Simulation Report – Group 6

Carlos Pedro Cabral de Sousa Pinto

Applied Statistics

Masters in Modeling, Data Analysis and Decision Support Systems

University of Porto, Faculty of Economics, Portugal

2021 – 11 – 26

1 Introduction

This report was made in the context of the simulation topic from the unit of Applied statistics belonging to the Master Data Analysis from the Faculty of Economics of University of Porto. The goal of this report is to evaluate the results from multiple simulation methods and the impact of the number of samples on the models.

On the first exercise, with the Monte Carlo method, the goal is to find the approximated value of an integral with different sized samples. On the second exercise also with different sized samples the goal is to generate samples of a normal distribution using the Polar method and analyzing the quality of adjusting these results to a QQ-plot graphic.

All the exercises will be developed and analyzed with the software RStudio, with the R language.

2 Problem 1

The first simulation problem on this report is to find an approximated value, using the Monte Carlo method and considering different sized samples for:

$$\int_0^1 \int_0^1 yx^2 + 3\ln(xy+1)\,dydx$$

2.1 Monte Carlo method

Monte Carlo method is a mathematical technique which goal is to obtain a numerical result by using repeated random sampling. It is used mainly for three types of problems: numerical integration, optimization and generating draws from a probability distribution.

This method follows the following formula:

$$\theta = \int g(x)f(x)\,dx$$

So, $\theta = E\,[g(X)]$ where X has a probability function f(x). The algorithm to implement this follows these steps:

1. Generate X1, …, Xn with density f(x)
2. Compute $\hat{\theta} = \frac{1}{n}\ \sum_{i=1}^{n} g(Xi)$
3. Set $\hat{\theta}$ for an approximation of θ

In this problem, a multivariate case there are some adjustments to be made in the algorithm where:

$$\theta = \int_0^1 \dots \int_0^1 g(x1, \dots, xn) dx1 \dots dxn$$

So, $\theta = E\ [g(U1, \dots, Un)]$ where U1, …, Un ~ U [0;1] and independent.

2.2 Monte Carlo Script in R for Problem1

Following the steps described in the previous point the first step to take will be to generate two independent samples following an uniform distribution (X and Y) and after that estimate the expected value of g(X, Y). The R script used to solve this problem is visible in the picture below.

```
mean.estimated <-function(nvalues){
  x <- runif(nvalues)
  y<-runif(nvalues)
  theta.hat <- mean( (y*(x^2)) + (3*log(x*y+1)) )

}

monte.carlo <- function(nreps,nvalues) {
  estimativas <- NULL
  for (i in 1:nreps){
    estimativas[i] <- mean.estimated(nvalues)
  }
  return(estimativas)
}

montecarlo.runSeveralSamples<-function(nreps, nvalues){
  for (size in nvalues){
    simvalues <- monte.carlo(nreps ,size)
    hist(simvalues,xlim = c(0.2,1.5), main=paste(nreps, "estimations with n=",size))

  }
}

sampleSizes= c(10, 50, 100, 500, 1000, 5000,10000)
montecarlo.runSeveralSamples(200,sampleSizes)
```

Fig. 1. RScript with the functions that allow the estimation of the value for a certain number of values and repetitions

2.3 Results of applying Monte Carlo method

In order to solve this problem, we need to input two different values, the number of repetitions the monte carlo method will be performed and the size of the sample. Due to performance issues and since that is already possible to get good results, the number of repetitions will be 200 and the sample size will be variable in order to understand the impact on estimating the value of the integral.

The current problem was solved with 7 different simulations, considering different sample sizes (10, 50, 100, 500, 1000, 5000, 10000) and the values for the mean estimated value and standard deviation can be seen in the following table:

Table 1. Mean and standard deviation for different sample sizes with Monte Carlo method

Reps	n=	Average	Standard Deviation
200	10	0,8123	0,2211
	50	0,7960	0,0981
	100	0,7852	0,0710
	500	0,7891	0,0286
	1000	0,7924	0,0214
	5000	0,7925	0,0097
	10000	0,7929	0,0071

From the table above we can see that with small sample sizes we already get a good estimation for the estimated value and depending on the number of decimal cases needed for the final values, different size samples would be needed. Also, we can see that the standard deviation as a bigger impact when increase the size sample than the mean has which means that every time, we run the monte carlo method the value we will get will be closer to the mean.

