Statistical Methods for Data Science

Project 1

By:

Chirag Shahi cxs180005

Krishnan Vijayaraghavan kxv190006

Contribution:

Krishnan: Question 1

Chirag: Question 2

Section 1

Answers:

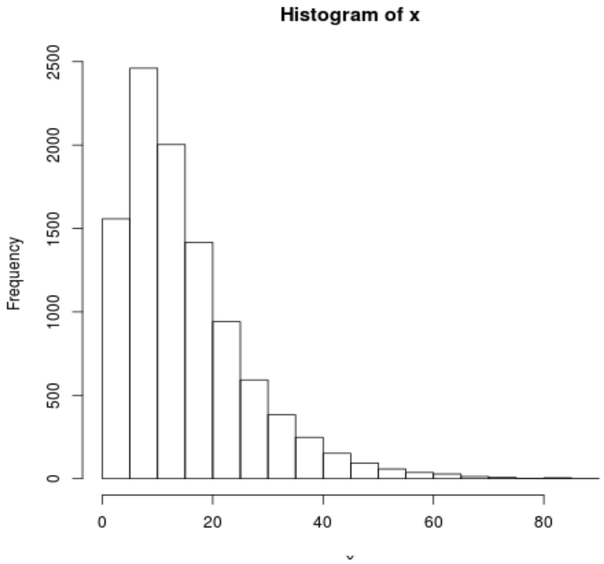
1.

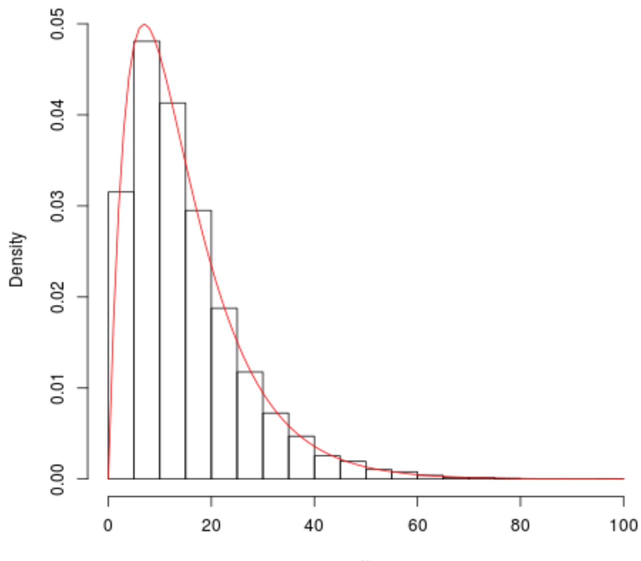
a) 0.3964733

b)

i) 30.61059

ii) 10,000 draws from the distribution of T iii)





iv) 15.11982

v) 0.3978

vi) 15.11982 15.11982 15.11982 15.11982

0.3978 0.3978 0.3978 0.3978

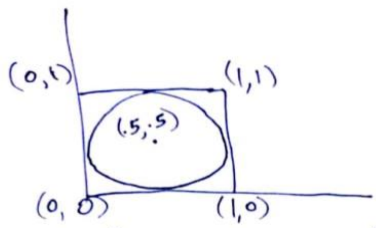
c)

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	15.26502	14.79926	15.56796	14.9543	14.7198
[2,]	0.42600	0.38700	0.41700	0.3850	0.3530

	[,1]	[,2]	[,3]	[,4]	[,5]
[1,]	14.99510	14.94204	15.00806	15.04241	14.99053
[2,]	0.39738	0.39399	0.39634	0.39663	0.39395

2. (10 points) Use a Monte Carlo approach estimate the value of π based on 10, 000 replications. [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 center (0.5, 0.5) inscribed in the square. Then, estimate this probability, and go from there.]

