Long, Ervin HCSE Simulation Replication

AUTHOR
Elise Dixon

Introduction

In their seminal paper, "Correcting for Heteroscedasticity with Heteroscedasticity Consistent Standard Errors in the Linear Regression Model: Small Sample Considerations," Long and Ervin (2000) reviewed and compared several commonly-used standard errors in OLS. Redo their simulation and replicate Figures 1–4.

Long and Ervin's (2000) goal in this work was to assess the performance of several heteroskedasticity consistent covariance matrices (HCCMs), denoted H0-H3, and provide further evidence to existing literature that HC3 arguably yields the most desirable properties as a HCCM among proposed estimators. Heteroskedastic models rely on asymptotic inference which require large samples to validate the underlying limiting theory toward robust application of normal methods. However, Long and Ervin show that HC3 is a highly versatile covariance estimator which performs well in small samples and for data with both heteroscedastic and homoscedastic errors.

Data Simulation

Long and Ervin (2000) assume the standard linear regression model

$$y_i = \beta_1 + \beta_2 x_{i2} + \ldots + \beta_K x_{iK} + \epsilon_i \qquad (1)$$

with classic notation and assumptions of a standard linear regression, including:

1. Linearity of the relationship between the response and predictors
2. Lack of collinearity between predictors
3. Expectation of error terms is 0
4. Homoscedasticity of error terms (a primary assumption to be challenged throughout the analysis)
5. Uncorrelated error terms

To simulate the underlying population used to examine the CM estimates, Long and Ervin simulated explanatory variables produced from a diverse selection of underlying distributions to mimic real-world, cross-sectional data, which is collected from many subjects at a single point in time, as opposed to a longitudinal or time-series. Independant variables are denoted $\delta_i,\ i = 1, \ldots, 5$; the δ_i's were then combined to construct four explanatory variables, $x_i,\ i = 1, \ldots, 5$ and x_D denotes a dummy variable for $x_2 > 1.6$ vs $x_2 \leq 1.6$. Likewise, several homoscedastic error structures were implemented with various choice distributions toward the same goal.

The error terms were combined with the data to form the following model:

$$y_i = 1 + 1x_{1i} + 1x_{2i} + 1x_{3i} + 0x_{4i} + \tau\epsilon_i \qquad (2)$$

where τ can be varied to adjust the R^2 of the regression model.

Steps 1-3: These steps of the simulation involve generating the population data as outlined above with true coefficients corresponding to model (2). The population must be adjustable to the precise model and error structure specifications for each figure.

```
# Population simulation
# 1: 100,000 observations for explanatory variables with varying properties
library(sandwich)
library(lmtest)

set.seed(23456)

# 100000 observations for independent variables
n = 100000
d1 = runif(n, 0, 1)
d2 = rnorm(n, 0, 1)
d3 = rchisq(n, 1)
d4 = rnorm(n, 0, 1)
d5 = runif(n, 0, 1)

# Combinations of di's for predictors
x1 = 1+d1
x2 = 3*d1 + 0.6*d2
x3 = 2*d1 + 0.6*d3
x4 = 0.1*d1 + 0.9*d3 - 0.8*d4 + 4*d5
Xpop = data.frame(x1, x2, x3, x4)

# 2: Generate error terms

# Homoscedastic error
#e_X = rchisq(n, 5)
#e_X = e_X - mean(e_X)

# Adjusting tau for chosen error structure based on Rsq

tausel = function(X, err, targetR2, betas = c(1, 1, 1, 0)){

  yhat = as.matrix(X) %*% betas
  tauseq = seq(0.01, 2, length.out = 1000)

  R2vals = sapply(tauseq, function(taus){
    y = as.numeric(yhat + taus*err)
    summary(lm(y~x1+x2+x3+x4, data = X))$r.squared
  })

  best_tau = tauseq[which.min(abs(R2vals - targetR2))]

  return(best_tau)
```

```
}

# 3: Construct dependent variable based on error structure

# Function to generate dependent variable y based on error
# structure selected
popgen = function(structure){

  # Homoscedastic error
  e_X = rchisq(n, 5)
  e_X = e_X - mean(e_X)

  # Heteroscedastic error
  if(structure == 0){
    e = e_X
  } else if(structure == 2){
    e = sqrt(pmax(Xpop$x3+1.6, 0))*e_X
  } else if(structure == 3){
    e = sqrt(pmax(Xpop$x3, 0))*sqrt(pmax(Xpop$x4+2.5, 0))*e_X
  }

  # Adjust tau, construct dependent variable
  tau = tausel(Xpop, e, 0.4)
  y = with(Xpop, 1+x1+x2+x3+tau*e)

  # Return population with noise, true betas
  list(pop = data.frame(y, Xpop),
       t = tau,
       pop_betas = c(1, 1, 1, 1, 0))
}
```

Step 4: Using Monte Carlo simulations to demonstrate the evolution of each covariance matrix estimator at varying sample size after generating population data specific to the type of error structure being studied, randomly sample from the population without replacement, conduct linear regressions for each sample size and Monte Carlo iteration, and perform relevant size and power hypothesis tests for each covariance matrix type.

