Report of the assignment for the simulation work

Applied Statistics

Victor Malheiro

For the first part of the work, I had to get samples from a target distribution,

$$f(x) = \frac{2}{\sqrt{\pi}} x^{\frac{1}{2}} e^{-x}, x > 0$$

And because it is difficult to get those samples directly, I had to sample from one easy to sample from distribution $g(x)$.

By the rejection method, as long as there is some proposal distribution that is easy to sample from that can be multiplied by a constant k and if $k.g(x) \geq f(x)$ than I can use a random sample from $k.g(x)$ to get a random sample for $f(x)$.

Therefore, the idea is that I need to create an envelope over the target distribution $f(x)$ *and* if the proposal distribution is the uniform distribution that is multiplied by a big enough positive k, then it will envelop the target distribution.

The reason I want to create this envelope for all the support of $f(x)$ is that when I am going to sample, I accept or reject that sample according to where it is on the envelope. If the sample is above the target distribution and below the proposal distribution, I reject the sample but if it is below the target distribution, I accept it.

By doing that I have a distribution from where I can easily sample from and then I accept a subset of those samples and that subset will be distributed according to our target distribution.

Because $f(x)$ have a continuous distribution, $g(x)$ can come from the uniform distribution.

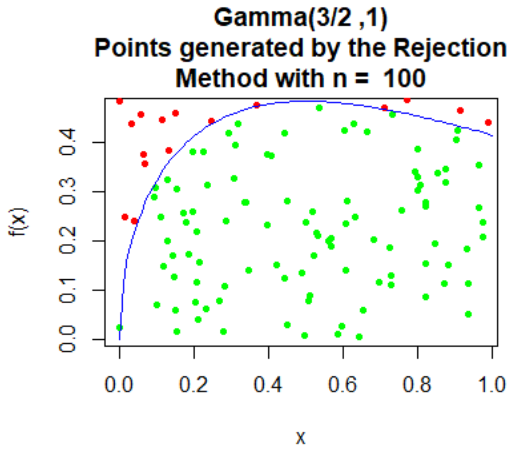
Taking the given suggestion of,

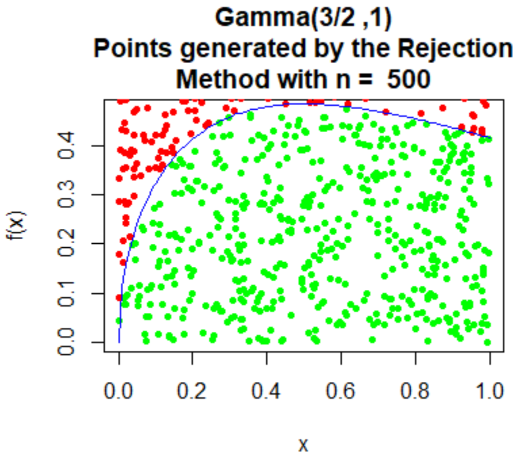
$$f(x) \leq 3\sqrt{\frac{3}{2\pi e}} \frac{2}{3} e^{-\frac{2}{3}x}$$

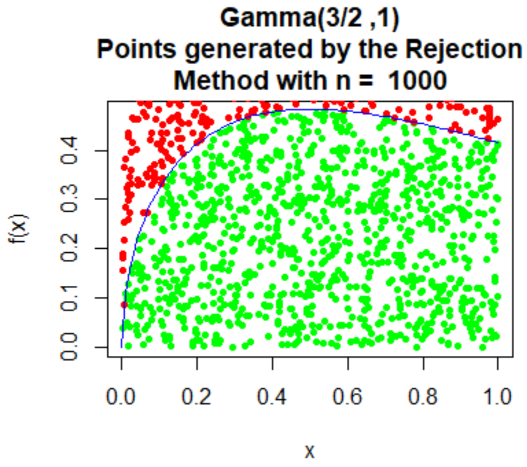
And knowing that $f(x) \leq k.g(x)$, then to generate the sample I used the *reject.gamma* function, that uses the while loop to generate values for a given size of the sample that generates Y $\sim U_{[0,1]}$ and U $\sim U_{[0,k.g(Yi)]}$ and this variable uses the variable Z $\sim U_{[0,1]}$.

