International Journal of Mathematics and Computational Science
Vol. 4, No. 1, 2018, pp. 18-33
http://www.aiscience.org/journal/ijmcs
ISSN: 2381-7011 (Print); ISSN: 2381-702X (Online)

A Practical Guide for Creating Monte Carlo Simulation Studies Using R

Mohamed Reda Abonazel[*]

Department of Applied Statistics and Econometrics, Institute of Statistical Studies and Research, Cairo University, Cairo, Egypt

Abstract

This paper considers making Monte Carlo simulation studies using R language. Monte Carlo simulation techniques are very commonly used in many statistical and econometric studies by many researchers. So, we propose a new algorithm that provides researchers with basics and advanced skills about how to create their R-codes and then achieve their simulation studies. Our algorithm is a general and suitable for creating any simulation study in statistical and econometric models. Moreover, we provide some empirical examples in econometrics as applications on this algorithm.

Keywords

Autocorrelation Problem, Econometric Modeling, Graphical Presentation Methods, Ridge Estimation, Seemingly Unrelated Regressions Model

Received: March 15, 2018 / Accepted: April 3, 2018 / Published online: May 28, 2018

1. Introduction

Frequently, the properties of statistical and econometric models cannot easily be developed through statistical and econometric theories alone, due to complexity or gaps in understanding, necessitating the use of empirical methods. How do we empirically study a theoretical model? Well we have to find an actual instance of it that is conducive to study. But this is usually an impossible proposition. So, Monte Carlo simulation (MCS) techniques have been used to generate data indistinguishable from data collected from actual phenomena that adhered to the specifications of our model. MCS techniques are thus uniquely suited to empirically studying the properties of theoretical models and this is how they're most often used (See [14]).

Historically, MCS method proved to be successful and was an important instrument in the Manhattan Project of World War II. After the war, 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 MCS method (See [25]). Now it is used routinely in many different fields such as business, engineering, science and finance. This paper focuses on the usage of MCS techniques in statistics and econometrics.[1]

We can summarize the general advantages or goals of MCS techniques in the following four points: First, evaluate the performance of an inference method. Second, evaluate the robustness of parametric inference to assumption violations. Third, compare the statistical properties of different estimators. Finally, make inferences when weak statistical theory exists for an estimator.

Mooney [19] presented the general methodology of MCS, which mentions in most statistical literature, in the following five steps:

Step 1: Specify the pseudo-population in symbolic terms in

1 For more details about MCS methods that used in statistics and econometrics, see, e.g., [9], [7], [25], [13], [22], and [3].

* Corresponding author
E-mail address: mabonazel@hotmail.com

such a way that it can be used to generate samples by writing a code to generate data in a specific method.

Step 2: Sample from the pseudo-population in ways that show the topic of interest.

Step 3: Estimate the parameter using a pseudo-sample and store it in a vector.

Step 4: Repeat steps 2 and 3 L-times where L is the number of trials.

Step 5: 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.

The above algorithm is commonly used in most books and paper that discuss the methodology of MCS because it summarizes the basic steps to create MCS study. However, this algorithm has not provided to the researcher sufficient details about how to conduct each step in this algorithm. Moreover, it not contains important assumptions about the population, sample, model, and estimation method. While that these assumptions are very effective on the performance of MCS. Therefore, we will propose a comprehensive algorithm to create a professional MCS study using R-language (one of the best programming languages).

This paper is organized as follows. Section 2 provides the proposed algorithm. Section 3 presents two empirical examples in econometrics as applications on our algorithm. In Section 4, the graphical presentation methods for simulation results have been discussed. Finally, Section 5 offers the concluding remarks.

2. The Proposed Algorithm: The Full Steps to Create a MCS Study

In this section, we provide a complete algorithm of MCS study. We explain our algorithm through an application in regression framework, especially; we will use Monte Carlo technique to prove that ordinary least squares (OLS) estimator of classical linear regression (CLR) model is unbiased and efficient. Our algorithm contents five main stages as follows:

Stage one: Planning for the study

In this stage, we should put the plan for our simulation study; this plane contains two important subjects: First, specify our goals of the study (*in our example the goal is: prove that OLS estimators of CLR model is unbiased and efficient*). Second, studying and understanding the model that will use in the study. (*Study the theoretical framework of CLR model*).

The CLR model is given as:

$$Y = X\beta + u \tag{1}$$

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

Assumptions:

A1.1: $E(u) = 0, E(uu') = \sigma_u^2 I_n$.

A1.2: X is non-stochastic matrix and $cov(X, u) = 0$.

A1.3: X is full column rank matrix, i.e., $rank(X) = k$.

The OLS estimator of β is given as:

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

Specify the simulation controls (sample size (n), number of the independent variables $(k - 1)$, standard deviation of the error term (σ_u), theoretical assumptions of *CLR model (A1 to A3 above), and so on*).

Specify the criteria that will calculate in the simulation study (*Bias and variance of OLS estimators, that are given as*):

$$bias(\hat{\beta}_{ols}) = \hat{\beta}_{ols} - \beta; \tag{3}$$

$$var(\hat{\beta}_{ols}) = \sigma_u^2 (X'X)^{-1}. \tag{4}$$

We would like to point out here that the simulation criteria, in all MCS studies, are divided into two types; theoretical and empirical criteria. The theoretical criteria are defined as the measures that proved theoretically, for example, the formula of the variance of OLS estimator that given in equation (4). While the empirical criteria are defined as the measures that not proved theoretically and have been concluded from a simulated data and they based on the true parameters of population, for example, the formula of the bias of OLS estimator that given in equation (3), despite the fact that the OLS estimator is theoretically unbiased.[2]

Stage two: Building the model

We can build our model by generate all 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 β.

Step 2: Choose the sample size n.

Step 3: Generate the random values of the error vector u under the model assumptions.

Step 4: Generate the fixed values of the independent variables matrix X under A1.2 and A1.3.

2 The empirical criteria are commonly used in recent studies that discuss advanced models, see, e.g., [1], [4], [29], and [28].

Step 5: Calculate the values of dependent variable Y using the regression equation, since $\beta, u,$ and X are known.

Stage three: The treatment

Once we obtain Y vector plus X matrix, thus we successes to build our model under the specified assumptions. Now, we ready to make the treatment that is specified in planning stage. The treatment is exactly correlated with goals of our study. In our example, the treatment is estimating the parameters by OLS method, and proving that OLS estimator is unbiased and efficient. We can summarize this stage in the following steps:

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

Step 2: Calculate the criteria that are specified in planning stage. Then we calculate $bias(\hat{\beta}^{1}_{ols})$ and $var(\hat{\beta}^{1}_{ols})$ using equations (3) and (4), respectively.

