Limited Information Maximum Likelihood

Introduction

The LIML estimator is commonly used to estimate parameters in linear regression models with endogenous regressors and instrumental variables. It is particularly useful when the instrumental variables are weak or when the number of instrumental variables is large.

The LIML estimator aims to address the potential inefficiencies of the Two-Stage Least Squares (2SLS) estimator, especially in scenarios where there are many instrumental variables or when these instruments are weakly correlated with the endogenous regressors. In such cases, 2SLS might produce biased and inconsistent estimates of the coefficients.

```
#importing the mvrnorm function into this environment
mvrnorm <- MASS::mvrnorm

#The function that estimates the LIML estimator,
liml <- function(y, x, z){
  Y <- cbind(y,x)
  Pz <- z %*% solve(t(z) %*% z) %*% t(z)
  Id <- diag(nrow(x))
  lambda_hat <- min(solve(t(Y) %*% (Id - Pz) %*% Y) %*% t(Y) %*% Pz %*%Y) #just using the fo
  Beta <- solve(t(x) %*% Pz %*% x - lambda_hat %*% t(x) %*% (Id - Pz) %*% x) %*% (t(x) %*% P
  return(Beta)
}
```

This function **liml** estimates the LIML (Limited Information Maximum Likelihood) estimator given the dependent variable **y**, the endogenous regressor **x**, and the instrumental variables **z**.

```r
#make a function with different input variables
aprox_distr <- function(R, N, K, B, P){

  #Creating empty containers
  b_tsls <- c()
  b_liml <- c()

  for(i in 1:R){
    #generate the data
    ev <- mvrnorm(N, mu = c(0,0), Sigma = matrix(c(1,.4,.4,1),nrow = 2))
    z <- matrix(rnorm(N * K),nrow = N)
    Pz <- z %*% solve(t(z) %*% z) %*% t(z)
    x <- z %*% P + ev[,2]
    y <- x %*% B + ev[,1]

    #Estimating the TSLS & LIML estimator
    Eb_tsls <- solve(t(x) %*% Pz %*% x) %*% (t(x) %*% Pz %*% y)
    Eb_liml <- liml(y, x, z)

    b_tsls <- c(b_tsls, Eb_tsls)
    b_liml <- c(b_liml, Eb_liml)
  }


  return(list(b_tsls,b_liml))
}
```

This function **aprox_distr** performs a simulation to estimate the distribution of coefficient estimates using both TSLS and LIML estimators.

It takes as input the number of iterations **R**, the number of observations **N**, the number of instrumental variables **K**, the true coefficient **B**, and the instrumental variable coefficients **P**.

For each iteration, it generates synthetic data with random errors using **mvrnorm**, calculates the projection matrix **Pz**, and estimates the coefficients using both TSLS and LIML methods.

It stores the estimated coefficients in two lists **b_tsls** and **b_liml**, and returns them.

```r
#Creating some simulation data
R <- 1000
N <- 100
K <- list(1,1,10,10)
B <- 1 #real Beta
```

```r
P <- list(1,10,rep(.1,10),rep(10,10))



#2 empty lists to store the value
results_tsls <- list()
results_liml <- list()


# seed for replicability
set.seed(54321)
for(i in 1:4){
  results <- aprox_distr(R,N,K[[i]],B,P[[i]])

  #In above variable resultss
  results_tsls[i] <- results[1]
  results_liml[i] <- results[2]

}
```

