



Munich Personal RePEc Archive

# **How to Create a Monte Carlo Simulation Study using R: with Applications on Econometric Models**

Mohamed R. Abonazel

30. December 2015

Online at <https://mpra.ub.uni-muenchen.de/68708/>

MPRA Paper No. 68708, posted 9. January 2016 20:08 UTC



Annual Conference on  
Statistics, Computer Sciences  
and Operations Research 2015



## Workshop

# How to Create a Monte Carlo Simulation Study using R: with Applications on Econometric Models

**Dr. Mohamed Reda Abonazel**

Department of Applied Statistics and Econometrics

Institute of Statistical Studies and Research

Cairo University

*[mabonazel@hotmail.com](mailto:mabonazel@hotmail.com)*

2015

## Summary

In this workshop, we provide the **main steps** for making the Monte Carlo simulation study using R language. A Monte Carlo simulation is very common used in many statistical and econometric studies by many researchers. We will extend these researchers with **the basic information** about how to create their R-codes in an easy way. Moreover, this workshop provides **some empirical examples** in econometrics as applications. Finally, **the simple guide** for creating any simulation **R-code** has been produced.

## Contents of the workshop

1. Introduction to Monte Carlo Simulation.
2. The history of Monte Carlo methods.
3. The advantages of Monte Carlo methods.
4. The methodology of Monte Carlo methods in literatures.
5. The full steps to create a Monte Carlo simulation study (*the proposed technic*).
6. *The Application*: Multiple linear regression model with autocorrelation problem.
7. General notes on simulation using R.

# 1. Introduction to Monte Carlo Simulation

- **Gentle (2003)** defined the Monte Carlo methods, in general, are the experiments composed of random numbers to evaluate mathematical expressions
- To apply the Monte Carol method, the analyst constructs a mathematical model that simulates a real system.
- A large number of random sampling of the model is applied yielding a large number of random samples of output results from the model.
- For each sample, random data are generated on each input variable; computations are run through the model yielding random outcomes on each output variable.
- Since each input is random, the outcomes are random.

## 2. The history of Monte Carlo methods

- The Monte Carlo method proved to be successful and was an important instrument in the Manhattan Project. After the World War II, during the 1940s, the method was continually in use and became a prominent tool in the development of the **hydrogen bomb**.
- The **Rand Corporation** and the **U.S. Air Force** were two of the top organizations that were funding and circulating information on the use of the Monte Carlo method.
- Soon, applications started popping up in all sorts of situations in **business, engineering, science** and **finance**.

### 3. The advantages of Monte Carlo methods

We can summarize the public advantages (goals) of Monte Carlo methods in the following points:

- Make **inferences** when weak statistical theory exists for an estimator
- **Test null hypotheses** under a variety of conditions
- **Evaluate the quality** of an inference method
- **Evaluate the robustness** of parametric inference to assumption violations
- **Compare** estimator's properties

## 4. The methodology of Monte Carlo methods in literatures

Mooney (1997) presents **five steps** to make a Monte Carlo simulation study:

**Step1: Specify** the **pseudo-population** in symbolic terms in such a way that it can be used to generate samples by writing a code to generate data in a specific method.

**Step2: Sample** from the pseudo-population in **ways** that show the topic of interest

**Step3: Calculate**  $\theta$  in a pseudo-sample and **store** it in a vector



**Step4: Repeat** steps 2 and 3  $t$ -times where  $t$  is the number of trials

**Step5:** Construct a relative **frequency distribution** of resulting values which is a Monte Carlo estimate of the **sampling distribution** of under the *conditions specified* by the pseudo-population and the sampling procedures

❖ **For more details** about Monte Carlo methods, you can review the following references: Thomopoulos (2012), Gentle *et al.* (2012), and Robert and Casella (2013).

## 5. *The proposed technic*: The full steps to create a Monte Carlo simulation study

- In this section, we proved the completed algorithm of Monte Carlo simulation study.
- We explain our algorithm through an application in regression framework, especially; we will use the Monte Carlo technic to prove that OLS estimators of GLR model are BLUEs.

