SCRIPT MOD4S2A: GLS VIA SIMULATION

INSTRUCTOR: KLAUS MOELTNER

Generate Data

We will first create a non-spherical error matrix. As an example, we will use a generic heteroskedastic matrix with separate variance for each error term.

```
R> n<-1000 #sample size
R> x1<-rep(1,n)
R> x2<-rnorm(n,2,3)
R> x3<-rnorm(n,5,1.5)
R> #
R> bvec<-c(1.2,-0.8,1.7)
R> #
R> X<-cbind(x1,x2,x3)
R> k<-ncol(X)
R> #
R> #KEY: draw heteroskedastic errors
R> sig2true<-rnorm(n,1,0.5)^2 #"true" error variance
R> eps<-rnorm(n,0,sqrt(sig2true))
R> Om<-diag(as.vector(sig2true))
R> #
R> y<-X %*% bvec + eps
```

OLS

```
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y)
R> e<-y-X%*%bols
R> SSR<-(t(e)%*%e)
R> s2<-(t(e)%*%e)/(n-k)
R> Vb<-s2[1,1]*solve((t(X))%*%X)
R> se=sqrt(diag(Vb))
R> tval=bols/se
R> #
R> ttols<-data.frame(col1=c("constant","x2","x3"),
                col2=bols,
                col3=se,
                col4=tval)
R> colnames(ttols)<-c("variable","estimate","s.e.","t")
```

OLS with correct V(b)

```
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y)
R> e<-y-X%*%bols
R> #
```

```
R> Vb<-solve((t(X))%*%X) %*% t(X) %*% Om %*% X %*% solve((t(X))%*%X)
R> se=sqrt(diag(Vb))
R> tval=bols/se
R> #
R> ttolscorr<-data.frame(col1=c("constant","x2","x3"),
                col2=bols,
                col3=se,
                col4=tval)
R> colnames(ttolscorr)<-c("variable","estimate","s.e.","t")
```

GLS

```
R> bgls<-solve((t(X)) %*% solve(Om) %*% X) %*% (t(X) %*% solve(Om) %*% y)
R> e<-y-X%*%bgls
R> Vb<-solve((t(X))%*% solve(Om) %*% X)
R> se=sqrt(diag(Vb))
R> tval=bgls/se
R> #
R> ttgls<-data.frame(col1=c("constant","x2","x3"),
                col2=bgls,
                col3=se,
                col4=tval)
R> colnames(ttgls)<-c("variable","estimate","s.e.","t")
```

Robust OLS

```
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y)
R> e<-as.vector(y-X%*%bols)
R> # The "as.vector" coercion is apparently needed for the following diag command
R> # to work properly
R> S<-diag(e^2)
R> Vb<-solve((t(X))%*%X) %*% t(X) %*% S %*% X %*% solve((t(X))%*%X)
R> se=sqrt(diag(Vb))
R> tval=bols/se
R> #
R> ttolsrob<-data.frame(col1=c("constant","x2","x3"),
                col2=bols,
                col3=se,
                col4=tval)
R> colnames(ttolsrob)<-c("variable","estimate","s.e.","t")
```

Table 1. Basic OLS

variable	estimate	s.e.	t
constant	1.378199	0.132457	10.404857
x2	-0.813061	0.012256	-66.338912
x3	1.686679	0.024492	68.866152

Table 2. OLS with correct V(b)

variable	estimate	s.e.	t
constant	1.378199	0.126227	10.918408
x2	-0.813061	0.011075	-73.411565
x3	1.686679	0.023586	71.510621

Table 3. GLS

variable	estimate	s.e.	t
constant	1.205437	0.008252	146.073748
x2	-0.800433	0.000657	-1218.778395
x3	1.699338	0.001010	1681.753417

Table 4. Robust OLS

variable	estimate	s.e.	t
constant	1.378199	0.134327	10.260026
x2	-0.813061	0.011589	-70.156631
x3	1.686679	0.025105	67.184678

```
R> proc.time()-tic
   user  system elapsed
   1.01    0.17    1.20
```