# Statistical Methods for Data Science: Mini Project 1 Solved

**Mini Project #:** 1
**Group #:** 9
**Names of group members:** Bhargaw Rajnikant Patel, Nikita Ahuja
**Contribution of each group member:** Both team members worked together to solve the two problems. Both learnt the basics of R programming together, wrote the code and finished the report in a timely manner. Both partners worked equally to complete the project requirements.

**Problem 1:**

a) Use the above density function to analytically compute the probability that the lifetime of the satellite exceeds 15 years.

The Probability Density Function form given information can be FT(t).

P (T > 15) = 1 – P (T <= 15)

= 1 – F (T <= 15)

= 1 – $\int_0^{15} FT(t)$

= 1 – $\int_0^{15}(0.2 * e^{-0.1t} - 0.2 * e^{-0.2t})$

= 1 – $[0.2\ (\left(\frac{(e^{-0.1t})}{(-0.1)}\right) - \left(\frac{(e^{-0.2t})}{(-0.2)}\right)]\begin{matrix}15\\0\end{matrix}$

= 1 – $[-2 * e^{-0.1t} + e^{-0.2t}]\begin{matrix}15\\0\end{matrix}$

= 1 – $[\ (-2 * e^{-0.1*15} + e^{-0.2*15}\ ) - (-2 * e^{-0.1*0} + e^{-0.2*0}\ )\ ]$

= 1 – $[\ (-2 * e^{-1.5} + e^{-3}\ ) + (\ 2 * e^{0} - e^{0}\ )\ ]$

= 1 – $[\ e^{-3} - 2 *\ e^{-1.5} + 1\ ]$

= 1 – [ 0.049787 – 0.0446260 + 1 ]

= 1 – 0.603527

= 0.396473

b) Use the following steps to take a Monte Carlo approach to compute E(T) and P(T > 15).
   i) Simulate one draw of the block lifetimes XA and XB. Use these draws to simulate one draw of the satellite lifetime T.

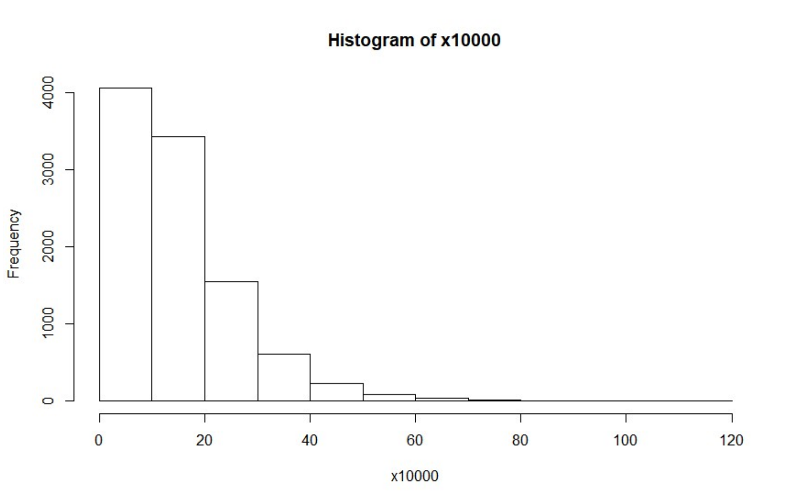
```
>pdf.T <- function(x){return(0.2*exp(-0.1*x)-0.2*exp(-0.2*x))}
```

ii) Repeat the previous step 10,000 times. This will give you 10,000 draws from the distribution of T. Try to avoid `for' loop. Use `replicate' function instead. Save these draws for reuse in later steps.

```
> x10000 = replicate(10000, max(rexp(n=1,rate = 1/10), rexp(n=1,rate = 1/10)))
```

```
Values
  x10000                  num [1:10000] 6.76 6.93 5.59 29.08 4.21 ...
```

iii) Make a histogram of the draws of T using `hist' function. Superimpose the density function given above. Try using `curve' function for drawing the density.




iv) Use the saved draws to estimate E(T). Compare your answer with the exact answer given above.

Analytically

```
> mean(x10000)
[1] 14.9337
```

v) Use the saved draws to estimate the probability that the satellite lasts more than 15 years. Compare with the exact answer computed in part (a).