That algorithm contents five main stages as follows:

### *Stage one: Planning for the study*

In this stage, we should put the plan to our simulation study; the plane contains many important subjects:

- Satisfy our **goals** of the study (*prove that OLS estimators of GLR model are BLUEs*)
- Studying and **understanding the model** that will use in the study. (*Studying theoretically framework of the GLR model*)

The GLR model is given as:

$$Y = X\beta + u \quad (1)$$

where  $Y$  is  $n \times 1$  dependent vector,  $X$  is  $n \times k$  independent variables matrix,  $\beta$  is  $k \times 1$  unknown parameters vector, and  $u$  is  $n \times 1$  error term vector.

### Assumptions:

A1:  $E(u) = 0, E(uu') = \sigma_u^2 I_n$ .

A2:  $X$  is non-stochastic matrix and  $cov(X, u) = 0$ .

A3:  $X$  is full column rank matrix, i.e.,  $rank(X) = k$ .

The OLS estimator of  $\beta$  is given as:

$$\hat{\beta}_{ols} = (X'X)^{-1}X'Y \quad (2)$$

- Satisfy the **simulation controls** (sample size ( $n$ ), number of the independent variables ( $k - 1$ ), standard deviation of the error term ( $\sigma_u$ ), theoretical assumptions of the *GLR model (A1 to A3 above)*, and so on )
- Satisfy the **criteria** that will calculate in the simulation study (*Bias and variance of OLS estimators, that are given as*):

$$bias(\hat{\beta}_{ols}) = \hat{\beta}_{ols} - \beta; \quad var(\hat{\beta}_{ols}) = \sigma_u^2(X'X)^{-1} \quad (3)$$

- **Note that** these criteria are given in econometric literature, but if they are not given theoretically, we can calculate them by simulation. As an example, see Abonazel (2014a), Youssef *et al.* (2014), and Youssef and Abonazel (2015).

### **Stage two: Building the model**

We can build our model by generate all the simulation controls. In this stage, we must follow the following steps by order:

**Step 1:** Suppose any values as true values of the parameters vector  $\beta$ .

**Step 2:** Choose the sample size  $n$ .

**Step 3:** generate the random generate the of the error vector  $u$  under the model assumptions.

**Step 4:** Generate the fixed values of the independent variables matrix  $X$  under A2 and A3.

**Step 5:** Generate the values of dependent variable  $Y$  by using the regression equation, since we well know  $\beta, u,$  and  $X$ .

### **Stage three: The treatment**

- Once we obtain  $Y$  vector plus  $X$  matrix, thus we successes to build our model under the satisfied assumptions.
- Now we ready to make the treatment has been satisfied in planning stage. The treatment is exactly correlated with the goals of our study. In our example, the treatment is the estimation of the regression parameters by using OLS method and then proves that OLS estimators are BLUEs.

- We can summarize the treatment stage in the following steps:

**Step 1:** Regress  $Y$  on  $X$  by using the OLS formula in equation (2), then obtain the OLS estimations  $\hat{\beta}_{ols}^1$ .

**Step 2:** Calculate the criteria that have been satisfied planning stage. Then **we calculate**  $bias(\hat{\beta}_{ols}^1)$  and  $var(\hat{\beta}_{ols}^1)$  by using equation (3).

