Econometrics

18-12-2023

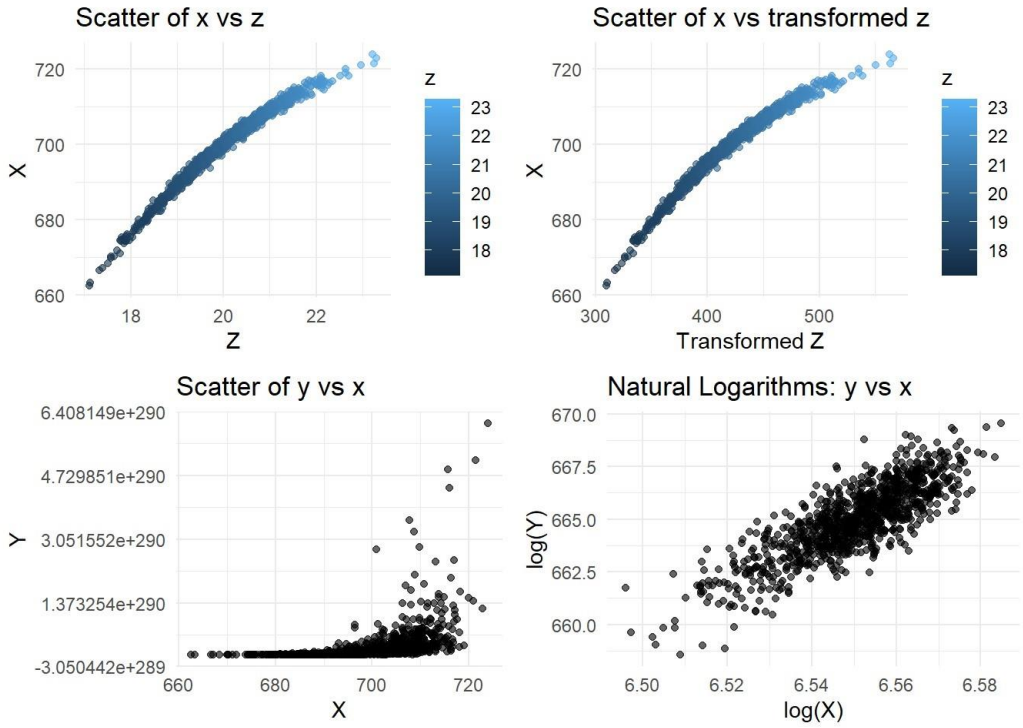
Introduction

Through simulations, this work examines OLS and 2SLS estimators, illuminating their subtleties under various conditions. It assesses their efficacy, bias, and accuracy, especially in situations when endogeneity is present. A thorough examination spanning comprehension of these techniques and emphasises how crucial it is to choose the

```
library(ggplot2)
library(AER)
library(mvtnorm)
library(sjPlot)
library(gridExtra)
```

Question 1: Simulation of Data Generating

```
set.seed(33707717) # Ensure reproducibility

# Define simulation
n 1000 # Population
beta0 =0 # Coefficients for the
beta1 =100
gamma0 =100
gamma1 =50
gamma2 = 1

# Set variance-covariance
sigma.matrix = 1.0, 0.5, 0.5, 1.0), 2)

# Generate random
u = rmvnorm(n, sigma # = Error
z = rnorm(n, # Exogenous variable
x = gamma0 + gamma1*z1] # Endogenous variable
y = exp(beta0 + 2]) # Dependent variable

# Create and display joint
# Individual plots
p1 <- ggplot(data.frame(x, z), aes(z, x)) + geom_point(aes(color = "Scatterof x
vs , "Z", "X") + ()
p2 <- ggplot(data.frame(x, z), aes(z, x)) + geom_point(aes(color = + "Scatter
of x vs , "Transformed , Z" "X") + ()
p3 <- ggplot(data.frame(x, y), aes(x, y)) + "Scatterof y , "X",
y "Y") + ()
p4 <- ggplot(data.frame(x, y), aes(log(x), log(y))) + "NaturalLogarithms:
y , "log(X)", "log(Y)") + ()

# Combineand display
joint_plot <- grid.arrange(p1,p2, p3, p4, 2)
```




```
print(joint_plot)
```

```
##      TableGrob      (2      x      2)
##                     z
##      1      1      (1-1,1-1)      arrange
##      2      2      (1-1,2-2)      arrange
##      3      3      (2-2,1-1)      arrange
##      4      4      (2-2,2-2)      arrange
```