Analytically

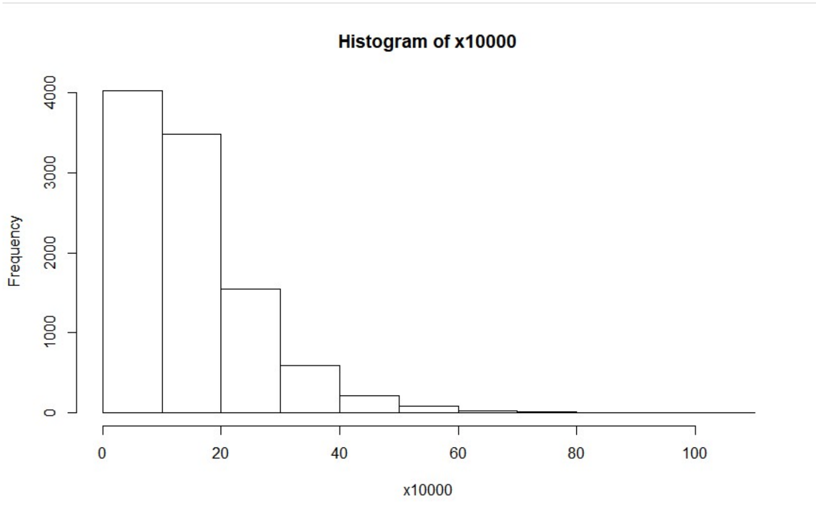
```
> 1 - pexp(15, rate = 1 / mean(x10000))
[1] 0.3662497
```

vi) Repeat the above process of obtaining an estimate of E(T) and an estimate of the probability four more times. Note what you see.

Analytically

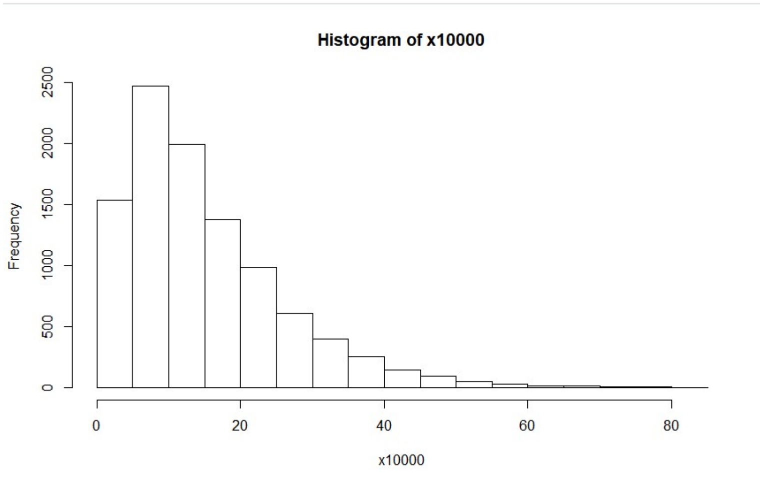
**Test: 2**

```
> x10000 = replicate(10000, max(rexp(n=1,rate = 1/10), rexp(n=1,rate = 1/10)))
> hist(x10000)
> mean(x10000)
[1] 14.92074
> 1 - pexp(15, rate = 1 / mean(x10000))
[1] 0.3659303
```



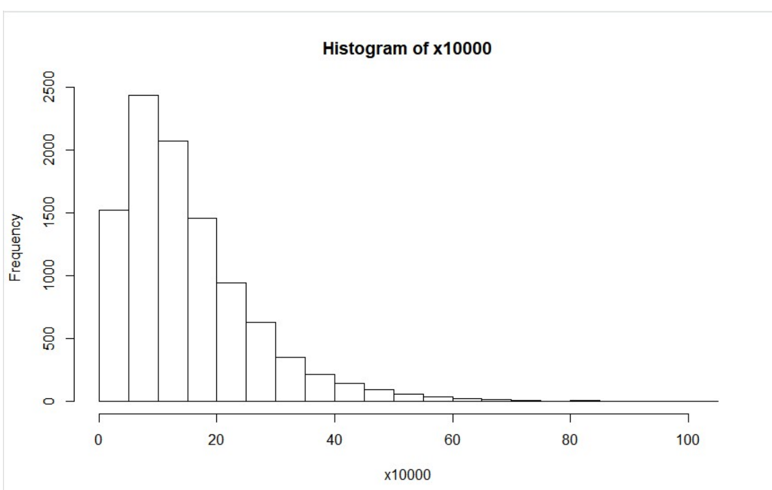

**Test: 3**

```
> x10000 = replicate(10000, max(rexp(n=1,rate = 1/10), rexp(n=1,rate = 1/10)))
> hist(x10000)
> mean(x10000)
[1] 15.08261
> 1 - pexp(15, rate = 1 / mean(x10000))
[1] 0.3699
```




