hw3-YifanZheng-7

Yifan Zheng

3/4/2020

Lecture 8, Problem 3: Empirical comparison of the standard errors

Long and Ervin (2000) reviewed and compared several commonly-used standard errors in OLS. Redo their simulation and replicate their Figures 1-4.

```
library(car)
```

```
## Loading required package: carData
```

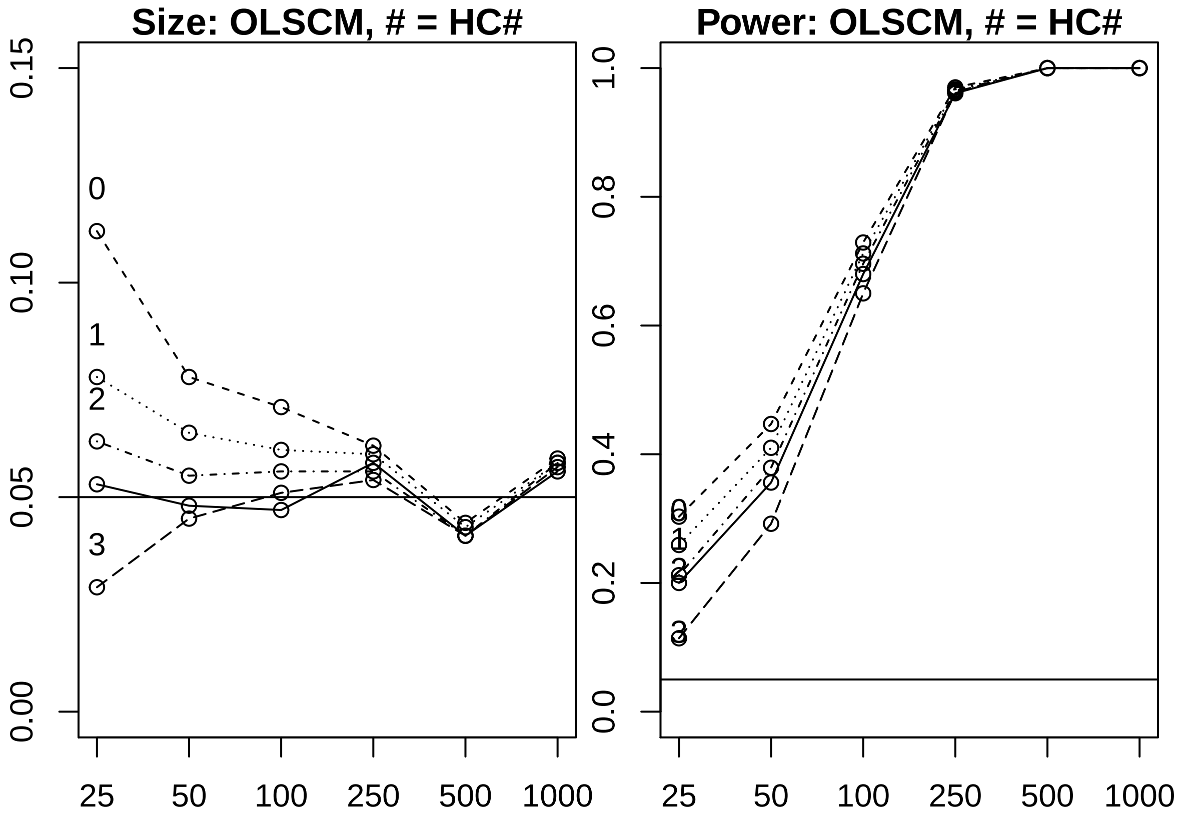
```
set.seed(123)
n=100000

## independent variables
delta1 = runif(n)
delta2 = rnorm(n)
delta3 = rchisq(n,1)
delta4 = rnorm(n)
delta5 = runif(n)
#uniform
x1 = 1+delta1
#bell-shaped
x2 = 3*delta1 + 0.6* delta2
#skewed
x3 = 2*delta1 + 0.6* delta3
#bell-shaped
x4 = 0.1*delta1 + 0.9*delta3 - 0.8*delta4 + 4*delta5
#binary
xd = ifelse(x2 > 1.6, 1, 0)

## errors
en = rnorm(n)
echi = rchisq(n,5)
et = rt(n,5)

## dependent variables: a linear combination of 3 of 5 independent variables plus error term
# tau, let r2 approx 0.4
tau = 0.8
# figure 1 use chi-squared(5) errors
e = echi
y1 = 1 + x1 + x2 + x3 + tau * e

model1 = lm(y1~x1+x2+x3+x4)
covar_pop = summary(model1)$coef
# notice here r2 is approx 0.4
r.squared = summary(model1)$r.squared
r.squared
```

```
## [1] 0.3953686
```

```
## simulations

# a function for simulation 1000 times
simulation <- function(x1, x2, x3,x4,y,size,n, beta){

# simulate 1000 times
  i = 0
  rej_ols_size = 0
  rej_ols_power = 0
  rej_hc0_size = 0
  rej_hc0_power = 0
  rej_hc1_size = 0
  rej_hc1_power = 0
  rej_hc2_size = 0
  rej_hc2_power = 0
  rej_hc3_size = 0
  rej_hc3_power = 0

  while (i<1000) {

    # sample
    data = cbind(x1,x2,x3,x4,y)
    sample = data[sample(n,size),]


    # fitting linear model
    model = lm(sample[,5]~sample[,1]+sample[,2]+sample[,3]+sample[,4])
    # hc
    model.hc0 = sqrt(diag(hccm(model, type = "hc0")))
    model.hc1 = sqrt(diag(hccm(model, type = "hc1")))