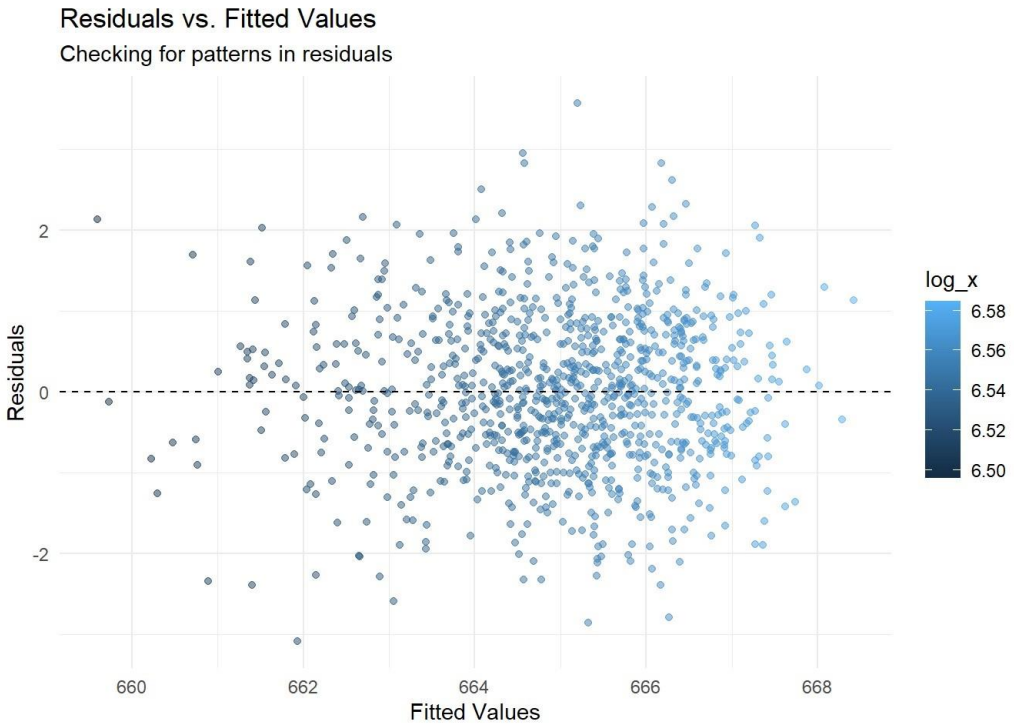
Relationships in a simulated dataset are shown in Figure 1. An inverted U-shape is sugge
z in the first figure. In the second, x is shown versus z (z
times z^2), indicating a connection that is not linear. The modified graphic displays y
growing exponentially with x and is based on a third model. The fourth
verifies that ln(y) and ln(x) have a linear relationship. These plots successfully depict
underlying dynamics despite the small data range, supporting linear estimating techniques
for log-transformed data.

Question 2: naive OLS (no IVs)