In order to have a better comprehension of the results we get every time we run the method, we plot the results into a histogram for the different size samples and use the same range of values in the x axis to have an easier comparison between each one of them.

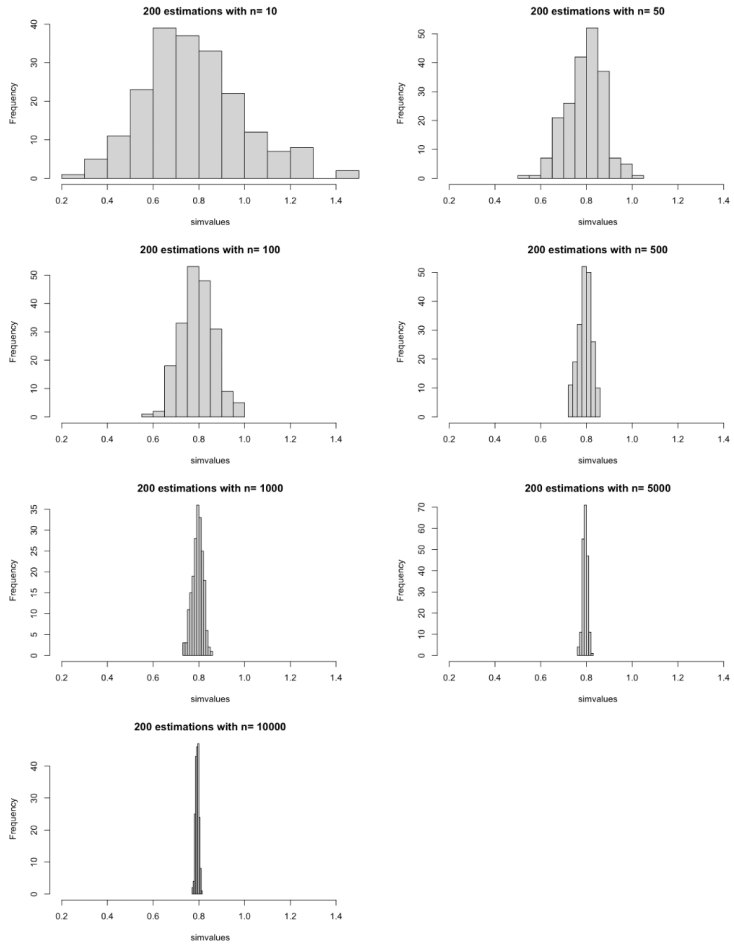

Fig. 2: Histogram with the values from monte carlo method for different size samples

After plotting the data, we get a better idea of the impact of increasing the number of samples. With the increase of the number of samples till n=500 we observe a big modification in the histograms, with the increase of n we can see that all the values are much closer to the mean reducing the minimum and maximum values having a less disperse data. Depending on the precision needed for the value we may not need to use samples bigger than 500 since the increase of this value increases the time it takes to perform the simulation.

3 Problem 2

The second problem on this report consists in generating samples of different sizes from a normal distribution with mean = 5 and variance = 4 using the Polar method. After this, the quality of this samples is analyzed through the QQ-plot graphic.

3.1 Polar Method

The Polar method is a number sampling method for generating random values that follows a Normal distribution N (0,1). It is based on the transformation of the Box-Muller method and is a superior. The algorithm to apply this method follows these steps:

1. Generate V1, V2 ~ U [-1; 1]
2. If $W = V_1^2 + V_2^2 > 1$, then go to step 1
3. Else, $C = \sqrt{-2W^{-1}\ln(W)}$
4. $X = C\,V_1$ and $Y = C\,V_2$

X and Y are two independent random variables with normal distribution.

In order to get a normal distribution with mean μ and variance σ^2, the following logic is applied:

$$Z \sim N(0,1) \rightarrow X = \mu + \sigma Z \sim N(\mu, \sigma^2)$$

3.2 Polar Method Script in R for Problem 2

Once again to solve the simulation problem the software used was R and the steps to solve the problem were the ones described in the previous point and to a better comprehension the variables used in R have the same name as described in the point 3.1.