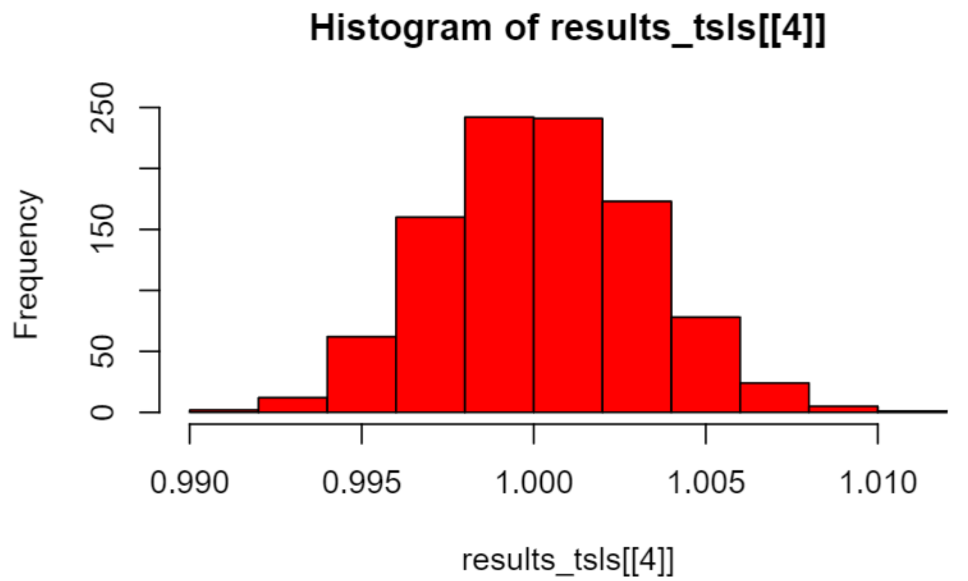
This section sets up parameters for the simulation, such as the number of iterations **R**, the number of observations **N**, and the characteristics of instrumental variables (**K**) and their coefficients (**P**).

It initializes empty lists **results_tsls** and **results_liml** to store the results of the simulation.

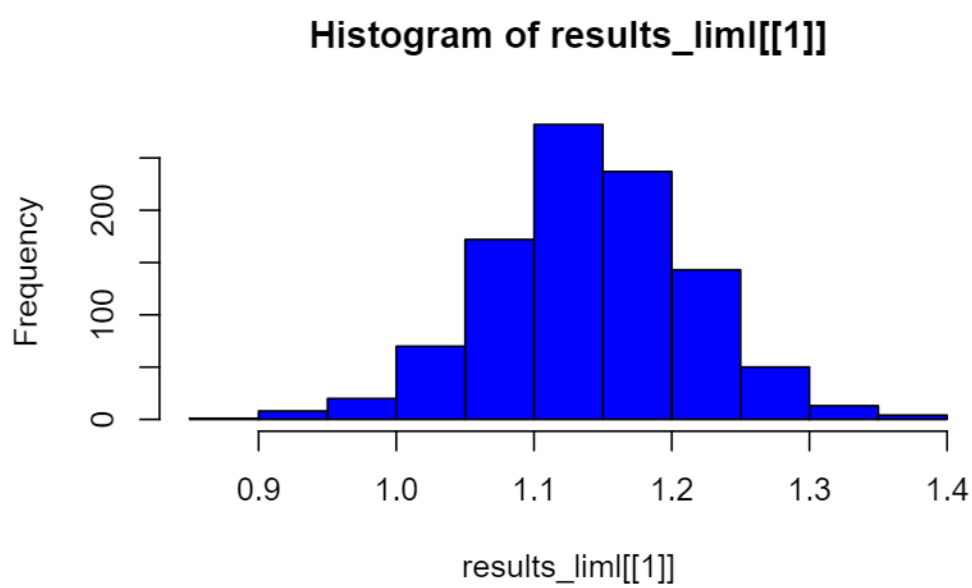
```r
# Set the size of the plot window
options(repr.plot.width=6, repr.plot.height=4)

# Adjusting plot margins
par(mar = c(5, 4, 4, 2) + 0.1)

# Create histograms
hist_tsls <- hist(results_tsls[[4]], col = 'red', plot = TRUE)
```




```
hist_liml <- hist(results_liml[[1]], col = 'blue', plot = TRUE)
```



In conclusion, both the TSLS and LIML estimators provide coefficient estimates centered around 1.00, indicating a consistent estimation of the true coefficient. However, the LIML estimator appears to offer a slightly wider range of estimates compared to TSLS, suggesting a potentially higher variability in the estimates.