Stage four: The Replications

Once you have completed the treatment stage, we obtain the values of bias and variance for only one experiment (one sample). However, we cannot rely upon these values only. So, we must make the following:

Step 1: Repeat this experiment (L-1) times. In each experiment, the same values of the parameters and independent variables are used, if n and k are not changed. Of course, the values of u are varying from experiment to experiment even though n and k are not changed. In the end, we have L-values of bias and variance.[3]

Step 2: Take the averages of these values and call them Monte Carlo estimates:

$$mean.bias(\hat{\beta}_{ols}) = \tfrac{1}{L}\textstyle\sum_{l=1}^{L}\hat{\beta}^{l}_{ols} - \beta;$$

$$mean.var(\hat{\beta}_{ols}) = \tfrac{1}{L}\textstyle\sum_{l=1}^{L} var(\hat{\beta}^{l}_{ols}).$$

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 simulation 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 the program code, there may be programmatic mistakes.

After this evaluation, we can repeat calculate the simulation criteria again in different situations (depending on simulation factors), this step is very important because it gives us general image and more analysis of the studied model. In the end, the simulation results should be displayed using a properly method. The tables and graphs are the main methods to present any simulation results. The researcher chooses between them based on the contribution made by each method.

3. Applications on Econometric Models

In this section, we apply the proposed algorithm to conduct two simulation studies in two different econometric models. All details of these studies (applications) are given in Table 1.

Table 1. The details of applications.

Item	Application I	Application II
The model	Linear regression model	Seemingly unrelated regressions model
Model type	Single-equation model	Multi-equation model
Econometric problem	Autocorrelation	Multicollinearity
Study objective	Compere between OLS and generalized least squares (GLS) estimators	Compere between the performance of Zelner's and ridge estimators, and provide new efficient estimators

Application I: Linear Regression Model with Autocorrelation Problem

In this application, we apply above algorithm of Monte Carlo technique to compere between OLS and GLS estimators in linear regression model when the errors are correlated with first-order autoregressive (AR(1)). In each stage, we provide an R-code that achieves it.[4]

Stage one: Planning for the study

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

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

2. Study theoretical framework of the model: this model is given in equation (1), where A1.2 and A1.3 are still valid, but A1.1 will be replaced to the following assumption:

A1.4: $u_i = \rho u_{i-1} + \varepsilon_i$; $|\rho| < 1$, where ρ is a first-order

3 In practice, many such experiments are conducted sometimes 1000 to 2000. See [12].

4 In this paper, we suppose that the reader is familiar with the basics of R-programing. If not, it can review the following references: [21], [10], and [2].

autocorrelation coefficient, ε_i is an error term; with $E(\varepsilon_i) = 0$, $E(\varepsilon_i u_{i-1}) = 0$, and

$$E(\varepsilon_i \varepsilon_s) = \begin{cases} \sigma_\varepsilon^2 & if\ i = s; \\ 0 & if\ i \neq s; \end{cases} i = s = 1, \dots, n.$$

The OLS and GLS estimators of β under A1.2 to A1.4 are:

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

where

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

Since the elements of Ω are usually unknowns, we develop a feasible Aitken estimator of β 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. Specify the simulation controls: Table 2 displays the full details about the simulation factors.

Table 2. The simulation factors of application I.

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

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

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

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

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

Stage two: Building the model

We can build our model by generate all simulation controls (factors) as given in Table 2. The R-code is:

```
#---- Stage two: Building the model
rm(list = ls(all = TRUE)) # Remove all objects in R console
set.seed(123456) # Set the seed for reproducible results
#---- Step 1: Suppose the true values of the parameters vector β:
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

In this stage, we will conduct the required treatments on the simulated model (X and Y) that generated in previous stage. Implementing it on our application, we will create an R-function to calculate bias and variance values of OLS and GLS estimators using the following R-code:

```
#---- Stage three: The treatment (by creating an 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:
   n=length(Y)
   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

From previous stage, we get the values of bias and variance for only one experiment (one sample). Therefore, we repeat this experiment $(L - 1)$ times, and then we take the average of these L estimates as in the following code:

```
#---- Stage four: The Replications
set.seed(123456) # Set the seed for reproducible results
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
```

Table 3. Simulation results when n = 5, $\beta = (1, 1)'$, $\sigma_\varepsilon^2 = 1$, and $\rho = 0.50$.

	Bias OLS	Bias GLS	Var OLS	Var GLS
Beta 0	0.00051	0.00071	16.76125	16.22938
Beta 1	0.00598	0.00819	4.61890	3.46068

Stage five: Evaluating and presenting the results

In this stage, we 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 3 indicates that the variance of GLS estimates is less than the variance of OLS estimates. While the bias of OLS and GLS estimates are very close to zero. This result is consistent with the theoretical framework, and 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 repeat calculate the values of bias and variance under different simulation factors that were given in Table 2. Therefrom, the complete program and the final table of this study are:

```
#---- The complete program based on estimation function
set.seed(123456) # Set the seed for reproducible results
n = c(5,15,30,50)
rho = c(0.50, 0.90)
sigma.epsilon  = sqrt(c(1,5))
True.Beta <- c(1,1)
L = 10000
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))
         Sim.results  = matrix (0,nrow = 2,ncol = 4)
         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
Final.table
```

Table 4. The results of the Monte Carlo study when the replications equal to 10000.

	β_0	β_1	β_0	β_1	β_0	β_1	β_0	β_1
n	**5**	**5**	**15**	**15**	**30**	**30**	**50**	**50**
				$\sigma_\varepsilon^2 = 1, \rho = 0.50$				
Bias OLS	-0.019	-0.007	0.006	0.000	-0.004	-0.003	-0.002	0.001
Bias GLS	-0.017	-0.005	0.006	0.001	-0.005	0.000	-0.002	0.001
Var OLS	4.899	2.560	0.375	0.120	0.156	0.123	0.088	0.084
Var GLS	4.855	2.287	0.355	0.094	0.151	0.081	0.086	0.050
				$\sigma_\varepsilon^2 = 5, \rho = 0.50$				
Bias OLS	-0.010	-0.044	0.006	-0.010	0.007	0.000	-0.003	-0.011
Bias GLS	-0.008	-0.036	0.005	-0.008	0.005	0.001	-0.003	-0.004
Var OLS	23.958	11.527	1.960	1.172	0.781	0.429	0.444	0.418
Var GLS	23.482	10.059	1.867	0.906	0.755	0.283	0.433	0.262
				$\sigma_\varepsilon^2 = 1, \rho = 0.90$				
Bias OLS	0.014	0.000	-0.009	0.007	-0.002	0.001	0.003	0.004
Bias GLS	0.017	0.002	-0.009	0.001	-0.003	0.003	0.002	0.003
Var OLS	24.811	1.169	15.031	0.401	5.703	0.188	2.622	0.174