### **Stage four: The Replications**

- Once we end the treatment stage, we obtain the values of biases and variances for only one experiment (one sample), then we cannot dependence with these values. To solve that, we should make the following:

**Step 1: Repeat** this experiment  $(L-1)$  times, each time using the same values of the parameters and independent variables, if  $n$  and  $k$  are not changed. Of course, the  $u$  values will vary from experiment to experiment even though  $n$  and  $k$  are not changed. Therefore, in all we have  $L$  experiments, thus generating  $L$  values each of biases and variances.<sup>1</sup>

**Step 2:** Take the **averages** of these  $L$  estimates and call them Monte Carlo estimates:

$$\text{mean. bias}(\hat{\beta}_{ols}) = \frac{1}{L} \sum_{l=1}^L \hat{\beta}_{ols}^l - \beta ; \quad (4)$$

---

<sup>1</sup> In practice, many such experiments are conducted sometimes 1000 to 2000. See Gujarati (2003).



$$\text{mean. var}(\hat{\beta}_{ols}) = \frac{1}{L} \sum_{l=1}^L \text{var}(\hat{\beta}_{ols}^l). \quad (5)$$

### **Stage five: Evaluating and presenting the results**

- After ending the treatment stage, we must check and evaluate the simulation result before put or discuss (display) it in our paper (research).
- The evaluation process aims to answer an important question: Are the results consistent with the theoretical framework or not?
- If the answer is yes, thus these results can be relied upon.
- But in a case of the results are inconsistent with the theoretical framework, we must review and/or

repeat the four stages with more accuracy to catch the mistake and correct it.

- The reviewing process contains two branches. First, *review the theoretical framework* of the model from different books or papers. Second, *review your software program*, there may be programmatic mistakes.
- After this evaluation, we can repeat calculate the simulation criteria again in different situations (apply the simulation factors), this step is very important because it gives us general image and more analysis of the studied model.

- In the end, the results should be consistent with the theoretical framework. And then, we should display these results using a properly method.
  - There are two main methods, to provide any simulation results, are *tables* and *graphs*. The researcher chooses between tables and graphs based on the contribution made by each method.
- ❖ **For more details** about the simulation technics that used in econometrics, you can review the following references: Craft (2003) and Barreto and Howland (2005).

## 6. **The Application: Multiple linear regression model with autocorrelation problem**

In this application, we apply the above algorithm of Monte Carlo technic to compare between OLS and GLS estimators in multiple linear regression model when the errors are correlated with first-order autoregressive (AR(1)). In each stage, we provide R-code to create it. In this workshop, we suppose that the reader is familiar with R-programming basics. If you are not satisfied that, you can review the following references: Robert and Casella (2009), Crawley (2012), and Abonazel (2014b).

### **Stage one: Planning for the study**

Now we apply the first stage, so we satisfy four factors as follows:

1. ***Satisfy our goal of the study***: The goal is compare between the performance of OLS and GLS estimators in multiple linear regression model when the errors are correlated with first-order.

2. *Studying theoretically framework of the model:* this model is given in equation (1), where A2 and A3 are still valid, but A1 will be replaced to the following assumption:

$$\mathbf{A4}: u_t = \rho u_{t-1} + \varepsilon_t; \text{ where } \varepsilon_t \sim N(0, \sigma_\varepsilon^2), \\ E(\varepsilon_t \varepsilon_s) = 0 \quad \forall t \neq s, \text{ and } E(\varepsilon_t u_{t-1}) = 0.$$

The OLS and GLS estimators of  $\beta$  under A2 to A4 are:

$$\hat{\beta}_{ols} = (X'X)^{-1}X'Y; \quad \hat{\beta}_{gls} = (X'\Omega^{-1}X)^{-1}X'\Omega^{-1}Y$$

where

$$\Omega = E(uu') = \frac{\sigma_\varepsilon^2}{1 - \rho^2} \begin{pmatrix} 1 & \rho & \rho^2 & \dots & \rho^{n-1} \\ \rho & 1 & \rho & \dots & \rho^{n-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{n-1} & \rho^{n-2} & \rho^{n-3} & \dots & 1 \end{pmatrix}.$$

Since the elements of  $\Omega$  are usually unknowns, we develop a feasible Aitken estimator of  $\beta$  based on consistent estimators of it:

$$\hat{\rho} = \frac{\sum_{i=2}^n \hat{u}_i \hat{u}_{i-1}}{\sum_{i=2}^n \hat{u}_{i-1}^2},$$

where  $\hat{u}_i$  are the residuals from apply OLS, and

$$\hat{\sigma}_{\varepsilon}^2 = \frac{\sum_{i=1}^n \hat{\varepsilon}_i^2}{n - K},$$

where  $\hat{\varepsilon}_1 = \hat{u}_1 \sqrt{1 - \hat{\rho}^2}$ , and  $\hat{\varepsilon}_i = \hat{u}_i - \hat{\rho} \hat{u}_{i-1}$ ; for  $i = 2, \dots, n$ .

