Distributions of Regression Model Parameters using Monte Carlo Simulations

Paritosh Tiwari

***Abstract*—Estimation of parameters in regression models can often be difficult or near impossible via analytical methods, sometimes owing to the complex differnetial equations and related functions, or due to lack of appropriatedata. In this paper, we explore the estimation of parameters via Monte Carlo simulations and compare the accuracy of the estimation under various conditions of the simulations. Here we cover the simple linear regression model, a dynamic regression model and a non-linear least squares regression model.**

***Index Terms*—Estimators, Monte Carlo Simulation, Linear Regression, Sampling Distribution.**

I. Introduction

In order to grasp the concept of a sampling distribution, we first need to understand what is meant by a "statistic".

A statistic is just some function of the random data in our sample. Each individual sample value is a statistic. So is the the sum of the sample values, the difference between any two sample values, the variance of the sample numbers, etc. Because a statistic is a function of the random sample values, it itself is a random variable. The sample average, the sample standard deviation, etc., are all random variables, and as such they are all described by a probability distribution. The special name "sampling distribution" is used when we're talking about the probability disrtibution of a statistic, rather than just any general random variable.

Estimators and test statistics are (almost always) constructed form random sample data. For example, when we use the sample average as an estimator of the population mean, or when we apply the formula for the least squares regression estimator which uses the data for the (random) dependent variable. Estimators are statistics and their random behaviour is described by their sampling distributions.

Monte Carlo (MC) methods, or Monte Carlo experiments, are a broad class of computational algorithms that rely on repeated random sampling to obtain numerical results. The underlying concept is to use randomness to solve problems that might be deterministic in principle. Monte Carlo methods are mainly used in three problem classes: optimization, numerical integration, and generating draws from a probability distribution.

Monte Carlo methods vary, but tend to follow a particular pattern:

1. Define a domain of possible inputs
2. Generate inputs randomly from a probability distribution over the domain
3. Perform a deterministic computation on the inputs
4. Aggregate the results

Paritosh Tiwari, is with the Computer Science Department, International Institute of Information Technology, Naya Raipur, Chhattisgarh 493661, India (e-mail: paritosh19300@iiitnr.edu.in).

When we conduct an MC simulation experiment what we're trying to do is to replicate (simulate) a sampling distribution. In principle we need to use an infinite number of replications. Obviously, this would be infeasible . By using only a finite number of replications (1,000 or even 100,000) we incur some simulation error, which is inevitable. If that error is quite small (in some sense) then there's probably not too much of its effect on the desired result.

Let's relate our introduction so far to the specific problem of determining the bias, and efficiency, of some estimator. In the case of its bias we want to compare the expected value of the estimator with the true value of the parameter that we're trying to estimate. In this specific case, we need to compare the mean of the sampling distribution of the estimator with the true parameter value.

Suppose that S is an estimator (say θ^*) of some parameter (θ), and we construct θ^* from some random sample data (y) that are drawn from a population that depends on θ, then what we're trying to do is get an accurate approximation of $Bias[\theta^*] = E[\theta^*] - \theta = \int \theta^*(y)p(y|\theta)dy - \theta$.

The steps involved in conducting the most basic MC experiment would be something like the following:

1. Assign a realistic, or interesting, value to θ. (The results will be conditioned on this choice, so the whole exercise may need to be repeated lots of times, for different values of theta, because in reality θ is unknown.)
2. Generate a random sample of fixed size (say, n) from $p(y \mid \theta)$.
3. Compute the value of θ^*, based on this particular y vector, and store this value.
4. Repeat steps 2 and 3 several times. Suppose the number of replications that we decide to use is N.
5. We will now have N different values of θ^*, each based on a different (independent) sample of size n.

When we look at the empirical distribution of these N values in a table, or in a histogram, what we're looking at is an approximation to the sampling distribution of the statistic, θ^* that we're interested in. It's an approximation, because N is finite. And of course, this sampling distribution will (in general) depend on our choices of values for θ and for n. We can now vary these values to get a more complete picture of what's going on.