	β_0	β_1	β_0	β_1	β_0	β_1	β_0	β_1
n	5	5	15	15	30	30	50	50
Var GLS	24.660	1.089	14.734	0.199	5.390	0.074	2.415	0.035
				$\sigma_\varepsilon^2 = 5, \rho = 0.90$				
Bias OLS	-0.023	-0.040	-0.039	0.015	0.029	-0.019	-0.002	-0.008
Bias GLS	-0.017	-0.037	-0.018	0.002	0.031	-0.010	-0.006	-0.010
Var OLS	97.974	22.108	62.250	3.054	24.718	0.918	14.549	2.362
Var GLS	96.886	15.198	60.711	0.527	23.194	0.266	13.517	0.200

In general, Table 4 indicates that the variance of GLS estimates is less than the variance of OLS estimates in all simulation situations. Moreover, the variances of OLS and GLS estimates are increasing when $\boldsymbol{\sigma_\varepsilon^2}$ and $\boldsymbol{\rho}$ are increased. But when $\boldsymbol{n}$ is increasing, the variances of estimates are decreased. While the values of bias for all estimates are very close to zero in all simulation situations.

In this application, we have studied the estimation properties of single-equation regression model. However there are many studies are used the MCS techniques in multi-equation regression models (such as, panel data models, and simultaneous equation models), see, e.g., [27] and [20]. In multi-equation regression models, the creating of MCS study is difficult in most statistical and econometric studies. So, we will present an application on a multi-equation regression model.

Application II: Ridge Estimation of SUR Model

The seemingly unrelated regressions (SUR) model has been proposed by Zellner [30] on the assumption that the explanatory variables in the model are independent. But this assumption is very hard in economic models, because in most of the empirical works the researchers are often concerned about problem with the data, namely multicollinearity problem. This problem arises in situations when the explanatory variables are highly inter-correlated. Then it becomes difficult to disentangle the separate effects of each of the explanatory variables on the dependent variable. As a result, the estimated parameters may be statistically insignificant and/or have, unexpectedly, different signs. Thus, conducting a meaningful statistical inference would be difficult for the researcher. Srivastava and Giles [24] proposed the general ridge estimator for this model when the independent variables are affected by multicollinearity. Several ridge estimators are proposed and compared by [6].

In general, the studies on ridge regression estimators are pioneered by [16-17], and later followed by [26], [8], [24], [15], [23], [11], [18], and [5]. In most of these studies, the authors are used the simulation techniques to study the properties of some new proposed estimators and compared their properties with other popular existing estimators.

Stage one: Planning for the study

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

1. Specify our goal of the study: The goal here is compere between the performance of Zelner's (GLS) and ridge estimators of SUR model when the independent variables are affected by multicollinearity.

2. Study theoretical framework of the model: Consider the SUR model as m-system equations given by:

$$\underset{mn\times 1}{Y} = \underset{mn\times K}{X}\,\underset{K\times 1}{B} + \underset{mn\times 1}{U} \tag{5}$$

where Y is the vector of endogenous variable, and $X = diag\{X_i\}$; with X_i $(i = 1,2,\dots,m)$ is the matrix of the exogenous variables of equation number i with dimension $n \times k_i$, and B is the parameters vector with $K = \sum_{i=1}^{m} k_i$, while U is the errors vector.

Assumptions:

A2.1: $E(U) = 0$,

$$E(UU') = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1m} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{m1} & \sigma_{m2} & \cdots & \sigma_{mm} \end{bmatrix} \otimes I_n = \Sigma \otimes I_n = \Psi.$$

A2.2: X is non-stochastic matrix and $cov(X, U) = 0$.

A2.3: X is full column rank matrix, i.e., $rank(X) = K$.

Under these assumptions, Zellner's estimator of B in (5) and mean square error (MSE) of it are given by:

$$\hat{B}_Z = (X'\Psi^{-1}X)^{-1}X'\Psi^{-1}Y,$$
$$MSE(\hat{B}_Z) = (X'\Psi^{-1}X)^{-1}. \tag{6}$$

While, the ridge estimation of this model and MSE of it are given by:

$$\hat{B}_R = (X'\Psi^{-1}X + R)^{-1}X'\Psi^{-1}Y,$$
$$MSE(\hat{B}_R) = \begin{bmatrix} (X'\Psi^{-1}X + R)^{-1} \\ \times (X'\Psi^{-1}X + RBB'R') \\ \times (X'\Psi^{-1}X + R)^{-1} \end{bmatrix}, \tag{7}$$

where R is a $K \times K$ matrix with nonnegative elements. Note that if R is a diagonal matrix then the estimator $\hat{B}_R$ is regarded as an extension of the general ridge regression estimator. Further, if $R = rI_K$ with r is nonstochastic scalar then the estimator $\hat{B}_R$ is like the ordinary ridge regression estimator (see [24]).

The canonical version of model (5) is given by:

$$Y^* = Z\alpha + U^*, \tag{8}$$

where $Y^* = (\Sigma^{-.5} \otimes I_n)Y$, $U^* = (\Sigma^{-.5} \otimes I_n)U$, $Z = X^*\Phi$ with $X^* = (\Sigma^{-.5} \otimes I_n)X$ and Φ eigenvectors of $(X^{*\prime}X^*)$, then $Z'Z = \Phi' X^{*\prime} X^* \Phi = \Lambda$, where Λ the eigenvalues of $(X^{*\prime}X^*)$.

The corresponding GLS $(\hat{B}_Z)$ estimator of (8) is:

$$\hat{\alpha} = (Z'Z)^{-1} Z'Y^*$$

Srivastava and Giles [24] and Firinguetti [11] proved that the optimum values of $R = diag\{R_1, \dots, R_m\}$ with $R_i = diag\{r_{i1}, \dots, r_{ik_i}\}$ are:

$$r_{ij} = \frac{1}{\hat{\alpha}_{ij}^2};\ i = 1, \dots, m;\ j = 1, \dots, k_i, \tag{9}$$

if and only if the following conditions are holds: $\alpha' \Lambda \alpha < 1$ and $\alpha' R\, \alpha < 2$. Note that if $R = rI_K$ then these conditions are reduced to one condition: $0 < r < (2/\alpha'\alpha)$.

