Mini-Project duo group 7

Venkata Bapanapalli

Karthik Meyyappan

Contributions: Both members equally contributed to analytically solve and implement the code of the given two questions.

1 a. P(T>15) = 1 – P(T<=15)

Analytically solved the above problem using integration and algebraic simplification.

**TEST 1:**

Xa = rexp(n=1, rate=1/10)

Xb = rexp(n=1, rate=1/10)

x = max(Xa, Xb)

ten\_thousand= replicate(10000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))

hist(ten\_thousand, prob = TRUE)

pdf = function(x){

return (0.2\*exp(-0.1\*x)-0.2\*exp(-0.2\*x))

}

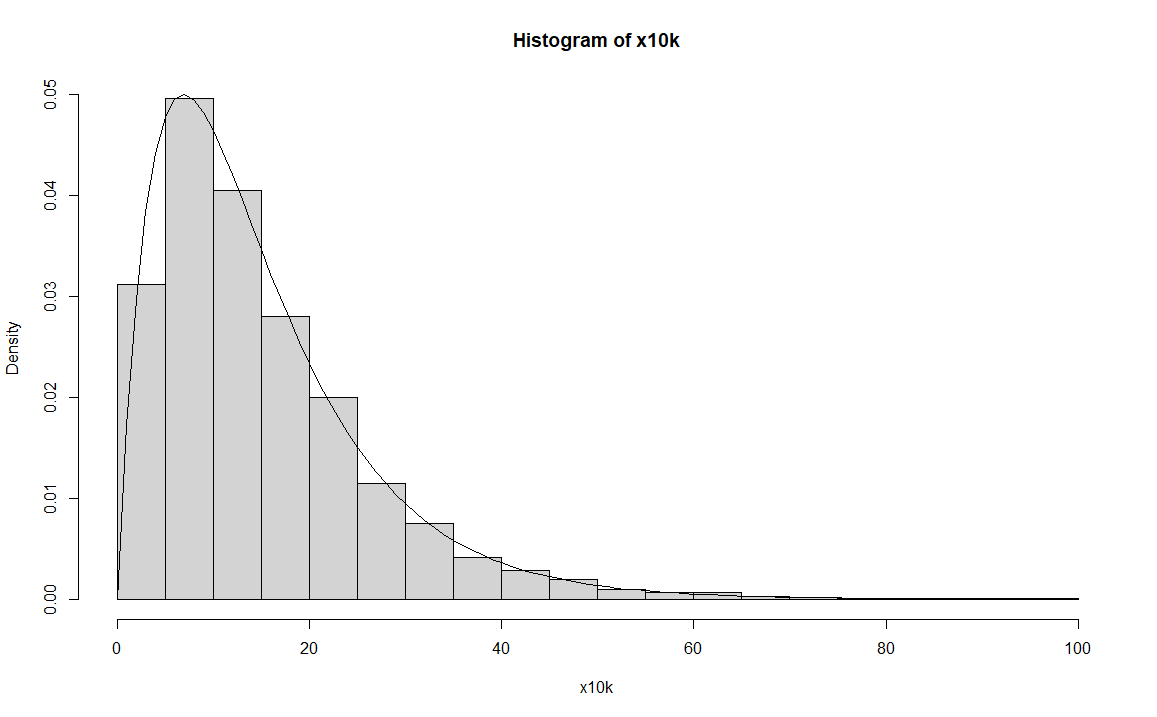
curve(pdf(x), add = TRUE)

mean(ten\_thousand)

**[1] 14.93673**

1-pexp(15, rate = 1/(mean(ten\_thousand)))

**[1] 0.3663244**



**TEST 2:**

Xa = rexp(n=1, rate=1/10)

Xb = rexp(n=1, rate=1/10)

x = max(Xa, Xb)

ten\_thousand= replicate(10000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))

hist(ten\_thousand, prob = TRUE)

pdf = function(x){

return (0.2\*exp(-0.1\*x)-0.2\*exp(-0.2\*x))