To approximate $E[\theta^*]$ we can just use the arithmetic average of N values of θ^* that we've generated. This is because the (strong) Law of Large Numbers ensures that the sample average converges in probability to the population expectation as N becomes infinitely large, given that we have independent sampling.

The difference between this empirical average and the value that we assigned to θ itself will give us an approximation to

the bias of θ*. Remember, the value of this bias that we come up with may very well change as we change n or θ.

II. Related Work

In their paper titled Goodness of fit in restricted measurement error models, C. -L. Cheng et al. explored the goodness of fit statistics for regression models using Monte Carlo simulations. However, their work was limited to measurement error models. They studied the differences between classical linear regression and linear regression with measurement errors and tried to determine how the goodness of fit procedure could be adjusted to incorporate measurement errors.

In another paper titled Learning models from data with measurement error: tackling underreporting, Roy Adams et al. explored the effects of measurement error is logistic regression models and how to compensate for them.

III. Methodology

A. Simple Linear Regression

Consider a basic linear regression model of the following form:

$$y = a + b\,x + e \tag{1}$$

The observed values for the regressor, x will be non-random. They'll be what we can call "fixed in repeated samples". The coefficients of the model a and b will be fixed parameters. In practice, we won't know their true values and indeed, one purpose of the model is to estimate these values. The random error term, e is the most important part of the model as without it, the model would be purely deterministic, and there would be no statistical problem for us to solve.
The random values of e are pair-wise uncorrelated, and come from a normal distribution with a mean of zero, and a variance of σ^2. This variance is also an unknown constant parameter that's positive and finite in value. Being uncorrelated, and also normally distributed, the values of the random error term are actually statistically independent. The values taken by y will be realized values of a random process, because of the presence of the e random variable. In reality, the values of e won't be observed. They're coming from the underlying population.

The steps that we need to go through are as follows:

1. Assign values to the parameters, a, b and σ.
2. Generate n independent random e values from a N[0, σ^2] distribution. Then use these values, the values for the parameters, and a set of n values for x to generate a random sample of y values, using equation (1).
3. Compute the Ordinary Least Squares (OLS) estimates of the regression parameters, based on this particular y vector, and store these estimates.
4. Repeat steps 2 and 3 several times. Suppose the number of replications that we decide to use is N. In each case the same parameter values and sample values for x will be used. However, because different e values will be used each time, the sample of y values will also differ at each replication.
5. We'll now have N different estimates of the parameters, each based on a different (independent) sample of size n.

The empirical distributions of these N estimates are an approximation to the sampling distribution of the corresponding OLS estimator of the parameters. The values of the regression coefficients are set to 1 and 3 respectively and the variance of the error term is set to 1.

B. Dynamic Model (Time Series, Stochastic Process)

$$y_t = a + b\,y_{t-1} + e \tag{2}$$

The assumptions and conditions from the previous simple linear regression model apply here. Also, the procedure remains the same.

However, in this case, the (non-constant) regressor in the model is no longer "fixed in repeated samples". Each time that we generate a new sample of y_t values, we'll also have a new sample of y_{t-1} values. So, in this case we have a regressor that is non-random. But it's not correlated with the error term, because we're explicitly forcing the latter to take values that are independent over time i.e., serially independent.