    model.hc2 = sqrt(diag(hccm(model, type = "hc2")))
    model.hc3 = sqrt(diag(hccm(model, type = "hc3")))

    # covariates
    covar = summary(model)$coef

    # critical value
    c = qt(0.975, size - 4)

    # test t statistic
    diff = (covar[, 1] - covar_pop[, 1])[beta+1]
    se = c(covar[beta+1,2],model.hc0[beta+1],
           model.hc1[beta+1], model.hc2[beta+1], model.hc3[beta+1])

    rej_ols_size = ifelse((abs(diff/se[1]))>c,
                          rej_ols_size+1,rej_ols_size)
    rej_hc0_size = ifelse((abs(diff/se[2]))>c,
                          rej_hc0_size+1, rej_hc0_size)
    rej_hc1_size = ifelse((abs(diff/se[3]))>c,
                          rej_hc1_size+1, rej_hc1_size)
    rej_hc2_size = ifelse((abs(diff/se[4]))>c,
                          rej_hc2_size+1, rej_hc2_size)
    rej_hc3_size = ifelse((abs(diff/se[5]))>c,
                          rej_hc3_size+1, rej_hc3_size)

    rej_ols_power = ifelse(abs(covar[beta+1,1]/se[1])>c,
                           rej_ols_power+1, rej_ols_power)
    rej_hc0_power = ifelse(abs(covar[beta+1,1]/se[2])>c,
                           rej_hc0_power+1, rej_hc0_power)
    rej_hc1_power = ifelse(abs(covar[beta+1,1]/se[3])>c,
                           rej_hc1_power+1, rej_hc1_power)
    rej_hc2_power = ifelse(abs(covar[beta+1,1]/se[4])>c,
                           rej_hc2_power+1, rej_hc2_power)
    rej_hc3_power = ifelse(abs(covar[beta+1,1]/se[5])>c,
                           rej_hc3_power+1, rej_hc3_power)

    i = i + 1
  }

  size = c(rej_ols_size,rej_hc0_size,rej_hc1_size,
           rej_hc2_size, rej_hc3_size)/1000
  names(size) = c("ols","hc0","hc1","hc2","hc3")
  power = c(rej_ols_power,rej_hc0_power,rej_hc1_power,
            rej_hc2_power,rej_hc3_power)/1000
  names(power) = c("ols","hc0","hc1","hc2","hc3")