Size: The size of a test is the probability of making a type I error, or falsely rejecting the correct null hypothesis. Size plots show the proportion of times the true null hypothesis $H_0 : \beta_k = \beta_k^*$, where β_k^* is the true parameter determined by population simulation, is rejected for each covariance estimator.

Power: The power of a test is the probability of correctly rejecting the false null hypothesis. Power plots display the power of hypothesis tests at varying sample sizes for the false null hypothesis $H_0 : \beta_k = 0$.

```
# 4: Randomly sample w/o replacement, estimate regression, and conduct hypothesis test for 1000 r

# Function to calculate CM by type
CM = function(model, type = "HC0"){
  if(type == "OLS"){
    return(vcov(model))
```

```
  } else {return(vcovHC(model, type = type))}
}

# Hypothesis testing function
HT = function(model, k, cm_type = "HC0", truebeta, bhatnum){

  # Store estimated betas and CMS
  b_hats = coef(model)
  cov_mtx = CM(model, type = cm_type)

  # Calculate test statistic t and corresponding p value
  se = sqrt(cov_mtx[bhatnum + 1, bhatnum + 1])       #Std err for t
  tstat = (b_hats[bhatnum+1]-truebeta)/se
  pval = 2*pt(-abs(tstat), k-4)

  # Return boolean for proportion rejected in simulation step
  reject = (pval < 0.05)
  return(reject)
}

# Simulation function for empirical size and power of relevant pop

simulation = function(population, bnum){

  # Initialize preliminaries to store results for each CM
  sampsize = c(25, 50, 100, 250, 500, 1000)
  N = 1000                    #Number of Monte Carlo iterations

  # Columns have each CM with row-wise obs for each sample size
  testsize = testpow = matrix(NA, nrow = length(sampsize), ncol = 5)
  colnames(testsize) = colnames(testpow) = c("OLS", "HC0", "HC1",
                                             "HC2", "HC3")
  # Call popgen, save population and true betas
  relpop = population$pop
  b_star = population$pop_betas


  # Conduct regression analysis, size and power tests
  for(i in 1:length(sampsize)){

    n = sampsize[i]       # Implement inner loop for each size

    # Initialize structures for HT results on each iteration
    size_reject = power_reject = rep(0, ncol(testsize))

    # 1000 MCMC iterations for each samp size
    for(j in 1:N){

      # Sample from population
      samp = relpop[sample(1:nrow(relpop), n, replace = FALSE),]
      modl = lm(y~x1+x2+x3+x4, data=samp)     #LM from sample
```

```
      # Size test results
      size_reject = size_reject + sapply(
        c("OLS", "HC0", "HC1", "HC2", "HC3"), function(type){
          HT(modl, n, cm_type = type,
             truebeta = b_star[bnum + 1], bhatnum = bnum)
        })

      # Power test results
      power_reject = power_reject + sapply(
        c("OLS", "HC0", "HC1", "HC2", "HC3"), function(type){
          HT(modl, n, cm_type = type, truebeta = 0,
             bhatnum = bnum)
        })
    }

    # Proportion rejected for each sample size and CM
    testsize[i, ] = size_reject/N
    testpow[i, ] = power_reject/N

  }
  list(size = testsize, power = testpow)
}
```

Figure 1

Long and Ervin (2000) compare the four proposed versions of HCCMs to the OLSCM toward assessing their comparative performance with homoscedastic χ^2_5 error structures at varying sample sizes.

Figure 1 shows how the size and power of hypothesis tests vary with sample sizes from $n = 1$ through $n = 1000$ for each version of covariance matrix type.

Size: Plot 1.1 shows the proportion of times the true null hypothesis $H_0 : \beta_3 = \beta_3^*$, where β_3^* is the true parameter determined by population simulation ($\beta_3^* = 1$), is rejected for each covariance estimator.

Power: Plot 1.2 displays the power of tests at varying sample sizes for the false null hypothesis $H_0 : \beta_3 = 0$.