3. **Satisfy the simulation controls:** Table 1 displays the full details about the simulation factors.

Table (1): The simulation factors

| No. | Simulation Factor   | Levels                               |
|-----|---|--------------------------------------|
| 1   | The true values of the parameters ( $\beta$ )               | $\beta = 1, 1$ (where $k = 2$ )      |
| 2   | Sample size ( $n$ )   | $n = 5, 15, 30,$ and $50$            |
| 3   | The AR(1) coefficient ( $\rho$ )                            | $\rho = 0.50$ and $0.90$             |
| 4   | The variance of the error term ( $\sigma_{\varepsilon}^2$ ) | $\sigma_{\varepsilon}^2 = 1$ and $5$ |

4. **Satisfy the study criteria:** The *criteria* here are the bias and variance of *OLS and GLS estimators* that are given in this model as:

$$\text{bias}(\hat{\beta}_{ols}) = \hat{\beta}_{ols} - \beta; \quad \text{bias}(\hat{\beta}_{gls}) = \hat{\beta}_{gls} - \beta$$

$$\text{var}(\hat{\beta}_{ols}) = (X'X)^{-1}X'\Omega X(X'X)^{-1}$$

$$\text{var}(\hat{\beta}_{gls}) = (X'\Omega^{-1}X)^{-1}$$

### **Stage two: Building the model**

We can build our model by generate all the simulation controls (factors) as given in table (1). The R-code is:

```
#---- Stage two: Building the model
#---- Step 1: Suppose the true values of the parameters vector  $\beta$  :
True.Beta<- c(1,1)
#---- Step 2: Choose the sample size n:
n=5
#---- Step 3: generate the random generate the of the error vector u
under A1:
sigma.epsilon = sqrt(1)
rho=0.50
epsilon= rnorm(n,0, sigma.epsilon)
u=c(0)
u[1]=epsilon[1]/((1-(rho)^2)^0.5)
for(i in 2:n) u[i]=rho*u[i-1]+epsilon[i]
#---- Step 4: Generate the fixed values of the independent variables
matrix X under A2 and A3:
X = cbind(1,runif(n,-1,1))
#---- Step 5: Generate the values of dependent variable Y :
Y=X%%True.Beta+u
```

## Stage three: The treatment

```
#---- Stage three: The treatment (by cerate estimation function):
estimation<-function(Y=Y,X=X){
#---- Step 1: calculate OLS and GLS estimators
##1 ---- OLS estimator:
Beta.hat.ols = solve(t(X) %*% X) %*% t(X) %*% Y
## 2 ----GLS estimator:
rho.hat = (t(u[-n]) %*% u[-1])/sum(u[-1]^2)
dim(rho.hat)=NULL
if(rho.hat>1) rho.hat=0.99; if(rho.hat<0) rho.hat=0.005
#-----
epsilon.hat=NA
epsilon.hat[1]= u[1]*(1 - (rho.hat) ^2) ^0.5
epsilon.hat[2:n]= u[-1]+rho.hat * u[-n]
sigma2.epsilon.hat= sum(epsilon.hat^2)/(n-2)
dim(sigma2.epsilon.hat)=NULL
#-----
v <- matrix(NA,nrow = n,ncol = n)
for (i in 1:n) for (j in 1:n) v[i,j] = (rho.hat) ^ abs(i - j)
omega <- (sigma2.epsilon.hat / (1 - (rho.hat) ^ 2)) * v
Beta.hat.gls = solve(t(X) %*% solve(omega) %*% X) %*% (t(X) %*%
solve(omega) %*% Y)
#---- Step 2: Calculate the Simulation criteria (bias and variance)
bias.ols = Beta.hat.ols - True.Beta
bias.gls = Beta.hat.gls - True.Beta
var.Beta.hat.gls = diag(solve(t(X) %*% solve(omega) %*% X))
var.Beta.hat.ols = diag (solve(t(X) %*% X) %*% t(X) %*% omega
%*% X %*% solve(t(X) %*% X))
BV= cbind(bias.ols, bias.gls, var.Beta.hat.ols, var.Beta.hat.gls)
rownames (BV) = c("Beta0", "Beta1")
colnames(BV) = c("Bias OLS", "Bias GLS", "Var OLS", "Var GLS")
return (BV) }
```