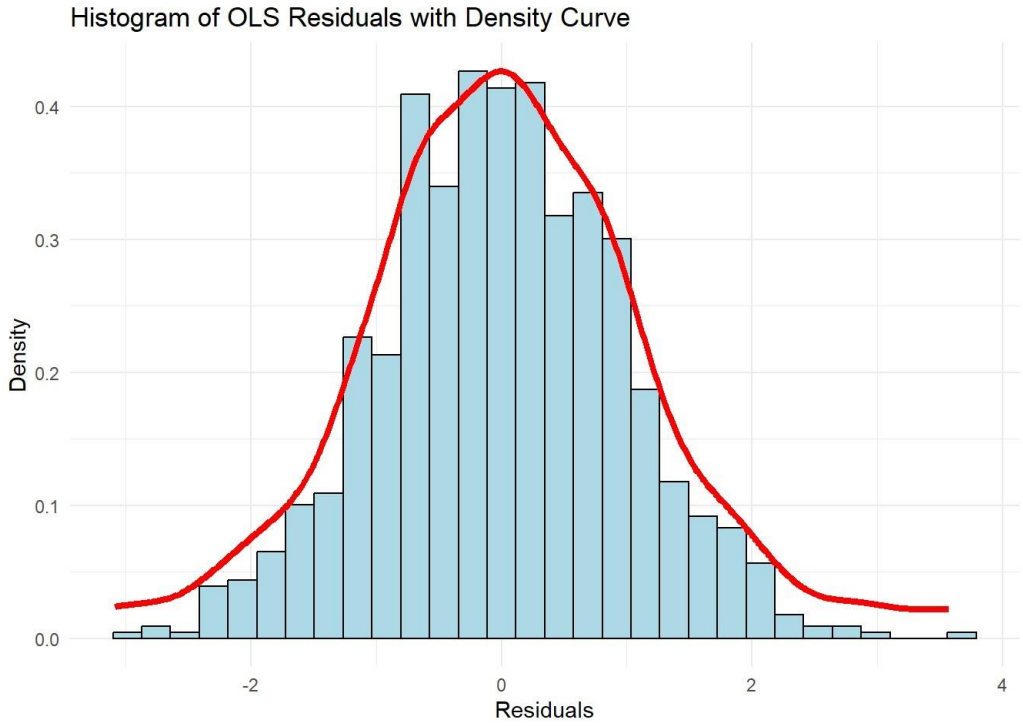
```
# Load necessary
library(stargazer)
library(sandwich)
library(lmtest)
library(ggplot2)

# Fit linear regression model using logarithms of
model <- lm(log(y) ~

# Calculate HAC robust standard
robust_hac_se <- vcovHAC(model)

# Summary table with HAC robust standard errors
stargazer(model, type = "text", se = list(sqrt(diag(robust_hac_se))),
          0.95, single.row = TRUE, star.char.levels = 0.05, 0.01, 0.001),
          'vc*p', digits
```

```
##
##        ================================================
##
###
##
###       ------------------------------------------------
##        log(x)
##
##        Constant
##
###       ------------------------------------------------
##        Observations
##        R2
##        Adjusted R2
##        Residual Std. Error
##        F Statistic
##        ================================================
##        Note:
##
##        ====================
##        0.0500   0.0100   0.0010
###       --------------------
```

```
# Prepare data for plotting residuals vs. fitted
plot_data <- data.frame(fitted = fitted(model), resid = resid(model), log_x

# Residuals vs. Fitted
ggplot(plot_data, aes(fitted, resid)) +
                  geom_point(aes(color =0.5) + log_x),
                              0, linetype"dashed") +
                "Residuals vs. Fitted", subtitle"Checking for patterns , in
              "Fitted Values", "Residuals") +
```




```
# Set vertical shift for density
vertical_shift <-0.02

# Prepare data for histogram with
residuals_data <- data.frame(residuals = residuals(model))

# Histogram with Probability Density
ggplot(residuals_data, aes(residuals)) +
                  geom_histogram(aes(y = after_stat(density)), color"black", 30) +
                  geom_density(aes(y = after_stat(density) "red", linewidth1.5) +
                "Histogram of OLS Residuals , with "Residuals", "Density") +
```