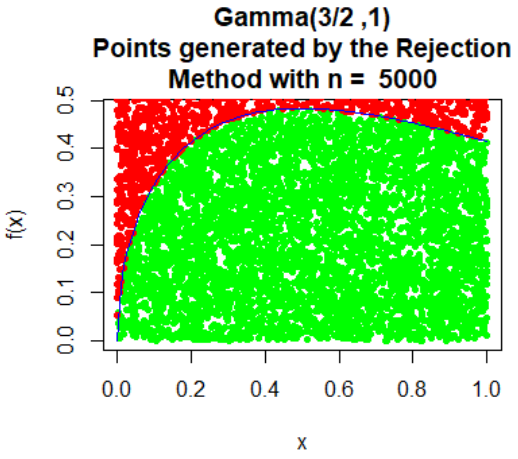
After that, is used a conditional function to test if the value of U is superior or not to the target distribution.

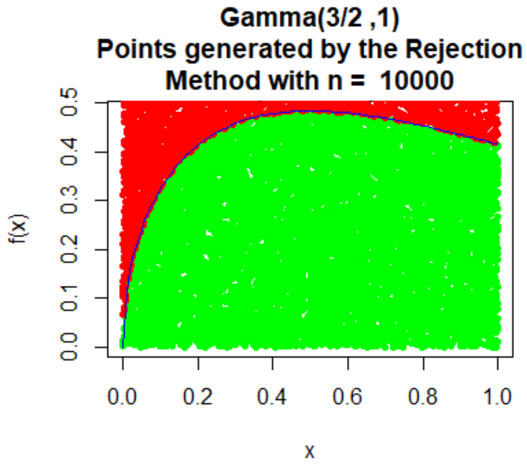
In the end, I represented it graphically for five different size samples, that are below, the points that are rejected have color red, the ones that are not have color green and the graphic have the line of the Gamma (3/2, 1) distribution function separating them.



