SCRIPT MOD4S2E: GROUP-WISE HETEROSKEDASTICITY

INSTRUCTOR: KLAUS MOELTNER

Load and Prepare Data

We will use the hotel / water data from `mod4_2c`.

```
R> load('c:/Klaus/AAEC5126/R/data/water.rda')
R> attach(data)
R> n<-nrow(data)
```

Variable definitions:

```
% Contents of Data (columns)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% 1     id          hotel id (1:120)
% 2     year        1993 - 2000 (not complete for all hotels)
% 3     ln_cons     log of (annual consumption in 1000 gallons)
% 4     age         age of establishment
% 5     stories     number of stories
% 6     sf000       square footage in 1000 units
```

We will assign a different error variance to each hotel. Since we have an unequal number of observations per firm, so we need to first capture these "sub-sample" sizes. The following loop accomplishes this.

```
R> i<-1
R> g=0
R> for (i in 2:n){
     if (id[i]!=id[i-1]){
         g<-c(g,(i-1))
         }
 }
R> g1<-c(g,n)   #I use "g" later for LM tests, thus the re-labeling
R> s1<-diff(g1) #shows length of each panel
R> N<-length(s1)# number of hotels
```

There are 120 hotels in our sample. The largest panel size is 8. The smallest panel size is 4.

Generic OLS to get residuals

```
R> y<-lncons
R> n<-length(y)
R> X<-cbind(rep(1,n),age,age2,stories,sf000,d94,d95,d96,d97,d98,d99,d00)
R> k<-ncol(X)
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y)
R> e<-y-X%*%bols
```

```
R> yhat<-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","age","age2","stories",
        "sf000","d94","d95","d96","d97","d98","d99","d00"),
               col2=bols,
               col3=se,
               col4=tval)
R> colnames(ttols)<-c("variable","estimate","s.e.","t")
```

TABLE 1. OLS Results

variable	estimate	s.e.	t
constant	4.374	0.225	19.436
age	0.067	0.009	7.885
age2	-0.001	0.000	-5.617
stories	0.194	0.037	5.174
sf000	0.034	0.002	17.498
d94	0.054	0.120	0.447
d95	0.048	0.118	0.410
d96	0.023	0.117	0.194
d97	-0.007	0.117	-0.058
d98	-0.066	0.118	-0.562
d99	0.006	0.120	0.048
d00	-0.046	0.122	-0.381

BREUSCH-PAGAN TEST FOR GROUPWISE HSK

```
R> #Create an indicator variable for each hotel
R> D<-matrix(0,n,N)
R> jj<-1
R> for (j in 1:N) {
     D[jj:(jj+s1[j]-1),j]<-rep(1,s1[j],1)
     jj<-jj+s1[j]
     }
R> int<-(t(e)%*% e)/n
R> g<-(e^2/(int[1,1]))-1
R> #capture variables you think may be related to HSK
R> Z<-cbind(rep(1,n), D[,1:(ncol(D)-1)]) #drop one column of D
R> kz<-ncol(Z)
R> LM<-(1/2)*(t(g) %*% Z %*% solve(t(Z) %*% Z) %*% t(Z) %*% g)
R> pval=1-pchisq(LM,kz-1)
```

The BP-statistic for this test is 1721.098. The degrees of freedom for the test are 119. The corresponding p-value is 0.

Robust Breusch-Pagan test (Koenker, 1981)

```
R> int<-(t(e)%*% e)/n
R> V<-mean((e^2-(int[1,1]))^2)
R> LM<-(1/V)*
 (t(e^2-int[1,1]) %*% Z %*% solve(t(Z) %*% Z) %*% t(Z) %*% (e^2-int[1,1]))
R> pval=1-pchisq(LM,kz-1)
```

The robust BP-statistic for this test is 760.306. The degrees of freedom for the test are 119. The corresponding p-value is 0.

FGLS

```
R> #place OLS residuals into separate cells for each panel
R> sig2vec<-0
R> ecell<-vector('list',N)#create S empty cells
R> for (i in 1:N) {
       ecell[[i]]<-e[(g1[i]+1):g1[i+1]]   #fill ith cell
       int<-ecell[[i]]    #call content of ith cell
       int<-(t(int) %*% int)/s1[i] #estimate variance for ith panel
       int<-rep(int,s1[i]) #replicate variance for all observations in panel
       sig2vec<-c(sig2vec,int) #add to the variance vector for the sample
       }
R> sig2vec<-as.vector(sig2vec[2:length(sig2vec)]) #drop leading placeholder
R> Om<-diag(sig2vec)
R> #Step 2: 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","age","age2","stories",
         "sf000","d94","d95","d96","d97","d98","d99","d00"),
                col2=bgls,
                col3=se,
                col4=tval)
R> colnames(ttgls)<-c("variable","estimate","s.e.","t")
```

TABLE 2. FGLS Results

variable	estimate	s.e.	t
constant	4.523	0.101	44.622
age	0.061	0.004	16.247
age2	-0.000	0.000	-11.611
stories	0.177	0.015	11.658
sf000	0.034	0.001	36.865
d94	0.044	0.047	0.924
d95	0.066	0.046	1.428
d96	0.034	0.046	0.734
d97	0.022	0.046	0.470
d98	-0.039	0.047	-0.830
d99	0.006	0.048	0.126
d00	-0.050	0.048	-1.031

ROBUST OLS (FOR COMPARISON)

```
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> ttols<-data.frame(col1=c("constant","age","age2","stories",
         "sf000","d94","d95","d96","d97","d98","d99","d00"),
                col2=bols,
                col3=se,
                col4=tval)
R> colnames(ttols)<-c("variable","estimate","s.e.","t")
```

TABLE 3. Robust OLS Results

variable	estimate	s.e.	t
constant	4.374	0.254	17.199
age	0.067	0.008	8.076
age2	-0.001	0.000	-5.828
stories	0.194	0.035	5.491
sf000	0.034	0.002	17.609
d94	0.054	0.123	0.438
d95	0.048	0.125	0.385
d96	0.023	0.121	0.186
d97	-0.007	0.117	-0.058
d98	-0.066	0.120	-0.552
d99	0.006	0.121	0.047
d00	-0.046	0.117	-0.397

```
R> proc.time()-tic
   user  system elapsed
   0.39    0.09    0.49
```