So, the first step was to generate de values of V1 and V2 following a uniform distribution between -1 and 1, then checking the result of W, if the value is higher than 1, repeat the first step otherwise calculate C, the next step was to calculate X(x[nx+1]) and Y(x[nx+2]). Finally considering that the mean in this problem is 5 and the variance is 4 we calculate $X = 5 + 2 * Z$

The code used can be seen in the Figure 3 below:

```r
polar.normal <- function(nvalues) {
  x <- NULL
  nx <- 0

  while (nx < nvalues) {

    U1 <- runif(1)
    U2 <- runif(1)

    V1 <- U1*2 - 1
    V2 <- U2*2 -1

    W = V1^2 + V2^2

    if (W <= 1) {

      C <- sqrt(-2*log(W)/W)
      x[nx+1] <- C*V1
      x[nx+2] <- C*V2
      nx <- nx+2
    }
  }
  return(x)
}

Z <- polar.normal(1000)
X <- 5 + 2*Z
```

Fig. 3: RScript to generate random samples that follow a normal distribution with mean=5 and variance=4

3.3 Results of applying Polar method

After adapting the polar method to the problem, we choose different sized samples in order to understand not only the behavior of this algorithm but also the impact that increasing or decreasing the number of samples have in the final output. The size of the samples that were chosen are the following: 50, 100, 500, 1000, 5000, 10000, 50000, 100000 and the results in terms of mean values can be found in table 2.

Table 2. Mean for different sample sizes with Polar method

n =	Mean	Variance
50	4,0309	5,7484
100	5,2286	3,6532
500	4,8793	4,3266
1000	4,9513	4,1722
5000	4,9689	4,0585
10000	5,0256	4,0395
50000	4,9950	3,9985
100000	4,9983	3,9961

When analyzing the results of the simulation we can see that when we increase the values of the samples the mean and the variance are closer to the ones established on the problem for the normal distribution.

What we can conclude by only looking to the mean and the variance is that with the increase of the number of samples this simulation method gets a better approximation to the normal distribution with the specified parameters.

Another way to analyze if the generated samples follow a normal distribution is by plotting the values with a histogram and drawing a line that that represents the values of a normal distribution with mean = 5 and variance = 4. When we do this for the multiple sized samples, we see different behaviors that we can separate between small and big sized samples.

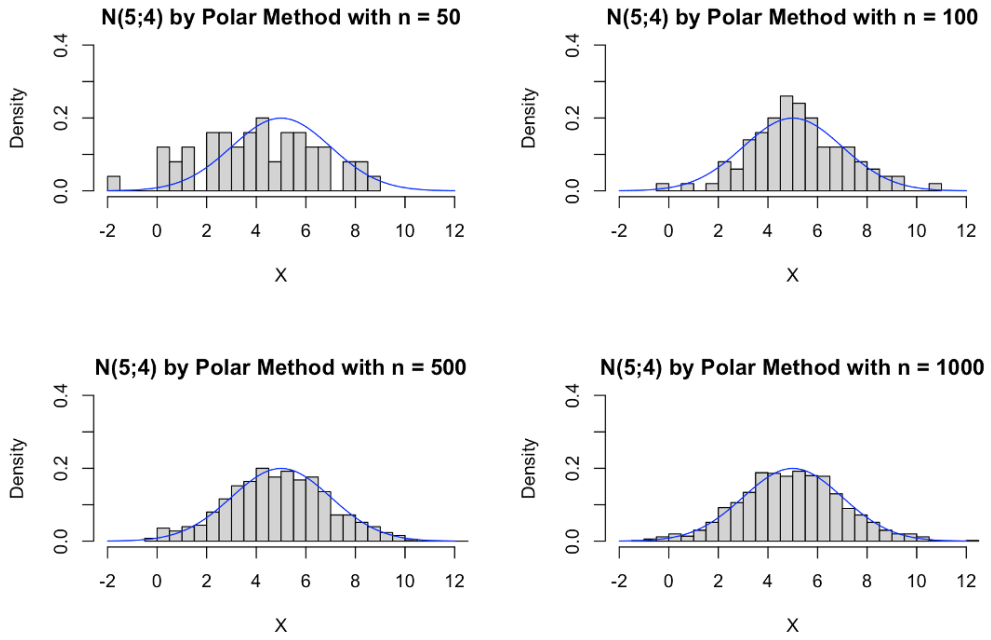

Fig. 4: Distribution of the simulation for n=50, 100, 500 and 1000

When plotting the results of the simulation for small sized samples is not clear that it follows a normal distribution but is possible to observe that with the increase of n the results get closer to the normal distribution with the frequency of the values closer to the mean of the normal distribution increasing.