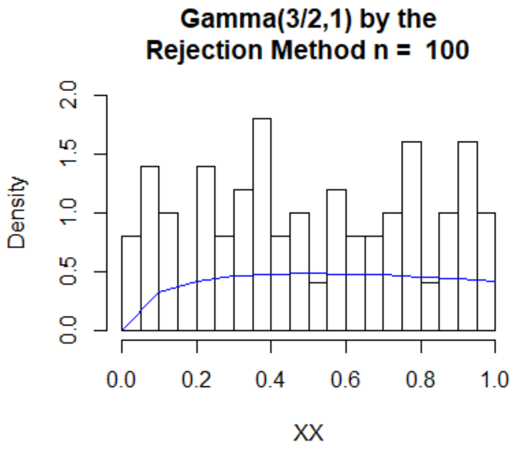


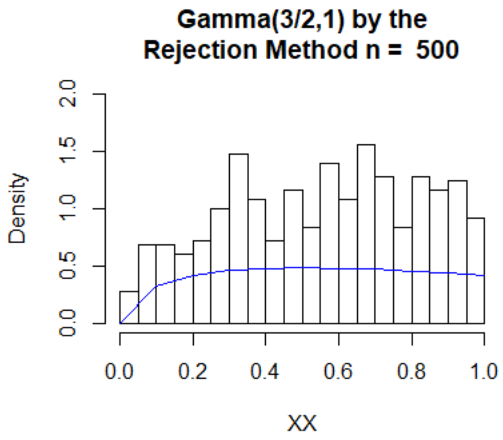


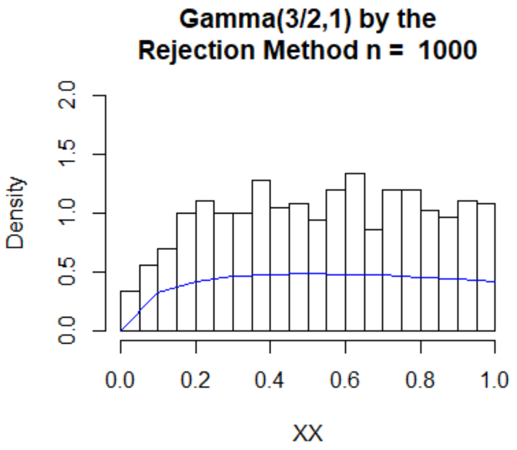




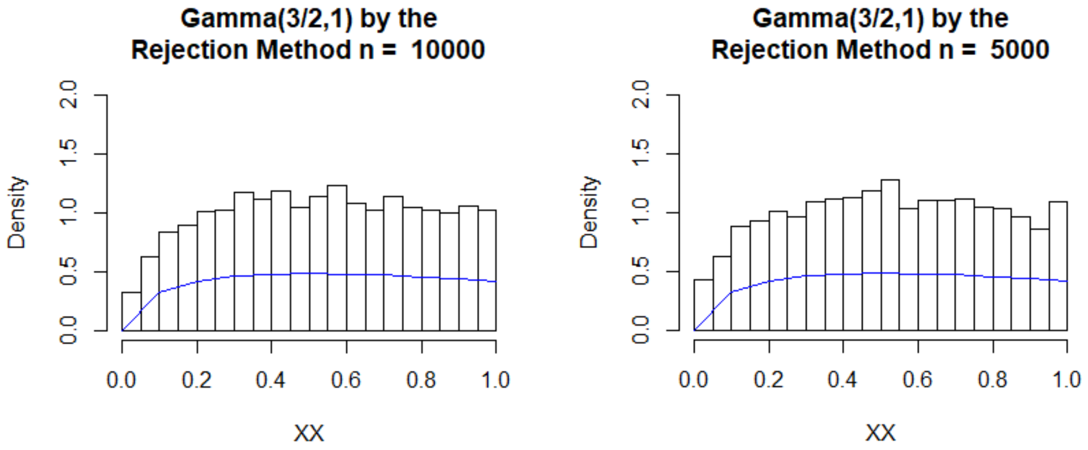
Analyzing the graphical representations of the Gamma distribution function for the five different sample sizes, I can say that for the sizes of 100, 500 and 1000 observations the fit is not very good, although for the samples of size 500 and 1000 observations it can be seen some similarities for the tendency of the two representations, but I can't say for sure that the samples can come from a Gamma (3/2, 1) distribution.







For the bigger sample sizes, with size of 5000 and 10 000 observations it is possible to say that the samples can come from a population with Gamma (3/2, 1) distribution because the tendency of the values is almost the same in the sample and in the line of the Gamma distribution.

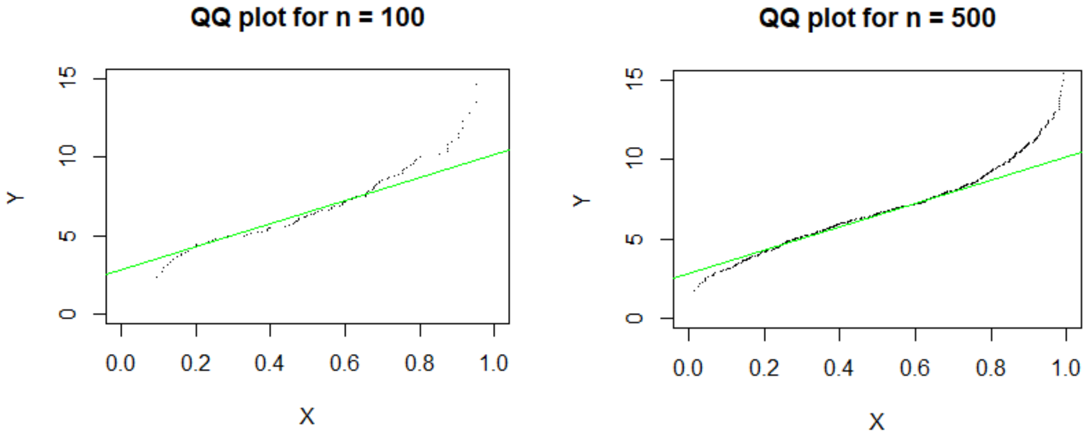


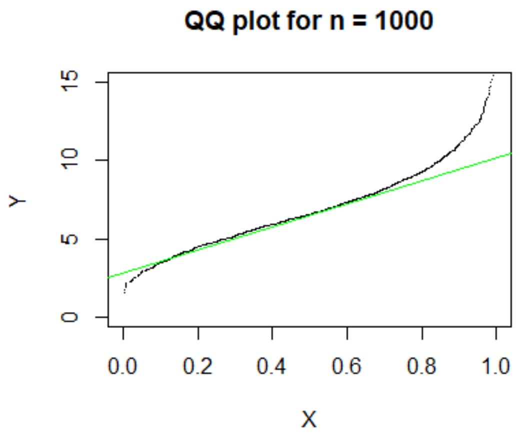
To have a visual confirmation from the quality of the adjustment of the model to the data I have created a QQ plot.