}

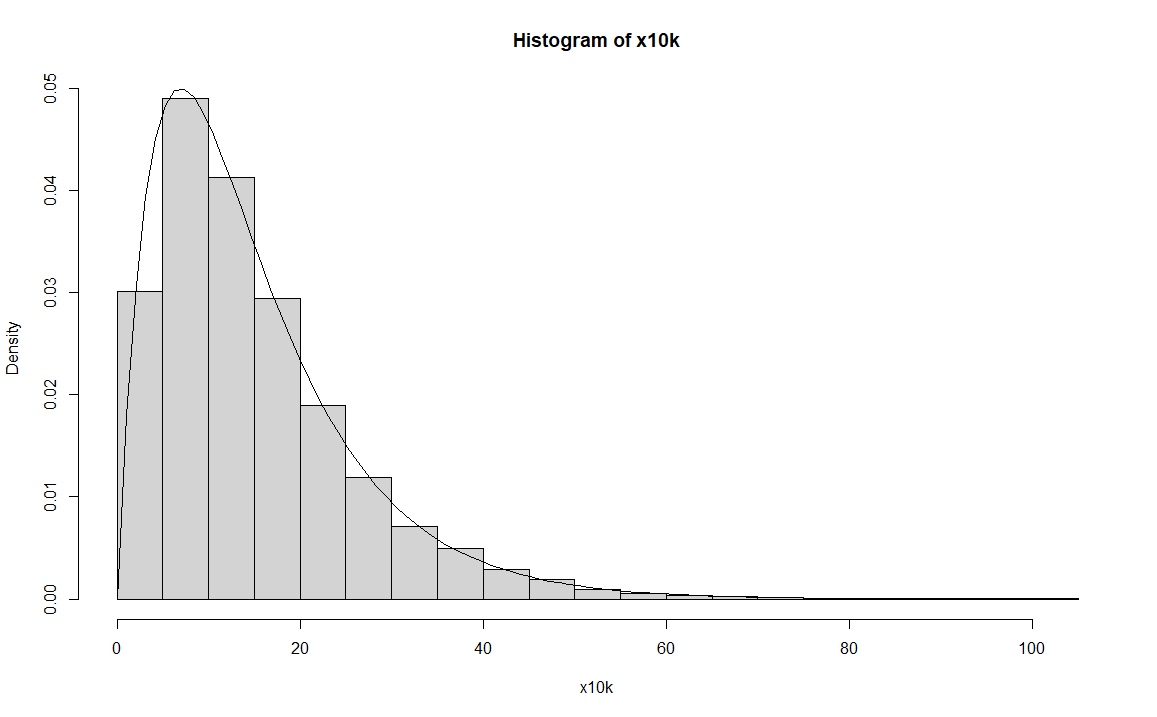
curve(pdf(x), add = TRUE)

mean(ten\_thousand)

**[1] 15.0346**

1-pexp(15, rate = 1/(mean(ten\_thousand)))

**[1] 0.3687271**



**TEST 3:**

Xa = rexp(n=1, rate=1/10)

Xb = rexp(n=1, rate=1/10)

x = max(Xa, Xb)

ten\_thousand= replicate(10000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))

hist(ten\_thousand, prob = TRUE)

pdf = function(x){

return (0.2\*exp(-0.1\*x)-0.2\*exp(-0.2\*x))

}

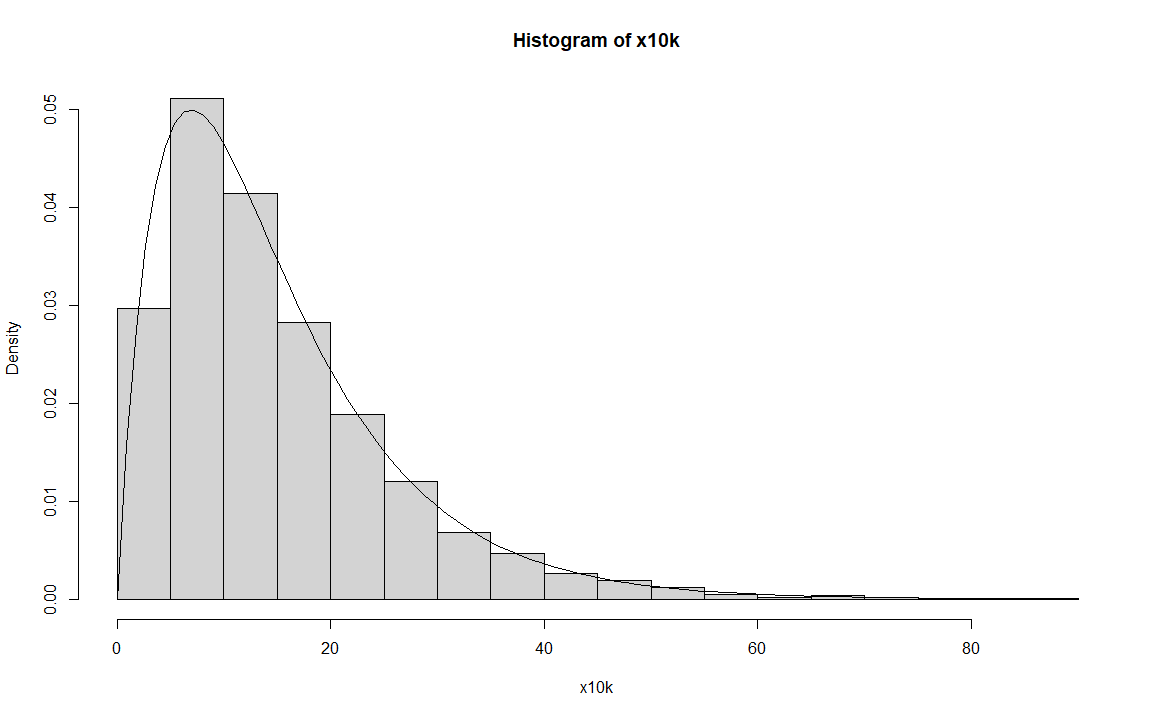
curve(pdf(x), add = TRUE)