## **Stage four: The Replications**

Once we end the **treatment stage**, we obtain the values of biases and variances for only one experiment (one sample). Therefore, we **Repeat** this experiment (L-1) times, and then take the **averages** of these L estimates as follows:

```
#---- Stage four: The Replications
L=5000
Sim.results=matrix(0,nrow=2,ncol=4)

for (l in 1:L) {
  epsilon= rnorm(n,0, sigma.epsilon)
  u=c(0)
  u[1]=epsilon[1]/((1-(rho)^2)^0.5)
  for(i in 2:n) u[i]=rho*u[i-1]+epsilon[i]
  Y=X%%True.Beta+u
  results.matrix = estimation (Y=Y,X=X)
  Sim.results = Sim.results + results.matrix
}
average= Sim.results /l
average
write.table(average, "clipboard", sep="\t", col.names=NA )
```

### Stage five: Evaluating and presenting the results

After ending the treatment stage, we must check and evaluate the simulation result. The evaluation process aims to answer an important question: Are the results consistent with the theoretical framework or not?

Table (2): Simulation results when  $n = 5$ ,  $\beta = (1,1)'$ ,  $\sigma_{\varepsilon}^2 = 1$ , and  $\rho = 0.50$

|              | Bias OLS | Bias GLS | Var OLS  | Var GLS  |
|--------------|----------|----------|----------|----------|
| <b>Beta0</b> | -0.00063 | -0.00608 | 48.84283 | 47.66144 |
| <b>Beta1</b> | 0.029276 | 0.042476 | 16.0963  | 10.58763 |

**Note that:** Bias and variance for GLS estimates are *less than* the bias and variance of OLS estimates, this result is consistent with the theoretical framework, then we can rely on these results.

**After this evaluation,** we can repeat calculate the simulation criteria again in different situations. In other words, we calculate the values of biases and variances under different simulation factors as given in table (1). And then, we should display these results using a properly method. Here we will use the tables.

```

#---- Complete Program after definition our function (estimation)
###-----Not Fixed-----
n = c(5,15,30,50)
rho = c(0.50, 0.90)
sigma.epsilon = sqrt(c(1,5))
#-----Fixed -----
True.Beta <- c(1,1)
L = 1000
Sim.results = matrix (0,nrow = 2,ncol = 4)
Final.table = array(NA,c(16,8))
colnames(Final.table) = c(
  "n = 5","n = 5","n = 15","n = 15","n = 30","n = 30",
  "n = 50","n = 50")
#-----
ro=0
for (rhoi in 1:2) {
  se = 0
  for (sigma in 1:2) {
    sz = 0
    for (ni in 1:4) {
      X = cbind(1,runif(n[ni],-1,1))
      for (l in 1:L) {
        epsilon = rnorm(n[ni],0, sigma.epsilon[sigma])
        u = c(0)
        u[1] = epsilon[1] / ((1 - rho[rhoi]) ^ 2) ^ 0.5
        for (i in 2:n[ni])
          u[i] = rho[rhoi] * u[i - 1] + epsilon[i]
        Y = X %%% True.Beta + u
        results.matrix = estimation (Y = Y,X = X)
        Sim.results = Sim.results + results.matrix
      } ## for l
      average = Sim.results / l
      Final.table[(ro+se + 1):(ro+se + 4),(sz + 1):(sz + 2)] <- t(average)
      sz = sz + 2 } ##for ni
      se = se + 4 } ## for sigma
    ro=ro+8} ## for rhoi
  } ## for rhoi
} ## for sigma
} ## for rhoi
Final.table
write.table(Final.table, "clipboard", sep="\t", col.names=NA )