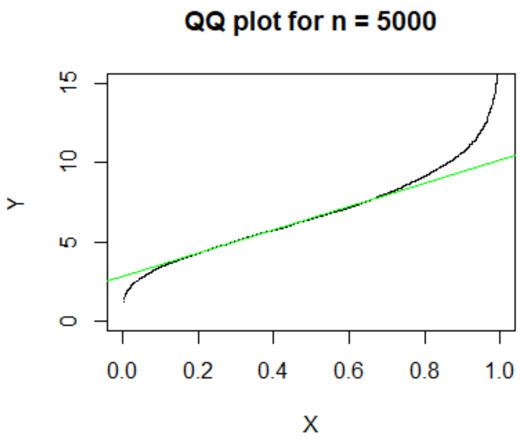
To do this I have the *quantiles.pop* function with the number of quantiles equal to *nvals*, then I created the matrix *qq*, with the sorted values accepted in the function *reject.gamma*, and the quantiles.

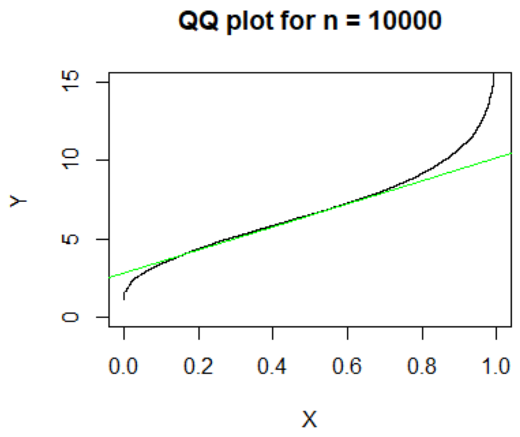
After that, I created the QQ plot that shows some points that fall along a line in the middle of the graphic but curve a bit on the left and more on the right side.

This means that the largest and smallest values are not as extreme as would be expected, so I can assume that the distribution is skewed to the right and have a good adjustment on the middle points that follow the line represented, the adjustment is better on the bigger sample sizes than on the smaller ones, as expected.









On the second part of the work, I had to compute the probability,

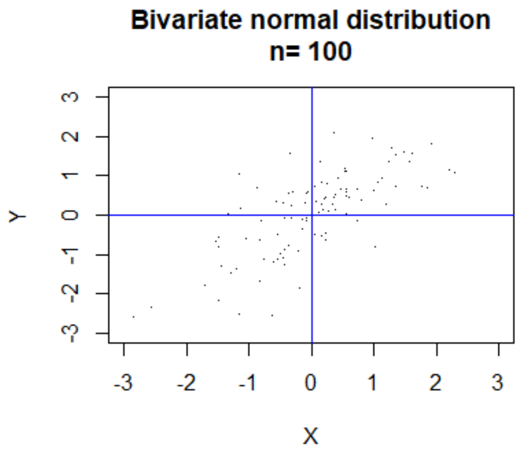
$$\theta = P(X^3 + 2Y^2 > 3)$$

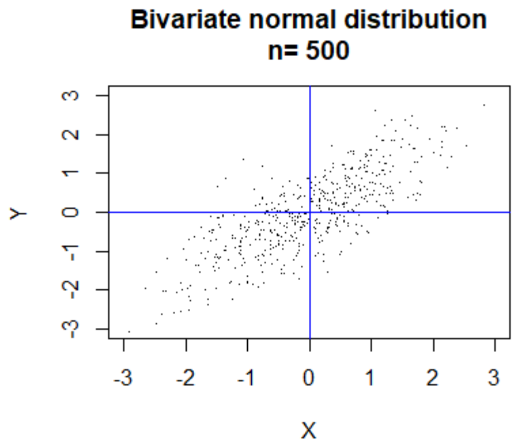
Of a random variable with bivariate normal distribution $N(\mu, \sum)$ with $\mu = (0,0)$ and $\sum = \begin{matrix} 1 & 0.75 \\ 0.75 & 1 \end{matrix}$, where $\rho = cov(X, Y) = 0.75$.

