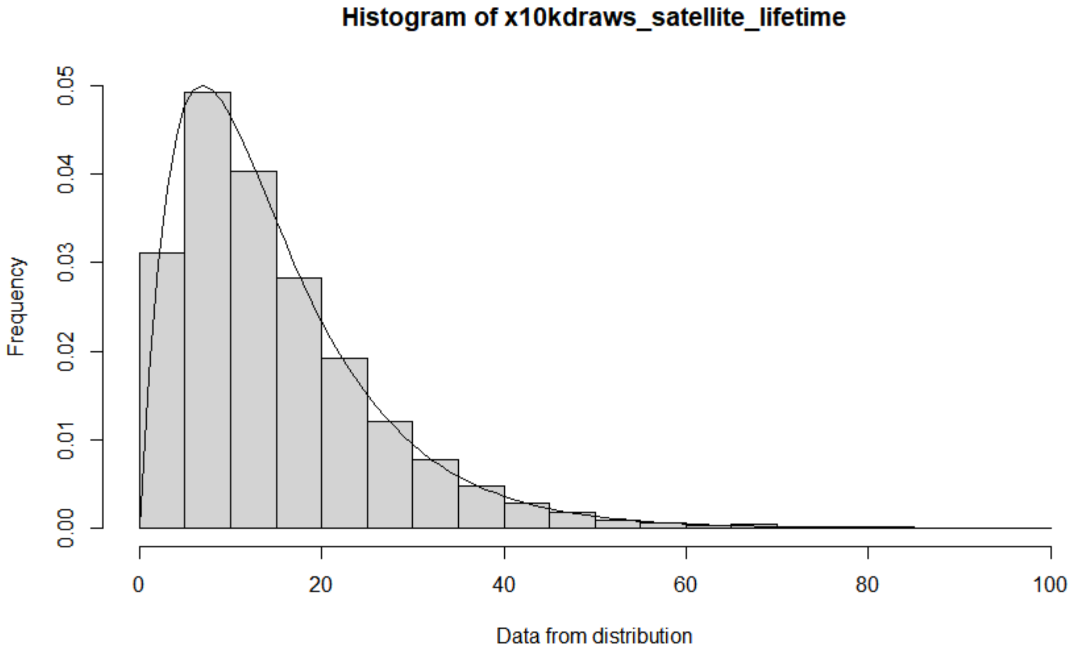
RCODE:

```
#To get estimate of expected values and probability of T with 10k replications by running 4 times

for(i in 1:4){
x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)));

#histogram for previously draws

hist(x10kdraws_satellite_lifetime,xlim=c(0,100),ylim=c(0,0.05)
          ,xlab="Data from distribution",ylab="Frequency", prob= TRUE);

curve(0.2*exp(-0.1*x)-0.2*exp(-0.2*x), 0, 100,add= TRUE)

#saved draws to calculate mean
mean_10Kdraws=mean(x10kdraws_satellite_lifetime);
Probability_10kdraws=1-pexp(15, rate=1/mean_10Kdraws)

#Trying to show all the outcomes from each iteration using data frames
df <- data.frame(Iteration=rep(c(i)),
                 Probability_of_10kdraws=rep(c(Probability_10kdraws)),
                 expected_value_of_10kdraws=rep(c(mean_10Kdraws)))

print (df)
}


```

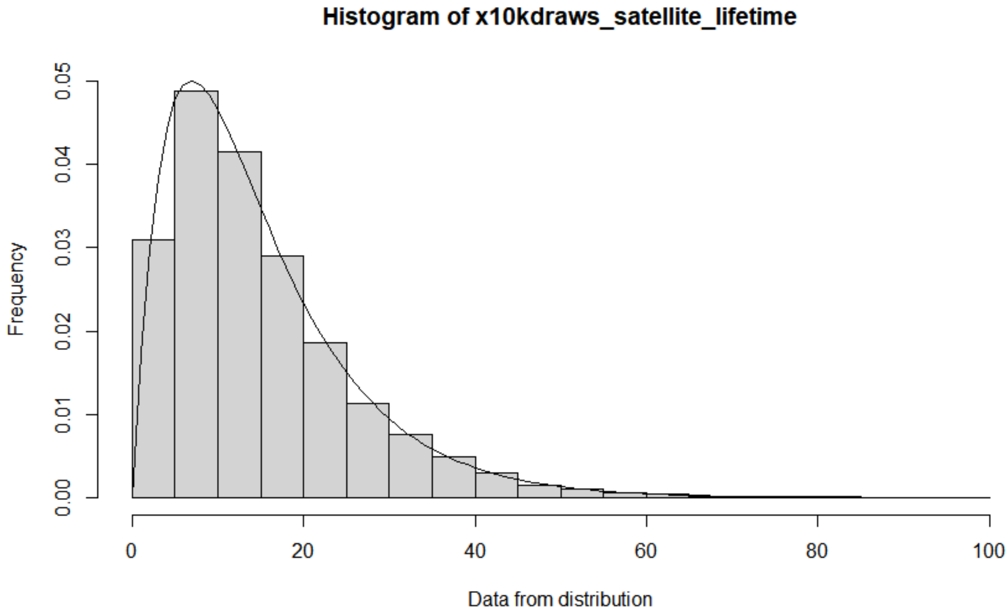
OUTPUT:

```
+
+ print (df)
+ }
  Iteration Probability_of_10kdraws expected_value_of_10kdraws
1         1               0.3701891                   15.09447
  Iteration Probability_of_10kdraws expected_value_of_10kdraws
1         2               0.3668292                   14.95724
  Iteration Probability_of_10kdraws expected_value_of_10kdraws
1         3               0.3663686                   14.93852
  Iteration Probability_of_10kdraws expected_value_of_10kdraws
1         4               0.3687296                    15.0347
>
```

