## Bootstrap regression

- Consider the linear regression model

$$\boldsymbol{y_t = X_t\beta + u_t},\ \ \boldsymbol{E(u_t|X_t) = 0},\ \ \boldsymbol{u_t \sim IID(0, \sigma^2)},$$

- where there are n observations and $\boldsymbol{k}$ regressors. Regressors may include lagged dependent variables, but $\boldsymbol{y_t}$ is not explosive and does not have a unit root.
- There are many ways to bootstrap the above regression model. In general, making stronger assumptions results in better performance if those assumptions are satisfied, but it leads to asymptotically invalid inferences if they are not.
- The assumptions we make in bootstrapping include
  1. Are the errors independent?
  2. Are the errors identically distributed?

### Parametric bootstrap

- Assume that $ut$ is independent and follows a specific distribution, say normal distribution.
- Steps of parametric bootstrap include

1. Use OLS to obtain $\widehat{\boldsymbol{\beta}}$ and $\hat{u}_t$.
2. Generate a typical observation using

$$y_t^* = \boldsymbol{X}_t\widehat{\boldsymbol{\beta}} + u_t^*,\ \ u_t^* \sim NID(0, s^2),$$

where $s^2$ denotes the sample variance of $\hat{u}_t$.

### Residual bootstrap

- Required that the errors be independent of contemporaneous regressors and IID, but with minimal distributional assumptions.
- Steps of residual bootstrp

1. Use OLS to obtain $\widehat{\boldsymbol{\beta}}$ and $\hat{u}_t$.

2. (Optional) rescale residuals so that they have correct variance. For example, the simplest rescaled residual is

$$\ddot{u}_t \equiv \left({}^{n}/_{n-k}\right)^{1/2} \hat{u}_t.$$

3. Generate a typical observation of the bootstrap sample as

$$y_t^* = \boldsymbol{X}_t\widehat{\boldsymbol{\beta}} + u_t^*, \;\; u_t^* \sim EDF(\ddot{u}_t).$$

The $u_t^*$ are often to said to be resampled from $\ddot{u}_t$.

### Block/pair bootstrap

- Resample from the matrix with typical row $[y_t, X_t]$. We no longer condition on the $X_t$, since each bootstrap sample now has a different $X$ matrix. A typical observation of the bootstrap sample is $[y_t^*, X_t^*]$.
  1. The pairs bootstrap is valid even when the errors display heteroskedasticity of unknown form.
  2. It works even for dynamic models. If regressors include lagged dependent variables, we treat them like any other element of $X_t$.
  3. Pairs bootstrap can be applied to an enormous range of models.
  4. In the case of multivariate models, we can combine the pairs and residual bootstraps. Organize residuals as a matrix and apply the pairs bootstrap to its rows. This preserves cross-equation correlations.

## Bootstrap dependent time series

### Parametric bootstrap

- Simulate unconditional ARMA (p,q) model (Mcleod and Hipel, 1978)
- Consider a stationary AR(1) model

$$X_t = \alpha + \phi X_{t-1} + a_t, \;\; a_t \sim NID(0, \sigma_a^2). \qquad (1)$$

- The unconditional distribution of $X_t$ is given by

$$X_t \sim N\left(\frac{\alpha}{1-\phi}, \frac{\sigma_a^2}{1-\phi^2}\right), \qquad (2)$$

- The conditional distribution of $X_t$ given $X_{t-1}$ is given by

$$X_t | X_{t-1} \sim N(\phi X_{t-1}, \sigma_a^2). \qquad (3)$$

- The (unconditional) simulation procedure may be summarized as follow:

1) Simulate $X_0$ by drawing a random number from eqn. (2);
2) Simulate $X_1 = \alpha + \phi X_0 + a_t$, where $X_0$ is obtained from Step 1;
3) Simulate $X_t = \alpha + \phi X_{t-1} + a_t,\ t = 1,2, \ldots$, recursively.

## Sieve bootstrap

- Suppose that the error term $u_t$ in a regression model follow an unknown, stationary process with homoskedastic innovations.
- The sieve bootstrap approximates this process using an $AR(p)$ process with $p$ chosen by some sort of model selection criterion (like AIC or BIC), or by sequential testing.

1. Estimate the model to obtain residuals $\hat{u}_t$;

2. Estimate AR(p) model

$$\hat{u}_t = \sum_{i=1}^{p} \phi_i \hat{u}_{t-i} + \varepsilon_t, \quad (1)$$

for several values of $p$ and choose best one. [need to ensure stationarity]

3. Generate bootstrap error terms

$$u_t^* = \sum_{i=1}^{p} \hat{\phi}_i u_{t-i}^* + \varepsilon_t^*, \quad (2)$$

where the $\varepsilon_t^*$ are resampled from the (rescaled) residuals from eqn. (1).