  solution <- list("size" = size, "power" = power)
  return (solution)
}

## replicate figure 1
# beta3
size = c(25,50,100,250,500,1000)
rej_size = list()
rej_power = list()
for (i in 1:length(size)){
  rej_size[[i]] = simulation(x1, x2, x3, x4,y1,size[i],n, 3)$size
  rej_power[[i]] = simulation(x1, x2, x3, x4,y1,size[i],n, 3)$power
}
```

```
# plot
par(mfrow = c(1,2),oma = c(1,1,1,1) + 0.1,mar = c(1,1,1,1) + 0.1)
# size
plot(sapply(rej_size,"[[",1),type = "o", lty = 1,xaxt = "n", ylim=c(0,0.15),
     ylab = "Percent Rejected", xlab = "Sample Size")
axis(1, at=1:6, labels = size)
lines(sapply(rej_size,"[[",2), type = "o", lty = 2)
lines(sapply(rej_size,"[[",3), type = "o", lty = 3)
lines(sapply(rej_size,"[[",4), type = "o", lty = 4)
lines(sapply(rej_size,"[[",5), type = "o", lty = 5)
text(1,sapply(rej_size,"[[",2)[1]+0.01,"0")
text(1,sapply(rej_size,"[[",3)[1]+0.01,"1")
text(1,sapply(rej_size,"[[",4)[1]+0.01,"2")
text(1,sapply(rej_size,"[[",5)[1]+0.01,"3")
abline(h = 0.05)
```

```
title("Size: OLSCM, # = HC#")
# power
plot(sapply(rej_power,"[[",1),type = "o", lty = 1,xaxt = "n",ylim=c(0,1),
     ylab = "", xlab = "Sample Size")
axis(1, at=1:6, labels = size)
lines(sapply(rej_power,"[[",2), type = "o", lty = 2)
lines(sapply(rej_power,"[[",3), type = "o", lty = 3)
lines(sapply(rej_power,"[[",4), type = "o", lty = 4)
lines(sapply(rej_power,"[[",5), type = "o", lty = 5)
text(1,sapply(rej_power,"[[",2)[1]+0.01,"0")
text(1,sapply(rej_power,"[[",3)[1]+0.01,"1")
text(1,sapply(rej_power,"[[",4)[1]+0.01,"2")
text(1,sapply(rej_power,"[[",5)[1]+0.01,"3")
abline(h = 0.05)
title("Power: OLSCM, # = HC#")
```

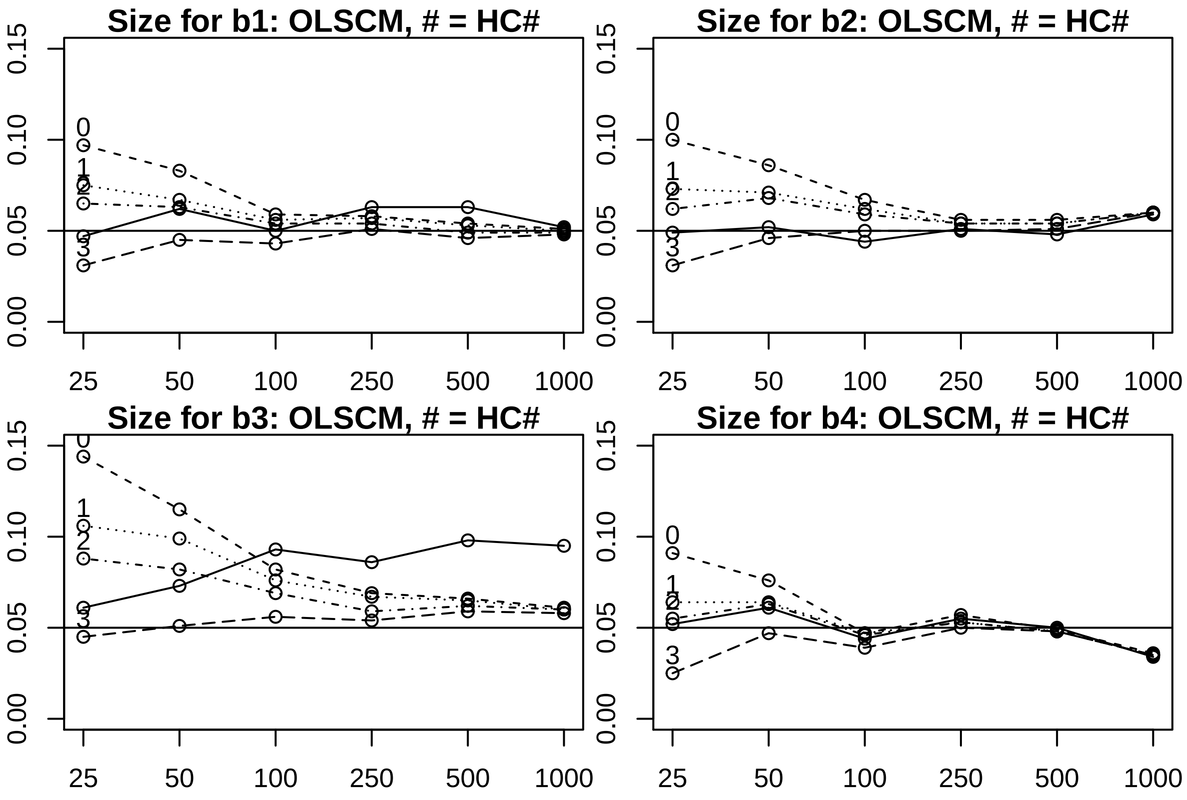


```
# replicate figure 2
e.new = 0.6*(sqrt(x3+1.6))*e
y2 = 1 + x1 + x2 + x3 + e.new
model2 = lm(y2~x1+x2+x3+x4)
covar_pop = summary(model2)$coef