**Test: 4**

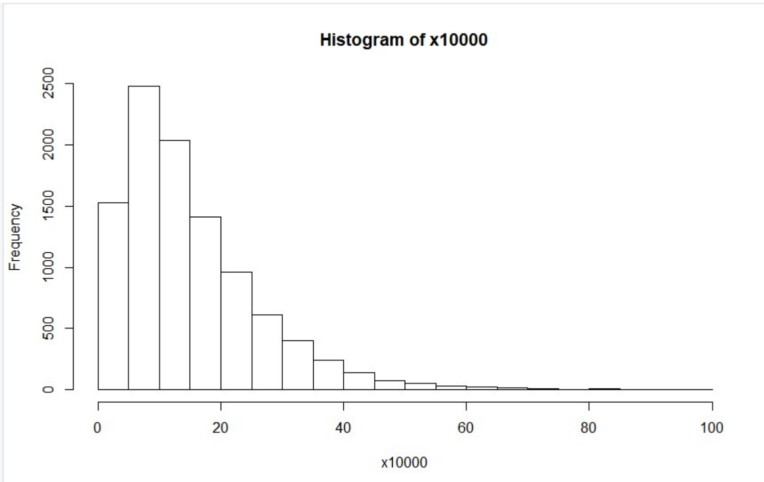
```
> x10000 = replicate(10000, max(rexp(n=1,rate = 1/10), rexp(n=1,rate = 1/10)))
> hist(x10000)
> mean(x10000)
[1] 15.01905
> 1 - pexp(15, rate = 1 / mean(x10000))
[1] 0.3683463
```




**Test: 5**

```
> x10000 = replicate(10000, max(rexp(n=1,rate = 1/10), rexp(n=1,rate = 1/10)))
> hist(x10000)
```

```
> mean(x10000)
[1] 14.98433
> 1 - pexp(15, rate = 1 / mean(x10000))
[1] 0.367495
```




| Test for Sample size 10,000 | E(T) | P(T>15) |
|---|---|---|
| Test 1 | 14.9337 | 0.3662497 |
| Test 2 | 14.92074 | 0.3659303 |
| Test 3 | 15.08261 | 0.3699 |
| Test 4 | 15.01905 | 0.3683463 |
| Test 5 | 14.98433 | 0.367495 |

Observation: From the output of test cases for sample size 10,000 the E(T) value is nearly 15 with small variation and same for P(T>15). These qualities match with the qualities that were logically determined in 1a. Thus, the definition of the Central Limit Theorem continues to be proven.

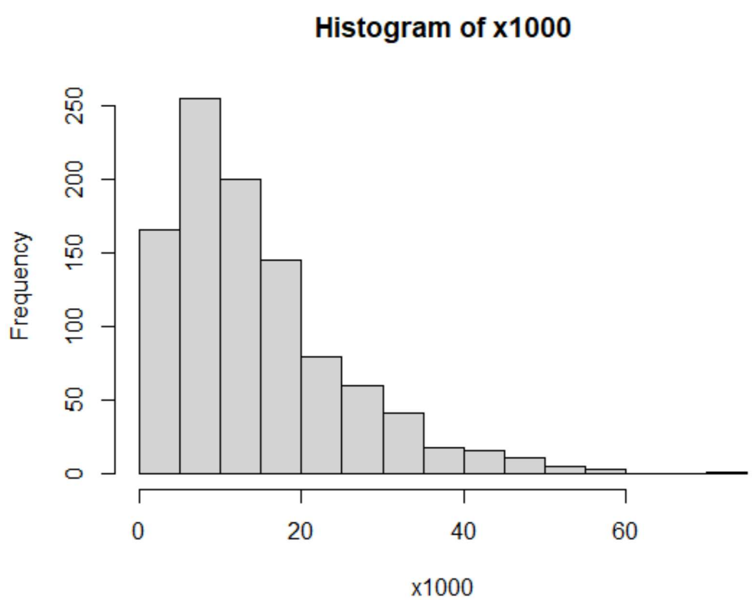
Problem 1c: Repeat part (vi) five times using 1,000 and 100,000 Monte Carlo replications instead of 10,000. Make a table of results. Comment on what you see and provide an explanation.