4. Generate the bootstrap data according to

$$y_t^* = \boldsymbol{X}_t \widehat{\boldsymbol{\beta}} + u_t^*.$$

**Remark:** Sieve bootsrap assume IID innovations, thus ruling out GARCH and other forms of heteroskedasticity.

```
## U of T teaching (Boostraping)
library(ggplot2)
library(MASS)
library(knitr)

## Simulate sample
set.seed(1234)
n<-15
a<-1
b<-3
df<-3
x<-rnorm(n, 3,1)
e<-rt(n,df)
y<-a+b*x+e

df.boots<-data.frame(x=x,y=y)
p<-ggplot(df.boots,aes(x=x,y=y))+geom_point(size=2)
p+geom_smooth()
p+geom_smooth(method=lm, color="red", lty="dashed")

mod1<-summary(lm(y~x))
kable(mod1$coefficients, digits = 3)
?loess
names(loess(y~x))

##Bootstrap simple regression
## block bootstrap
nBoot<-100
yX<-as.matrix(cbind(y,x),ncol=2)
dim(yX) # 15, 2
?sample
a.hat<-numeric(nBoot)
b.hat<-numeric(nBoot)

for(i in 1:nBoot){
        bootsID<-sample(x=1:dim(yX)[1],size=dim(yX)[1],
replace = TRUE)
        bootsMatrix<-yX[bootsID,] #create bootsMatrix using
subindexing
        yTemp<-bootsMatrix[,1]
        xTemp<-bootsMatrix[,2]
```

```
        coefTemp<-lm(yTemp~xTemp)$coefficients
        a.hat[i]<-coefTemp[1]
        b.hat[i]<-coefTemp[2]
}

par(mfrow=c(1,1))
truehist(a.hat, breaks=seq(-
0.1+min(a.hat),max(a.hat)+0.1,0.1),
         xlab="intercept", main="Histogram of bootstrap
intercept")
points(a,0,col=2)
points(lm(y~x)$coef[1],0, pch=4, col=4)
points(mean(a.hat),0,pch=10)
points(mean(a.hat)+sd(a.hat),0,pch=10)
points(mean(a.hat)-sd(a.hat),0,pch=10)

truehist(b.hat, breaks=seq(-
0.1+min(b.hat),max(b.hat)+0.1,0.1),
         xlab="slope", main="Histogram of bootstrap slope")
points(b,0,col=2)
points(lm(y~x)$coef[2],0, pch=4, col=4)
points(mean(b.hat),0,pch=10)
points(mean(b.hat)+sd(b.hat),0,pch=10)
points(mean(b.hat)-sd(b.hat),0,pch=10)

## residual bootstrap
nBoot<-100
a.hat.r<-numeric(nBoot)
b.hat.r<-numeric(nBoot)
yFit<-lm(y~x)$fitted
yRes<-lm(y~x)$residual


for(i in 1:nBoot){
        bootsID<-sample(x=1:dim(yX)[1],size=dim(yX)[1],
replace = TRUE)
        yTemp<-yFit+sqrt(15/14)*yRes[bootsID]
        coefTemp<-lm(yTemp~x)$coefficients
        a.hat.r[i]<-coefTemp[1]
        b.hat.r[i]<-coefTemp[2]
}
```

```
par(mfrow=c(1,1))
truehist(a.hat.r, breaks=seq(-
0.1+min(a.hat.r),max(a.hat.r)+0.1,0.1),
         xlab="intercept", main="Histogram of bootstrap
intercept")
points(a,0,col=2)
points(lm(y~x)$coef[1],0, pch=4, col=4)
points(mean(a.hat.r),0,pch=10)
points(mean(a.hat.r)+sd(a.hat.r),0,pch=10)
points(mean(a.hat.r)-sd(a.hat.r),0,pch=10)


truehist(b.hat.r, breaks=seq(-
0.1+min(b.hat.r),max(b.hat.r)+0.1,0.1),
         xlab="slope", main="Histogram of bootstrap slope")
points(b,0,col=2)
points(lm(y~x)$coef[2],0, pch=4, col=4)
points(mean(b.hat.r),0,pch=10)
points(mean(b.hat.r)+sd(b.hat.r),0,pch=10)
points(mean(b.hat.r)-sd(b.hat.r),0,pch=10)


out<-rbind(c(a, mod1$coef[1,1],mean(a.hat),mean(a.hat.r)),
           c(NA,mod1$coef[1,2],sd(a.hat),sd(a.hat.r)),
           c(b,mod1$coef[2,1],mean(b.hat),mean(b.hat.r)),
           c(NA, mod1$coef[2,2],sd(b.hat),sd(b.hat.r)))
colnames(out)<-c("True","LS","Block Boots", "Residual Boots")
row.names(out)<-c("Intercept","Intercept se", "Slope", "Slope
se")
kable(out, digits = 3)
```