Statistical Methods for Data Science
CS 6313.001: Mini Project #1

Due on Thursday January 31, 2019 at 10am

Instructor: Prof. Min Chen

Shyam Patharla **(sxp178231)**

Contents

Section 1 Answers

Problem 1.1 Analytically calculating the probability that the lifetime of the satellite will exceed 15 years

Based on the conditions in the problem, we have:

$$f_T(t) = \begin{cases} 0.2e^{-0.1t} - 0.2e^{-0.2t}, 0 \leq x < \infty, \\ 0, \text{otherwise} \end{cases} \tag{1}$$

where T is the random variable representing the lifetime of the satellite.

We can calculate the following using formulas

$$P(T > 15) = \int_{15}^{\infty} f(t)\,\mathrm{d}t = \int_{15}^{\infty} (0.2e^{-0.1t} - 0.2e^{-0.2t})dt = 0.3965 \tag{2}$$

Problem 1.2 Monte Carlo Simulation

(a) Simulate a Draw of X_A, X_B and T

We first simulate X_A. From inverse transform method, we have:

$$U = F(X_A) = 1 - e^{\lambda X_A} (X_A > 0) \tag{3}$$

$$X_A = F^{-1}(U) \tag{4}$$

Solving for X_A we get:

$$X_A = -10ln(1 - U) \tag{5}$$

Therefore, to simulate a draw from the distribution of X_A, we can call `runif()` function in R as follow:

```
> Xa = (-10) * 2.303 * log10(1 - runif(1))
> print(Xa)

## [1]  5.694788
```

Since X_B also has the same distribution as X_A (exponential distribution with the same mean lifetime of 10 years), we repeat the procedure we did with X_A:

```
> Xb =(-10) * 2.303 * log10(1 - runif(1))
> print(Xb)
```

```
## [1] 2.024587
```

We know the lifetime of a satellite in any draw is the sum of the lifetimes of we get in draws of X_A and X_B. Using the above draws to simulate a draw for T, we get:

```
> Xa + Xb

## [1]7.719375
```

(b) Repeat Step (a) 10,000 times

We simulate 10,000 draws of T repeating the procedure in (a).

```
> t <- replicate(10000, (-10) * (2.303) * log10(1 - runif(1)) +
                  (2.303) * log10(1 - runif(1)) * (-10))
```

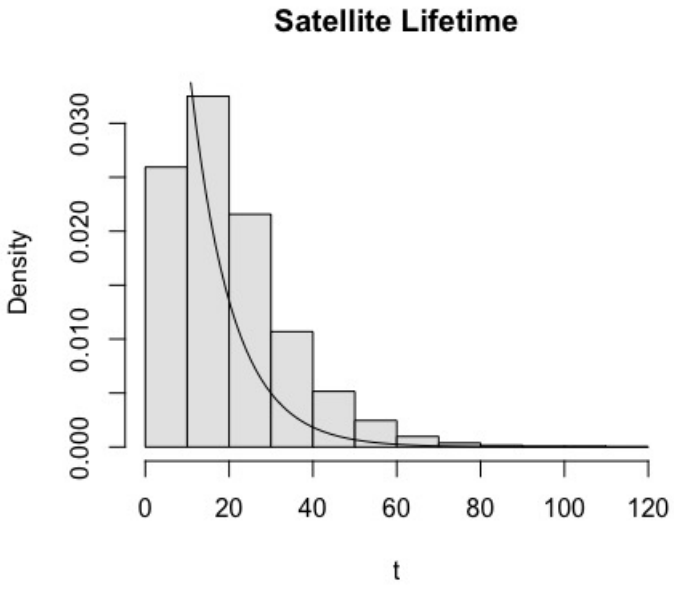
We save our draw results in vector t.

(c) Make a histogram of the 10,000 draws using 'hist' function

Since random variables X_A and X_B have the same exponential distribution with the same parameter λ, their sum Z = $X_A + X_B$ also has the same parameter λ=0.1. We plot the histogram using the following command in R:

```
> hist(x, probability = TRUE, col = gray(.9), main = "Satellite Lifetime")
> curve( dexp(x,0.1), add = T )
```

We also draw the curve function of the probability of T with $\lambda = 0.1$.