Solution 1c:

A. Sample size: 1000

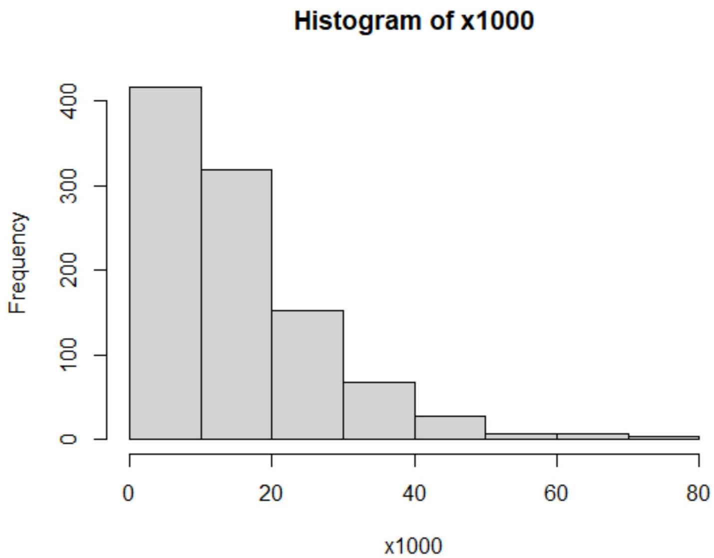
Test 1:

```
> #Sample size = 1000
> #Test 1
> x1000 = replicate(1000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))
> hist(x=x1000)
> mean(x1000)
[1] 14.54542
> 1 - pexp(15, rate=1/mean(x1000))
[1] 0.3565601
```



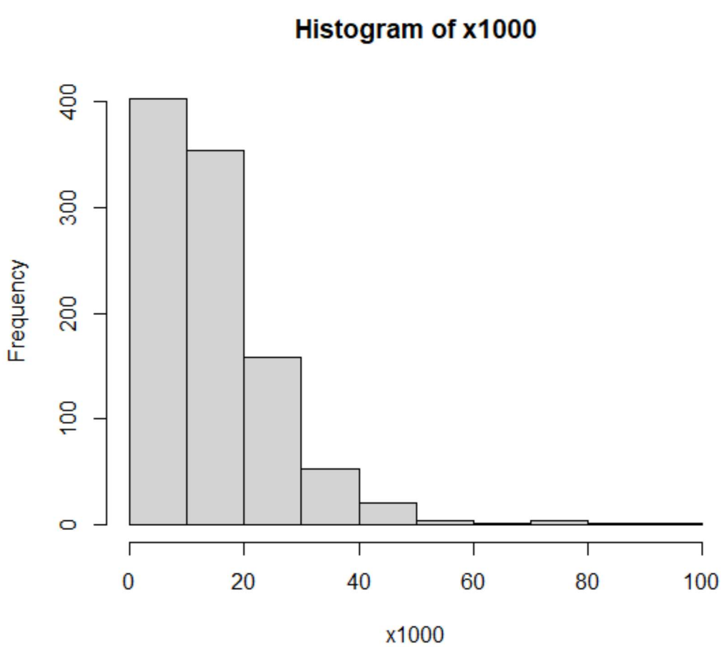

Test 2:

```
> #Sample size = 1000
> #Test 2
> x1000 = replicate(1000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))
> hist(x=x1000)
> mean(x1000)
[1] 15.36718
> 1 - pexp(15, rate=1/mean(x1000))
[1] 0.3767753
```