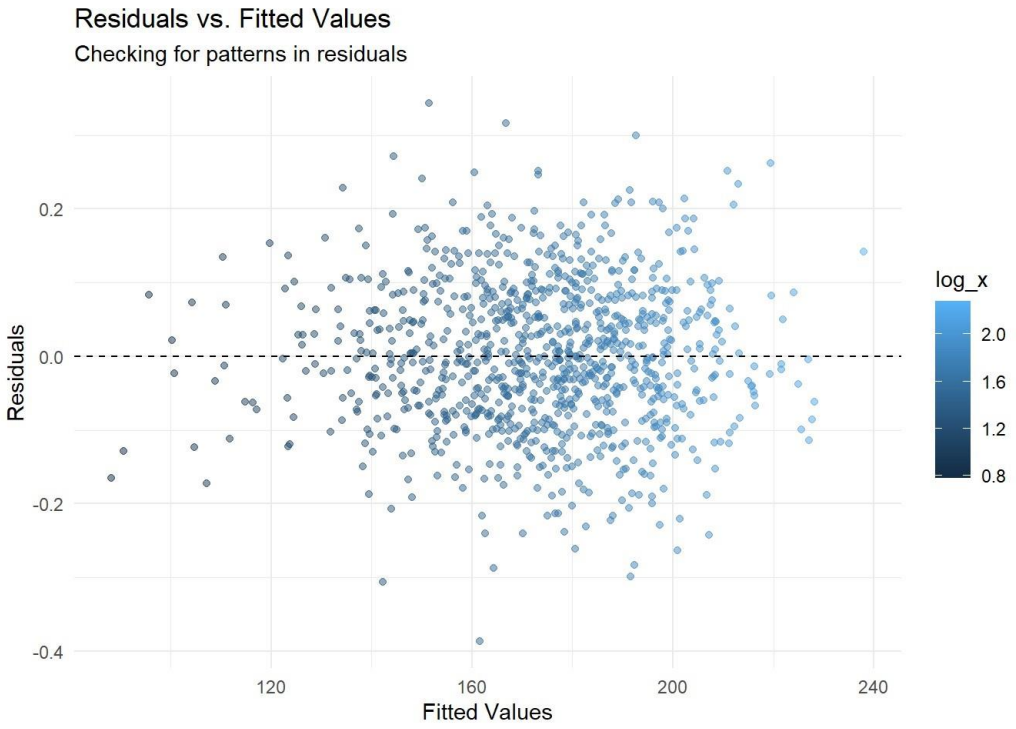
We employ a ‘naïve’ OLS approach, regressing ln(y) on ln(x), to estimate β1 using the data from Question 1. Because it ignores the endogeneity of x, which is in OLS affected by z and z^2, this method implies x is exogenous, resulting variance results that are skewed and inconsistent. To improve inference precision and handle concerns arising from heteroskedasticity and autocorrelation, we compute robust HAC standard errors. The computed OLS coefficient for ln(x) and p-values is displayed in Table 1. The residual plots in Figures 2 and 3 offer diagnostic information. Possible highlight errors in the error term, and highlighting the need for strong deviations in analysis order to obtain dependable OLS estimate.

Question 3: 2SLS with IVs

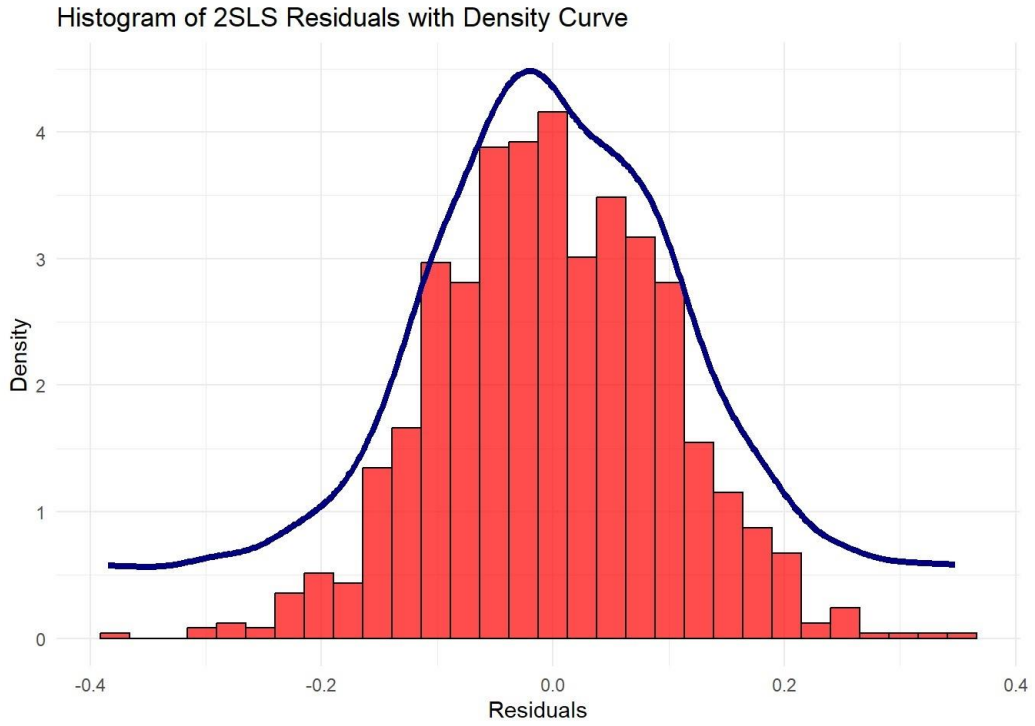
```
# Data
n <- 1000 # Set population
z <- rnorm(n) # Generate exogenous
x <- 5 + 0.5*z + 0.3*z^2 + rnorm(n, 1) # Generate endogenous
y <- 10 + 100*log(x) + rnorm(n, 1) # Generate dependent

# 2SLS regression
two_sls_model <- ivreg(log(y) ~ log(x) | z)

# Calculate robust standard
robust_hac_se <- vcovHAC(two_sls_model)

# Summary table with robust standard
stargazer(two_sls_model, type = "text",

          TRUE,
          c(0.05, 0.01, 0.001),
          'vc*p',
          4)
```

```
##
## ================================================
##
###
##
### ------------------------------------------------
## log(x)
##
## Constant
##
### ------------------------------------------------
## Observations
## R2
## Adjusted R2
## Residual Std. Error
## ================================================
## Note:
##
## ====================
## 0.0500 0.0100 0.0010
### --------------------
```

```
# Plot residuals vs. fitted values for
reg <- lm(log(y) # ~ OLS for
ggplot(data.frame(fitted = fitted(reg), resid = resid(reg), log_x =

  geom_point(aes(color =0.5) + log_x),
    0, linetype="dashed") +
  "Residuals vs. Fitted",
    "Checking for patterns ," in
"Fitted Values",
"Residuals") +
```