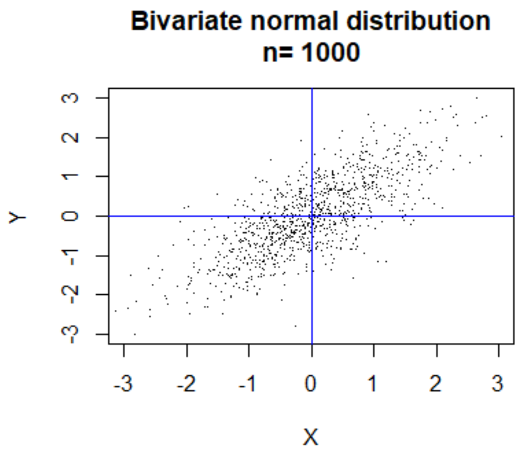
For this, I used the Monte Carlo method that is the process of generating independent random draws from a specified probabilistic model.

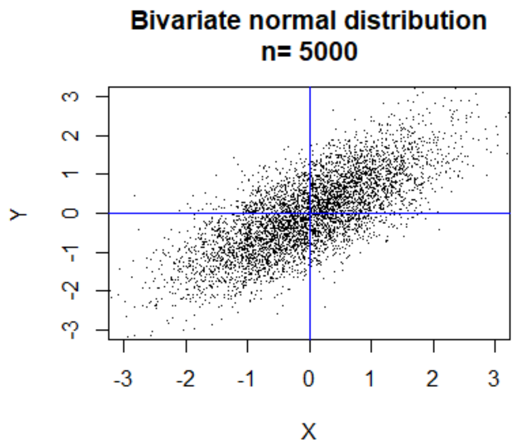
I started with a simulation model and then I computed that model several times with a randomly changing parameter and analyzed the results systematically.

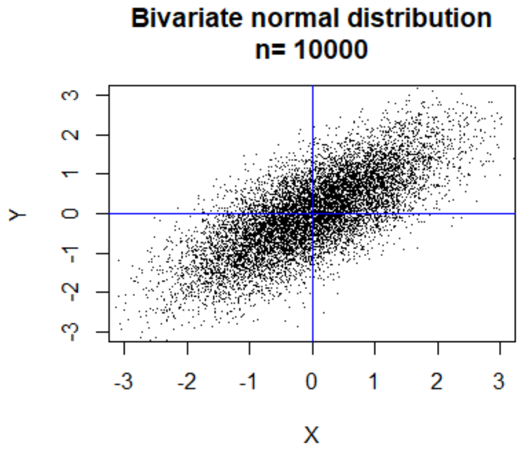
With *polar.normal* I simulated points of the bivariate normal distribution, then with *rbinorm.n* I computed a matrix of a sample of bivariate normal distribution with the expected values vector and the variance matrix given and, in the end, I plotted the sample on a two axis graphic.











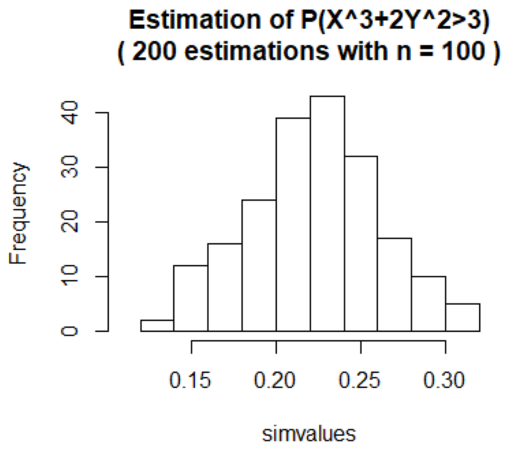
As seen above, because $\rho = 0.75$ the bell-shaped curve becomes more flattened on the 45 degree line than the one on the bivariate normal distribution with $\rho = 0$, for example, where we can see a perfectly bell-shaped curve, both of them in the three dimension curve but on the two dimension curve the flatness can be seen too.