mean(ten\_thousand)

**[1] 14.88006**

1-pexp(15, rate = 1/(mean(ten\_thousand)))

**[1] 0.364926**



**TEST 4:**

Xa = rexp(n=1, rate=1/10)

Xb = rexp(n=1, rate=1/10)

x = max(Xa, Xb)

ten\_thousand= replicate(10000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))

hist(ten\_thousand, prob = TRUE)

pdf = function(x){

return (0.2\*exp(-0.1\*x)-0.2\*exp(-0.2\*x))

}

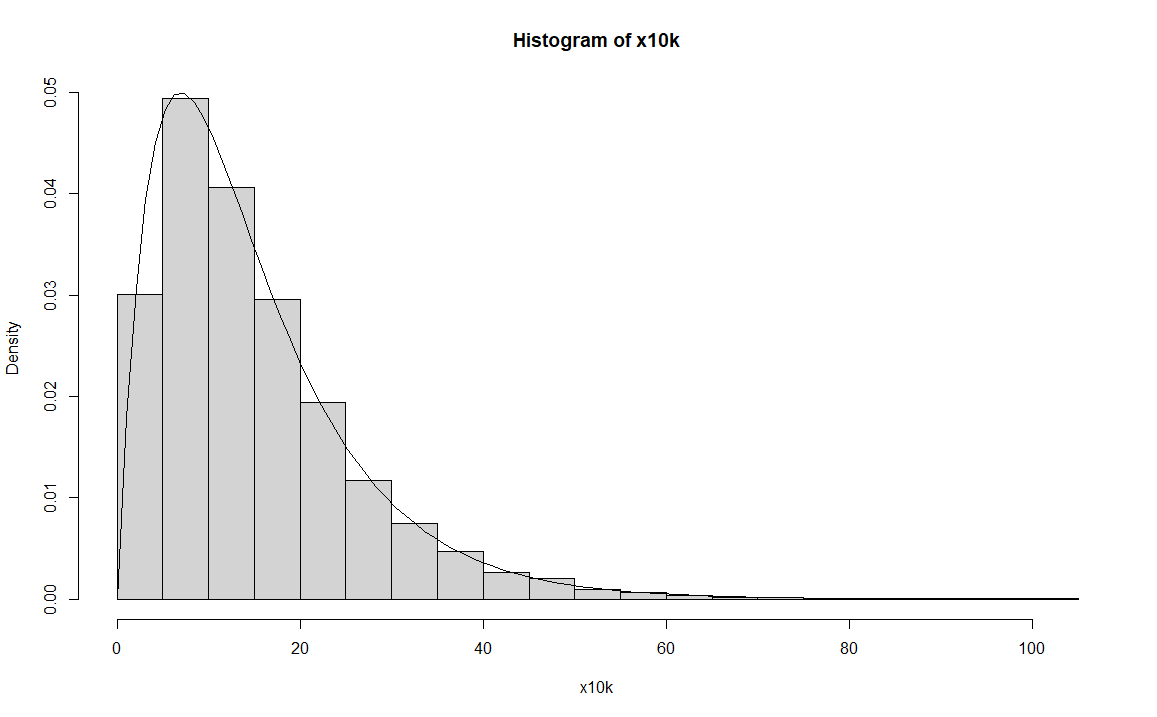
curve(pdf(x), add = TRUE)

mean(ten\_thousand)

**[1] 15.00983**

1-pexp(15, rate = 1/(mean(ten\_thousand)))

**[1] 0.3681205**



**TEST 5:**

Xa = rexp(n=1, rate=1/10)