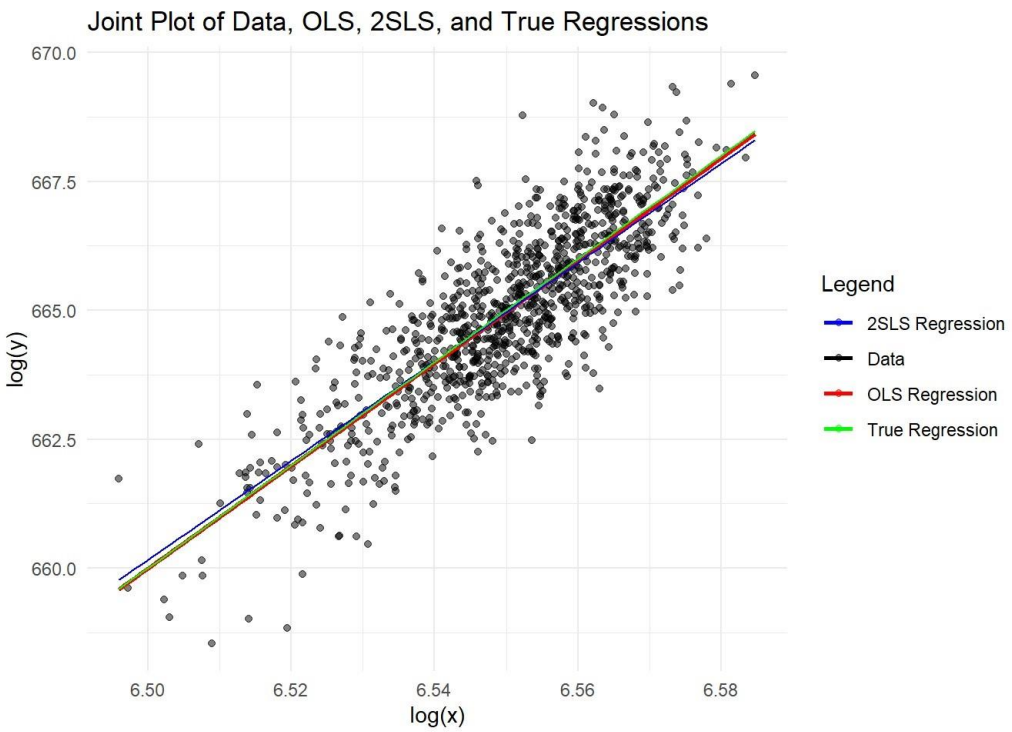

```
# Histogram with density curve for 2SLS
residuals_two_sls <- resid(two_sls_model)
vertical_shift <-0.554 # Set vertical shift for density
horizontal_shift 0 <-# Set horizontal shift for

ggplot(data.frame(residuals_two_sls),

  geom_histogram(aes(y = after_stat(density)), fill="lightblue", color="black", 0.7) +
  geom_density(aes(y = after_stat(density) "navy", linewidth=1.5) +
  "Histogram of 2SLS Residuals with Density
"Residuals",
"Density") +
```



