**MINI PROJECT #1**

**Name:** Omkar Nandkumar Dixit (ond170030)

**Name of group members:** Sushrut Patnaik (sxp175331)

**Contributions of each group members:**

We discussed both the questions together, but Omkar coded the first question and Sushrut coded the second question.

**Section 1:**

**Question 1 -**

(a)

= 0.3965

(b) Following are the steps taken in order to get the E(T) and P(T > 15) using Monte Carlo approach.

* The following code gives us the one Draw of the life times of Xa and Xb, then these draws are used to simulate one draw of the satellite lifetime T, as we need the one that has the longest or max lifetime of two.

*FirstSat.Xa <- rexp (1,0.1)*

*SecondSat.Xb <- rexp (1, 0.1)*

*Xt <- max (FirstSat.Xa, SecondSat.Xb)*

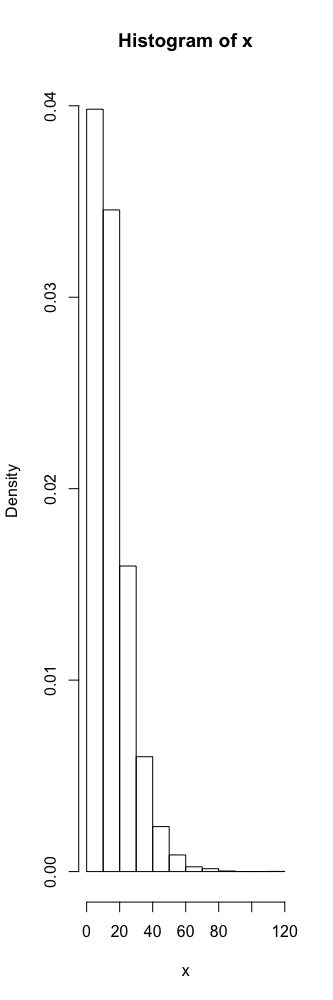
* Then we need to repeat the process of draws for 10,000 times so we use the replicate function to get the distribution of T. We save the draws for further use as well, apart from that we were asked not to use more than one line of code for steps (i) and (ii) so this line of code does the job of step I as well and the step ii. I separated two steps just to show the flow and the difference.

*x <- replicate (10000, max (rexp (1,0.1), rexp (1,0.1)))*

* In the Step (iii) we had to make a histogram of the draws of T (i.e x in our code),

*hist(x, probability = T)*

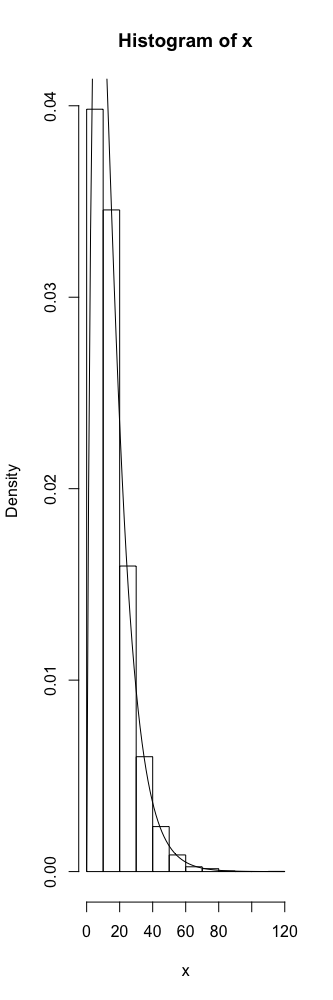
After executing the above command what we get is:



Then we have superimposed the density function.

*curve((0.2\*exp(-0.1\*x)-0.2\*exp(-0.2\*x)), add = T, xlab = "x", ylab = "density")*

After Executing the above command what we get is:



* In Step (iv) we have to estimate E(T), so we use

*mean(x)*

After executing what we get is

[1] 15.04581

Now the above answer that we get from the code is very close to the one that was provided to us i.e. E(T) = 15

* In Step (v), using the saved draws we have to estimate the probability that the satellite lasts more than 15 years.

*mean(abs(x) > 15)*

After executing this what we get is

[1] 0.401

Now the above output that we got is very close to 0.3965 that we got analytically in part a.

* In Step (vi), we have to repeat the process of estimating E(T) and Probability 4 times.

Following are the results that we got, first are the one that were observed in the first execution, and then followed by the repetitions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E(T) | 15.04581 | 15.03805 | 14.99007 | 15.03719 | 14.92586 |
| Probability | 0.401 | 0.3973 | 0.394 | 0.3958 | 0.3918 |

(c) In this part we were asked to change the number of Monte Carlo Estimations and then we tabulated the results for each to make some good observation.

