Statistical Methods for Data Science

Mini Project 1 (Solution)

---

1. From the given information, it follows that the lifetimes (in years) X_A and X_B of respective blocks A and B follow independent exponential distributions with $\lambda = 1/10 = 0.10$, and the satellite lifetime $T = \max\{X_A, X_B\}$. The density function $f_T(t)$ of T is given and it is also given that $E(T) = 15$.

   (a) To find $P(T > 15)$, we integrate $f_T(t)$ over 15 to ∞, giving

$$\begin{aligned} P(T > 15) &= \int_{15}^{\infty} f_T(t)dt = 0.2\int_{15}^{\infty} \exp(-0.1t)dt - 0.2\int_{15}^{\infty} \exp(-0.2t)dt \\ &= 0.2 * \frac{\exp(-0.1 * 15)}{0.1} - 0.2 * \frac{\exp(-0.2 * 15)}{0.2} \approx 0.3965. \end{aligned}$$

   (b) Figure 1 presents a histogram of 1000 draws of T superimposed with its probability density function $f_T(t)$. As expected, we observe good agreement between the histogram and the density.

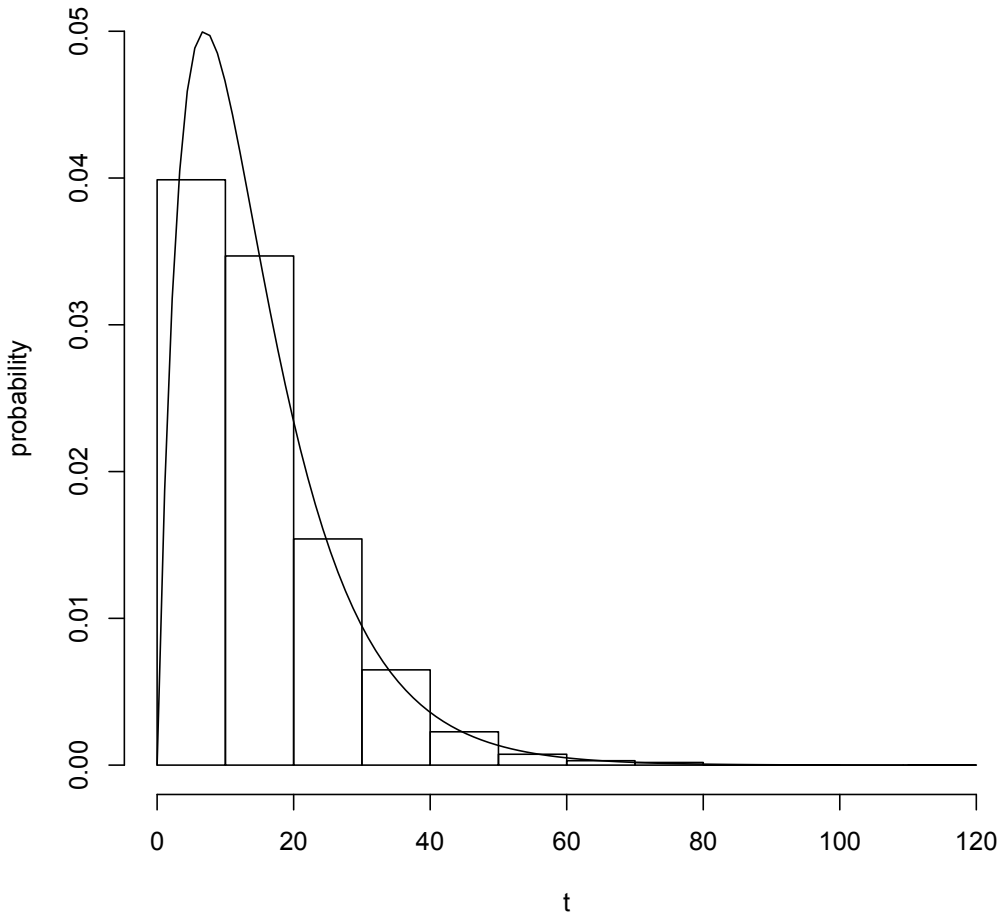

Figure 1: A histogram of 10000 draws of T superimposed with its probability density function $f_T(t)$.

The five Monte Carlo estimates for $E(T)$ and $P(T > 15)$ based on 10,000 draws and the corresponding exact answers are summarized in Table 1. The estimates based on 1,000 and 100,000 draws are also summarized therein. All estimates are rounded to three decimal places.

			replication number				
parameter	true value	# draws	1	2	3	4	5
$E(T)$	15	1,000	15.237	14.814	15.383	14.873	15.249
$E(T)$	15	10,000	14.986	15.075	14.836	14.988	15.011
$E(T)$	15	100,000	14.957	15.026	15.008	15.014	14.949
$P(T > 15)$	0.3965	1,000	0.397	0.363	0.426	0.392	0.406
$P(T > 15)$	0.3965	10,000	0.396	0.400	0.396	0.392	0.396
$P(T > 15)$	0.3965	100,000	0.395	0.398	0.398	0.396	0.395

Table 1: Summary of Monte Carlo estimates and true values.