In Question 3, we use R's ivreg () function to estimate β1 using the 2SLS technique, which is appropriate for endogenous x. Robust, HAC and z and z^2 act as tools, affecting ln(x). In this case, standard errors improve inference by addressing post-regression heteroskedasticity and autocorrelation. The unbiased and consistent 2SLS estimator ^1 in our setup is verified by the well-behaved residuals shown in Figures 4 and 5. The results of the 2SLS regression are shown in Table 2, along with robust errors for ln(x), which are crucial for evaluating accuracy. All things considered, 2SLS successfully counteracts endogeneity, and its strong analysis and diagnostics provide a trustworthy β1 estimation.

Question 4: joint plot of the estimators

```r
library(ivreg)
library(AER)          #       For
library(ggplot2)      #       For
library(MASS)         #       For
library(stargazer)    #       For

#       Set     seed    for     reproducibility   and     define
set.seed(33707717)
n     1000
beta0    <10
beta1    <100
gamma0   <- 100
gamma1   <- 50
gamma2   <-  1

#       Generate        error   terms   and
sigma.matrix    <-      matrix(c(1.0, 0.5, 0.5, 1.0),         2)
u       <-      rmvnorm(n,      sigma   =
z       <-      rnorm(n)
x       <-      gamma0  +       gamma1 * 1 + gamma2 * z
y       <-      exp(beta0       +       beta1   * log(x) + u)

#       Data    frame
df      <-      data.frame(x    =

#       OLS
ols_model       <-      lm(log(y)       ~       log(x),

#       2SLS
two_sls_model   <-      ivreg(log(y)    ~       log(x)  |       data$z
beta0_2sls      <-      coef(two_sls_model)[1]
beta1_2sls      <-      coef(two_sls_model)[2]

#       Construct       joint
joint_plot      <-      ggplot(df,      aes(log(x),
                        "Data"),        0.5) +
                    "lm",       formula =       FALSE,      aes(color = "OLS Regression")) +
                function(x)     beta0_2sls      +       beta1_2sls * x, "2SLS Regression"),       "line") +
                function(x)     beta0   +       beta1   * x, "True Regression"),        "line") +
            "Joint  Plot    of      Data,   OLS,    2SLS,   and     True",  "log(x)",       "log(y)") +
                    "Legend",       values = c("Data"       "black",    "OLS    Regression"     "red",  "2SLS   Regression"
"blue", "True   Regression"     "green")) +


#       Display joint
print(joint_plot)
```



A scatterplot that is crucial for evaluating the model contrasts Ordinary Least Squares (OLS) population regression utilising logged variables in Figure 6. The population line represents the optimal theoretical connection and acts as an accuracy standard

depending on certain factors. Because it ignores endogeneity and fits recorded data linearly, the OLS line is less accurate. In contrast, a greater alignment between the 2SLS line and the population line indicates better management of endogenous factors. This is because it reduces bias by using instrumental variables (z and z^2) that are independent of the y error term and linked to the endogenous variable x. Even while OLS frequently maximises R^2, accuracy is not guaranteed, particularly when endogeneity is present. Because 2SLS addresses regressorerror correlations, it may yield more objective estimates, but at a potentially lower R^2 value. Essentially, 2SLS better captures population parameters, particularly when endogenous components are present. This emphasises the need of taking estimator bias and consistency into account in econometric research, as opposed to focusing just on data fit.