size = c(25,50,100,250,500,1000)

# beta1
```

```
rej_size.1 = list()
for (i in 1:length(size)){
  rej_size.1[[i]] = simulation(x1, x2, x3, x4,y2,size[i],n, 1)$size
}
# beta2
rej_size.2 = list()
for (i in 1:length(size)){
  rej_size.2[[i]] = simulation(x1, x2, x3, x4,y2,size[i],n, 2)$size
}
# beta3
rej_size.3 = list()
for (i in 1:length(size)){
  rej_size.3[[i]] = simulation(x1, x2, x3, x4,y2,size[i],n, 3)$size
}
# beta4
rej_size.4 = list()
for (i in 1:length(size)){
  rej_size.4[[i]] = simulation(x1, x2, x3, x4,y2,size[i],n, 4)$size
}
```

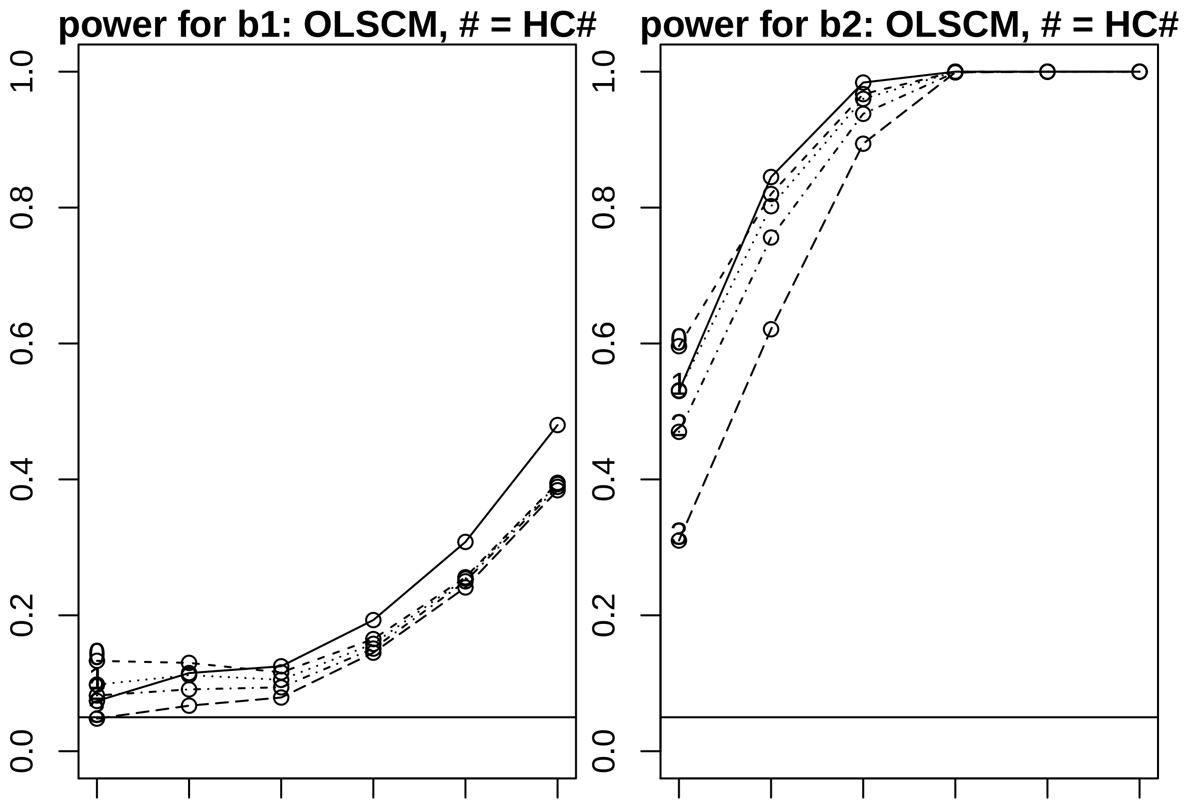
```
# plot
par(mfrow = c(2,2),oma = c(1,1,1,1) + 0.1,mar = c(2,1,1,1) + 0.1)
# beta1
plot(sapply(rej_size.1,"[[",1),type = "o", lty = 1,xaxt = "n", ylim=c(0,0.15),
     ylab = "Percent Rejected", xlab = "Sample Size")
axis(1, at=1:6, labels = size)
lines(sapply(rej_size.1,"[[",2), type = "o", lty = 2)
lines(sapply(rej_size.1,"[[",3), type = "o", lty = 3)
lines(sapply(rej_size.1,"[[",4), type = "o", lty = 4)
lines(sapply(rej_size.1,"[[",5), type = "o", lty = 5)
text(1,sapply(rej_size.1,"[[",2)[1]+0.01,"0")
text(1,sapply(rej_size.1,"[[",3)[1]+0.01,"1")
text(1,sapply(rej_size.1,"[[",4)[1]+0.01,"2")
text(1,sapply(rej_size.1,"[[",5)[1]+0.01,"3")
abline(h = 0.05)
title("Size for b1: OLSCM, # = HC#")
# beta2
plot(sapply(rej_size.2,"[[",1),type = "o", lty = 1,xaxt = "n", ylim=c(0,0.15),
     ylab = "Percent Rejected", xlab = "Sample Size")
axis(1, at=1:6, labels = size)
lines(sapply(rej_size.2,"[[",2), type = "o", lty = 2)
lines(sapply(rej_size.2,"[[",3), type = "o", lty = 3)
lines(sapply(rej_size.2,"[[",4), type = "o", lty = 4)
lines(sapply(rej_size.2,"[[",5), type = "o", lty = 5)
text(1,sapply(rej_size.2,"[[",2)[1]+0.01,"0")
text(1,sapply(rej_size.2,"[[",3)[1]+0.01,"1")
text(1,sapply(rej_size.2,"[[",4)[1]+0.01,"2")