(d) Estimate E(T)

We estimate the expected value of T using the following command:

```
> mean(x)

## 19.9922
```

Analytical calculation gives us a value for E(T) of 15. We get a value of around 20 using Monte Carlo simulation.

(e) Estimate the probability that the satellite will last more than 15 years

We run the following command to estimate P(T > 15):

```
> sum(x>15)/10000

## 0.5529
```

Analytical calculation gives us a value for P(T>15) of around 0.4. We get a value of 0.55 using Monte Carlo simulation.

(f) Taking 10,000 draws for 4 more times

Using Monte Carlo simulation with 10,000 draws taken 4 times we get the results in table 1.

Draw #	E(T)	P(T>15)
1	20.18618	0.5636
2	20.10753	0.5632
3	20.00512	0.5580
4	19.95078	0.5566

Table 1: 4 Times 10000 Draws

The value of E(T) remains more or less stable around 20. The value of P(T>15) stabilises around 0.55

Problem 1.3 Repeat part (e) 5 times with 1,000 and 100,000 draws instead of 10,000

Using Monte Carlo simulation with 1,000 draws taken 5 times we get the results in table 2.

Draw #	E(T)	P(T>15)
1	19.29913	0.541
2	20.48910	0.568
3	19.83506	0.548
4	19.79278	0.557
5	20.22468	0.543

Table 2: Taking 1,000 Draws for 5 times

The value of E(T) remains more or less stable around 20. The value of P(T>15) stabilises around 0.5.

Using Monte Carlo simulation with 100,000 draws taken 5 times comes the results in table 3

Draw #	E(T)	P(T>15)
1	20.10833	0.56098
2	19.95363	0.55437
3	19.97778	0.55673
4	20.02585	0.55765
5	19.96851	0.55651

Table 3: Taking 100,000 Draws for 5 times

The value of E(T) remains more or less stable around 20. The value of P(T>15) stabilises around 0.55. We get a greater stability in values compare with n=1000.

Problem 1.4 Estimating the value of pi using Monte Carlo Simulation

Consider a square of unit length with vertices at (0,0), (0,1), (1,0), and (1,1) and a circle circumscribed in it with centre at (0.5, 0.5) and a radius 0f 0.5 units. Let us look at the ratio of the area of the circle to the area of the square.

$$\frac{Area(Circle)}{Area(Square)} = \frac{\pi * (0.5)^2}{1^2} = 0.25\pi \tag{6}$$

Say we randomly choose a point inside the square. The probability that it falls inside the circle is 0.25π. So we can see that:

$\pi = 4$ * Probability (a random point chosen in the square falls inside the circle)

So we can randomly choose a point from the standard uniform distribution and see of it falls inside the circle, If we do this for 10,000 times and determine the probability above, we can approximate the value of pi.

To implement this, we first generate a random value from the standard uniform distribution for the value of x-coordinate. We repeat the same for the y coordinate. We then check if:

$$\sqrt{(x-0.5)^2 + (y-0.5)^2} <= 0.5 \tag{7}$$

i.e. if the point lies inside the circle. We repeat this for 10,000 times.

```
> x = runif(10000)
> y = runif(10000)
> z = sqrt((x-0.5)^2 + (y-0.5)^2)
> sum(z<=0.5)/10000*4

## [1] 3.1464
```