```

**Table (3): The results of the Monte Carlo study when the replications = 1000**

| $n$   | $\beta_0$ | $\beta_1$ | $\beta_0$ | $\beta_1$ | $\beta_0$ | $\beta_1$ | $\beta_0$ | $\beta_1$ |
|---|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| <b><math>\sigma_{\varepsilon}^2 = 1, \rho = 0.50</math></b> |           |           |           |           |           |           |           |           |
| <b>Bias OLS</b>   | -0.067    | 0.008     | -0.076    | 0.012     | -0.080    | 0.020     | -0.076    | 0.016     |
| <b>Bias GLS</b>   | -0.069    | -0.019    | -0.083    | -0.011    | -0.090    | 0.003     | -0.087    | -0.005    |
| <b>Var OLS</b>  | 41.513    | 3.740     | 43.376    | 4.891     | 43.903    | 5.128     | 44.161    | 5.276     |
| <b>Var GLS</b>  | 41.155    | 2.645     | 42.886    | 3.483     | 43.391    | 3.659     | 43.645    | 3.757     |
| <b><math>\sigma_{\varepsilon}^2 = 5, \rho = 0.50</math></b> |           |           |           |           |           |           |           |           |
| <b>Bias OLS</b>   | -0.103    | 0.126     | -0.014    | 0.146     | 0.018     | 0.138     | 0.042     | 0.131     |
| <b>Bias GLS</b>   | -0.116    | 0.101     | -0.040    | 0.120     | -0.005    | 0.106     | 0.021     | 0.104     |
| <b>Var OLS</b>  | 242.182   | 12.622    | 252.728   | 17.297    | 255.231   | 19.370    | 256.507   | 20.076    |
| <b>Var GLS</b>  | 240.451   | 10.678    | 250.363   | 13.426    | 252.779   | 14.489    | 254.026   | 14.978    |
| <b><math>\sigma_{\varepsilon}^2 = 1, \rho = 0.90</math></b> |           |           |           |           |           |           |           |           |
| <b>Bias OLS</b>   | 0.008     | 0.137     | 0.123     | 0.167     | 0.073     | 0.150     | 0.047     | 0.143     |
| <b>Bias GLS</b>   | -0.003    | 0.104     | 0.112     | 0.105     | 0.080     | 0.100     | 0.064     | 0.098     |
| <b>Var OLS</b>  | 995.257   | 29.104    | 1288.576  | 38.887    | 1478.919  | 41.730    | 1554.474  | 43.076    |
| <b>Var GLS</b>  | 988.125   | 22.141    | 1272.723  | 24.514    | 1453.568  | 25.450    | 1522.198  | 25.964    |
| <b><math>\sigma_{\varepsilon}^2 = 5, \rho = 0.90</math></b> |           |           |           |           |           |           |           |           |
| <b>Bias OLS</b>   | 0.313     | 0.215     | 0.267     | 0.252     | 0.283     | 0.209     | 0.276     | 0.215     |
| <b>Bias GLS</b>   | 0.327     | 0.182     | 0.280     | 0.208     | 0.264     | 0.181     | 0.272     | 0.173     |
| <b>Var OLS</b>  | 5151.000  | 316.166   | 7051.481  | 339.658   | 8519.216  | 352.380   | 8895.343  | 390.531   |
| <b>Var GLS</b>  | 5091.892  | 222.181   | 6950.816  | 236.186   | 8366.166  | 241.435   | 8705.211  | 244.697   |

❖ If you are want to display the simulation results in graphs, see, e.g., Abonazel (2009, Appendix B) for *2D graphs*, while for *3D graphs*, see, e.g., Abonazel (2014a, Appendix B) and Youssef and Abonazel (2015).