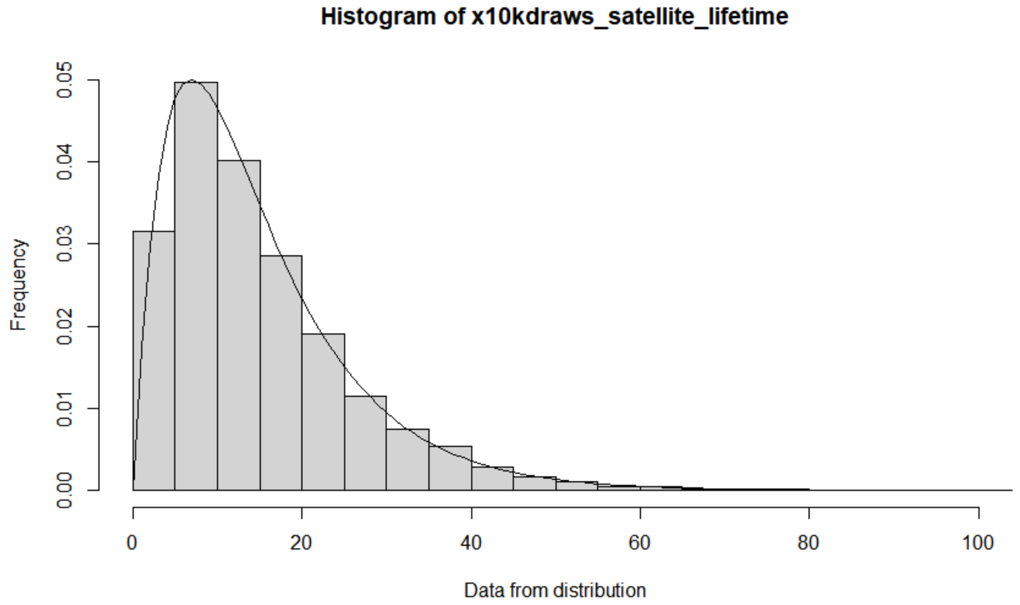
HISTOGRAM IN ITERATION-1:



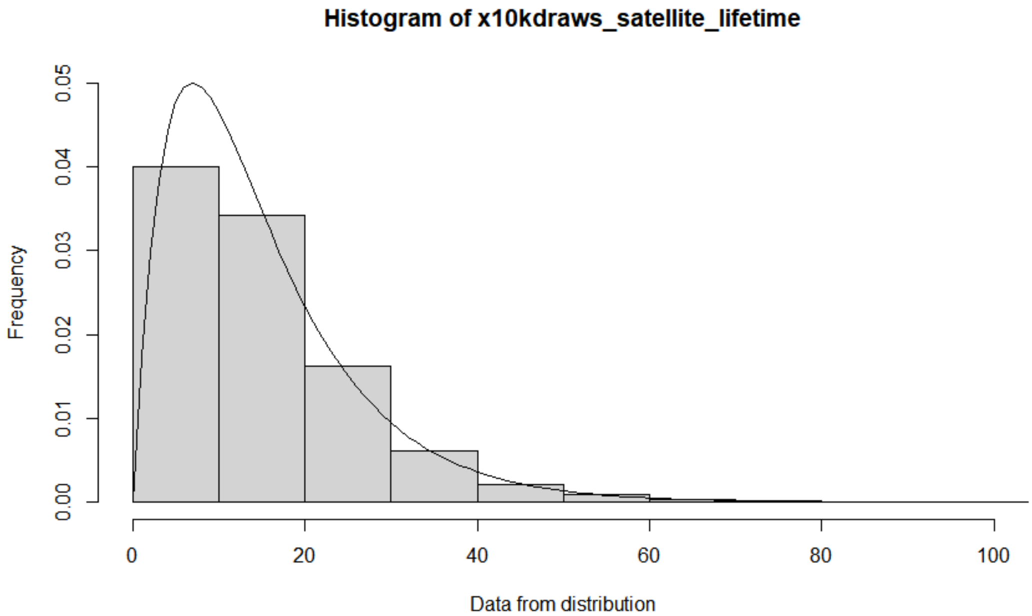
HISTOGRAM IN ITERATION-2:



HISTOGRAM IN ITERATION-3:



HISTOGRAM IN ITERATION-4:



TABULAR DATA:

Iterations for 10000 replications	P(T>15)	E(T)
Iteration-1	0.3701891	15.09947
Iteration-2	0.3668292	14.95724
Iteration-3	0.3663686	14.93852
Iteration-4	0.3687296	15.0347

Note: We can see the simulated mean with 10000 replications from the tabulated data (Sample size 10000) E(T) is approximately equal to 15 and probability from the simulated values correspond to the theoretically calculated values in part (a)

1.C) Instead of 10,000 Monte Carlo replications, repeating section vi) five times with 1,000 and 100,000 Monte Carlo replications.