Since the increase of the number of samples showed a positive impact on the results of the simulation, we should analyze also the histograms for the simulations with big sample sizes

With the histograms for the big sized samples, we can now see that the result from the simulation follows a normal distribution, and we can understand the power of this test to generate sample of values that follow normal distribution as it possible to see in the Figure 5 where the bars of the histogram go along with the blue line that, as previously said, represents a normal distribution.

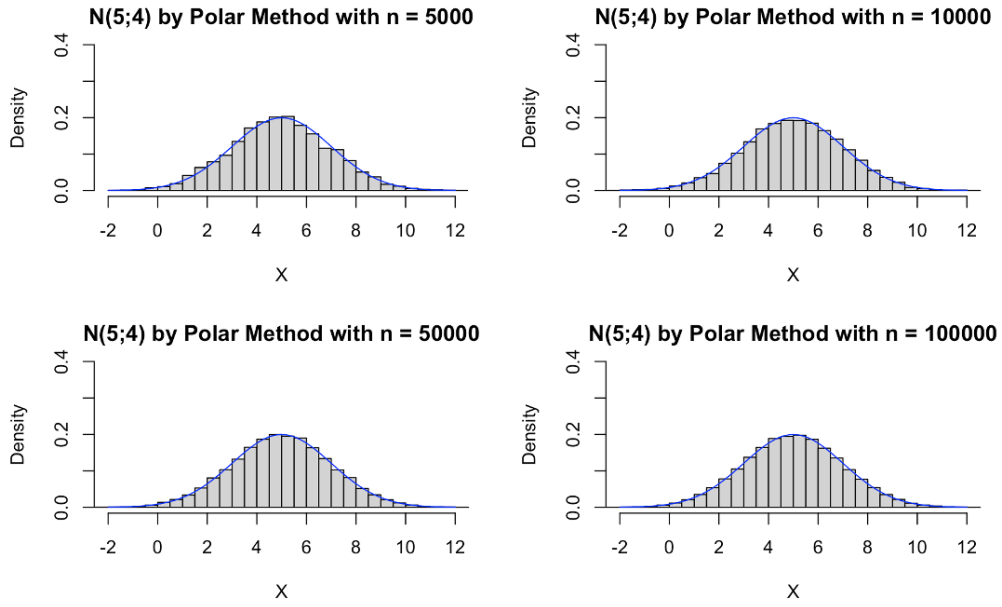

Fig. 5: Distribution of the simulation for n=5000, 10000, 50000 and 100000

3.4 QQ-Plot

The QQ-plot is an exploratory graphic that allows a fast visual understanding of the quality of adjustment of a model to the data, by building the quantiles of the data and comparing it to the quantiles of the distribution. When plotting the graphic if the points are distributed closer to the line, then the distribution is adequate adjusted, if the points were distant from the straight line the model is not adequate.

For this exercise we will test the quality adjustment of the Polar method to a normal distribution with N(5,4), using the same sample sizes as described in the previous point (50, 100, 500, 1000, 5000, 10000, 50000, 100000).

