<u>**CS-6313 – Statistical Methods for Data Science**</u>

<u>**Mini Project #1**</u>
Group No - 5
Manan Dalal (MUD200000)
Lipi Patel (LDP210000)

Contribution of Team Members:

Worked Together with each other to solve both the problems analytically and logically. Both learned ‘R’ programming language. Both solved and presented their solutions for both the questions and worked together to present the best of both the solutions.

Question-1

Consider Exercise 4.11 from the textbook. In this exercise, let XA be the lifetime of block A, XB be the lifetime of block B, and T be the lifetime of the satellite. The lifetimes are in years. It is given that XA and XB follow independent exponential distributions with mean 10 years. One can follow the solution of Exercise 4.6 to show that the probability density function of T is

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

and E(T) = 15 years.

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

Solution

- ⇨ To compute the probability that the lifetime of the satellite exceeds 15 years, we would compute the probability of the satellite having lifetime less than or equal to 15.
- ⇨ From the given information the cumulative distribution of function can be inferred as (f_T (t)).

$$
\begin{aligned}
P (T > 15) &= 1 – P (T <= 15) \\
&= 1 – F (T <= 15) \\
&= 1 – {}_0\int^{\mathbf{15}} (f_T (t)) \\
&= 1 – {}_0\int^{\mathbf{15}} (0.2e^{-0.1t} – 0.2e^{-0.2t}) \\
&= 1 – {}_0{}^{15} [0.2 (((e^{-0.1t}) / (-0.1)) – ((e^{-0.2t}) / (-0.2)))] \\
&= 1 – {}_0{}^{15} [-2e^{-0.1t} + e^{-0.2t}] \\
&= 1 – [(-2e^{-0.1x15} +e^{-0.2x15}) – (-2e^{-0.1x0} +e^{-0.2x0})] \\
&= 1 – [-2e^{-1.5} + e^{-3} +2e^{0} – e^{0}] \\
&= 1 – [e^{-3} - 2e^{-1.5} +1] \\
&= 1- [0.049787 – 0.446260 + 1] \\
&= 1 – 0.603527 \\
&= \underline{\mathbf{0.396473}}
\end{aligned}
$$

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.

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. [Bonus: 1 bonus point for not taking more than 1 line of code for steps (i) and (ii).]

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. Note what you see.**

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

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).**

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.**

<u>Solution:</u>

i. The code below shows the simulation of one draw of the satellite lifetime.

```
# (i)

pdfT <- function(x) { return (0.2*exp(-0.1*x) + (0.2*exp(-0.2*x))) }
Xt <- max(rexp(n=1, rate=0.1), rexp(n=1, rate=0.1))

```

Xt	20.4831765211167

ii. Replicating the same simulation logic for 10,000 draws.

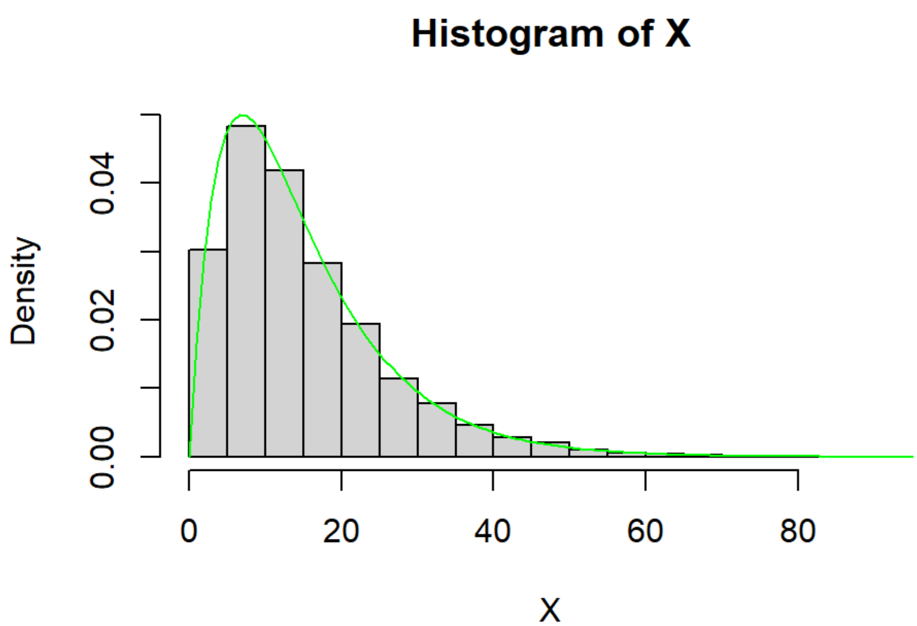
```
# (ii)
X <- replicate(10000, max(rexp(n=1, rate=0.1), rexp(n=1, rate=0.1)))
```

X	num [1:10000] 15.19 20.99 26.55 15.07 3.41 ...