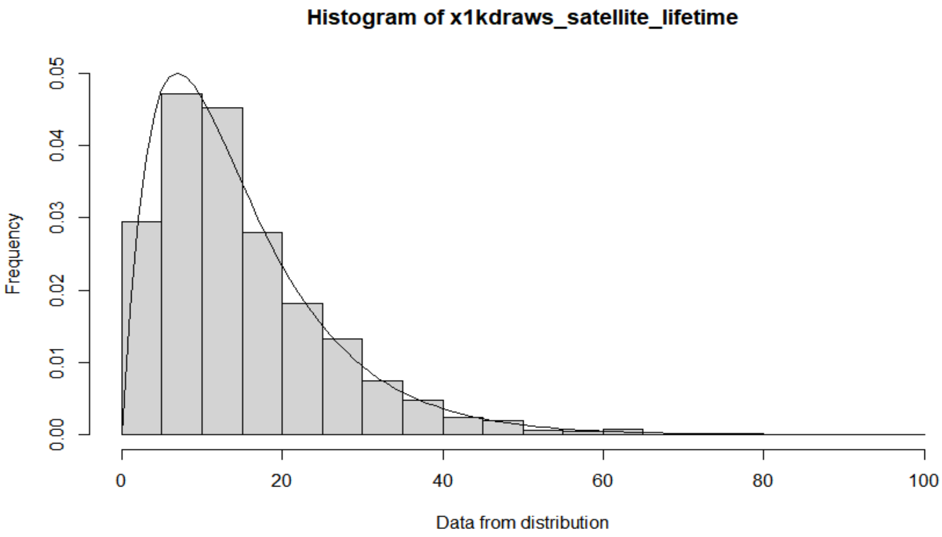
RCODE:

```
#To get estimate of expected values and probability of T with 1k replications by running 5 times
for(i in 1:5){
x1kdraws_satellite_lifetime=replicate(1000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)));

#histogram for previously draws

hist(x1kdraws_satellite_lifetime,xlim=c(0,100),ylim=c(0,0.05)
       ,xlab="Data from distribution",ylab="Frequency", prob= TRUE);


curve(0.2*exp(-0.1*x)-0.2*exp(-0.2*x), 0, 100,add= TRUE)

#saved draws to calculate mean
mean_1Kdraws=mean(x1kdraws_satellite_lifetime);
Probability_1kdraws=1-pexp(15, rate=1/mean_1Kdraws)


df <- data.frame(serial.no=rep(c(i)),
                 Probability_of_1kdraws=rep(c(Probability_1kdraws)),
                 expected_value_of_1kdraws=rep(c(mean_1Kdraws)))

print (df)
}
```

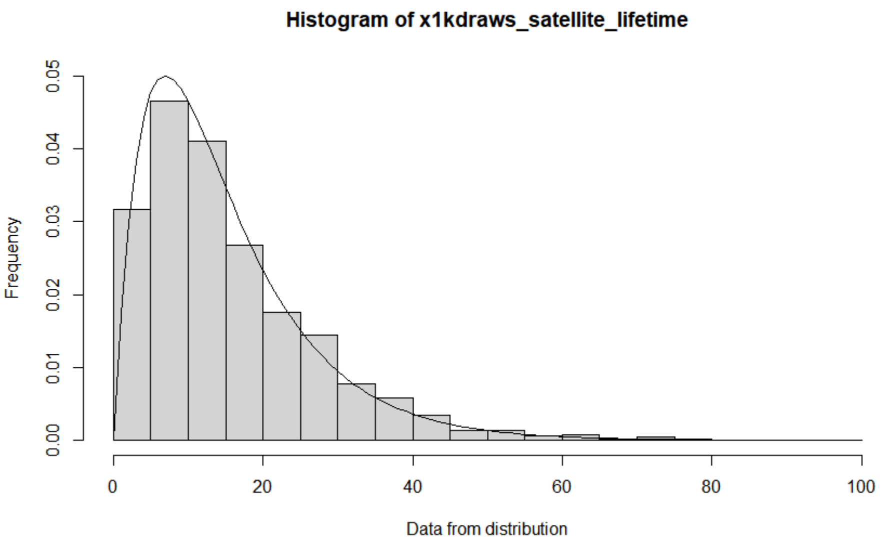
OUTPUT:

```
+ print (df)
+ }
  serial.no Probability_of_1kdraws expected_value_of_1kdraws
1         1              0.3680807                  15.00821
  serial.no Probability_of_1kdraws expected_value_of_1kdraws
1         2              0.3785466                  15.44137
  serial.no Probability_of_1kdraws expected_value_of_1kdraws
1         3              0.3734822                  15.23021
  serial.no Probability_of_1kdraws expected_value_of_1kdraws
1         4              0.3758073                  15.32679
  serial.no Probability_of_1kdraws expected_value_of_1kdraws
1         5              0.3703378                  15.10057
```