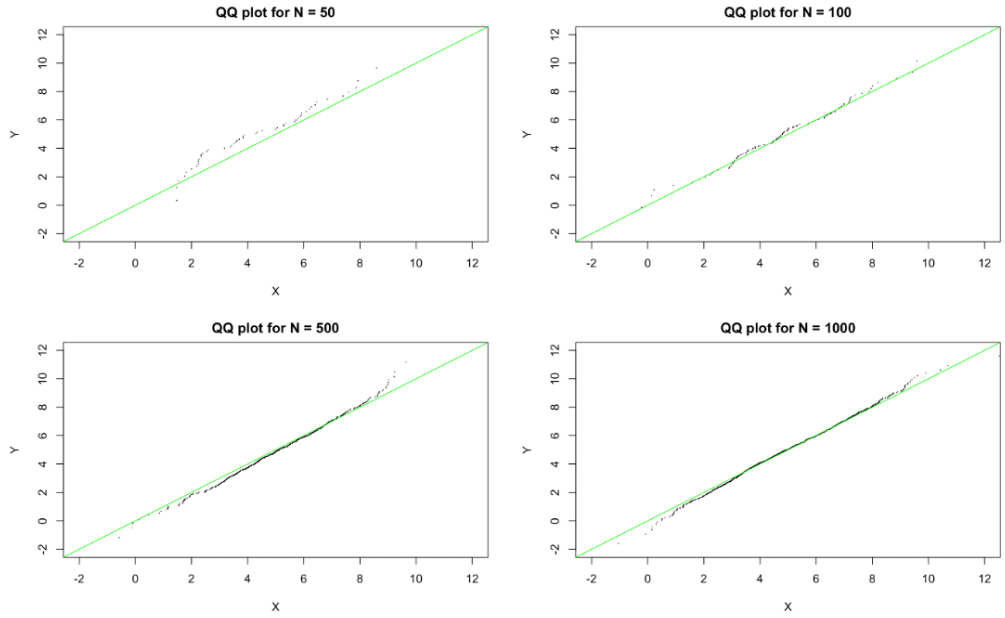

Fig. 6: QQ-plot of the simulation for n = 50, 100, 500, 1000

From the graphics of the figure 6 we can visualize what was referred in the point 3.3, that the simulations with small number of samples don' have a good approximation to the normal distribution, and that increasing n improves the quality of adjustment of the Polar method. We can also see that with the increase of n, the zones with more density, closer to the mean are the ones to get a better approximation.

For the big size sample simulations, we can see that results from the Polar method have a good adjustment to the normal distribution. In these cases, the zone with more density have already an excellent approximation to the line, and the increase in the number of samples move the points from the lower density zones closer to the line as showed in Figure7.

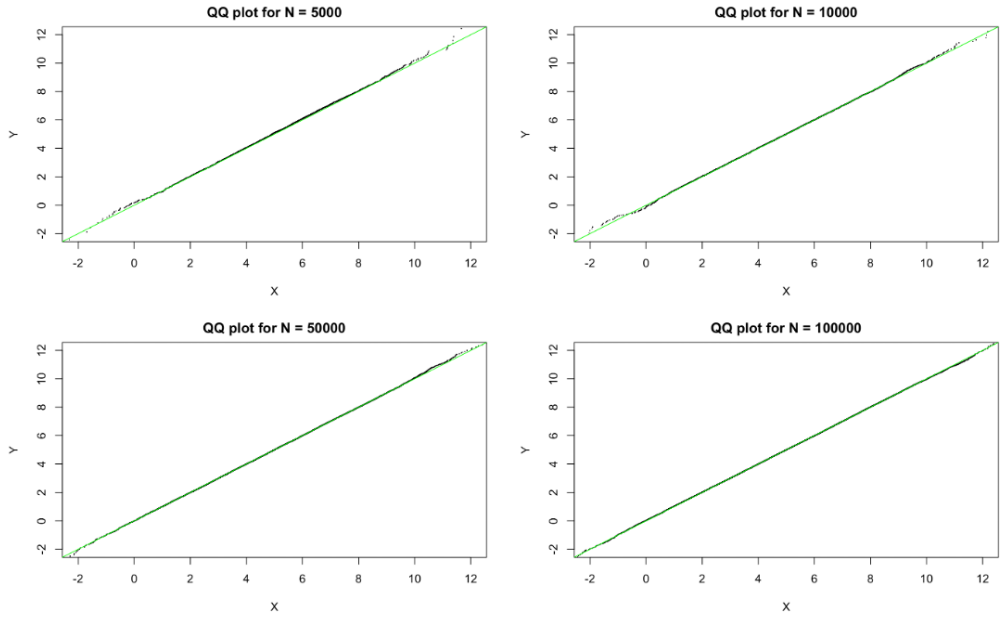

Fig. 7: QQ-plot of the simulation for n = 5000, 10000, 50000, 100000

4 Conclusions

This report has proved the importance of simulation in solving complex statistical problems, with the increase of computational capacity simulation is getting more and more useful and powerful since it allows a faster solution to solve problems.

The method used in this report was the Monte Carlo method and the Polar method. The first proved to be a good method estimate values, in this case estimate the values from a double integral with the variables following a uniform distribution. This method also showed very good results with small sample sizes. The second method, the Polar method, proved to generate distributions that have a good approximation to a normal distribution for big sample sizes (n > 5000).

In both cases the increase in the number of samples improve the results obtained in the experiences which indicates that with the evolution of technology and improvements in processing data, simulation methods will be more relevant and used more frequently to solve complex problems.