Question Description:

A program is divided into 3 blocks that are being compiled on 3 parallel computers. Each block takes an Exponential amount of time, 5 minutes on the average, independently of other blocks. The program is completed when all the blocks are compiled. Suppose X be the time it take the program to be compiled.
Conduct a Monte Carlo simulation study using R to obtain the answer for E(X2).

Part 1

Simulate the block execution times X_1, X_2 and X_3. Use the simulated values to simulate X^2, the execution time of the whole program.

Solution:
The command about describing exponential distribution in R language is
>rexp(n,rate)
X_1, X_2, X_3 follow the exponential distribution. In this question, we know that n=3, because there are three independent blocks. The rate is 0.2 because 5 minutes is the average value and rate = 1/average. From the question we also know that the program is completed when all the blocks are compiled. So we should select the maximum value from these three blocks. So the command is

>x=max(rexp(3,0.2))   # to get the maximum of Xi. i = 1, 2, 3.
>x^2   # to get the value of X^2

The result is X^2.

```
> x=max(rexp(3,0.2))
> x
[1] 16.00912
> x^2
[1] 256.2919
> 
```

Because my sample size is too small, so the result is not very accuracy. Now I will increase my sample size in Part 2. That is the process of Monte Carlo simulation.

Part 2

Repeat the previous step 10,000 times. This will give you 10,000 draws from the distribution of X.

Solution:

>replicate(10000, max(rexp(3,0.2))
repeat the previous step 10,000 times to get 10000 times of maximum of X_i

The result has so 10,000 values so I will not show the screenshot.

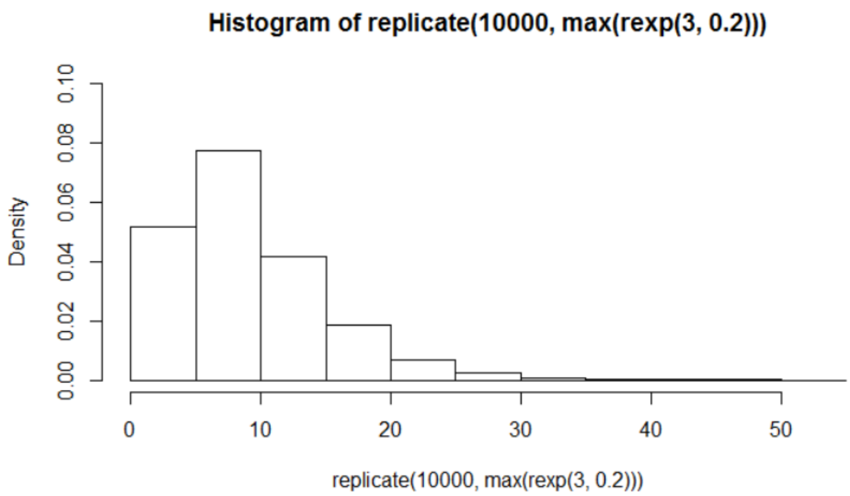
Part 3

Make a histogram of the draws of X. Superimpose the theoretical density function of X. Try using the R function 'curve' for drawing the density. Note what you see.

Solution:
First step is getting the function of f(x) and E(x).We know that F(X) = 1- exp(-λX). In this question we know X=max(X_1, X_2, X_3), so based on the knowledge of C.D.F, we should calculate: P(X≥X_1)* P(X≥X_2)*P(X≥X_3) = $[F(X)]^3$. So getting the P.D.F from C.D.F:f(x) = 0.6*(1-exp(-0.2*x))^2*exp(-0.2*x), x>0 ;
E(x) = ∫ x*f(x) dx
E(x) = (-3*x-15)*exp(-0.2*x)+(3*x+7.5)*exp(-0.4*x)+(-x-5/3)*exp(-0.6*x) ;
The command of making a histogram of draws of X is shown below:

>hist(replicate(10000,max(rexp(3,0.2))),ylim=c(0.0001,0.1),prob=TRUE)
draw histogram of 10000 times of maximum of exponential distribution, which Lambda=0.2, n=3.

The result of the histogram is shown below:

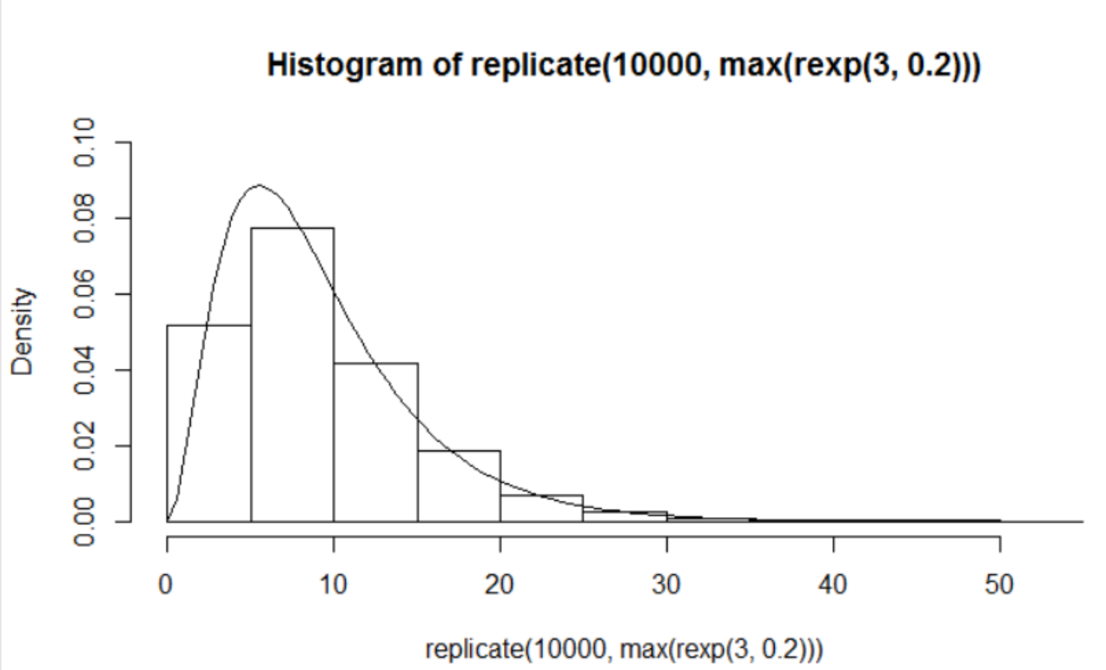


From the result I see that the highest density is between 5 to 10, which is same with what we guess. The execute time which more than 30 minutes close to none.

And then, using the R function ‘curve’ for drawing the density. The command is shown below:

```
>curve(0.6*(1-exp(-0.2*x))^2*exp(-0.2*x),add=TRUE)
#use curve function to draw the density
```

The result is:



The result shows more accuracy. It kind like using continuous function estimate the discrete function. From the screenshot I find that the highest density closes to 5 minutes, which is the average time.

Part 4

Use the draws to estimate $E(X^2)$. Compare your answer with the exact answer of $E(X^2)$. Note what you see.

Solution:
I have already got 10,000 draws from part 2. Now I want to use those data to calculate $E(X^2)$. First thing is calculate X^2 by using the command:

```
> z=replicate(10000,max(rexp(3,0.2)))
```

And then I use the function: D(X) = E[X-E(X)]^2 = D(X)=E(X^2)-[E(X)]^2
So E(X^2) = D(X)+[E(X)]^2

The answer is shown below:

```
> z=replicate(10000,max(rexp(3,0.2)))
> var(z)+mean(z)^2
[1] 118.6467
>
```

The answer is really close to 118. So the result is correct!

Part 5

Repeat the process of obtaining an estimate of $E(X^2)$ five times. Compare each estimate with the exact value. Note what you see.

Solution:

```
> mean((replicate(10000,max(rexp(3,0.2))))^2)
[1] 116.3591
> mean((replicate(10000,max(rexp(3,0.2))))^2)
[1] 119.6938
> mean((replicate(10000,max(rexp(3,0.2))))^2)
[1] 114.1429
> mean((replicate(10000,max(rexp(3,0.2))))^2)
[1] 118.2068
> mean((replicate(10000,max(rexp(3,0.2))))^2)
[1] 116.3226
>
```

From these results I find the values are close to 118, but not equal to 118 every time. This means even through the sample size is large, but still cannot shows the theoretical value.

Part 6

Comment on how your results would change if you use 1,000 Monte Carlo replications instead of 10,000. What if you use 100,000 replications? Justify your answers.

Solution:
First I use 1,000 Monte Carlo replications, repeat 5 times to obtain $E(X^2)$, the result is shown below:

```
>  mean((replicate(1000,max(rexp(3,0.2))))^2)
[1] 112.0888
> mean((replicate(1000,max(rexp(3,0.2))))^2)
[1] 119.8891
> mean((replicate(1000,max(rexp(3,0.2))))^2)
[1] 116.3478
> mean((replicate(1000,max(rexp(3,0.2))))^2)
[1] 108.7861
> mean((replicate(1000,max(rexp(3,0.2))))^2)
[1] 111.818
> 
```

From the result I find some value even equal to 108.7861, which is **far from** 118. The distribution is sparse. The reason is because my sample size is decrease.

Then, I use 100,000 Monte Carlo replications, repeat 5 times to obtain $E(X^2)$, the result is shown below:

```
> mean((replicate(100000,max(rexp(3,0.2))))^2)
[1] 118.4524
> mean((replicate(100000,max(rexp(3,0.2))))^2)
[1] 117.6289
> mean((replicate(100000,max(rexp(3,0.2))))^2)
[1] 118.1404
> mean((replicate(100000,max(rexp(3,0.2))))^2)
[1] 118.9318
> mean((replicate(100000,max(rexp(3,0.2))))^2)
[1] 118.2244
> 
```

From the result I find that those 5 values are really close to 118. That means the answer is more accuracy than 1,000 and 10,000 Monte Carlo replications. This is because the sample size is bigger.

Conclusion:

From this mini project I know the principle of Monte Carlo. When I use a large size of sample, the experiment can show the result which is very close to theoretical value. When the sample size increase, the result will more accurate. When the sample size decrease, the error will increase.