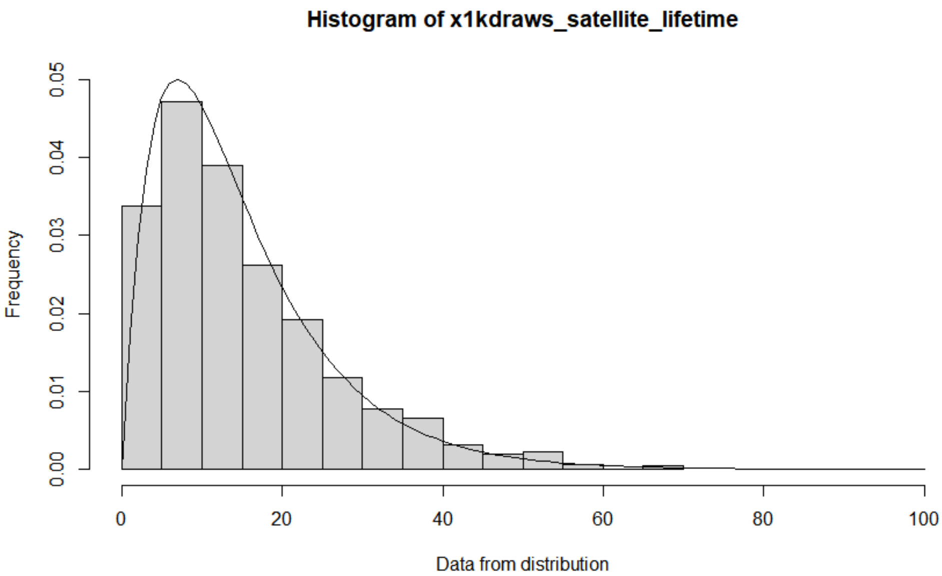
HISTOGRAM FOR ITERATION 1:



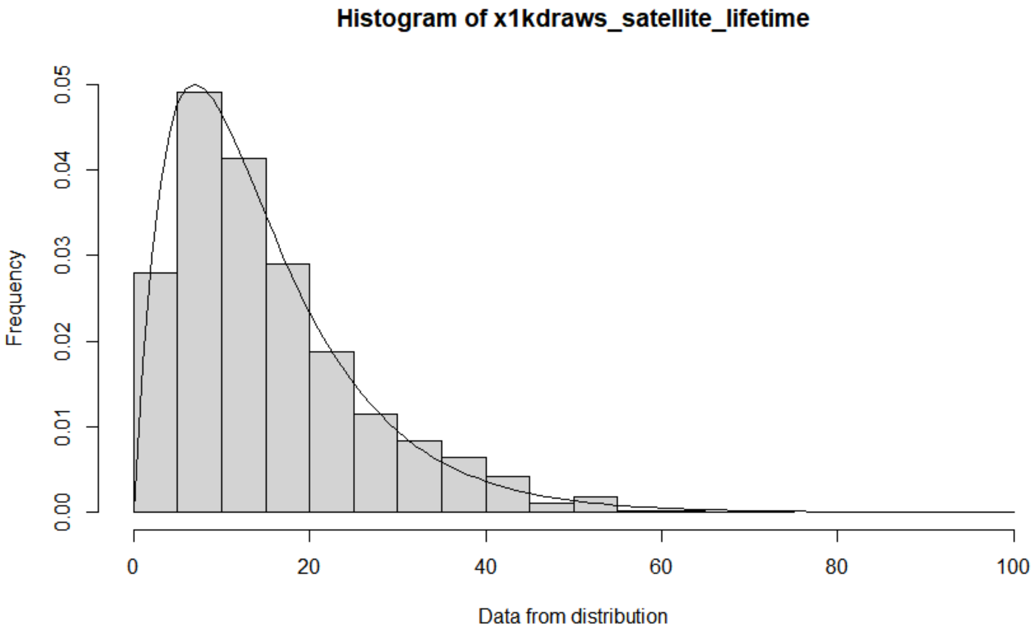
HISTOGRAM FOR ITERATION 2:



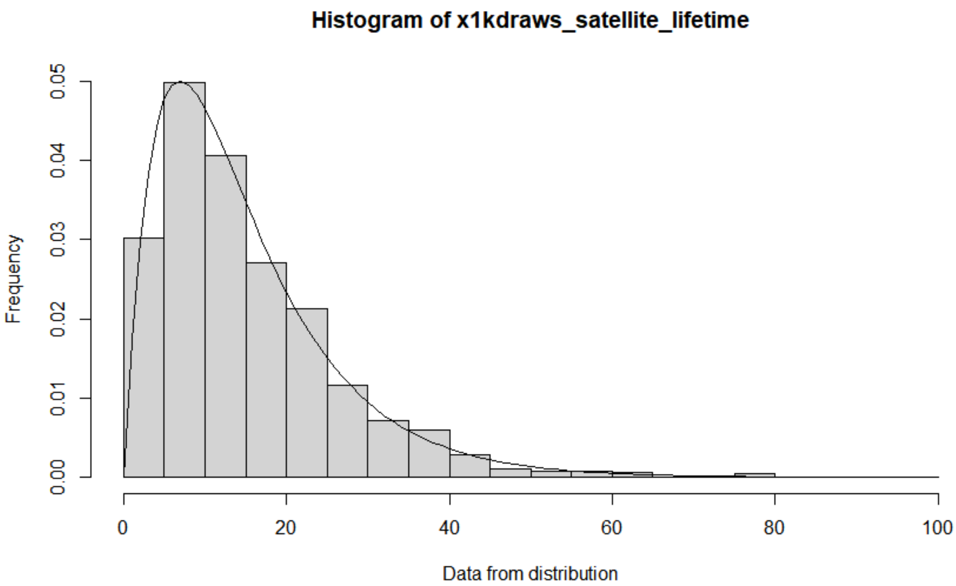
HISTOGRAM FOR ITERATION 3:



HISTOGRAM FOR ITERATION 4:



HISTOGRAM FOR ITERATION 5:



For 100k replications:

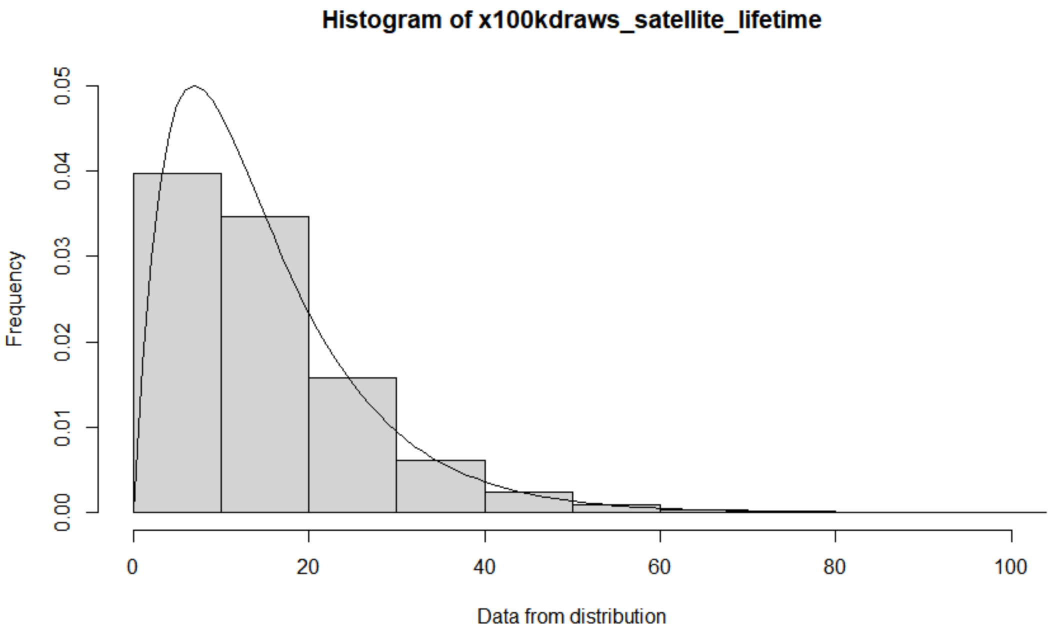
RCODE:

```r
#To get estimate of expected values and probability of T with 100k replications by running 5 times
for(i in 1:5){
x100kdraws_satellite_lifetime=replicate(100000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)));

#histogram for previously draws


hist(x100kdraws_satellite_lifetime,xlim=c(0,100),ylim=c(0,0.05),
       xlab="Data from distribution",ylab="Frequency", prob= TRUE);


curve(0.2*exp(-0.1*x)-0.2*exp(-0.2*x), 0, 100,add= TRUE)

#saved draws to calculate mean
mean_100Kdraws=mean(x100kdraws_satellite_lifetime);
Probability_100kdraws=1-pexp(15, rate=1/(mean_100Kdraws))

df <- data.frame(Iteration=rep(c(i)),
                 Probability_of_100kdraws=rep(c(Probability_100kdraws)),
                 expected_value_of_100kdraws=rep(c(mean_100Kdraws)))

print (df)

}
```

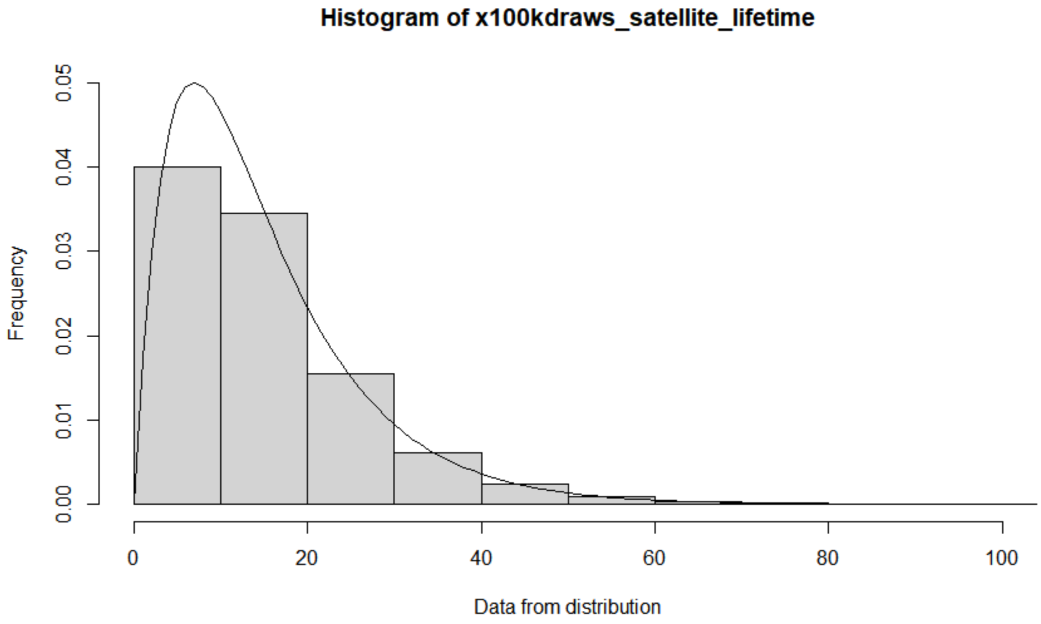
OUTPUT:

```
+ print (df)
+
+ }
  Iteration Probability_of_100kdraws expected_value_of_100kdraws
1         1                 0.370052                    15.08885
  Iteration Probability_of_100kdraws expected_value_of_100kdraws
1         2                0.3687065                    15.03376
  Iteration Probability_of_100kdraws expected_value_of_100kdraws
1         3                0.3685515                    15.02743
  Iteration Probability_of_100kdraws expected_value_of_100kdraws
1         4                0.3685036                    15.02547
  Iteration Probability_of_100kdraws expected_value_of_100kdraws
1         5                0.3687633                    15.03608
>
```