text(1,sapply(rej_size.2,"[[",5)[1]+0.01,"3")
abline(h = 0.05)
title("Size for b2: OLSCM, # = HC#")
# beta3
plot(sapply(rej_size.3,"[[",1),type = "o", lty = 1,xaxt = "n", ylim=c(0,0.15),
     ylab = "Percent Rejected", xlab = "Sample Size")
```

```r
axis(1, at=1:6, labels = size)
lines(sapply(rej_size.3,"[[",2), type = "o", lty = 2)
lines(sapply(rej_size.3,"[[",3), type = "o", lty = 3)
lines(sapply(rej_size.3,"[[",4), type = "o", lty = 4)
lines(sapply(rej_size.3,"[[",5), type = "o", lty = 5)
text(1,sapply(rej_size.3,"[[",2)[1]+0.01,"0")
text(1,sapply(rej_size.3,"[[",3)[1]+0.01,"1")
text(1,sapply(rej_size.3,"[[",4)[1]+0.01,"2")
text(1,sapply(rej_size.3,"[[",5)[1]+0.01,"3")
abline(h = 0.05)
title("Size for b3: OLSCM, # = HC#")
# beta4
plot(sapply(rej_size.4,"[[",1),type = "o", lty = 1,xaxt = "n", ylim=c(0,0.15),
     ylab = "Percent Rejected", xlab = "Sample Size")
axis(1, at=1:6, labels = size)
lines(sapply(rej_size.4,"[[",2), type = "o", lty = 2)
lines(sapply(rej_size.4,"[[",3), type = "o", lty = 3)
lines(sapply(rej_size.4,"[[",4), type = "o", lty = 4)
lines(sapply(rej_size.4,"[[",5), type = "o", lty = 5)
text(1,sapply(rej_size.4,"[[",2)[1]+0.01,"0")
text(1,sapply(rej_size.4,"[[",3)[1]+0.01,"1")
text(1,sapply(rej_size.4,"[[",4)[1]+0.01,"2")
text(1,sapply(rej_size.4,"[[",5)[1]+0.01,"3")
abline(h = 0.05)
title("Size for b4: OLSCM, # = HC#")
```



```r
## replicate figure 3
# check the minimum of x4 is -2.59
e.new = 0.32*sqrt(x3)*sqrt((x4 + 2.6))*e
y3 = 1 + x1 + x2 + x3 + e.new
model3 = lm(y3~x1+x2+x3+x4)
covar_pop = summary(model3)$coef

# beta1
rej_power.1 = list()
for (i in 1:length(size)){
  rej_power.1[[i]] = simulation(x1, x2, x3, x4,y3,size[i],n, 1)$power
}
# beta3
rej_power.3 = list()
for (i in 1:length(size)){
  rej_power.3[[i]] = simulation(x1, x2, x3, x4,y3,size[i],n, 3)$power