Section 2 R Code

```
############################################################
#R Code for problem 1
############################################################

# Problem 1 (b)
# part (i) : Simulate a draw of Xa, Xb and T
```

```
simulatedraw <- function(){

  # Simulate a draw of Xa using the function Xa = -10 ln (1-U)
  #Xa = (-10) * 2.303 * log10(1 - runif(1))

  # Print the value of the draw
  print(Xa)

  # Simulate a draw of Xb using the function Xb = -10ln(1-U)
  Xb = (-10) * 2.303 * log10(1 - runif(1))

  # Print the value of the draw
  print(Xb)

  # Simulate the draw of T using the above values of Xa and Xb
  print(Xa+Xb)

}


# > simulatedraw()
# [1] 3.821385
# [1] 17.6624
# [1] 21.48379


# Part (ii) Replicate the draw in part (a) for 10,000 times and
#store the results

getdraw <- function(n){

  # Store the values of T in the variable t
  t <- replicate(n, (-10) * (2.303) * log10(1 - runif(1)) +
                   (2.303) * log10(1 - runif(1)) * (-10))

  return (t)

}



# > getdraw(10000)

# [1] 10.0029291 13.0884744 20.1432919  7.9062688 13.7474493
# [6]  3.2303715  3.1979952 14.0537767 36.2692867 14.9240531
# [11] 63.3234714 16.0721903 20.3507367  0.8239364 27.7695047
# [16] 45.9597216 18.7721382  9.4141846 10.4780209 49.1960118
# [21]  2.3129311 14.7745887 12.8468813 25.9557017 14.9179431
...
```

```
...
...


# Part (iii) : Draw a histogram and a add curve for the exponential
# density function
# We call the following commands:

 t = getdraw(10000)
 hist(t, probability = TRUE, col = gray(.9), main = "Satellite Lifetime")
 curve( dexp(x,0.1), add = T )


# Part (iv) : Estimate the value of E(T)

# > m = getdraw(10000)
# > mean(m)

# [1] 19.99373

# Part (v) : Estimate the probability that the lifetime of the
# satellite exceeds 15 years i.e. P(T>15)

# > m = getdraw(10000)
# > sum(m>15)/10000

 # [1] 0.5532


# Part (vi): Simulating 10,000 draws for 4 times
# We add another paramter k to our getdraw(n) function
# and write a new function getdraw (n, k)

getdraws <- function(n,k){

   # Getting n draws for k times
   t <- replicate(k, (-10) * (2.303) * log10(1 - runif(n)) +
                     (2.303) * log10(1 - runif(n)) * (-10))

   # Finding the mean value in each column of the result
   m <- apply(t,2,mean)

   # Finding the probability of T>15 in each column of the result
   p  <- apply(t,2,function(x) sum(x>15)/sum(x>=0))

   # Combining the mean and probability results
   result <- cbind(m,p)

   # Naming the columns
```

```
   colnames(result) <- c("E(T)","P(T>15)")

   return (result)

 }

# We simply call getdraw(10000,4):

# > getdraws(10000,4)

#       E(T)   P(T>15)
# [1,] 20.18618  0.5636
# [2,] 20.10753  0.5632
# [3,] 20.00512  0.5580
# [4,] 19.95078  0.5566


# Problem 1, Part (c) : Monte Carlo replications for k=5 times using
# n=1000 and  n=100000, where n is the number of draws taken

# > getdraws (1000, 5)

#       E(T)     P(T>15)
# [1,] 19.29913   0.541
# [2,] 20.48910   0.568
# [3,] 19.83506   0.548
# [4,] 19.79278   0.557
# [5,] 20.22468   0.543

# > getdraw (100000, 5)

#       E(T)    P(T>15)
# [1,] 20.10833 0.56098
# [2,] 19.95363 0.55437
# [3,] 19.97778 0.55673
# [4,] 20.02585 0.55765
# [5,] 19.96851 0.55651
```

```
############################################################
# R Code for problem 2
############################################################


# Compute the approximate value of pi using Monte Carlo Simulations
# We write a function approxpi(), which uses the
# concept of a circle of radius 0.5 with centre at (0.5,0.5)
# circumscribed in a square of unit length with vertices at
# (0,0), (0,1), (1,0) and (1,1).
```

```
approxpi <- function(){

  # Generate 10,000 draws for x-coordinate of point inside unit square
  x = runif(10000)

  # Generate 10,000 draws for y-coordinate of point inside unit square
  y = runif(10000)

  #Computing distance of the point from the centre (0.5, 0.5)
  x = sqrt((x-0.5)^2 + (y-0.5)^2)

  # Finding the number of points which fall inside the circle
  sum(x<=0.5)/10000*4

}


# > approxpi()

# [1] 3.1404
```