In this paper, we propose new ridge estimators of SUR model. These estimators are based on ridge parameters (r_{ij}). The following remark presents some ridge parameters:

Remark 1:

1. R_{SF}. The *ij*-th component of the matrix is given by (9).

2. R_{SK}. The median of r_{ij} that proposed by [18] for single equation version and Alkhamisi and Shukur [6] developed it for SUR model:

$$r_{ij}(SK) = median\left(\frac{1}{\hat{\alpha}_{ij}^2}\right).$$

3. R_{AS}. The max of r_{ij} that proposed by Alkhamisi and Shukur [6]:

$$r_{ij} = max\left(\frac{1}{\hat{\alpha}_{ij}^2}\right).$$

4. R_{new1}. The $K^* = K/m$ root of r_{ij} as a following:

$$r_{ij}(new1) = \left(\frac{1}{\hat{\alpha}_{ij}^2}\right)^{1/K^*}.$$

5. R_{new2}. The K^* root of sum r_{ij} as a following:

$$r_{ij}(new2) = \left[sum\left(\frac{1}{\hat{\alpha}_{ij}^2}\right)\right]^{1/K^*}.$$

6. R_{new3}. The K^* root of max r_{ij} as a following:

$$r_{ij}(new3) = \left[max\left(\frac{1}{\hat{\alpha}_{ij}^2}\right)\right]^{1/K^*}.$$

3. *Specify the simulation controls*: Table 5 displays the full details about the simulation factors.

Table 5. The simulation factors of application II.

No.	Simulation Factor	Levels
1	The true values of the parameters (B) Where $B = (\beta_1, \dots, \beta_m)'$, without intercept	$\beta_i = (1,1,1,1)'\ \forall\ i = 1, \dots, m$; where $k_i = 4$
2	Number of equations (m)	$m = 3$ (in table and 2D plots) and 6 (in 3D plots)
3	Sample size for each equation (n)	$n = 10, 25, 50$, and 100
4	The covariance matrix of X (Σ_x) It is defined as *diag* $(\Sigma_x) = 1$ and *off-diag* $(\Sigma_x) = \rho_x$	$\rho_x = .70, .80, .90$, and $.98$
5	The covariance matrix of U $(\Psi = \Sigma \otimes I_n)$ It is defined as *diag* $(\Sigma) = 1$ and *off-diag* $(\Sigma) = \rho_e$	$\rho_e = .75$

4. *Specify the study criteria*: The criterion is the trace mean squared error (TMSE) of Zellner's and several ridge estimators that are defined in Remark 1. To calculate TMSE of Zellner's and ridge estimators, we will use equations (6) and (7), respectively.

Stage two: Building the model

We can build our model by generate all simulation factors as given in Table 5. The R-code is:

```
#---- Prepare the R console
rm(list = ls(all = TRUE))
library ("MASS")
library ("simex")
set.seed(15344321)
#---- Stage two: Building the model
True.B <- c(1,1,1,1)
m = 3; n = 10; N = m * n; ki = 4; K = ki * m; roo_x = .90; roo_e = .75
##Generate X
sigma_x <- diag(1,nrow = ki)
sigma_x  [upper.tri(sigma_x)]  <- roo_x
sigma_x  [lower.tri(sigma_x)]  <- roo_x
LX <- list()
for (ix in 1:m)
  LX [[ix]] <- mvrnorm(n, mu = rep(1,ki), sigma_x)
X <- diag.block(LX)
#---- Generate Error
sigma_e <- diag(1,nrow = m)
sigma_e  [upper.tri(sigma_e)]  <- roo_e
sigma_e  [lower.tri(sigma_e)]  <- roo_e
error <- mvrnorm(n, mu = rep(0,m), sigma_e)
dim(error) <- c(N,1)
#---- Calculate Y
B_vector <- rep(True.B,m)
Y <- (X %*% B_vector) + error
```

Stage three: The treatment

In this stage, we will create R-functions to calculate TMSE values of Zellner's and several ridge estimators using the following R-code:

```
#---- Stage three: The treatment
#----1- Ridge.parameters
Ridge.parameters    <- function(Y,X,sigma_e,K) {
   m = nrow(sigma_e);      n = length(Y) / m
   transformation.factor =    kronecker ((sigma_e) ^ .5 , diag(1,n))
   X.star = transformation.factor %*% X
   Y.star = transformation.factor %*% Y
   phi <- eigen(t(X.star) %*% X.star)$vec
   Z = X.star      %*% phi
   alpha.hat = ginv(t(Z) %*% Z) %*% t(Z) %*% Y.star
   #------------------
   #---- Firinguetti [11]
   r.SF = 1 / (alpha.hat) ^ 2
   R.SF <- diag (as.numeric (r.SF))
   #- Kibria [18] and it has been developed by Alkhamisi and Shukur [6]
   r.SK = median(r.SF)
   R.SK <- diag (r.SK, K)
   #---- Alkhamisi and Shukur [6]
   r.AS = max(r.SF)
   R.AS <- diag (r.AS, K)
   #---- new1
   K.bar = K / m
   r.new1 = r.SF ^ (1 / K.bar)
   R.new1 <- diag (as.numeric (r.new1))
   #---- new2
   r.new2 = sum(r.SF) ^ (1 / K.bar)
   R.new2 <- diag (as.numeric (r.new2),K)
   #---- new3
   r.ZSK = max(r.SF) ^ (1 / K.bar)
   r.new3 = r.new2 * r.ZSK
   R.new3 <- diag (as.numeric (r.new3),K)
   #------------------
F, R.SK = R.SK, R.AS = R.AS, R.new1 = R.new1,R.new2 = R.new2,R.new3 = R.new3)
   return (Ridge.parameters)
}
#----2- SUR.estimation
SUR.estimation <- function(Y,X, sigma_e , B_vector, estimation.type, RP) {
      m = nrow(sigma_e); n = length(Y) / m
      transformation.factor =    kronecker ((sigma_e) ^ .5 , diag(1,n))
      X.star = transformation.factor %*% X
      Y.star = transformation.factor %*% Y
      phi <- eigen(t(X.star) %*% X.star)$vec
      Z = X.star      %*% phi
      alpha_vector =    t(phi) %*% B_vector
      if (estimation.type == "GLS") {
         RP = NULL
         R = matrix(0,ncol = length (B_vector),nrow = length (B_vector)) }
      if (estimation.type ==    "Ridge")
         R = RP
      part1 <- ginv (t(Z) %*% Z + R)
      part2 <-
         t(Z)    %*% Z + R %*% alpha_vector %*% t(alpha_vector) %*% t(R)
      alpha.hat.R = part1    %*% t(Z) %*%      Y.star
      MSE.B.hat.R = part1    %*% part2    %*% part1
      TMSE.R = sum (diag(MSE.B.hat.R))
      Ridge.est = list( Ridge.estimets = alpha.hat.R, Var_cov.matrix = MSE.B.hat.R, TMSE = TMSE.R)
      return (Ridge.est)
   }
```