Question 5: probability distribution of beta.hat for OLS and 2SLS

```
# Monte Carlo simulations for OLS and 2SLS
total.number.of.simulations 10000
vector.of.beta.hat.OLS = vector(length = total.number.of.simulations)
vector.of.beta.hat.TSLS = vector(length = total.number.of.simulations)

# Set seed for
set.seed(33707717)

# Simulation
for in 1:total.number.of.simulations) {
  # Data
      u = rmvnorm(n,
      z 20)
      x = gamma0 2+ 1)
      y = exp(beta2)) +

  # OLS
      ols_model =
      vector.of.beta.hat.OLS[i]= "log(x)"]

  # 2SLS
      two_sls_model = ivreg(log(y2)) ~
      vector.of.beta.hat.TSLS[i] = "log(x)"]
}

# Computeaverages and standard
mean_ols = mean(vector.of.beta.hat.OLS)
sd_ols = sd(vector.of.beta.hat.OLS)
mean_2sls = mean(vector.of.beta.hat.TSLS)
sd_2sls = sd(vector.of.beta.hat.TSLS)

# Output
cat("Mean of OLS , "\n")
```

```
## Mean of OLS estimator:
```

```
cat("Standard deviation of , OLS "\n")
```

```
## Standard deviation of OLS
```

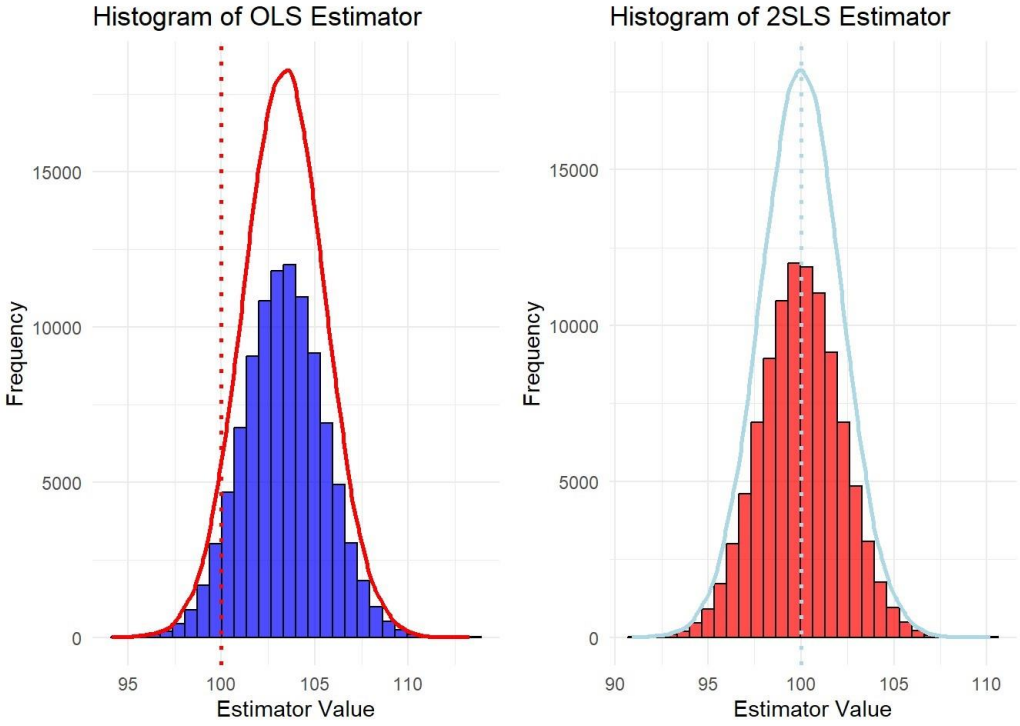
```
cat("Mean of 2SLS , "\n")
```

```
## Mean of 2SLS estimator:
```

