

Cheap computing power has led to the wide use of simulations in econometrics. We'll take a quick look at two of these: Monte-Carlo studies and bootstrapped standard errors.

Sometimes simulations are used to study the effects of different data structures or estimation methods. Typically the data used in such simulations are randomly generated in such a way that they contain the structure one wishes to examine. These simulations are called **Monte-Carlo studies**, and the sample programs given earlier in the semester—examining multicollinearity, omitted variable bias, autocorrelation, and endogeneity—are examples of typical Monte-Carlo simulations. The data are randomly generated and a model is formed with known coefficients. The coefficients and their standard errors are then estimated and stored. New data are randomly generated, and a new model is formed with the same coefficients as before, and the coefficients and standard errors are again estimated and stored. The process is repeated many times, perhaps thousands of times, giving a large set of estimated coefficients which can be compared with the “known” coefficients actually used to produce the model.

- The mean value of the estimated coefficients should lie very near the values actually used to produce the model—if the mean is very different from the true value then the estimates are biased.
- The standard deviation of the estimated coefficients is the same thing as the standard error of the estimated coefficient. One can compare the standard deviation of the collected estimated coefficients with the mean of the collected standard errors. If these are about the same, then one can say that the standard error estimate is unbiased.

Sometimes simulation methods are used on an actual dataset, rather than on randomly generated data. The objective here is not to study abstract econometric problems, but to improve estimates in a particular applied study. There are several **bootstrap methods**, but one widely used technique simply creates a new dataset by randomly sampling—*with replacement*—from the original dataset, then estimating and storing the coefficients. The process is repeated many times, and the collected estimated coefficients are then examined. The standard deviation of these collected coefficients can be used as the standard error, or selected percentiles can be used as the bounds of a confidence interval. Bootstrap methods are particularly useful when one wishes to learn the distribution of a nonlinear combination of estimated coefficients, such as the long-run multiplier.

Homework:

1. Using the Monte-Carlo program for omitted variable bias and multicollinearity, demonstrate that omitted variable bias is more severe the more highly correlated the omitted variable with the included variables.
2. Using the Monte-Carlo program for autocorrelation, demonstrate that the standard error is biased in the presence of autocorrelation, but that the estimated coefficient is unbiased.
3. Using the Monte-Carlo program for endogeneity, demonstrate that the estimated coefficient is biased in the presence of endogeneity. Are all coefficients biased, or only that for the endogenous independent variable? Are the standard errors biased?
4. Using the bootstrap standard error program below, calculate the standard error for returns to scale for manufacturing production in urban counties. Find the pvalue for the null hypothesis that there are constant returns to scale. Find also the 90% confidence interval for returns to scale.

```

#--Bootstrapping (S:\TEFF\662\R\rl1a.R)--
#--1997 Manufacturing, by County--
#--Cobb Douglass production function--

rm(list=ls(all=TRUE))
setwd("s:/TEFF/662/R/")
#setwd("d:/class/662/R/")
library(foreign)
library(AER)
library(boot)
sessionInfo()

#--read in the 1997 county-level manufacturing data--
gg<-read.dbf("manuf1997.dbf")
row.names(gg)<-gg$NAME
names(gg)

#--make instrument for log of kapital---
Kf<-lm(log(kap)~I(EST20/EST)+log(EST)+log(EMP)+log(HRS/npe)+
MEDYRBLT+psewer+hwyppop+hwylnnd+CORE+CD93+factor(REG)+FM,data=gg)
summary(Kf)
vif(Kf)
range(Kf$fitted.values)
gg$Kf<-exp(Kf$fitted.values)

zz<-lm(log(vla)~log(Kf)+log(HRS)+log(npe),data=gg)
summary(zz)
ncvTest(zz)

#--test for returns of scale--
RtS<-sum(zz$coefficients[-1])
RtS
linearHypothesis(zz,"log(Kf)+log(HRS)+log(npe)=1",white.robust=TRUE)

#--Get the SEs via bootstrap--
CDcoef<-function(data,i){coef(lm(log(vla)~log(Kf)+log(HRS)+log(npe),data=data[i,]))}
bs<-boot(gg,CDcoef,1000)
#--look at output--
bs
names(bs)
bs$t0
head(bs$t)
tail(bs$t)

bsRtS<-rowSums(bs$t[,-1])
hist(bsRtS,breaks=30)
abline(v=RtS,col="blue",lty=2)
abline(v=quantile(bsRtS,c(.05,.95)),col="red",lty=2)

t<-(RtS-1)/sd(bsRtS)
pval<-pt(t,zz$df.residual,lower.tail=FALSE)

#--recreate output--
se<-apply(bs$t,2,sd)
names(se)<-pp
mnbeta<-apply(bs$t,2,mean)
names(mnbeta)<-pp
t(rbind(bs$t0,mnbeta,se,(mnbeta-bs$t0)))
bs

#---can bootstrap by directly sampling and replacing, if one wants---

library(Hmisc)

betas<-NULL
for(j in 1:10){
i<-ceil(runif(NROW(gg))*NROW(gg))
betas<-rbind(betas,coef(lm(log(vla)~log(Kf)+log(HRS)+log(npe),data=gg[i,])))
}
betas

table(table(i)) #--see how often some observations are repeated--

```