}
```

```r
# plot
par(mfrow = c(1,2),oma = c(1,1,1,1) + 0.1,mar = c(0,1,1,1) + 0.1)
# beta1
plot(sapply(rej_power.1,"[[",1),type = "o", lty = 1,xaxt = "n", ylim=c(0,1),
     ylab = "Percent Rejected", xlab = "Sample Size",cex.main=1.25)
axis(1, at=1:6, labels = size)
lines(sapply(rej_power.1,"[[",2), type = "o", lty = 2)
lines(sapply(rej_power.1,"[[",3), type = "o", lty = 3)
lines(sapply(rej_power.1,"[[",4), type = "o", lty = 4)
lines(sapply(rej_power.1,"[[",5), type = "o", lty = 5)
text(1,sapply(rej_power.1,"[[",2)[1]+0.01,"0")
text(1,sapply(rej_power.1,"[[",3)[1]+0.01,"1")
text(1,sapply(rej_power.1,"[[",4)[1]+0.01,"2")
text(1,sapply(rej_power.1,"[[",5)[1]+0.01,"3")
abline(h = 0.05)
title("power for b1: OLSCM, # = HC#")
# beta3
plot(sapply(rej_power.3,"[[",1),type = "o", lty = 1,xaxt = "n", ylim=c(0,1),
     ylab = "", xlab = "Sample Size")
axis(1, at=1:6, labels = size)
lines(sapply(rej_power.3,"[[",2), type = "o", lty = 2)
lines(sapply(rej_power.3,"[[",3), type = "o", lty = 3)
lines(sapply(rej_power.3,"[[",4), type = "o", lty = 4)
lines(sapply(rej_power.3,"[[",5), type = "o", lty = 5)
text(1,sapply(rej_power.3,"[[",2)[1]+0.01,"0")
text(1,sapply(rej_power.3,"[[",3)[1]+0.01,"1")
text(1,sapply(rej_power.3,"[[",4)[1]+0.01,"2")
text(1,sapply(rej_power.3,"[[",5)[1]+0.01,"3")
abline(h = 0.05)
title("power for b2: OLSCM, # = HC#")
```



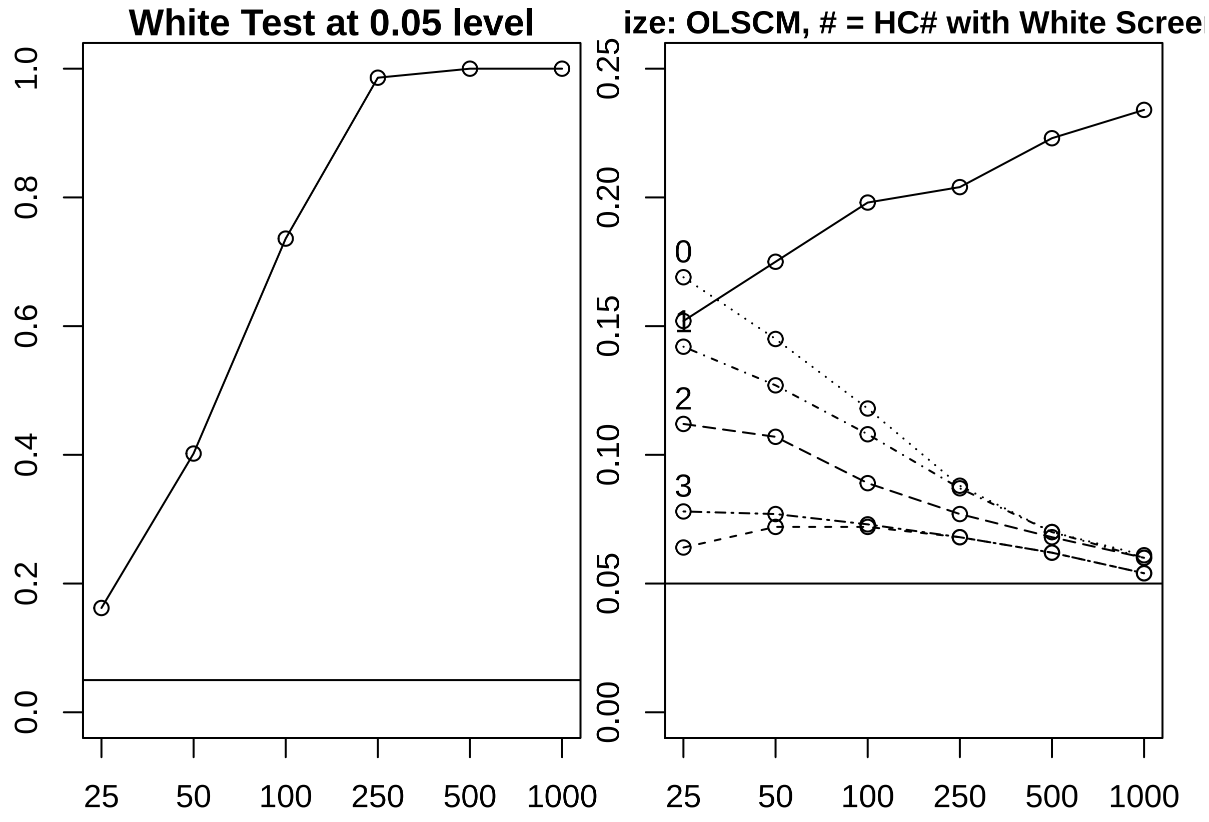
```r
## replicate figure 4
# function to simulate the power of white test
library(lmtest)
```

```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
simulation_wt <- function(x1,x2,x3,x4,y,size){
  i = 0
  j = 0
  while (i<1000) {
    data = cbind(x1,x2,x3,x4,y)
    # sample
    sample = data[sample(n,size),]
    # fitting linear model
    model = lm(sample[,5]~sample[,1]+sample[,2]+sample[,3]+sample[,4])