- ❖ In the previous example, we have studied the estimation properties of *single-equation regression* model. However there are studies are used the Monte Carlo simulation technics for *multi-equation regression* models (such as panel data models), see, e.g., Youssef and Abonazel (2009) and Mousa *et al.* (2011).

## 7. General notes on simulation using R

- R is considered one of the **fastest packages** for simulation.
- If the simulation time took too long or you want to end the processing, you can press the red icon "**STOP**" in the tool menu anytime.
- Two way to **reduce the bias** (bias = mean of experiments – true value):
  - By increase the **sample size**.
  - By increase the number of **iterations** but it will not be as effective.
- In loops, we can **create nested loops** (a loop inside a loop) very easily. For example: loop for i and loop for j inside it, i.e. **for i (for j)**

- In iterations, it is highly **recommended to omit the first 50** iterations from the calculations (such as bias or variances values)

## References

1. Abonazel, M. R. (2009). Some properties of random coefficients regression estimators. MSc thesis. Institute of Statistical Studies and Research. Cairo University.
2. Abonazel, M. R. (2014a). Some estimation methods for dynamic panel data models. PhD thesis. Institute of Statistical Studies and Research. Cairo University.
3. Abonazel, M. R. (2014b). Statistical analysis using R, Annual Conference on Statistics, Computer Sciences and Operations Research, Vol. 49. Institute of Statistical Studies and research, Cairo University. DOI: 10.13140/2.1.1427.2326.
4. Barreto, H., Howland, F. (2005). *Introductory econometrics: using Monte Carlo simulation with Microsoft excel*. Cambridge University Press.
5. Craft, R. K. (2003). Using spreadsheets to conduct Monte Carlo experiments for teaching introductory econometrics. *Southern Economic Journal*, 726-735.
6. Crawley, M. J. (2012). *The R book*. John Wiley & Sons.
7. Gentle, J. E. (2003). *Random number generation and Monte Carlo methods*. Springer Science & Business Media.
8. Gujarati, D. N. (2003) *Basic econometrics*. 4th ed. McGraw-Hill Education.
9. Gentle, J. E., Härdle, W. K., Mori, Y. (2012). *Handbook of computational statistics: concepts and methods*. Springer Science & Business Media.

10. Mooney, C. Z. (1997). *Monte Carlo simulation*. Sage University Paper Series on Quantitative Applications in the Social Sciences, series no. 07-116. Thousand Oaks, CA: Sage.
11. Mousa, A., Youssef, A. H., Abonazel, M. R. (2011). *A Monte Carlo study for Swamy's estimate of random coefficient panel data model*. Working paper, No. 49768. University Library of Munich, Germany.
12. Robert, C., Casella, G. (2009). *Introducing Monte Carlo Methods with R*. Springer Science & Business Media.
13. Robert, C., Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.
14. Thomopoulos, N. T. (2012). *Essentials of Monte Carlo Simulation: Statistical Methods for Building Simulation Models*. Springer Science & Business Media.
15. Youssef, A. H., Abonazel, M. R. (2009). *A comparative study for estimation parameters in panel data model*. Working paper, No. 49713. University Library of Munich, Germany.
16. Youssef, A. H., Abonazel, M. R. (2015). Alternative GMM estimators for first-order autoregressive panel model: an improving efficiency approach. *Communications in Statistics-Simulation and Computation* (in press). DOI: 10.1080/03610918.2015.1073307.
17. Youssef, A. H., El-sheikh, A. A., Abonazel, M. R. (2014). New GMM estimators for dynamic panel data models. *International Journal of Innovative Research in Science, Engineering and Technology* 3:16414–16425.