Simulation:

```
# Homoscedastic error structure 0 for Figure1

# Generate population
p1 = popgen(0)
#p1$t              # Optional: check tau
```

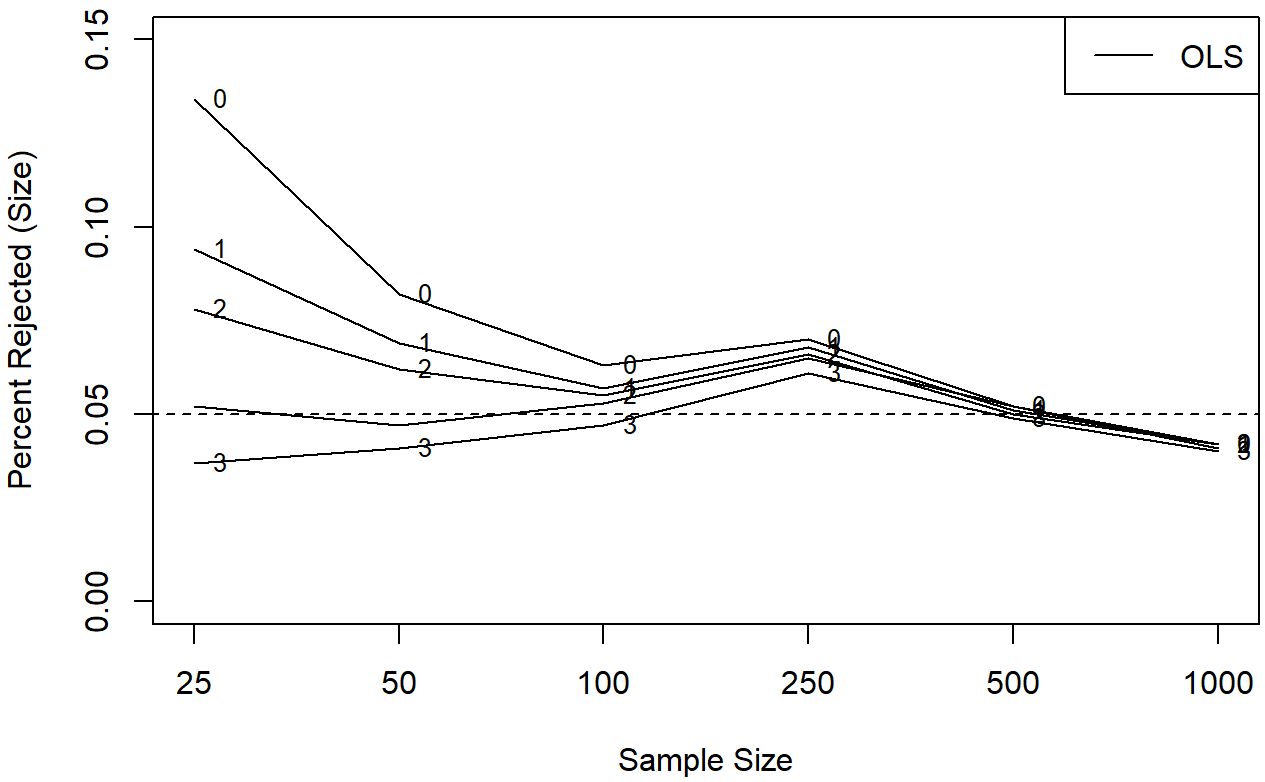
```r
# Conduct hypothesis tests
fig1 = simulation(p1,
                  # sampsize, N,
                  3)

# Plot size results
sampsize = c(25, 50, 100, 250, 500, 1000)
sizetic = c(seq(1:length(sampsize)))

matplot(sizetic, fig1$size, type = "l", lty = 1,
        col = "black",
        xlab = "Sample Size",
        ylab = "Percent Rejected (Size)",
        xaxt = "n", ylim = c(0, 0.15))
matlines(c(0, 1000), c(0.05,0.05), lty = 2)
axis(1, at = sizetic, labels = sampsize)

# Label lines
for (i in 0:3) {
  text(sizetic, fig1$size[,i+2], labels = i, pos = 4, cex = 0.8)
}

legend("topright", legend = "OLS", lty = 1, col = "black")
```



```r
# Plot power results
matplot(sizetic, fig1$power, type = "l", lty = 1,
        col = "black",
        xlab = "Sample Size",
        ylab = "Percent Rejected (Power)",
        xaxt = "n", ylim = range(fig1$power))
abline(h = 0.05, lty = 2)
axis(1, at = sizetic, labels = sampsize)

# Label lines
for (i in 0:3) {
  text(sizetic, fig1$power[,i+2], labels = i, pos = 4, cex = 0.8)
}

legend("topright", legend = "OLS", lty = 1, col = "black")
```