Here we made a function to which we send a number that is the number of replications for that particular trial run to calculate the E(T) and Probability

*calculateProbabilityExpectedValue <- function(n) {*

*y <- replicate (n, max (rexp (1,0.1), rexp (1,0.1)))*

*tempEV<- mean(y)*

*tempP <- mean(abs(y) > 15)*

*return (c (tempEV, tempP))*

*}*

*replicate (5, calculateProbabilityExpectedValue (1000))*

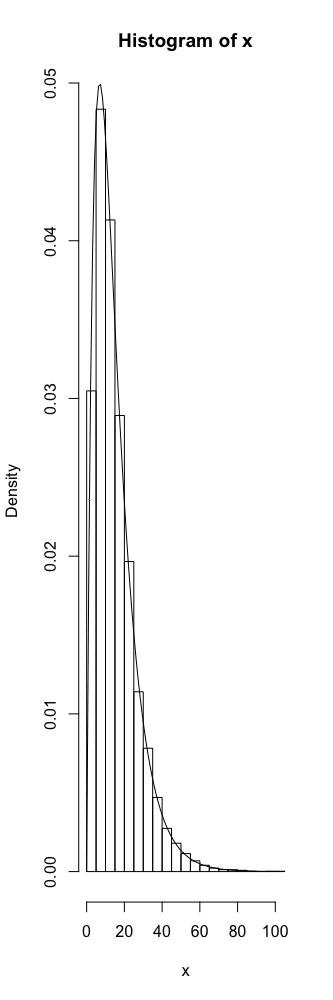
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E(T) | 15.10062 | 15.15041 | 15.71351 | 14.97937 | 14.88877 |
| Probability | 0.40400 | 0.39800 | 0.41900 | 0.38300 | 0.38900 |

*replicate (5, calculateProbabilityExpectedValue (100000))*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| E(T) | 15.06316 | 15.01315 | 15.00384 | 14.99368 | 15.0500 |
| Probability | 0.39666 | 0.39676 | 0.39607 | 0.39775 | 0.3994 |

From the results that we obtained from 10000, 1000 and 100,000, we can see that the results of 1000 replications the results are far away from what it was expected. But as the Law of Large numbers says more the number of replications or draws better will be the estimate, so if we see that the values in 100k table are closer to the estimated value than the ones in 1000 replications or 10k replications.

Note: Sometimes while executing the histogram, the tip of the curve might get cut, but its works correct for some.



E(T) = 15.07751

Probability Estimate = 0.3993

These were the Values were obtained during this execution for 10k replications.

**Question 2 –**

(10 points) Use a Monte Carlo approach estimate the value of π based on 10,000 replications. [Ignorable hint: First, get a relation between π and 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 the square. Then, estimate this probability, and go from there.]

Answer: We can consider a circle of radius 0.5, centred at (0.5, 0.5) inscribed in the unit square with coordinates (0, 0), (0, 1), (1, 0) and (1, 1). The Area of the square is 1 and the area of the circle will be πr2 which will be 0.25π or π/4.

Π = 4\*

We have generated random points that are uniformly distributed within the given area. We can check the number of points that lie within the circle and the total number of points. The ratio of the number of points within the circle and within the square will be approximately equal to the ratio of the areas of the circle and the square.

Π = 4\*

With a large number of points, we can have an estimate that is closer to the actual value of Pi.

**Section 2: R Code**

**Question 1 –**

*s set.seed(5)*

*#Q1.b.i*

*#first satelite*

*FirstSat.Xa <- rexp(1,0.1)*

*#second satelite*

*SecondSat.Xb <- rexp(1, 0.1)*

*#we need the one that lasts the longest so*

*Xt <- max(FirstSat.Xa, SecondSat.Xb)*

*#Q1.b.ii*

*x <- replicate(10000, max(rexp(1,0.1), rexp(1,0.1)))*

*#Q1.b.iii*

*hist(x, probability = T)*

*curve((0.2\*exp(-0.1\*x)-0.2\*exp(-0.2\*x)), add = T, xlab = "x", ylab = "density")*

*#Q1.b.iv*

*#here we are calculating E(T) Expected value*

*mean(x)*

*#Q1.b.v*

*#estimation of probability that the satellite lasts more than 15 years.*

*mean(abs(x) >15)*

*#Q1.b.vi*

*#run the program till here 4 more times as asked in the question*

*#this is the expected value*

*mean(x)*

*#this is the monte carlo estimate*

*mean(abs(x) > 15)*

*#Q1.c.*

*# here we were asked to change the monte carlo replications and five times*

*# so rather than running the program 5 times we have created a function to that job*

*calculateProbabilityExpectedValue <- function(n){*

*y <- replicate(n, max(rexp(1,0.1), rexp(1,0.1)))*

*tempEV<- mean(y)*

*tempP <- mean(abs(y) > 15)*

*return(c(tempEV, tempP))*

*}*

*#changing the monte carlo estimates to 1000 and repeating 5 times*

*replicate(5, calculateProbabilityExpectedValue(1000))*

*#changing the monte carlo estimates to 100000 and repeating 5 times*

*replicate(5, calculateProbabilityExpectedValue(100000))*

**Question 2 -**

*#randomly generates 10000 X-coordinates*

*x=runif(10000)*

*#randomly generates 10000 Y-coordinates*

*y=runif(10000)*

*#generates the distance of point (x,y) from centre(0.5,0.5)*

*distance=sqrt((x-0.5)^2+(y-0.5)^2)*

*#finds the proportion of points inside the circle to the total number of points or points inside square*

*ratio = length(which(distance<=0.5))/length(distance)*

## *#the ratio is multiplied by 4 to get the value of pi as per the equation stated above*

*val=ratio\*4*

*val*

*[1] 3.144*