Stage four: The Replications

In this stage, we repeat this experiment $(L - 1)$ times, and then we take the average of these L estimates as in the following code:

```
#---- Stage four: The Replications
set.seed(15344321)
True.B <- c(1,1,1,1)
m = 3;n = 10;N = m * n;ki = 4;K = ki * m;roo_x = .90;roo_e = .75
L = 10000
Sim.results = matrix (0,nrow = 1,ncol = 7)
results.matrix = matrix (0,nrow = 1,ncol = 7)
for (l in 1:L) {
  sigma_e <- diag(1,nrow = m)
  sigma_e  [upper.tri(sigma_e)]  <- roo_e
  sigma_e  [lower.tri(sigma_e)]  <- roo_e
  error <- mvrnorm(n, mu = rep(0,m), sigma_e)
  dim(error) <- c(N,1)
  B_vector <- rep(True.B,m)
  Y <- (X %*% B_vector) + error
  GLS.Z = SUR.estimation (Y,X, sigma_e , B_vector, estimation.type = "GLS")$TMSE
  results.matrix [,1] = GLS.Z
  Ridge.par =  Ridge.parameters  (Y,X,sigma_e ,K)
   for (RP in 1:6) results.matrix [,RP + 1] = SUR.estimation (Y,X, sigma_e , B_vector, estimation.type =
"Ridge",RP = Ridge.par[[RP]])$TMSE
  Sim.results = Sim.results + results.matrix  }
average = Sim.results / l
colnames(average) = c("Zelner","SF","SK","AS","new1","new2","new3")
rownames(average) = "TMSE"
average
```

Table 6. Simulation results when $m = 3$, $n = 10$, and $\rho_x = .90$.

	Zelner	SF	SK	AS	new 1	new 2	new 3
TMSE	103.918	41.485	6.112	1.839	2.549	0.569	0.442

Stage five: Evaluating and presenting the results

In this stage, we check and evaluate the results by answer the following question: Are the results consistent with the theoretical framework or not? Table 6 indicates that TMSE values of all ridge estimators are less than TMSE values of Zelner's estimator, and the results are consistent with the theoretical framework, then we can rely on these results. Therefrom, the complete program and the final table of this study are:

```
#---- The Complete Program
#---- Simulation Factors
m = 3; n = c(10,25,50,100)
roo_x = c(.70, .80, .90, .98)
#---- Fixed
set.seed(15344321)
L = 10000
True.B <- c(1,1,1,1)
B_vector <- rep(True.B,m)
ki = 4; K = ki * m; roo_e = .75
sigma_e <- diag(1,nrow = m)
sigma_e    [upper.tri(sigma_e)]    <- roo_e
sigma_e    [lower.tri(sigma_e)]    <- roo_e
#-----------------
Final.table = array(NA,c(16,7))
rownames(Final.table) = rep(c("n = 10","n = 25","n = 50","n = 100"),4)
```

```
colnames(Final.table) = c("Zelner" , "SF" ,"SK" ,"AS" , "new1" , "new2" , "new3")
#------------------
f = 0
for (rxi in 1:4) {
  for (ni in 1:4) {
    ##Generate X
    sigma_x <- diag(1,nrow = ki)
    sigma_x   [upper.tri(sigma_x)]    <- roo_x[rxi]
    sigma_x   [lower.tri(sigma_x)]    <- roo_x[rxi]
    LX <- list()
    for (ix in 1:m)
      LX [[ix]] <- mvrnorm(n [ni], mu = rep(1,ki), sigma_x)
    X <- diag.block(LX)
    #------------------
    Sim.results = matrix (0,nrow = 1,ncol = 7)
    results.matrix      = matrix (NA,nrow = 1,ncol = 7)
    for (l in 1:L) {
      error <- mvrnorm(n [ni], mu = rep(0,m), sigma_e)
      dim(error) <- c(m * n [ni],1)
      Y <- (X %*% B_vector) + error
      GLS.Z = SUR.estimation (Y,X, sigma_e , B_vector, estimation.type = "GLS")$TMSE
      results.matrix [,1] = GLS.Z
      Ridge.par =   Ridge.parameters   (Y,X,sigma_e ,K)
      for (RP in 1:6)
        results.matrix [,RP + 1] = SUR.estimation   (Y,X, sigma_e , B_vector, estimation.type =
"Ridge",R = Ridge.par[[RP]])$TMSE
       Sim.results = Sim.results + results.matrix
    }
    average = Sim.results / l
    Final.table[ni + f,] <- average
  } #for ni
  f = f + 4
} #for rxi
Final.table
```

Table 7. TMSE values for the different estimators when $m = 3$.

n	Zellner	SF	SK	AS	new 1	new 2	new 3
			$\rho_x = .70$				
10	33.758	3.670	1.939	3.104	1.613	0.593	0.681
25	1.011	0.690	0.618	2.382	0.714	0.390	0.335
50	0.291	0.211	0.204	2.747	0.258	0.199	0.203
100	0.136	0.096	0.098	2.839	0.127	0.108	0.134
			$\rho_x = .80$				
10	75.078	10.887	2.990	2.189	2.025	0.508	0.432
25	1.205	0.808	0.742	2.651	0.858	0.445	0.334
50	0.466	0.318	0.309	2.760	0.395	0.267	0.226
100	0.186	0.127	0.140	1.970	0.171	0.144	0.119
			$\rho_x = .90$				
10	195.774	18.456	4.597	1.777	2.389	0.654	0.514
25	3.726	2.444	1.968	2.003	1.788	0.586	0.362
50	0.921	0.612	0.625	1.802	0.711	0.424	0.240
100	0.430	0.283	0.286	2.405	0.368	0.242	0.142
			$\rho_x = .98$				
10	899.949	91.424	16.424	1.272	3.400	0.514	0.421
25	19.969	13.050	8.136	1.452	3.621	0.471	0.204
50	5.326	3.472	3.244	1.967	2.411	0.454	0.161
100	2.043	1.341	1.351	1.596	1.335	0.455	0.153

In general, Table 7 indicates that TMSE values of all ridge estimators are always less than TMSE values of Zelner's estimator. Moreover, TMSE values of all estimators are decreasing when n is increased. And new 3 estimator is the best estimator because it has minimum TMSE in all simulation situations.

4. Graphical Presentation for Simulation Results

The standard presentation for the simulation results is the tables (as in above). However, in many studies the tables are not readable, so should be present the results by the graphs. Generally, the graphs that used in most studies can be summarized to two types; 2-dimansional (2D) and 3D graphs. In this section, we represent the results of the two applications (in above) using 2D and 3D graphs.