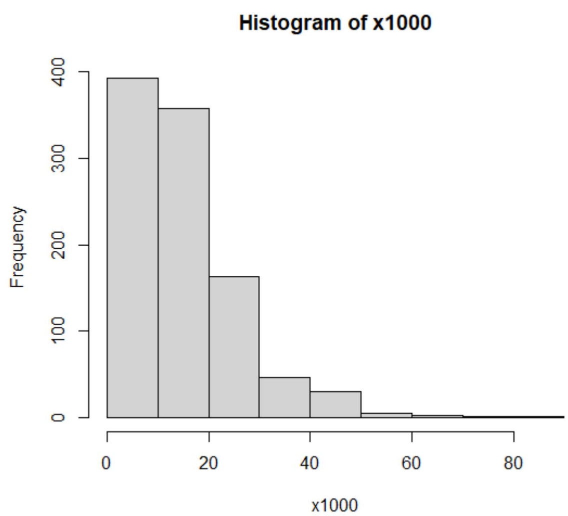

Test 3:

```
> #Sample size = 1000
> #Test 3
> x1000 = replicate(1000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))
> mean(x1000)
[1] 14.83317
> hist(x=x1000)
> 1 - pexp(15, rate=1/mean(x1000))
[1] 0.3637651
```



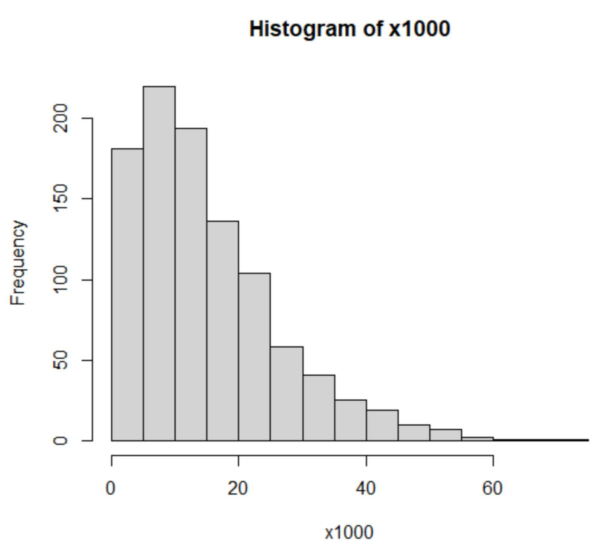
Test 4:

```
> #Sample size = 1000
> #Test 4
> x1000 = replicate(1000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))
> mean(x1000)
[1] 14.97857
> hist(x=x1000)
> 1 - pexp(15, rate=1/mean(x1000))
[1] 0.3673534
```




Test 5:

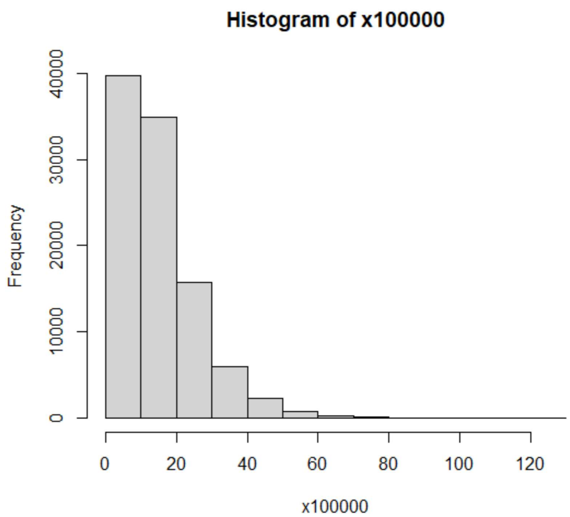
```
> #Sample size = 1000
> #Test 5
> x1000 = replicate(1000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))
> mean(x1000)
[1] 15.12636
> hist(x=x1000)
> 1 - pexp(15, rate=1/mean(x1000))
[1] 0.3709656
```