    # p value of Breusch-Pagan test
    p = bptest(model)$p.value
    j = ifelse(p < 0.05, j+1,j)
    i = i+1
  }
  return(j/1000)
}

# function to simulate size of t test after white test
simulation_wt_t <- function(x1, x2, x3,x4,y,size,n, beta){

  # simulate 1000 times
  i = 0
  rej_ols_size = 0
  rej_hc3_size = 0
  rej_hc0_size_wt = 0
  rej_hc1_size_wt = 0
  rej_hc2_size_wt = 0
  rej_hc3_size_wt = 0

  while (i<1000) {

    # sample
    data = cbind(x1,x2,x3,x4,y)
    sample = data[sample(n,size),]


    # fitting linear model
    model = lm(sample[,5]~sample[,1]+sample[,2]+sample[,3]+sample[,4])
    # hc
    model.hc0 = sqrt(diag(hccm(model, type = "hc0")))
    model.hc1 = sqrt(diag(hccm(model, type = "hc1")))
    model.hc2 = sqrt(diag(hccm(model, type = "hc2")))
    model.hc3 = sqrt(diag(hccm(model, type = "hc3")))

    # covariates
    covar = summary(model)$coef

    # critical value
    c = qt(0.975, size - 4)

    # test t statistic
    diff = (covar[, 1] - covar_pop[, 1])[beta+1]
    se = c(covar[beta+1,2],model.hc0[beta+1],
           model.hc1[beta+1], model.hc2[beta+1], model.hc3[beta+1])

    rej_ols_size = ifelse((abs(diff/se[1]))>c,
                      rej_ols_size+1,rej_ols_size)
    rej_hc3_size = ifelse((abs(diff/se[5]))>c,
                      rej_hc3_size+1, rej_hc3_size)

    # screen with a white test
    p = bptest(model)$p.value
    if(p<0.5){
      rej_hc0_size_wt = ifelse((abs(diff/se[2]))>c,
                          rej_hc0_size_wt+1,rej_hc0_size_wt)
      rej_hc1_size_wt = ifelse((abs(diff/se[3]))>c,
                          rej_hc1_size_wt+1, rej_hc1_size_wt)
      rej_hc2_size_wt = ifelse((abs(diff/se[4]))>c,
                          rej_hc2_size_wt+1,rej_hc2_size_wt)
      rej_hc3_size_wt = ifelse((abs(diff/se[5]))>c,