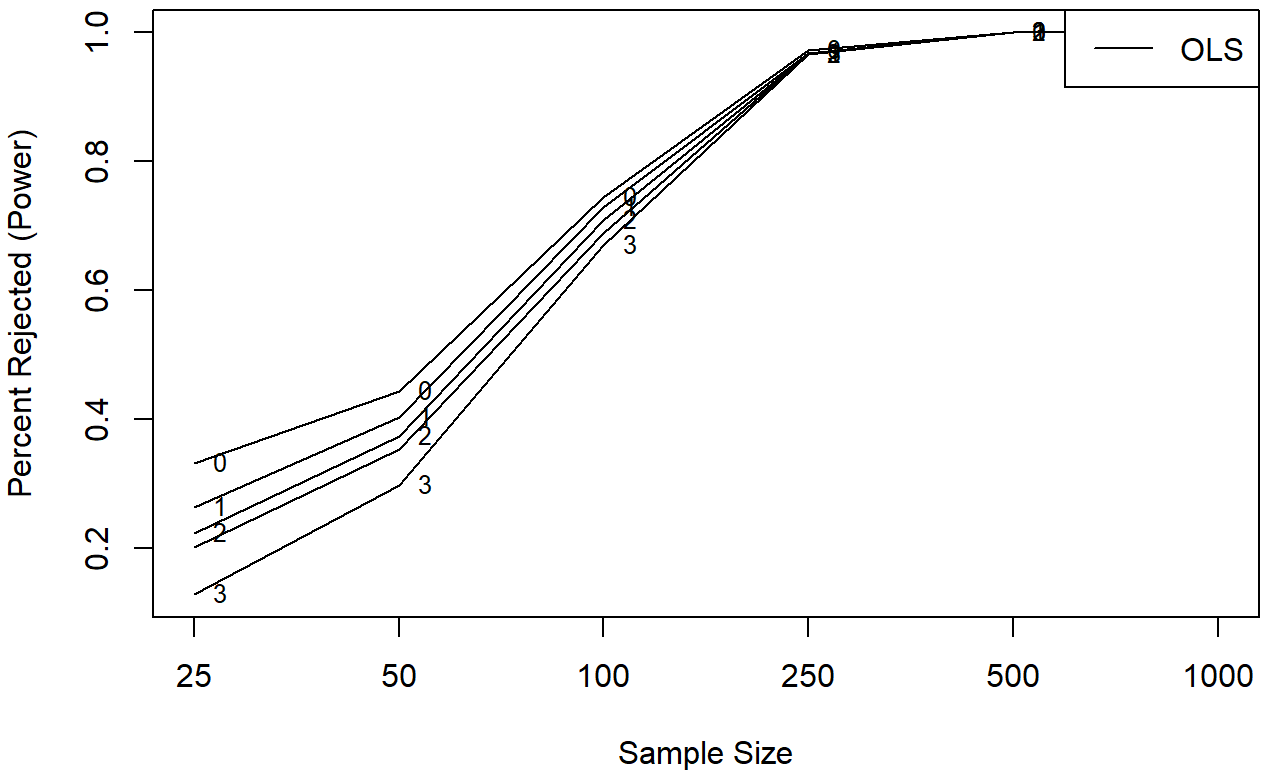


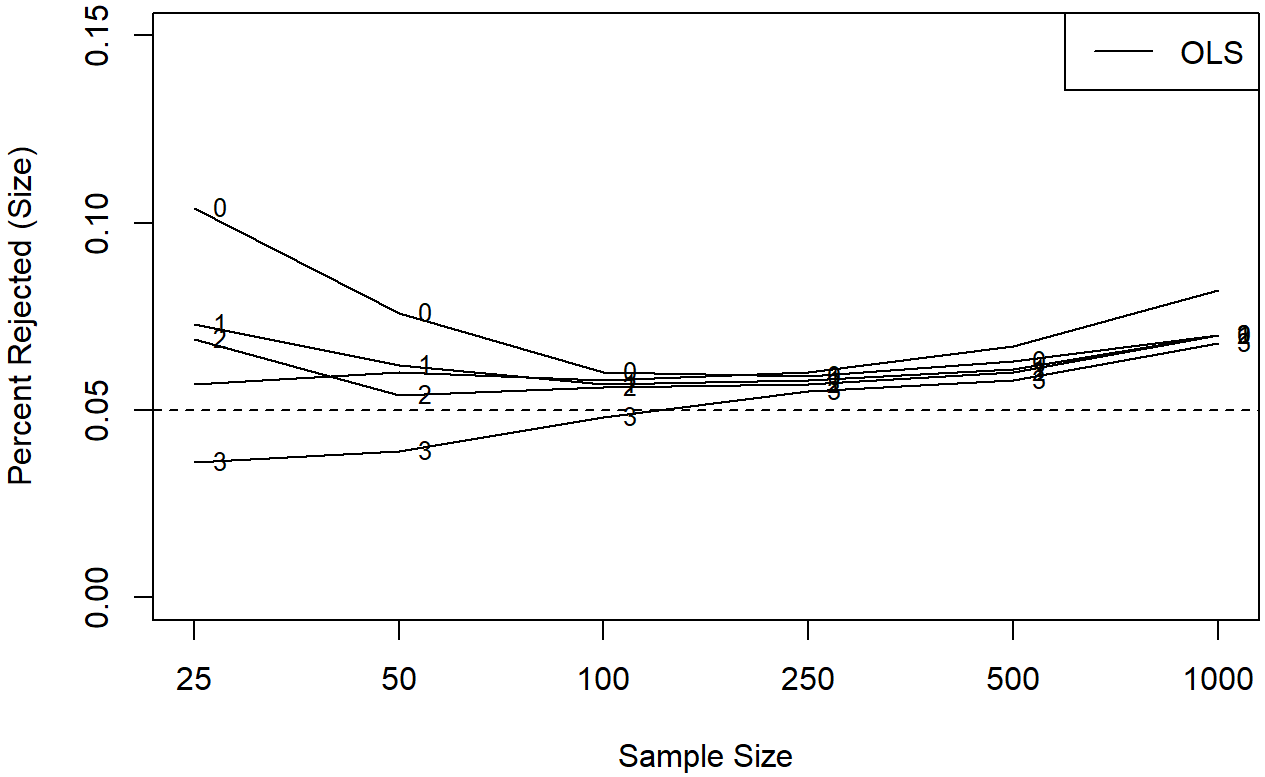
Figure 2

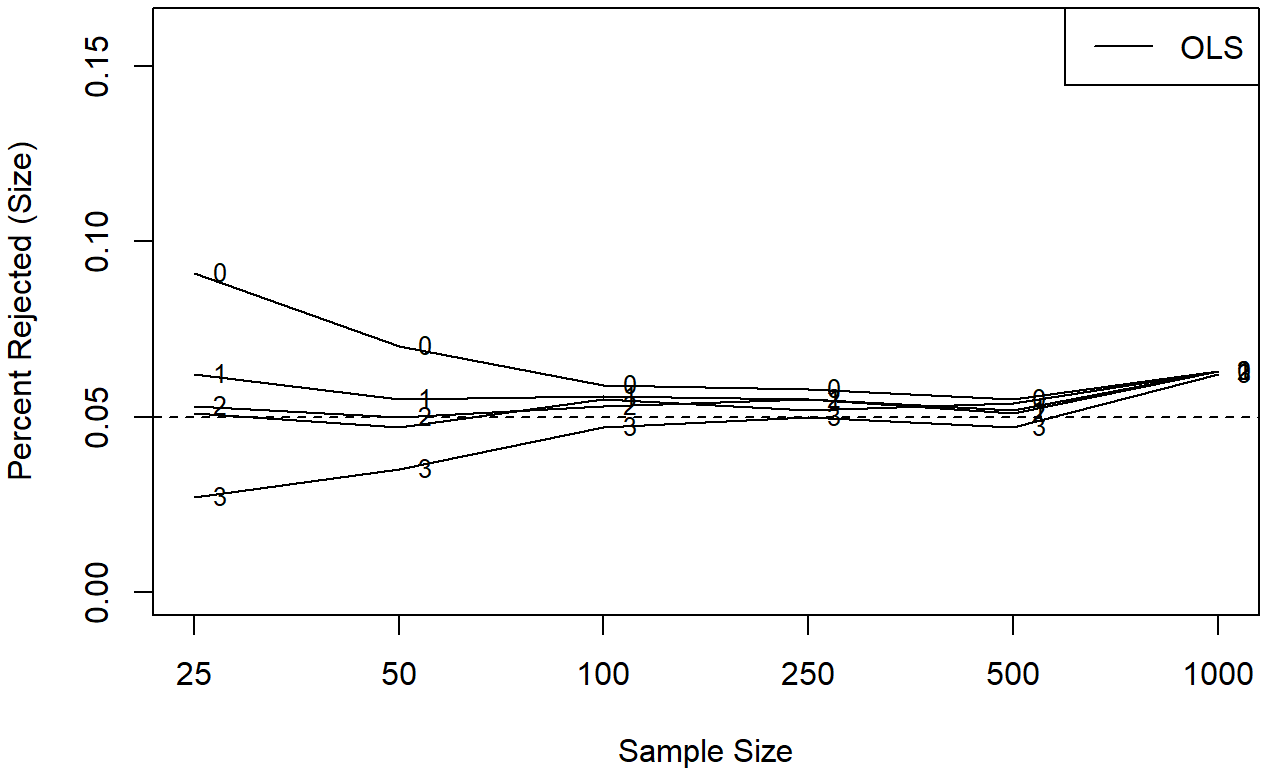
Long and Ervin (2000) compare the four proposed versions of HCCMs to the OLSCM toward assessing their comparative performance with heteroscedastic χ^2_5 error structures at varying sample sizes.