For the first application, we present the standard errors (SE) of OLS and GLS estimates for different n and ρ in bar graphs using the following R-code:

```
#---- Application I: 2D graphs (bar graphs)
var.rows = c(7,8,15,16)
beta0.data =  sqrt (Final.table [var.rows,seq(1,7,2)])
beta1.data = sqrt(Final.table [var.rows,seq(2,8,2)])
data.list = list(beta0.data,beta1.data)
windows(70,45)
par(mfcol = c(2,2))
leg.tx = c("OLS","GLS")
AR.fac = c(.5,.9)
beta = c(0,1)
for (j in 0:1) {
 a = 1;b = 2
 for (i in 1:2) {
  barplot(data.list [[j + 1]][a:b,],beside = TRUE,
   col = c(2,3),legend.text = leg.tx, angle = 65,
   main = bquote(paste("SE","(",hat(beta[.(j)]),")" ,", ",
   rho == .(AR.fac[i]))))
  box()
  a = a + 2; b = b + 2  }}
```

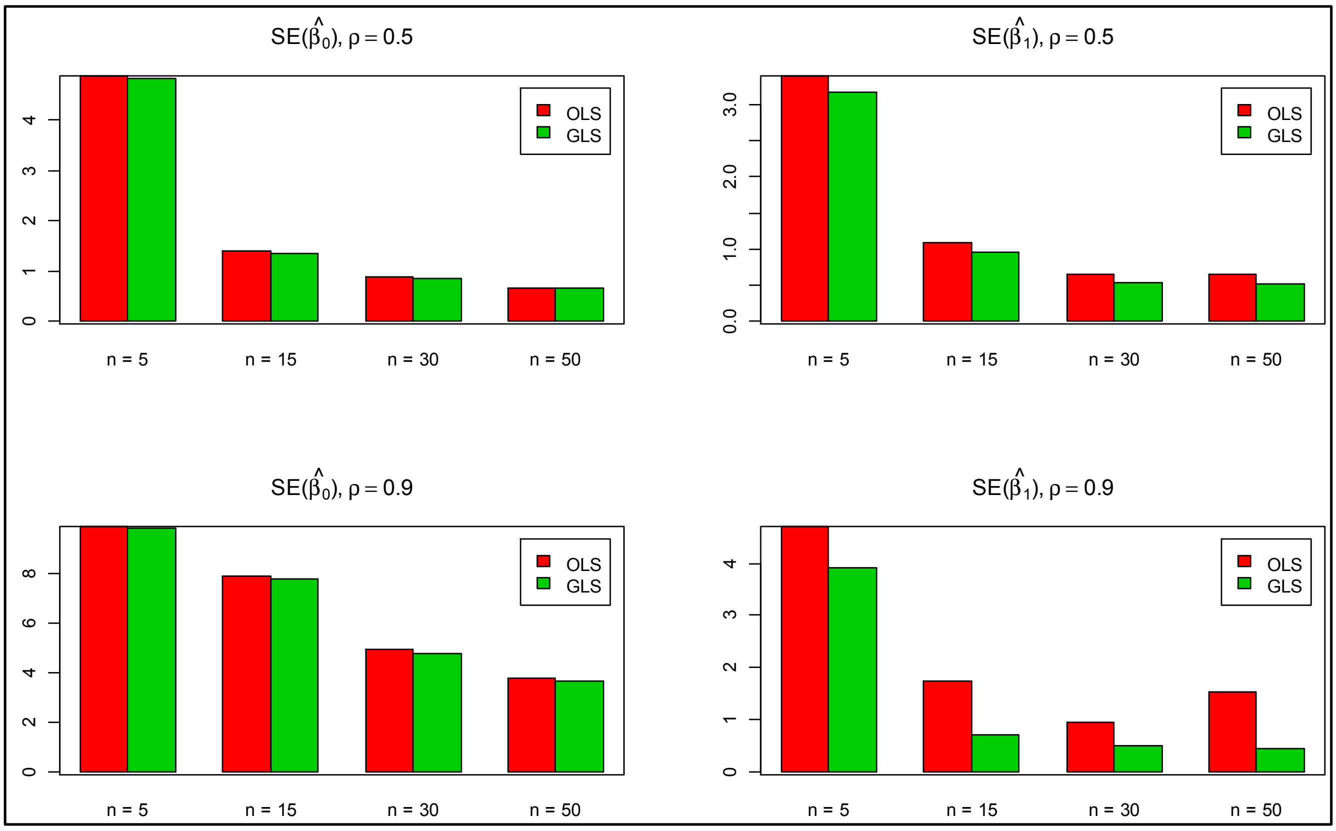

Figure 1. The standard errors of OLS and GLS estimates for different n and ρ when $\sigma_\varepsilon^2 = 5$.

Figure 1 confirms our conclusions from Table 4 and adds an important note: that the gain from applying GLS method is increasing in β_1 than β_0. In other words, SE of GLS estimates is always less than SE of OLS estimates for each of β_0 and β_1, but it is more noted in β_1 than β_0.

For the second application, we present the relative efficiency $[TMSE(\hat{B}_R)/TMSE(\hat{B}_Z)]$ of different ridge estimators for different n and ρ_x in line graphs using the following R-code:

```
#---- Application II: 2D graphs (line graphs)
data = Final.table / Final.table[,1]
data = data[,-c(1,4)]
##----------------------
yrange <- range(data)
yrange[2] <- yrange[2] + mean (yrange) * 0.5
y.values = seq(yrange[1], yrange[2],length = 4)
linetype <- c(1:5)
plotchar <- c(5,8,15,19,22)
colors <- c("black","red", "blue", "green", "DeepPink")
names <- c("SF" ,"SK" ,"new1","new2" , "new3")
##----------------------
windows(70,40)
par(mfrow = c(2,2), mar = c(4, 4, 2, 2) + 0.1)
sig = c(.70, .80,.90,.98)
a = 1; b = 4
for (j in 1:4) {
 plot(n, y.values, type = "n", xlab = "Sample", ylab = "Relative efficiency")
 mtext(bquote(rho[x] == .(sig[j])))
 for (i in 1:5) {
  lines(n, data [a:b,i], type = "b", lwd = 2, lty = linetype[i] , col = colors[i], pch = plotchar[i])
  legend ("top" , names, cex = 1.3, col = colors, horiz = TRUE, pch = plotchar, lty = linetype)
  } ; a = a + 4; b = b + 4 }
```

