1.

(b) (i) and (ii)

To simulate 10,000 draws from the distribution of T(Combining (i) and (ii) using replicate() which has better efficiency in memory management for large number of vector inputs compared to for loop)

t.sim10k = replicate(n = 10000, expr = max(rexp(1, 0.1), rexp(1, 0.1)))

here n = no.of.replications

expr = expression which will be executed repeatedly

(iii) To draw a histogram based on the above simulation

hist(x = t.sim10k, freq = FALSE, main = 'Simulation of 10000 draws', xlab = 't', ylab = 'probability density')

here x = input vector values

freq = if FALSE, plots probability density function

main = title of histogram

xlab = x-axis label

ylab = y-axis label

To superimpose density function using curve()

curve(expr = (0.2 \* exp(-0.1 \* x) - 0.2 \* exp(-0.2 \* x)), n=10000, from=0, to= 100 , add=TRUE, col='red')

here expr = expression/function for which the curve should be drawn

n = no.of values

from = lower limit

to = upper limit

add = if TRUE , adds to an already existing plot

col = color of the curve

Observations : ###update here

(iv) To estimate E(T) from saved draws

We wrote a function for estimation E(T) since it will be needed to call multiple times for various simulation count.

mean.function = function(count) {

t = replicate( n = count, expr = max(rexp(1, 0.1), rexp(1, 0.1)))

return (mean(t))

}

On executing this function,

mean.function(10000) , we got E(T) = 14.92805

Observations : ##update here(also try to attach image and check if it looks good)

(v) ##update same as (iv) modifying the function, I’ve attached image for this also