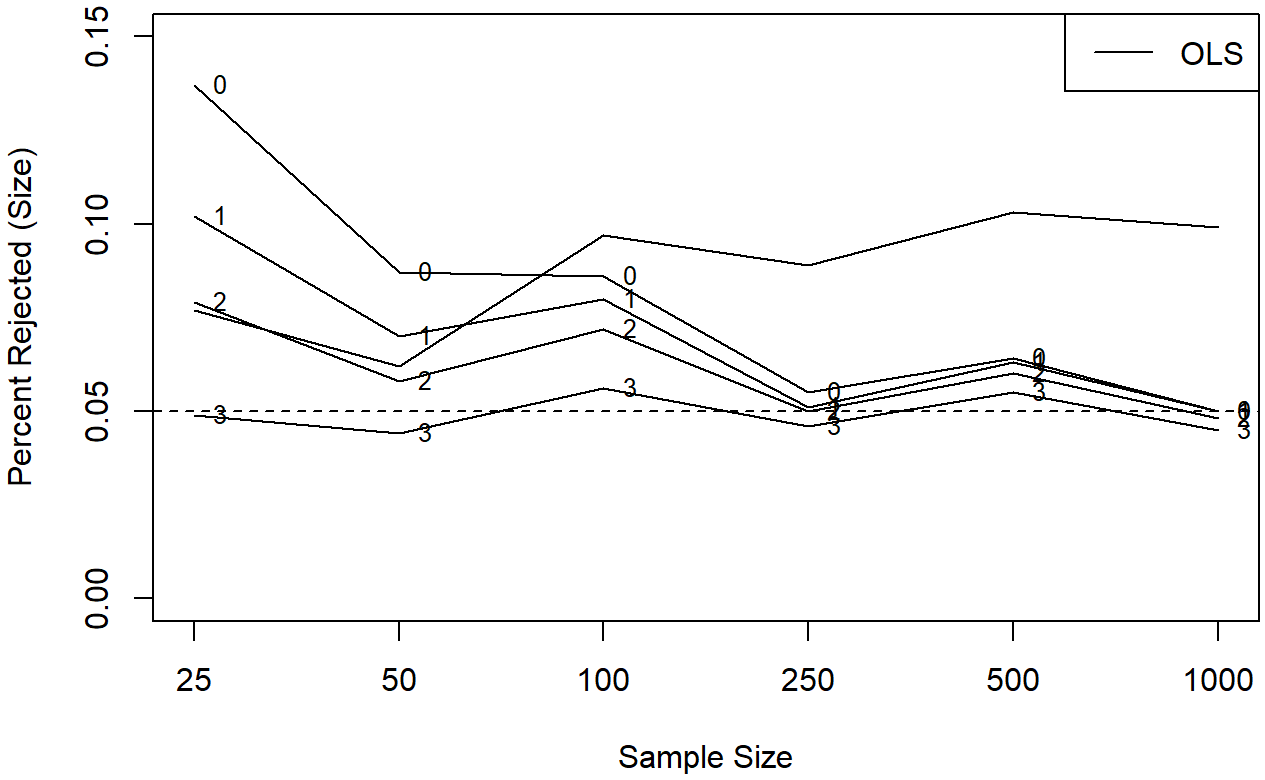
Figure2 shows the size of t-tests for each regression coefficient.

Size: Plots 2.1-2.4 show the proportion of times the true null hypothesis $H_0 : \beta_i = \beta_i^*,\ i = (1, 2, 3, 4)$ where β_i^* is the true parameter determined by population simulation is rejected for each covariance estimator, where $\boldsymbol{\beta} = (1, 1, 1, 0)$.

Simulation:




OLS
Percent Rejected (Size)
0.00
0.05
0.10
0.15
25
50
100
250
500
1000
Sample Size


OLS
Percent Rejected (Size)
0.00
0.05
0.10
0.15
25
50
100
250
500
1000
Sample Size