B. Sample Size: 100,000

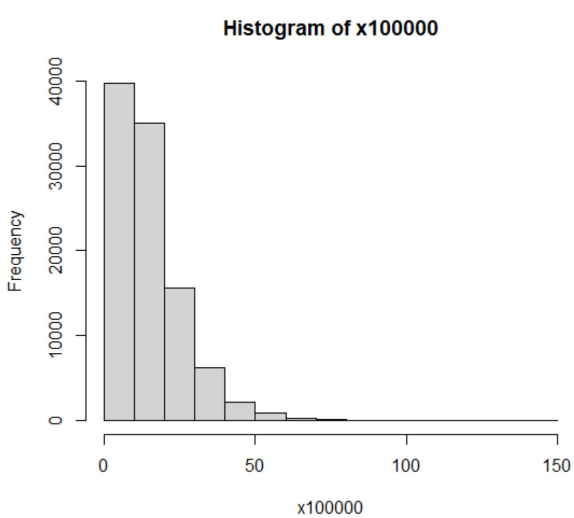
Test 1:

```
> #Sample size = 100000
> #Test 1
> x100000 = replicate(100000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))
> mean(x100000)
[1] 15.00775
> hist(x=x100000)
> 1 - pexp(15, rate=1/mean(x100000))
[1] 0.3680696
```



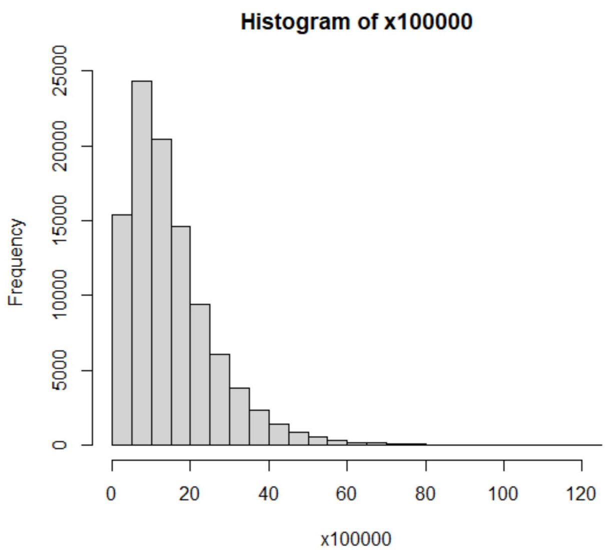

Test 2:

```
> #Sample size = 100000
> #Test 2
> x100000 = replicate(100000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))
> mean(x100000)
[1] 15.0032
> hist(x=x100000)
> 1 - pexp(15, rate=1/mean(x100000))
[1] 0.367958
```



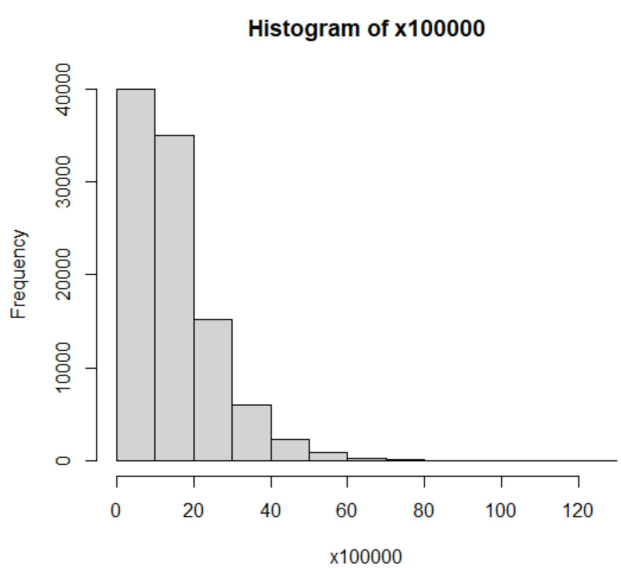
Test 3:

```
> #Sample size = 100000
> #Test 3
> x100000 = replicate(100000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))
> mean(x100000)
[1] 15.05803
> hist(x=x100000)
> 1 - pexp(15, rate=1/mean(x100000))
[1] 0.3692998
```