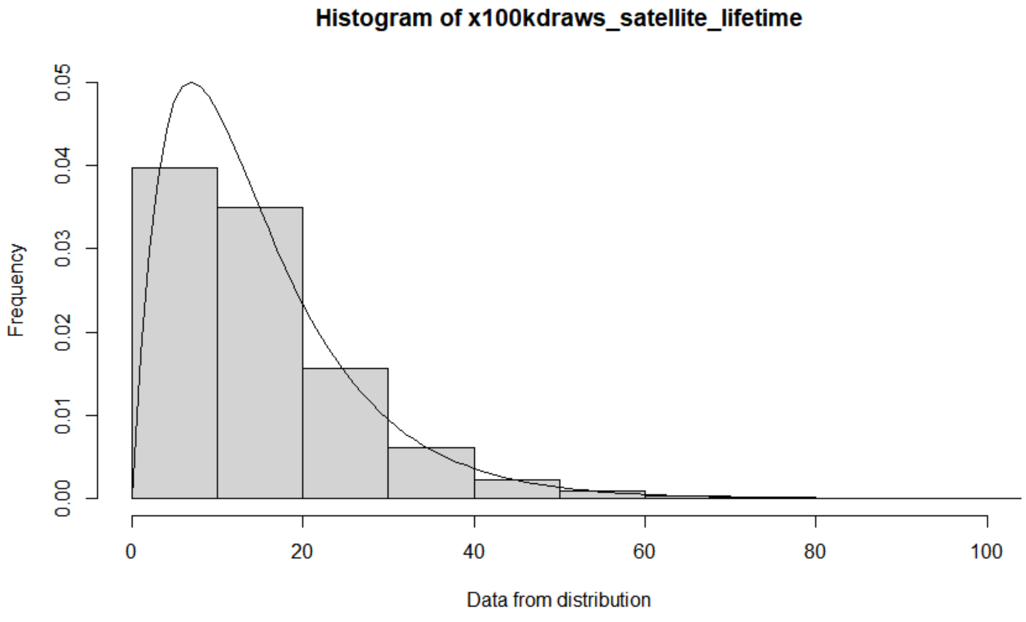
HISTOGRAM FOR ITERATION-1:



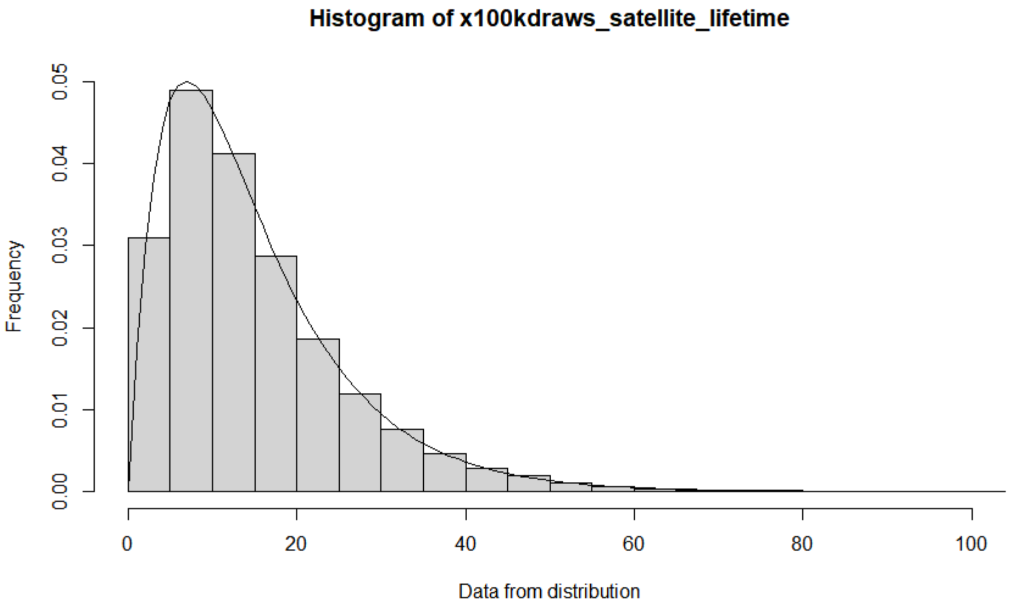
HISTOGRAM FOR ITERATION-2:



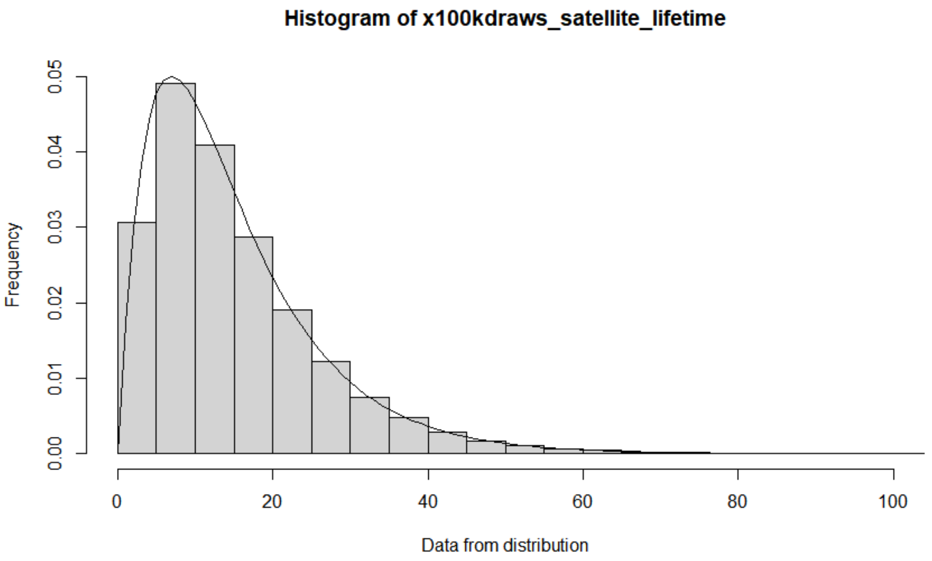
HISTOGRAM FOR ITERATION-3:



HISTOGRAM FOR ITERATION-4:



HISTOGRAM FOR ITERATION-5:



TABULAR DATA:

Iterations for 1000 iterations	P(T>15)	E(T)
Iteration-1	0.3680807	15.00821
Iteration-2	0.3785466	15.44137
Iteration-3	0.3734822	15.23021
Iteration-4	0.3758073	15.32679
Iteration-5	0.3703378	15.10057

Iterations for 100000 replications	P(T)	E(T)
Iteration-1	0370052	15.08885
Iteration-2	0.3687065	15.03376
Iteration-3	0.3685515	15.02743
Iteration-4	0.3685036	15.02547
Iteration-5	0.3687633	15.03608

Note: When replications are more (100000), we observed that the mean is closer to 15 which is the given value and probability has negligible change through the iterations.

When the replications are 1000, we observed that there is noticeable change in expected value and probability.

2)

Based on the information provided, we assume that the circle is inscribed within a square with coordinates (0,0), (0,1), (1,0), (1,1),

Also, the circle's centre is at (0.5,0.5), which represents the diameter of the circle = the side of the square.

Area of circle= πr^2; where r is the radius of the circle.

Area of square= side^2; where side=2r (diameter of the circle)

$=(2r)^2= \mathbf{4}r^2$

Now, determine the probability that a randomly chosen point in the square will fall within the circle P (Point falls in circle) =

$$\boxed{\text{P (Point falls in circle)} = \frac{\text{Probability that the selected point falls within the circle}}{\text{Total probability (Probability of randomly selecting a point inside the square)}}}$$

P (Point falls in circle) = $\pi r^2/\mathbf{4}r^2 = \pi/4$

Relation between π and P (Point falls in circle) is

$$\boxed{\Pi = \text{P (Point falls in circle)} * 4}$$

We have a mathematical equation to find the probability that a point is inside the circle with centre (a, b) i.e.

For centre with (0.5,0.5), it is= $\boxed{(x-a)^2+(y-b)^2<=r^2}$ **= $(x-0.5)^2+(y-0.5)^2<=r^2$**

RCODE:

```r
#From hint in question
replications=10000
#Picking values from uniform distribution in R in the range of co-ordinates of squares
x = runif(replications,min=0,max=1);
y = runif(replications,min=0,max=1);
#The probability that a point selected and it is inside the circle with radius rand center (a,b) is (x-a)^2+(y-b)^2<=r^2
#Here circle is with center (0.5,0.5)
circle= sqrt((x-0.5)^2 + (y-0.5)^2) <= 0.5
#Got this pi and probability relation from theoritical analysis
#piValue= probability that point lies inside a circle * 4
pivalue=(sum(circle)/replications)*4
print (pivalue)
```

OUTPUT:

```
> pivalue=(sum(circle)/replications)*4
> print (pivalue)
[1] 3.16
>
```

R ▾ Global Environment ▾

Values	
circle	logi [1:10000] TRUE TRUE FALSE TRUE TRUE TRUE ...
pivalue	3.16
replications	10000
x	num [1:10000] 0.6336 0.8734 0.0704 0.6035 0.6443 ...
y	num [1:10000] 0.65 0.425 0.204 0.296 0.486 ...