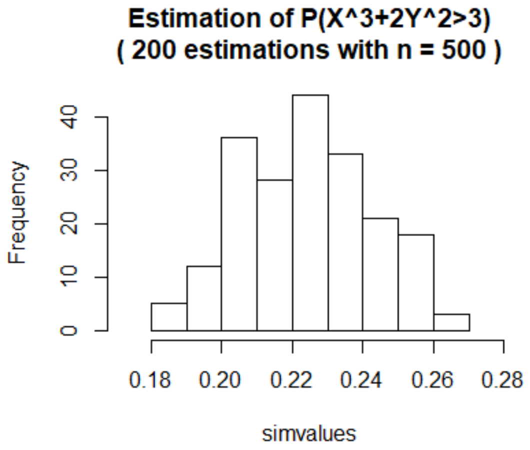
To compute the probability I generated the values of the bivariate normal distribution with $\rho = 0.75$ on the function *gera.binorm* that uses the *polar.normal* function and then it returns the probability value for the distribution of the sample generated

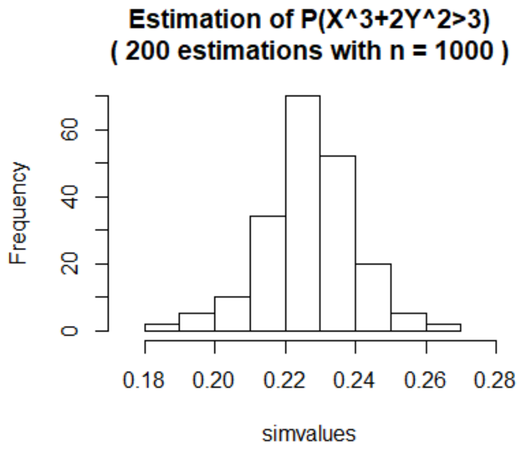
Then to return the probability of the expression being tested I used the function *mean.estimated.*

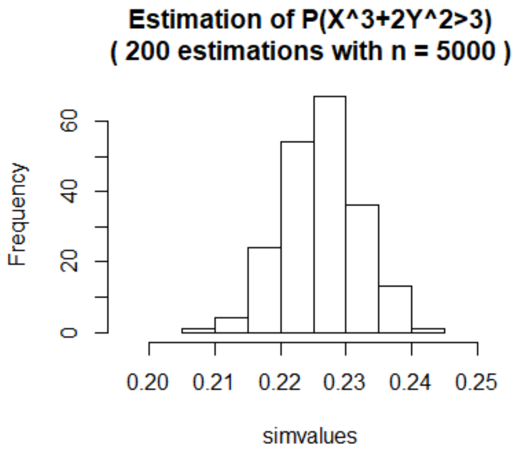
After that, I computed estimations using the Monte Carlo method where the samples of a probability with bivariate normal distribution is computed several times as stated on the variable *nreps*.

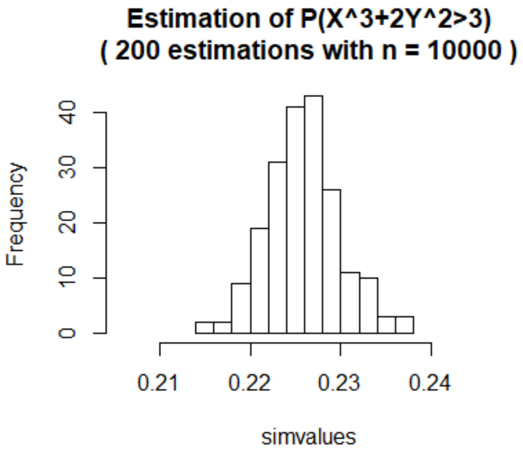
At the end it is created a histogram, where can be seen the probabilities of the values simulated and the frequencies of them, with different sized samples that were computed several times.