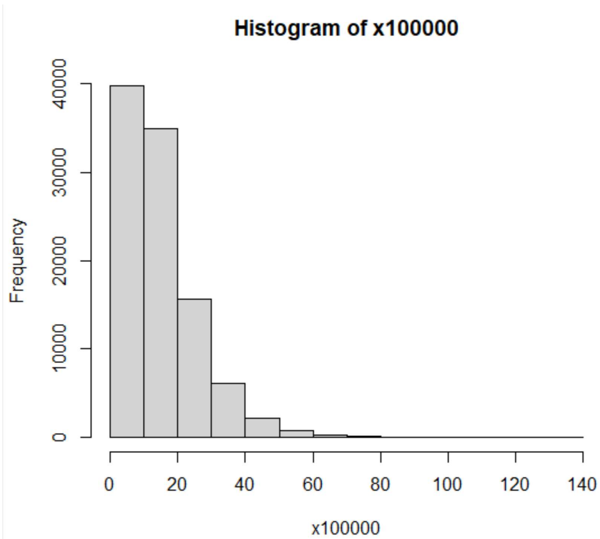

Test 4:

```
> #Sample size = 100000
> #Test 4
> x100000 = replicate(100000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))
> mean(x100000)
[1] 15.01342
> hist(x=x100000)
> 1 - pexp(15, rate=1/mean(x100000))
[1] 0.3682084
```



Test 5:

```
> #Sample size = 100000
> #Test 5
> x100000 = replicate(100000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))
> mean(x100000)
[1] 14.99997
> hist(x=x100000)
> 1 - pexp(15, rate=1/mean(x100000))
[1] 0.3678786
```




| Test for Sample Size 1000 | E(T) | P(T>15) |
|---|---|---|
| | | |
| Test 1 | 14.54542 | 0.3565601 |
| Test 2 | 15.36718 | 0.3767753 |
| Test 3 | 14.83317 | 0.3637651 |
| Test 4 | 14.97857 | 0.3673534 |
| Test 5 | 15.12636 | 0.3709656 |

| Test for Sample Size 100000 | E(T) | P(T>15) |
|---|---|---|
| | | |
| Test 1 | 15.00775 | 0.3680696 |
| Test 2 | 15.0032 | 0.367958 |
| Test 3 | 15.05803 | 0.3692998 |
| Test 4 | 15.01342 | 0.3682084 |
| Test 5 | 14.99997 | 0.3678786 |

Observation: Using the Monte Carlo approach to solve the problem for two test cases – 1000 and 100000 sample sizes and five tests for each sample size, it can be understood that as the sample size increases, the variation in values of E(T) and P(T>15) starts to reduce, and this corresponds directly to the Central Limit Theorem. E(T) ≈ 15 and P(T>15) ≈ 0.36 and both these values match the results derived in Problem 1(a) analytically.

Problem 2: Use a Monte Carlo approach to estimate the value of $\pi$ based on 10,000 replications.

Solution 2: The probability that a randomly selected point in a unit square with coordinates (0, 0), (0, 1), (1, 0), and (1,1) falls in a circle with centre (0.5, 0.5) inscribed in a square is equal to: Area of circle / Area of square

Area of circle = $\pi * (\frac{diameter}{2})$^2 , diameter = 1 (max value since our range is 0 to 1)
Area of circle = $\pi/4$
Area of square = 1 unit square
Area of circle / Area of square = $\pi/4$

A number has to be generated between 0 and 1 for both x and y in a uniform distribution, and this has to be iterated 10,000 times. This number must fall within the range of the circle inscribed in the square.

Total iterations (variable – sqpoints) = 10000
Variable – cpoints corresponds to the points generated within the circle

R Code with Monte Carlo Approach and resulting $\pi$ value:

```
> sqpoints <- 10000
> x <- runif(sqpoints, min=0, max=1)
> y <- runif(sqpoints, min=0, max=1)
> cpoints <- (x-0.5)^2 + (y-0.5)^2 <= 0.5^2
> mc.pi <- (sum(cpoints)/sqpoints) * 4
> mc.pi
[1] 3.1456
```

| Values | |
|---|---|
| cpoints | logi [1:10000] FALSE TRUE FALSE FALSE TRUE TRU… |
| mc.pi | 3.1456 |
| sqpoints | 10000 |
| x | num [1:10000] 0.183 0.7967 0.0358 0.2354 0.378… |
| y | num [1:10000] 0.07 0.4324 0.2835 0.0727 0.9277… |

Observation: The simulated value of $\pi$ is 3.1456 which is close to the true value of $\pi$ that is 3.14159. Therefore, the estimated value as calculated by the Monte Carlo approach is near or almost equivalent to the true value of $\pi$.