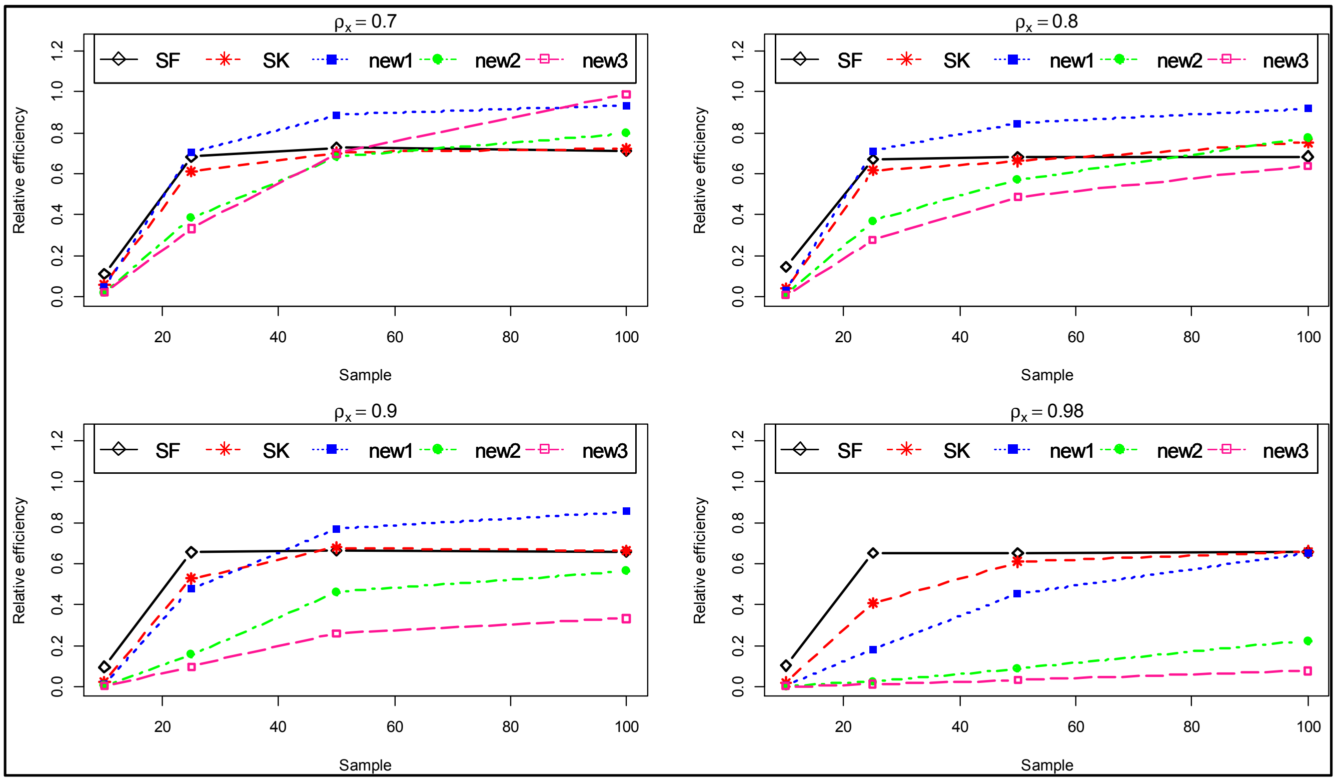

Figure 2. Relative efficiency ratios of different ridge estimators when $m = 3$.

Figure 2 confirms that new 3 estimator is the best estimator for this model, especially when ρ_x increases. Moreover, the efficiency of new 3 estimator is increasing (or the relative efficiency ratio is close to zero) in moderate samples. Alternatively, we can use 3D graphs as another way to present the relative efficiency ratios of different ridge estimators. The R-code is:

```
#---- Application II: 3D graphs (when m = 6)
library (lattice)
data = Final.table / Final.table[,1]
data = data[,-c(1,4,5)]
##----------------------
n = c(10,25,50,100)
Ridge.estimators <- c(rep("SF",16),  rep("SK",16), rep("new2",16) ,rep("new3",16))
rho_x.values = rep(c(.70, .80, .90, .98),4)
gg <- expand.grid(n = n,rho_x = rho_x.values)
my.data.es <- data.frame(gg,Ridge.estimators,TMSE.values = c(data))
##----------------------
windows(50,50)
wireframe(TMSE.values ~ n * rho_x | Ridge.estimators , data = my.data.es,
  scales = list(arrows = FALSE,  cex = 0.7,distance = 0.9 ,col = 1),screen = list(z = 40, x = -60),
  outer = TRUE, col.regions = heat.colors(100), col = rgb(1,5,10,max = 100),
  zoom = .85, panel.aspect = 1, drape = TRUE, main = "", xlab = expression (italic(n)),
  ylab = expression (italic(rho[x])),zlab = "")
```