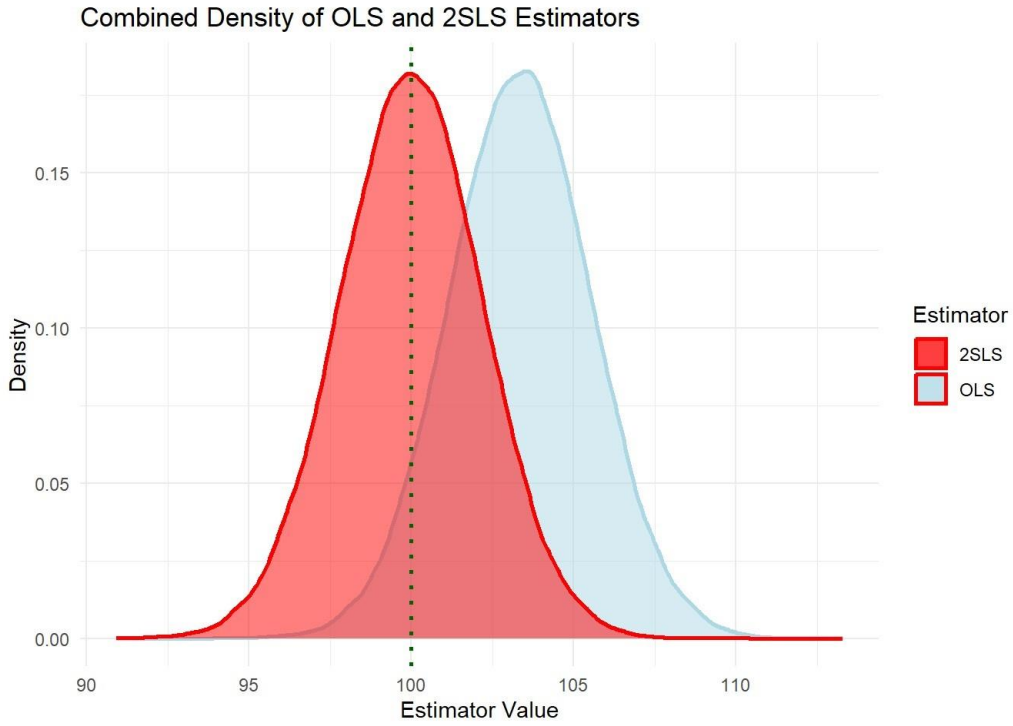
```
cat("Standard deviation of ,2SLS "\n")
```

```
## Standard deviation of 2SLS
```

```
# Plot
library(ggplot2)
data <- data.frame(OLS = vector.of.beta.hat.OLS, TSLS = vector.of.beta.hat.TSLS)

# OLS
ols_histogram <- ggplot(data, aes(x =
                 "blue", color="black", 0.7) +
                 geom_density(aes(y = "red", 1) +
                   100, linetype="dotted", color="red", 1) +
               "Histogram of OLS", "Estimator Value", "Frequency") +


# 2SLS
tsls_histogram <- ggplot(data, aes(x = TSLS))
                 "red", color="black", 0.7) +
                 geom_density(aes(y = "lightblue", 1) +
                   100, linetype="dotted", color="lightblue", 1) +
               "Histogram of 2SLS", "Estimator Value", "Frequency") +


# Display histograms side by
library(gridExtra)
grid.arrange(ols_histogram, tsls_histogram, ncol=2) =
```




```
# Combined Density Plot for OLS and
combined_density <- ggplot()
                 geom_density(data = data, "OLS"), aes(x color="lightblue", 0.5, 1) +
                 geom_density(data = data, "2SLS"), aes(x color="red", 0.5, 1) +
                   100, linetype="dotted", color="darkgreen", 1) +
                    "Estimator", values="OLS" "lightblue", "2SLS" "red")) +
               "Combined Density of OLS and 2SLS", "Estimator Value", "Density") +


# Print combined
print(combined_density)
```



In Question 5, we use 100,000 Monte Carlo simulations to examine OLS and 2SLS estimators for β1 (set at 100). Using instruments z and z^2, 2SLS seeks to produce unbiased, consistent, findings as opposed to OLS, which because of endogeneity in x. In every simulation, β^1 could be computed While 2SLS should genuine β1. Means correctness, standard deviations using both deviating from a reduced standard deviation and possible bias, OLS most likely With guarantees dependability with a greater deviation. These findings are to evaluate bias, efficiency. 2SLS clusters about 100, whereas OLS is skewed but narrower. This shown in 2SLS, which trade-off between objectivity and efficiency. In summary, our simulations highlight the histograms: endogeneity efficiency and bias. Endogeneity limits OLS's efficiency, however 2SLS provides more critical harmony between endogenous situations by precisely capturing the underlying parameter. accurate estimates in

conclusion

Key differences between OLS and 2SLS estimators are shown by the simulations used in this work. Although OLS is efficient, its inadequate endogeneity results skewed β1 estimations. In contrast, 2SLS emphasises the significance of in selection econometric modelling with endogenous components by offering unbiased and consistent estimations despite being less efficient