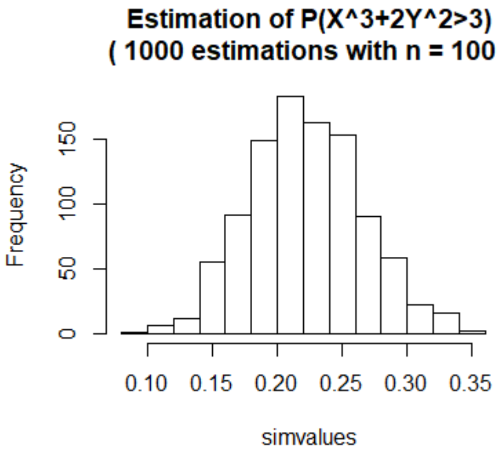


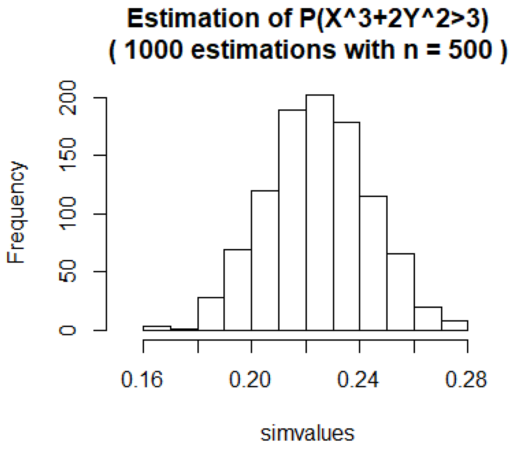


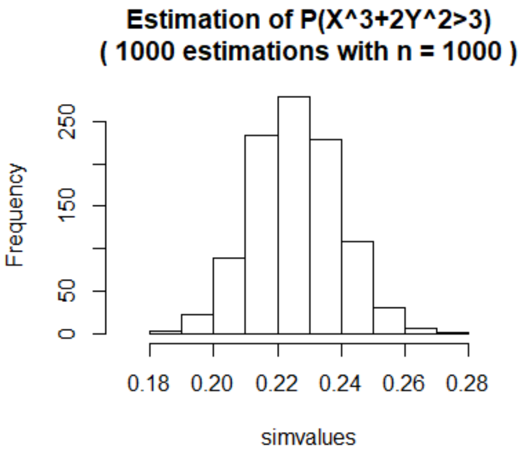


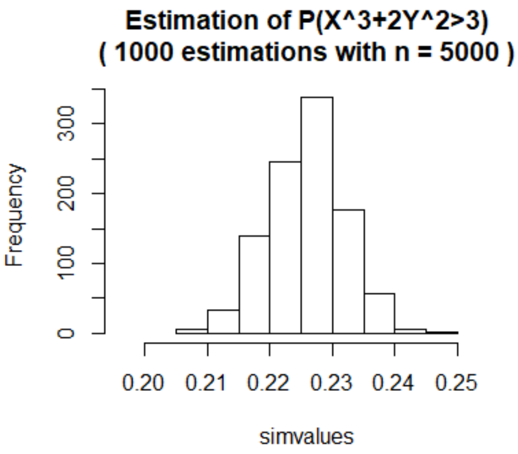


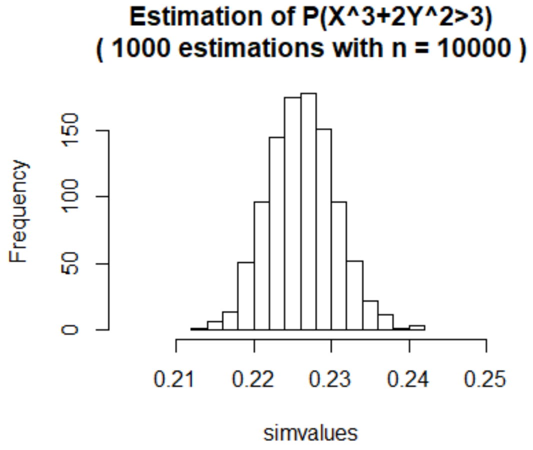
Seeing the graphics above I can confirm that the $P(X^3 + 2Y^2 > 3)$ is about 22%. Moreover, even with the number of estimations being different, for example 1000 instead of 200, the probability remains between 22% and 23 %.











As seen above the rejection sampling method is a basic, but good and very useful technique to generate observations from arbitrary distributions, from where it can be hard to take observations.

I understood that the Monte Carlo method is a simple but very powerful technique used to model the probability of different outcomes in a process that cannot be easily predicted due to the intervention of random variables.

I can conclude with the assessment that simulation modeling is able to solve real world problems efficiently, especially when conducting experiments on a real system is impossible or very hard, and provides important methods of analysis that can be easily verified and understood, providing valuable solutions and giving a very good insight into complex systems.