(c) From the law of large numbers, we expect each of the approximations to be close to the corresponding true value. We also expect the accuracy of approximation to improve as the number of draws increases until it reaches a value for which the approximation is quite good. Any further increase in the number of draws beyond this value may not lead to a noticeable improvement in accuracy. The results summarized in Table 1 are consistent with what we expect. In particular, the five approximations for each of $E(T)$ and $P(T > 15)$ are close the corresponding true value. There is variability in the approximations around the true value and the variability clearly decreases as the number of draws increases from 1,000 to 10,000. The increase in accuracy when the number of draws is further increased to 100,000 is less noticeable.

2. Let us consider the unit square and the circle inscribed in it as described in the hint. Let A be the event that a randomly selected point the square falls in the circle. From the given information, we can see that

$$P(A) = \frac{\text{area of the circle}}{\text{area of the square}} = \frac{\pi}{4},$$

implying that $\pi = 4P(A)$. Thus, to approximate the value of π, we can approximate $P(A)$ using a Monte Carlo approach and multiply the answer by 4. This will involve randomly generating a point in the square. This can be achieved by simulating the x coordinate and the y coordinate of the point as independent draws from a Uniform (0, 1) distribution. Next, the point (x, y) would fall in the given circle if its distance from the center of circle is less than or equal to the radius of the circle, i.e.,

$$\sqrt{(x - 0.5)^2 + (y - 0.5)^2} \leq 0.5.$$

Thus, to approximate $P(A)$, all we need to do now is to simulate 10000 points in the square and apply the above test to find the proportion of points that fall in the the circle. Using `R`, $P(A)$ is approximated as 0.7894. Therefore, the estimated value of π is $4 * 0.7894 = 3.1576$. This value may be contrasted with the exact value 3.1416 of π when rounded to 4 decimal places. Of course, we can get a more accurate approximation by increasing the number of simulated points to, say, 100,000.

R code:

```
##########################
# R code for Exercise 1  #
##########################

# simulate 10000 draws from the distribution of T

set.seed(123)
x <- replicate(10000, max(rexp(1, 0.1), rexp(1, 0.1)))

# make a histogram

hist(x, prob = T, ylim = c(0, 0.05), xlab = "t", ylab = "probability", main = "")

# superimpose the density function of T

# a function to compute the density function of T

density.fun <- function(x){
return(0.2*exp(-0.1*x) - 0.2*exp(-0.2*x))
}

curve(density.fun, from = 0, to = max(x), add = T)

# a function to simulate nsim draws from the given distribution
# and computing mean and P(X > 15)

sim.fun <- function(nsim, lambda.A = 0.10, lambda.B = 0.10){
x <- replicate(nsim, max(rexp(1, lambda.A), rexp(1, lambda.B)))
result <- c(mean = mean(x), prob = mean(x > 15))
return(result)
}

set.seed(123)

# get 5 approximations based on 1000 draws

# > round(replicate(5, sim.fun(1000)), 3)
       # [,1]   [,2]   [,3]   [,4]   [,5]
# mean 15.237 14.814 15.383 14.873 15.249
# prob  0.397  0.363  0.426  0.392  0.406
# >

# get 5 approximations based on 10000 draws

# > round(replicate(5, sim.fun(10000)), 3)
       # [,1]   [,2]   [,3]   [,4]   [,5]
# mean 14.986 15.075 14.836 14.988 15.011
# prob  0.396  0.400  0.396  0.392  0.396
# >
```

```
# get 5 approximations based on 100000 draws

# > round(replicate(5, sim.fun(100000)), 3)
       # [,1]   [,2]   [,3]   [,4]   [,5]
# mean 14.957 15.026 15.008 15.014 14.949
# prob  0.395  0.398  0.398  0.396  0.395
# >

##########################
# R code for Exercise 2  #
##########################

# a function for computing the probability that a randomly selected
# point in the given square falls in the given circle using nsim
# draws and multiplying it by four to get the value of pi

pi.fun <- function(nsim){
  x <- runif(nsim)
  y <- runif(nsim)
  prob <- mean(sqrt((x-0.5)^2 + (y-0.5)^2) <= 0.5)
  pi <- 4*prob
  return(c(prob = prob, pi = pi))
}

# get the value of estimated value of pi

set.seed(123)

# > pi.fun(10000)
  # prob     pi
# 0.7894 3.1576
# >

# value of pi given by R

# > pi
# [1] 3.141593
# >
```