                          rej_hc3_size_wt+1,rej_hc3_size_wt)
    }else{
      if(abs(diff/se[1])>c){
        rej_hc0_size_wt = rej_hc0_size_wt +1
        rej_hc1_size_wt = rej_hc1_size_wt +1
        rej_hc2_size_wt = rej_hc2_size_wt +1
        rej_hc3_size_wt = rej_hc3_size_wt +1

      }
    }

    i = i + 1
  }

  size_t = c(rej_ols_size,rej_hc3_size,rej_hc0_size_wt,rej_hc1_size_wt,
           rej_hc2_size_wt,rej_hc3_size_wt)/1000
  names(size_t) = c("ols","hc3","hc0_wt","hc1_wt","hc2_wt","hc3_wt")

  return (size_t)
}
```

```
# white test
res = list()
for (i in 1:length(size)){
  res[[i]] = simulation_wt(x1, x2, x3, x4,y3,size[i])
}

# t test for beta3
res.b3 = list()
for (i in 1:length(size)){
  res.b3[[i]] = simulation_wt_t(x1, x2, x3, x4,y3,size[i],n,3)
}
```

```
## plot figure 4
# plot
par(mfrow = c(1,2),oma = c(1,1,1,1) + 0.1,mar = c(1,1,1,1) + 0.1)
# White Test at 0.05 level
plot(sapply(res,"[[",1),type = "o", lty = 1,xaxt = "n", ylim=c(0,1),
     ylab = "Percent Rejected", xlab = "Sample Size")
axis(1, at=1:6, labels = size)
abline(h = 0.05)
title("White Test at 0.05 level")
# size test
plot(sapply(res.b3,"[[",1),type = "o", lty = 1,xaxt = "n", ylim=c(0,0.25),
     ylab = "", xlab = "Sample Size")
axis(1, at=1:6, labels = size)
lines(sapply(res.b3,"[[",2), type = "o", lty = 2)
lines(sapply(res.b3,"[[",3), type = "o", lty = 3)
lines(sapply(res.b3,"[[",4), type = "o", lty = 4)
lines(sapply(res.b3,"[[",5), type = "o", lty = 5)
lines(sapply(res.b3,"[[",6), type = "o", lty = 6)
text(1,sapply(res.b3,"[[",3)[1]+0.01,"0")
text(1,sapply(res.b3,"[[",4)[1]+0.01,"1")
text(1,sapply(res.b3,"[[",5)[1]+0.01,"2")
text(1,sapply(res.b3,"[[",6)[1]+0.01,"3")
abline(h = 0.05)
title("size: OLSCM, # = HC# with White Screen",cex.main=1)
```


White Test at 0.05 level
ize: OLSCM, # = HC# with White Scree