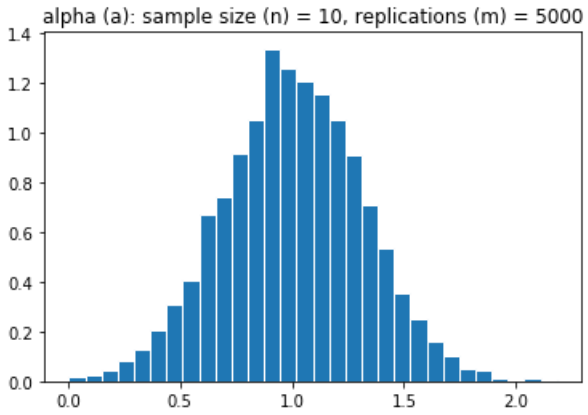
C. Non-Linear Least Squares Estimator

$$y = a + b\,x^p + e \tag{3}$$

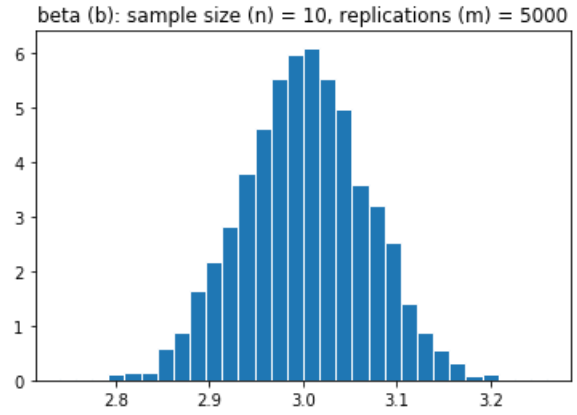
Once again, the values of x will be "fixed in repeated samples". Rest of the conditions and assumptions remain the same. In general, the non-linear least squares (NLLS) estimator is biased, although weakly consistent under standard conditions.

IV. Results

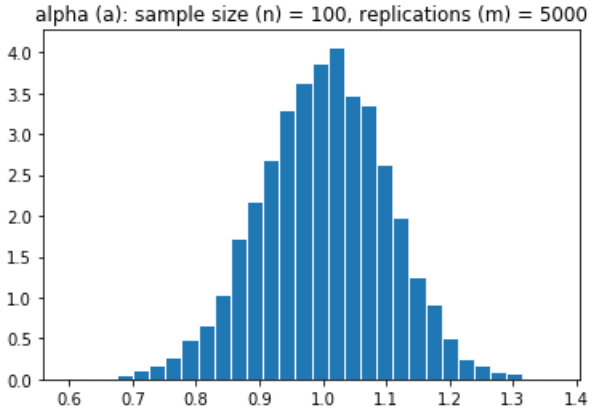
A. Simple Linear Regression



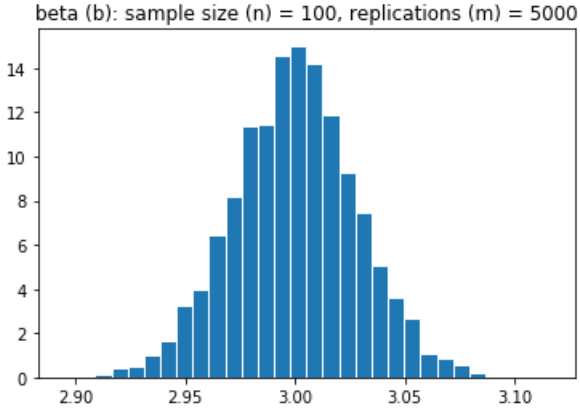
Median	1.003580
Mean	1.004324
Maximum	2.182226
Minimum	0.003571
Standard Deviation	0.313316



Median	3.001627
Mean	3.001454
Maximum	3.260459
Minimum	2.740233
Standard Deviation	0.066953



Median	1.003424
Mean	1.001097
Maximum	1.366985



Minimum	0.597768
Standard Deviation	0.101042

Median	3.000321
Mean	2.999882
Maximum	3.116210
Minimum	2.894256
Standard Deviation	0.028061

There are several things that are apparent from these two figures and the associated summary statistics.

First, regardless of the sample size, the mean value of the sampling distribution for a is extremely close to 1.0 and for b is extremely close to 3. That's the true value for a and b that we assigned previously. In fact, any discrepancy is simply due to sampling error, we've used only m = 5,000 replications, not an infinity of them.

Second, the standard deviation of the sampling distribution for a falls from 0.313316 to 0.101042 and for b falls from 0.066953 to 0.028061 as the sample size increases. If we made n *very large indeed*, this standard deviation would approach zero.