iii. Generating a histogram to represent the draws of T and the probability distribution function.

```r
# (iii)

X <- replicate(10000, max(rexp(n=1, rate=0.1), rexp(n=1, rate=0.1)))
hist(X, probability = TRUE)
curve(0.2*exp(-0.1*x)-0.2*exp(-0.2*x), col = "green",  add = TRUE)
```



iv. Estimating E(T) based on the simulated draws.

```
# (iv)

print(mean(X))
```

```
[1] 15.07592
```

Observation

- The estimated value of E(T) is slightly different than the calculated value. By increasing the number of draws, we can get a result that is more accurate than our current estimation.

v. Estimating P(T > 15) based on the estimated E(T).

```
# (v)

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

```
[1] 0.3697368
```

Observation

- The value of P(T > 15) is slightly different than the calculated value because of the slight variation in the value of E(T).

vi. Repeating all the above steps four times to note the randomness of the experiment.

```r
# (vi)

means <- c()
px <- c()

for (i in 1:4) {
  X <- replicate(10000, max(rexp(n=1, rate=0.1), rexp(n=1, rate=0.1)))
  means <- c(means, mean(X))
  px <- c(px, 1 - pexp(15, rate= 1/mean(X)))
}

data_frame = data.frame(
  means = means,
  px = px,
)

colnames(data_frame) = c("E(X)", "P(T > 15)")

print(data_frame)
```

<u>Output</u>

```
> print(data_frame)
      E(X) P(T > 15)
1 15.12115 0.3708387
2 15.08635 0.3699910
3 14.91132 0.3656980
4 15.06983 0.3695882
```

<u>Observation</u>

- By running the same experiment 4 more times for 10,000 draws each, we get different values of both E(T) and P(T > 15) but these values are pretty close to each other. This observation shows the effect of randomness of the Monte Carlo approach.

c) 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

```r
meansX <- meansY <- pxX <- pxY<- c()

for (i in 1:5) {
  X <- replicate(100000, max(rexp(n=1, rate=0.1), rexp(n=1, rate=0.1)))
  meansX <- c(meansX, mean(X))
  pxX <- c(pxX, 1 - pexp(15, rate= 1/mean(X)))

  Y <- replicate(1000, max(rexp(n=1, rate=0.1), rexp(n=1, rate=0.1)))
  meansY <- c(meansY, mean(Y))
  pxY <- c(pxY, 1 - pexp(15, rate= 1/mean(Y)))
}

data_frameX = data.frame(means = meansX, px = pxX)
data_frameY = data.frame(means = meansY, px = pxY)
colnames(data_frameX) = colnames(data_frameY) = c("E(X)", "P(T > 15)")

print(data_frameX)
print(data_frameY)
```

Output

⇨ Running Monte Carlo approach 5 times for 100,000 draws.

```
> print(data_frameX)
      E(X) P(T > 15)
1 15.04615 0.3690096
2 14.99784 0.3678263
3 15.00000 0.3678794
4 15.00844 0.3680863
5 15.04721 0.3690354
```

⇨ Running Monte Carlo approach 5 times for 100,000 draws.

```
> print(data_frameY)
      E(X) P(T > 15)
1 14.61995 0.3584395
2 15.53796 0.3808394
3 15.35191 0.3764098
4 14.06700 0.3442714
5 14.95088 0.3666727
```

Observation

⇨ By observing the estimations for 1000, 10,000 and 100,000 draws, we can conclude that the accuracy of the estimation increases with increasing number of draws.

Question-2

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 center (0.5, 0.5) inscribed in the square. Then, estimate this probability, and go from there.]

Solution

- ⇨ The probability of a randomly selected point falling in the circle under the space of the square is equal to the area of circle/area of square which is equal to (pi/4).
  - P(E) = Area of Circle/ Area of Square = pi/4.
    - E is the event of a point falling in the circle.
  - Thus,
    - Pi = 4 * P(E)

- ⇨ Using the Monte Carlo approach, we would estimate the value of P(E) in the following steps:
  - Generate 10,000 draws of X and Y co-ordinates from a Uniform Distribution between (0,1).
  - Find the proportion of points that fall in the circle.
    - A point is said to be inside the circle if:

$$(X - 0.5)^2 + (Y - 0.5)^2 <= 0.25$$

  - Calculate Estimated Value of Pi
    - Pi = 4 * P(E)

⇨ **R Code**

```
# Generating X,Y co-ordinates
x <- runif(10000)
y <- runif(10000)

# Calculating Probability of point falling in the circle
probE <- mean(1*((x - 0.5)^2 + (y - 0.5)^2 <= 0.25))

# Calculating the value of Pi
pi <- probE*4
```

⇨ **Output**

Values	
pi	3.1476
probE	0.7869
x	num [1:10000] 0.489 0.885 0.243 0.204 0.188 ...
y	num [1:10000] 0.5771 0.5501 0.5822 0.7335 0.0599 ...