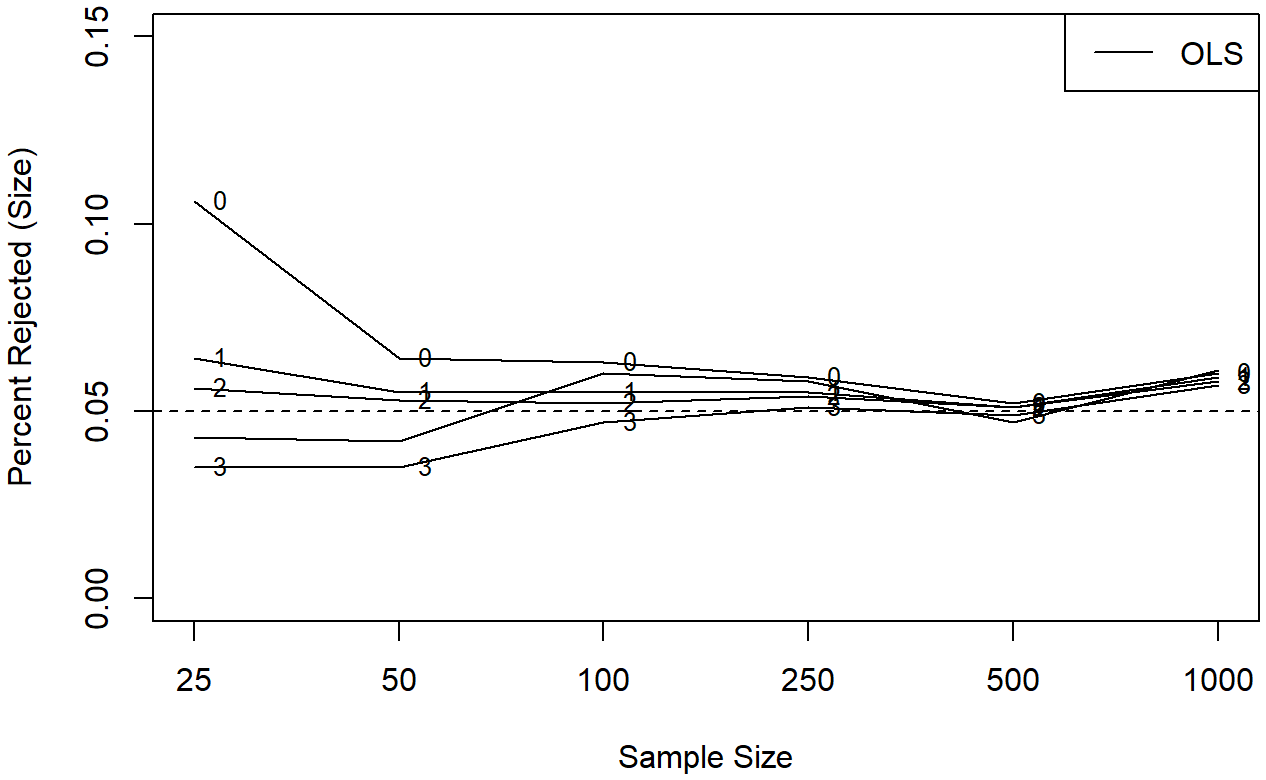
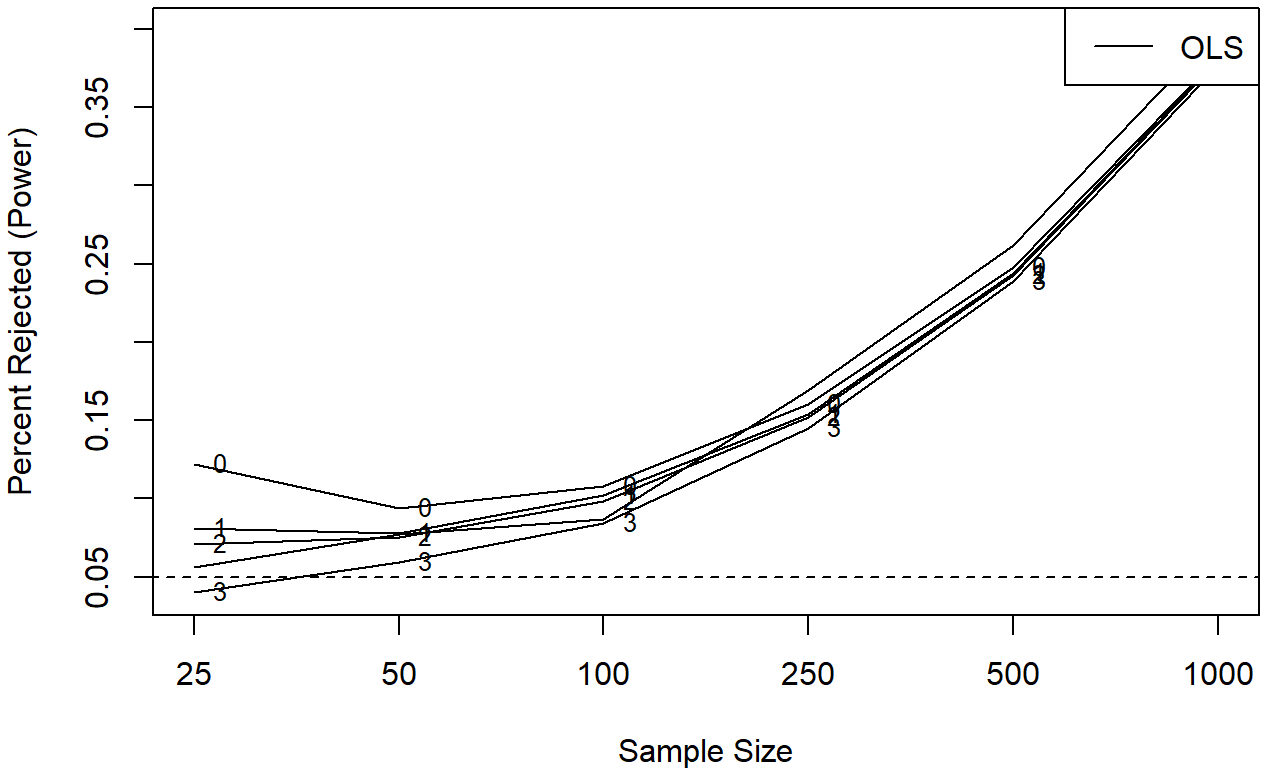