So, among the things that we've been able to demonstrate (not prove) by conducting this MC experiment are the following:

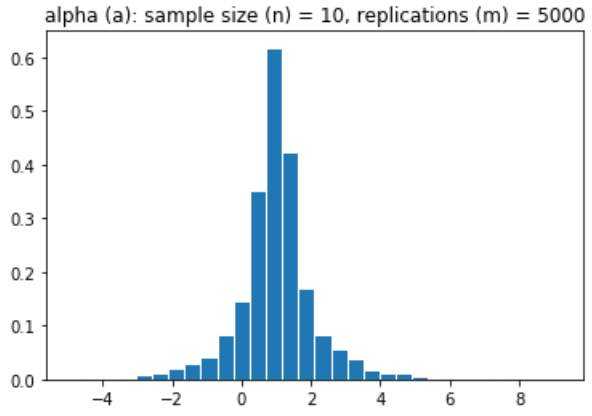
At least for the set of parameter values that we've considered, OLS seems to be an unbiased estimator of the regression coefficients under the conditions adopted in the MC experiment.

At least for the set of parameter values that we've considered, OLS seems to be a mean-square consistent (and hence weakly consistent) estimator of the regression coefficients under the conditions adopted in the MC experiment.

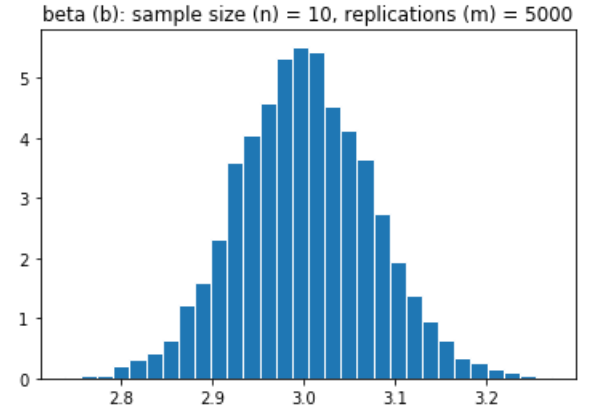
At least for the set of parameter values that we've considered, the OLS coefficient estimator seems have a sampling distribution that is Normal, even when the sample size, n, is very small.

If we were to repeat this experiment with different true values for the parameters, we'd keep coming to the same conclusion. This might lead us to suspect (correctly in this case) that these properties of the OLS estimator apply for any values of the parameters.

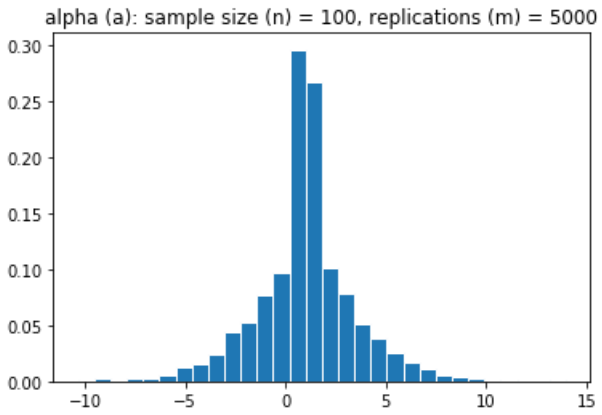
B. *Dynamic Model*



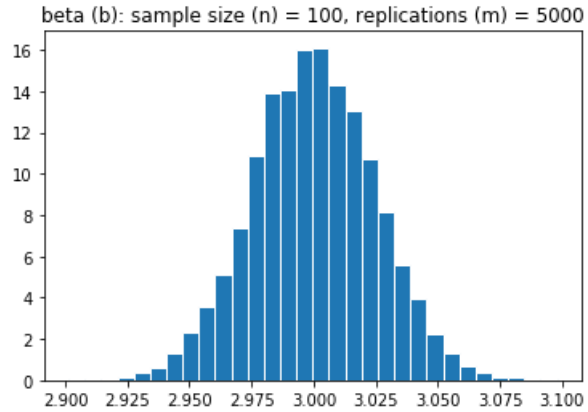
Median	0.994219
Mean	0.996284
Maximum	9.135775
Minimum	-4.920227
Standard Deviation	1.122909