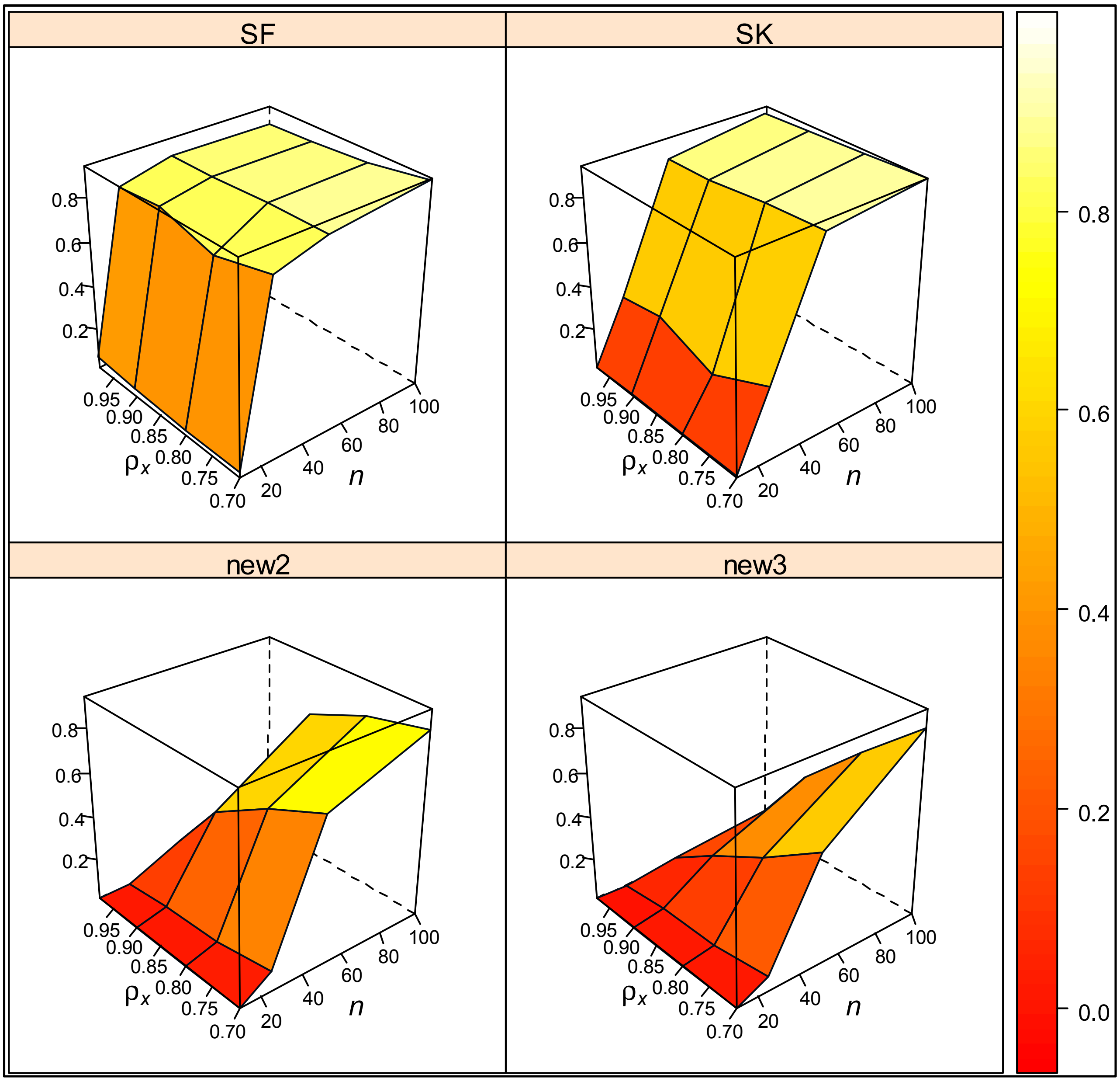

Figure 3. Relative efficiency ratios of different ridge estimators when $m = 6$.

Figure 3 displays the relative efficiency of four ridge estimators under the effect of n and ρ_x together, and it indicates that the efficiency of SF and SK is not improving when n and/or ρ_x are increasing. While the efficiency of new 2 and new 3 is very improving when ρ_x is increasing, but this efficiency is worsened when n is increasing. These conclusions are not clear in Figure 2.

5. Conclusion

In this paper, we proposed a complete algorithm to make professional MCS studies using R. This algorithm is a suitable for creating any simulation study in statistical and econometric models. This algorithm is considered as a practical guide for researchers to conduct their simulation studies in statistics and econometrics. Practically, empirical examples have been presented as applications on our algorithm. These applications proved that this algorithm is very easy and general for different studies even if the objectives of each study are different. Moreover, the graphical presentation methods for simulation results have been discussed. In future work, we publish this algorithm in an R-package so that more researchers will benefit, especially who are not proficient in programming.

References

[1] Abonazel, M. R. (2014). Some estimation methods for dynamic panel data models. PhD thesis. Institute of Statistical Studies and Research. Cairo University.

[2] Abonazel, M. R. (2014). 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.

[3] Abonazel, M. R. (2015). *How to Create a Monte Carlo Simulation Study using R: with Applications on Econometric Models*. Working paper, No. 68708. University Library of Munich, Germany. Online at: https://mpra.ub.uni-muenchen.de/68708

[4] Abonazel, M. R. (2017). Generalized estimators of stationary random-coefficients panel data models: Asymptotic and small sample properties, *Revstat Statistical Journal* (in press). Online at: https://www.ine.pt/revstat/pdf/GENERALIZEDESTIMATORSOFSTATIONARY.pdf

[5] Alkhamisi, M., Khalaf, G., Shukur, G. (2006). Some modifications for choosing ridge parameters. *Communications in Statistics-Theory and Methods*, 35 (11): 2005-2020.

[6] Alkhamisi, M., Shukur, G. (2008). Developing ridge parameters for SUR model. *Communications in Statistics-Theory and Methods*, 37 (4): 544-564.

[7] Barreto, H., Howland, F. (2005). *Introductory econometrics: using Monte Carlo simulation with Microsoft excel*. Cambridge University Press.

[8] Brown, P. J., Zidek, J. V. (1980). Adaptive multivariate ridge regression. *The Annals of Statistics*, 8 (1): 64-74.

[9] Craft, R. K. (2003). Using spreadsheets to conduct Monte Carlo experiments for teaching introductory econometrics. *Southern Economic Journal*, 69 (3): 726-735.

[10] Crawley, M. J. (2012). *The R book*. John Wiley & Sons.

[11] Firinguetti, L. (1997). Ridge regression in the context of a system of seemingly unrelated regression equations. *Journal of Statistical Computation and Simulation*, 56 (2): 145-162.

[12] Gujarati, D. N. (2003) *Basic econometrics*. 4th ed. McGraw-Hill Education.

[13] Gentle, J. E., Härdle, W. K., Mori, Y. (2012). *Handbook of computational statistics: concepts and methods*. Springer Science & Business Media.

[14] Haft, J. (2014). *A* Review *of Basic Monte Carlo Methods*. It is available online at: http://www.reed.edu/economics/parker/s14/312/Fin_Reports/4.pdf

[15] Haitovsky, Y. (1987). On multivariate ridge regression. *Biometrika*, 74 (3): 563-570.

[16] Hoerl, A. E., Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12 (1): 55-67.

[17] Hoerl, A. E., Kennard, R. W. (1970). Ridge regression: applications to nonorthogonal problems. *Technometrics*, 12 (1): 69-82.

[18] Kibria, B. G. (2003). Performance of some new ridge regression estimators. *Communications in Statistics-Simulation and Computation*, 32 (2): 419-435.

[19] 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.

[20] 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. Online at https://mpra.ub.uni-muenchen.de/49768

[21] Robert, C., Casella, G. (2009). *Introducing Monte Carlo Methods with R*. Springer Science & Business Media.

[22] Robert, C., Casella, G. (2013). *Monte Carlo statistical methods*. Springer Science & Business Media.

[23] Saleh, A. M. E., Kibria, B. M. G. (1993). Performance of some new preliminary test ridge regression estimators and their properties. *Communications in Statistics-Theory and Methods*, 22 (10): 2747-2764.

[24] Srivastava, V. K., Giles, D. E. (1987). *Seemingly unrelated regression equations models: estimation and inference* (Vol. 80). CRC Press.

[25] Thomopoulos, N. T. (2012). *Essentials of Monte Carlo Simulation: Statistical Methods for Building Simulation Models*. Springer Science & Business Media.

[26] Vinod, H. D. (1978). A survey of ridge regression and related techniques for improvements over ordinary least squares. *The Review of Economics and Statistics*, 60 (1): 121-131.

[27] 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.

[28] Youssef, A. H., Abonazel, M. R. (2017). Alternative GMM estimators for first-order autoregressive panel model: an improving efficiency approach. *Communications in Statistics-Simulation and Computation* 46 (4): 3112–3128.

[29] 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.

[30] Zellner, A. (1962). An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American statistical Association*, 57 (298): 348-368.