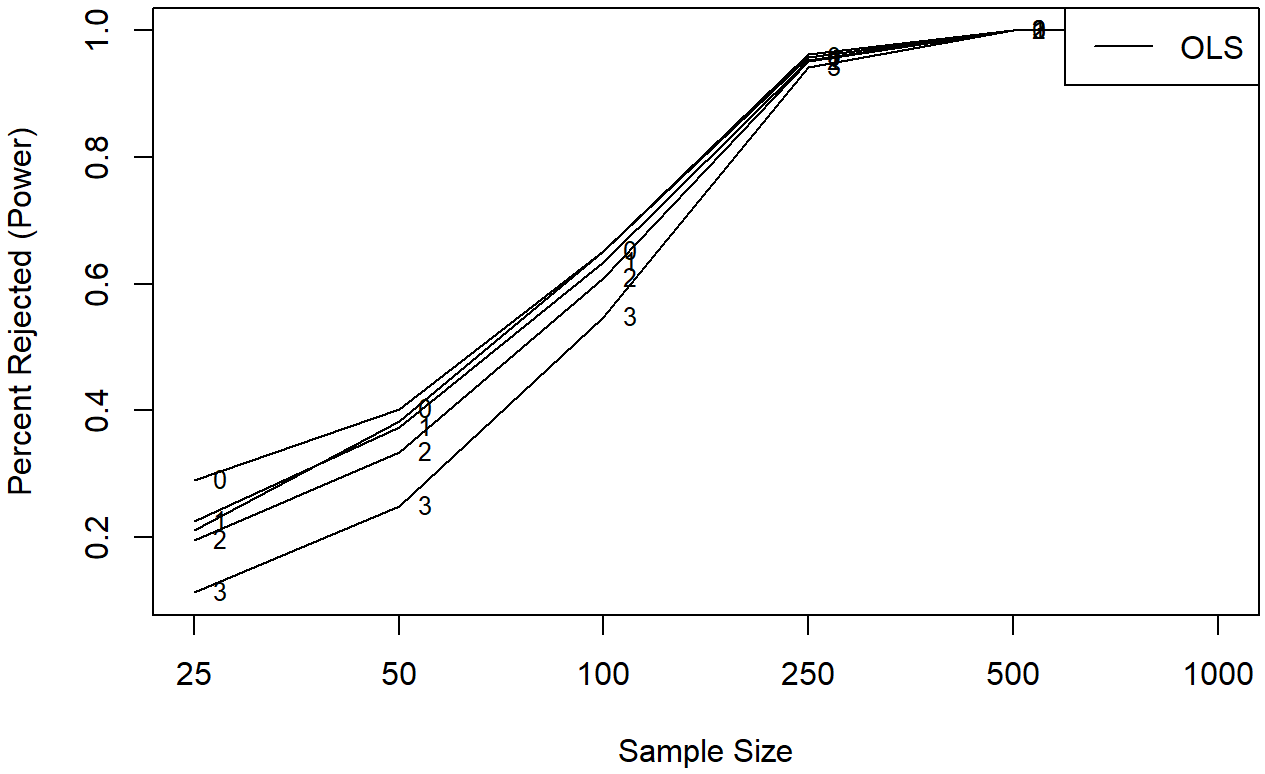
Figure 3

Long and Ervin (2000) compare the four proposed versions of HCCMs to the OLSCM toward assessing their comparative performance with heteroscedastic χ^2_5 error structures at varying sample sizes.

Figure3 shows the power of t-tests for regression coefficients β_1 and β_3.

Power: Plot 3.1 displays the power of tests at varying sample sizes for the true null hypothesis $H_0 : \beta_1 = 1$ and plot 3.2 displays power for the true null $H_0 : \beta_3 = 0$.


OLS
Percent Rejected (Power)
0.05
0.15
0.25
0.35
25
50
100
250
500
1000
Sample Size


Percent Rejected (Power)
0.2
0.4
0.6
0.8
1.0
25
50
100
250
500
1000
Sample Size
OLS