Xb = rexp(n=1, rate=1/10)

x = max(Xa, Xb)

ten\_thousand= replicate(10000, max(rexp(n=1, rate=1/10), rexp(n=1, rate=1/10)))

hist(ten\_thousand, prob = TRUE)

pdf = function(x){

return (0.2\*exp(-0.1\*x)-0.2\*exp(-0.2\*x))

}

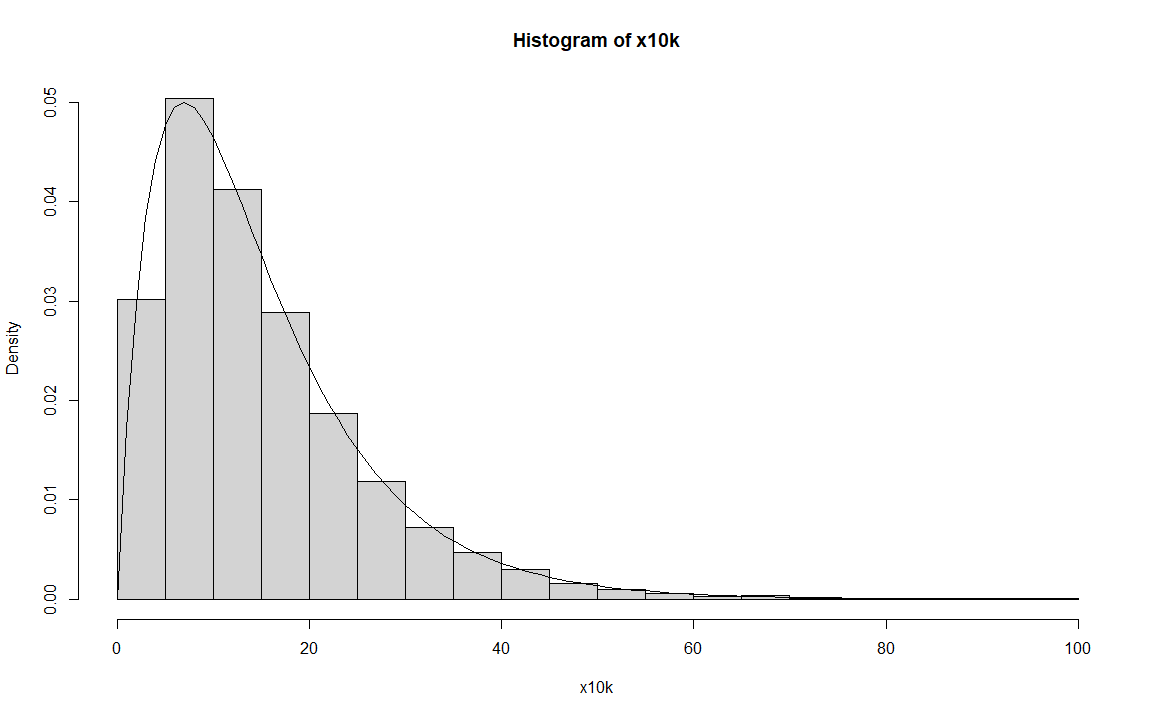
curve(pdf(x), add = TRUE)

mean(ten\_thousand)

**[1] 14.86456**

1-pexp(15, rate = 1/(mean(ten\_thousand)))

**[1] 0.3645427**



|  |  |  |
| --- | --- | --- |
| **1,000 replications:** | **Mean** | **Probability** |
| Test 1 | 15.27268 | 0.3745067 |
| Test 2 | 14.81808 | 0.3633907 |
| Test 3 | 14.99583 | 0.3677772 |
| Test 4 | 14.68535 | 0.3600811 |
| Test 5 | 14.6079 | 0.3581364 |

|  |  |  |
| --- | --- | --- |
| **100,000 replications:** | **Mean** | **Probability** |
| Test 1 | 14.99465 | 0.3677482 |
| Test 2 | 14.96038 | 0.3669065 |
| Test 3 | 14.96291 | 0.3669686 |
| Test 4 | 15.0092 | 0.3681049 |
| Test 5 | 14.98172 | 0.3674308 |

We can conclude from the collected data that as we increased the sample size from 1,000 to 100,000 the E(T) and P(T >15) reduces in variation. The larger the sample size the closer the mean is to 15.

2.

First, the probability of the number of points that fall within the circle that is inscribed in the square needs to be found. The probability of that would be the number of points that satisfy the function,

, and

divided by the number of total points generated, in this case 10000.

**Code:**

x = runif(10000, min=0, max=1)

y = runif(10000, min=0, max=1)

inscribed = (x-0.5)^2 + (y-0.5)^2 <= 0.5^2

4\*(sum(inscribed)/10000)