Note: The simulated Pi value is approximately equal to standard value (3.14).

RCODE:

1.b)

i)

```
#pdf function of satellite T= 0.2*exp(-0.1*x)-0.2*exp(-0.2*x)
#rate=1/mean; mean=10 for xA, xB;
#one draw of the block lifetimes
xA=rexp(n=1,rate=1/10);
xB=rexp(n=1,rate=1/10);

#1draw_-satellite_lifetime is one draw of lifetime of satellite
x1draw_satellite_lifetime=max(xA,xB)
```

ii)

```
#Repeat previous step 10000 times using replicate
x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/1
0)));
```

iii)

```
#pdf function of satellite T= 0.2*exp(-0.1*x)-0.2*exp(-0.2*x)
#To get 10000 draws fro distribution of T using replicate
x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1
/10)));

#histogram for previous x10kdraws_satellite_lifetime
hist(x10kdraws_satellite_lifetime,xlim=c(0,100),ylim=c(0,0.05),
        xlab="Data from distribution",ylab="Frequency", prob= TRUE);

#curve function for drawing the density function and superimposing it on histogram
curve(0.2*exp(-0.1*x)-0.2*exp(-0.2*x), 0, 100,add= TRUE)
```

iv)

```
#To get 10000 draws from distribution of T using replicate
x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1
/10)));
#saved draws to calculate mean
mean(x10kdraws_satellite_lifetime);
```

v)

```
#To get 10000 draws fro distribution of T using replicate
```

```
x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=
1/10)));

#the satellite lasts more than 15 years

#pexp is used to calculate the cumulate probabilty less than years

1-pexp(15, rate=1/mean(x10kdraws_satellite_lifetime))
```

vi) #To get estimate of expected values and probability of T with 10k replications by running 4 times

```
for(i in 1:4){

x10kdraws_satellite_lifetime=replicate(10000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=
1/10)));


#histogram for previously draws

hist(x10kdraws_satellite_lifetime,xlim=c(0,100),ylim=c(0,0.05)

        ,xlab="Data from distribution",ylab="Frequency", prob= TRUE);

curve(0.2*exp(-0.1*x)-0.2*exp(-0.2*x), 0, 100,add= TRUE)


#saved draws to calculate mean

mean_10Kdraws=mean(x10kdraws_satellite_lifetime);

Probability_10kdraws=1-pexp(15, rate=1/mean_10Kdraws)


#Trying to show all the outcomes from each iteration using data frames

df <- data.frame(Iteration=rep(c(i)),

           Probability_of_10kdraws=rep(c(Probability_10kdraws)),

           expected_value_of_10kdraws=rep(c(mean_10Kdraws)))

print (df)

}
```

1.C) #To get estimate of expected values and probability of T with 1k replications by running 5 times

```
for(i in 1:5){

x1kdraws_satellite_lifetime=replicate(1000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)));
```

```
#histogram for previously draws
hist(x1kdraws_satellite_lifetime,xlim=c(0,100),ylim=c(0,0.05)
     ,xlab="Data from distribution",ylab="Frequency", prob= TRUE);

curve(0.2*exp(-0.1*x)-0.2*exp(-0.2*x), 0, 100,add= TRUE)

#saved draws to calculate mean
mean_1Kdraws=mean(x1kdraws_satellite_lifetime);
Probability_1kdraws=1-pexp(15, rate=1/mean_1Kdraws)

df <- data.frame(serial.no=rep(c(i)),
           Probability_of_1kdraws=rep(c(Probability_1kdraws)),
           expected_value_of_1kdraws=rep(c(mean_1Kdraws)))
print (df)
}


#To get estimate of expected values and probability of T with 100k replications by running 5 times
for(i in 1:5){
x100kdraws_satellite_lifetime=replicate(100000,max(rexp(n=1,rate=1/10),rexp(n=1,rate=1/10)));

#histogram for previously draws
hist(x100kdraws_satellite_lifetime,xlim=c(0,100),ylim=c(0,0.05),
     xlab="Data from distribution",ylab="Frequency", prob= TRUE);
curve(0.2*exp(-0.1*x)-0.2*exp(-0.2*x), 0, 100,add= TRUE)

#saved draws to calculate mean
mean_100Kdraws=mean(x100kdraws_satellite_lifetime);
Probability_100kdraws=1-pexp(15, rate=1/(mean_100Kdraws))

df <- data.frame(Iteration=rep(c(i)),
```

```
        Probability_of_100kdraws=rep(c(Probability_100kdraws)),
        expected_value_of_100kdraws=rep(c(mean_100Kdraws)))
print (df)
}
```

2) #From hint in question

```
replications=10000

#Picking values from uniform distribution in R in the range of co-ordinates of squares
x = runif(replications,min=0,max=1);
y = runif(replications,min=0,max=1);

#The probability that a point selected and it is inside the circle with radius rand center (a,b) is (x-a)^2+(y-b)^2<=r^2
#Here circle is with center (0.5,0.5)
circle= sqrt((x-0.5)^2 + (y-0.5)^2) <= 0.5

#Got this pi and probability relation from theoritical analysis
#piValue= probability that point lies inside a circle * 4
piValue=(sum(circle)/replications)*4
print (piValue)
```