2)



Instead of taking a square of length 1, let us assume that the side of square is 'a' units long & thus radius of circle is a/2

area of square = a^2

area of circle = $\pi (a/2)^2 = \pi a^2/4$

Now we generate random points within the square that are uniformly distributed within the given area

And points $\propto$ area

$\therefore$ points in circle $\propto$ area of circle -①

points in square $\propto$ area of square -②

$\frac{①}{②} \Rightarrow \frac{\#\text{ points in circle}}{\#\text{ points in square}} = k \frac{\pi \cancel{a^2}/4}{\cancel{a^2}}$ (let k = 1) constant of proportionality

$\Rightarrow \pi = 4 \left(\frac{\#\text{ points in circle}}{\#\text{ points in square}} \right)$

Section 2

1.

a) Computing the probability that the lifetime of the satellite exceeds 15 years.

We compute the difference of 1 and probability that the lifetime of the satellite is less than 15 years that is CDF of distribution.
Since, it is given that the lifetime of the satellite T depends on the lifetime of the two blocks X_A and X_B, we multiply pexp function of two blocks.
R code:

*1 – [pexp (15, rate=0.1) * pexp ((15, rate=0.1)]*

Given

$$f_T(t) = \begin{cases} 0.2\,e^{(-0.1t)} - 0.2\,e^{(-0.2t)}, & 0 \le t < \infty \\ 0, & \text{otherwise.} \end{cases}$$

we need to find $P(T > 15)$

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

$$= 1 - (\text{CDF of } T)$$

$$= 1 - F(t)$$

$$F(t) = \int_0^\infty f(y)\,dy$$

$$= \int_0^\infty 0.2\,e^{-0.1t} - 0.2\,e^{-0.2t}$$

$$= 1 - 2e^{-0.1t} + e^{-0.2t}$$

$$P(T>15) = 1 - (1 - 2e^{-0.1t} + e^{-0.2t})$$

$$= (2e^{-0.1t} - e^{-0.2t})_{t=15}$$

$$= 2\,e^{-3/2} - e^{-3}$$

$$= 0.3964733$$

b)

i) Simulating one draw of the satellite lifetime T

Simulating one draw from of the block lifetimes

rexp function is used to simulate draws from a distribution

Syntax: rexp(n,rate)

R code:

rexp(1,0.1)

 # Simulating one draw of distribution of T

R code: max(rexp(1,0.1),rexp(1,0.1))

ii) Repeating previous step 10,000 times to get 10,000 draws from the distribution of T using replicate function.

R code:

x = replicate(10000, max(rexp(1,0.1),rexp(1,0.1)))

 # So, here x is the sample

iii) Using the hist and curve function

R code:

hist(x) (Histogram)

 # Since we are superimposing the density function, the probability density Function of T is taken as the first argument in the curve function

*R code: curve((0.2*exp(-0.1*x)-0.2*exp(-0.2*x)), from=0, to=100, add = T, xlab = "x", ylab = "density", col="red")*

iv) Estimating E(T) that is the mean of x

R code:

mean(x)

 # The estimated E(T) is similar to the computed E(T)

v) Estimating the probability that the satellite lasts more than 15 years

R code:

mean(x>15)

 # The estimated probability is similar to the computed probability

vi) Repeating above process 4 more times

R code:

replicate(4, mean(x))

replicate(4, mean(x>15))

 # The means and probabilities for 4 repetitions are the same.

c) Repeating part 6 five times using 1000 and 100000 Monte Carlo replications and making a table of results

Let us create a function to create a table and call the function later

```
Func = function(n)
{
y=replicate(n,max(rexp(1,0.1),rexp(1,0.1)))
ExpectedValue=mean(y)
Probability=mean(y>15)
return(c(ExpectedValue,Probability))
}
# 1000 Monte Carlo replications
R code: replicate(5,Func(1000))
#100000 Monte Carlo replications
R code: replicate(5,Func(100000))
# As the replications or the size of sampling increases, the accuracy of
  estimating E(T) and probability also increase that is there is less deviation
  from their actual values. The sample mean is approaching the theoretical
  mean. Thus, it follows law of large numbers.
```

2.

R Code:

```
x = runif(10000) #random points between 0,1 for x-axis

y = runif(10000) #random points between 0,1 for y-axis

z = sqrt((x - 0.5) ^ 2 + (y - 0.5) ^ 2) #distance of points from centre

pi = length(which(z <= 0.5)) * 4 / length(z) #ratio of points inside vs outside of
circle multiplied by 4

pi # print value
```