Median	2.999267
Mean	3.000678
Maximum	3.271745
Minimum	2.738654
Standard Deviation	0.074761



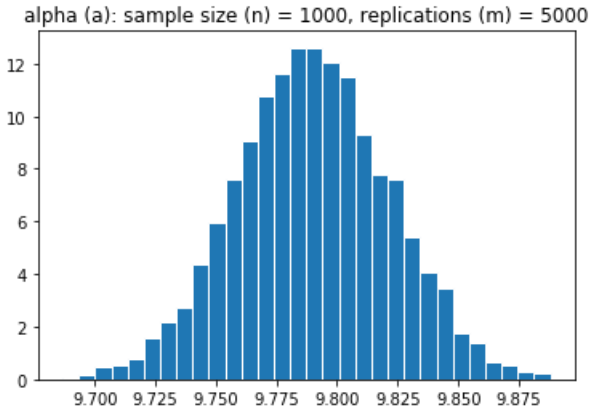
Median	0.991122
Mean	0.981777
Maximum	13.975604
Minimum	-10.345574
Standard Deviation	2.466551

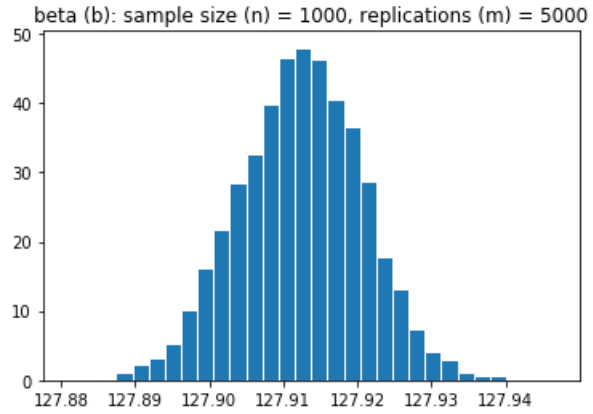


Median	2.999938
Mean	3.000017
Maximum	3.097902
Minimum	2.901379
Standard Deviation	0.024800

We see here that the alpha (a) distribution does not exhibit a normal distribution. It has a sharp rise in mean frequency with a high standard deviation. Also, the maximum and minimum values are extremely far apart form the mean. Increasing the sample size has no effect on the distribution. This suggests an error in our model assumptions and conditions, or in our approach to evaluate the data generated by the model in the first place. This could be rectified by making adjustments to our model and the corresponding analytical formulae, and also by studying the distribution of our data.

C. *Non-linear Least Squares Estimator*





As is glaringly obvious, our approach fails miserably when it comes to non-linear models. This suggests a study in the distribution of generated data and a completely new approach to the analytical formulae of estimators used to set up the MC experiment.

V. Conclusion

Here we have successfully demonstrated and/or surveyed various Monte Carlo simulations for different regression models. Although the variance in the simulations can be overcome with increased number of replications, we can explore other means of reducing variance like, antithetic method, control variates, moment matching method, stratified sampling and latin hypercube sampling. In general, we see that discrepancies can often be overcome by increasing the number replications, which is the same as drowning out the noise. Further work can be carried out on other regression models like, restricted models or measurement error models, and also on other statistics that measure the accuracy and robustness of a regression model.

References

[1] C. L. Cheng, Shalabh, G. Garg. Goodness of fit in restricted measurement error models. *Journal of Multivariate Analysis 145*, 2016 ,101-116.

[2] R. Adams, Y. Ji, X. Wang, S. Saria. Learning Models from Data with Measurement Error: Tackling Underreporting. *arXiv:1901.09060v1*, 2019.

[3] Econometrics Beat: Dave Gile's Blog. Monte Carlo Simulation Basics, I: Historical Notes

[4] Econometrics Beat: Dave Gile's Blog. Monte Carlo Simulation Basics, II: Estimator Properties

[5] Econometrics Beat: Dave Gile's Blog. Monte Carlo Simulation Basics, III: Regression Model Estimators

[6] Wikipedia. Simple Linear Regression