SCRIPT `MOD1_2C`: CONDITIONAL EXPECTATIONS AND ASSUMPTION 3 OF THE CLRM

INSTRUCTOR: KLAUS MOELTNER

Set basic R-options upfront and load all required R packages:

1. Independence of x and error

Generate an explanatory variable "x" and an error term "eps" independently:

```
R> n<-2000 #number of draws
R> x<-rnorm(n,2,1)
R> eps<-rnorm(n,0,1)
```

Create a scatterplot to examine the relationship between x and eps.

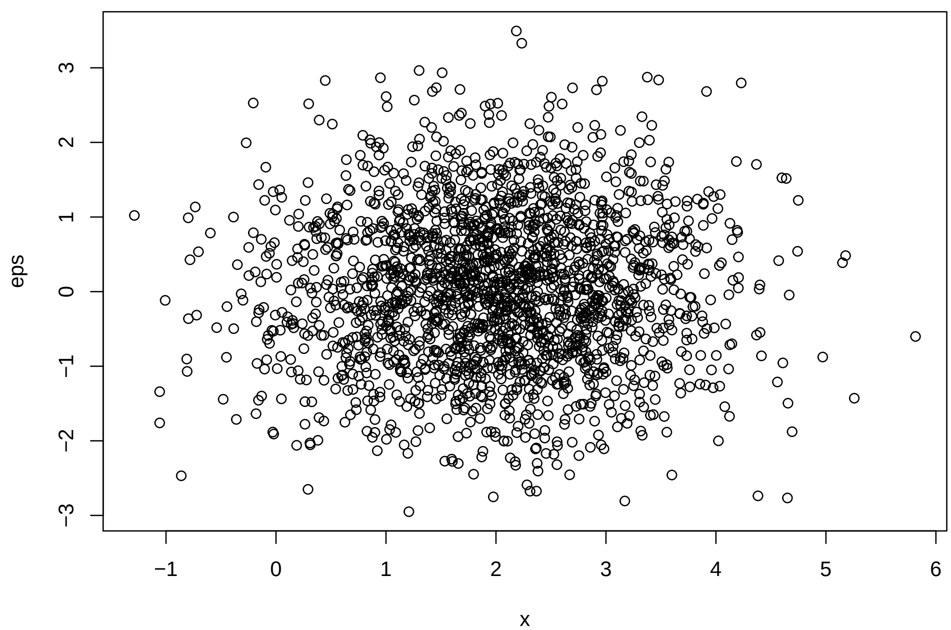

Figure 1. Scatterplot of x vs. epsilon

Sample statistics, including correlation:

```
R> tt<-data.frame(col1=c("x","eps"),
               col2=c(mean(x),mean(eps)),
               col3=c(sd(x),sd(eps)),
               col4=c(min(x),min(eps)),
               col5=c(max(x),max(eps)),
               col6=c(cor(x,eps)," "))
R> colnames(tt)<-c("variable","mean","std","min","max","corr")

R> ttx<- xtable(tt,caption="sample stats")
R> digits(ttx)<-3  #decimals to be shown for each column
R> print(ttx,include.rownames=FALSE,
 latex.environment="center", caption.placement="top",table.placement="!h")
```

TABLE 1. sample stats

variable	mean	std	min	max	corr
x	2.008	1.017	-1.288	5.815	0.0145891718224427
eps	0.035	1.032	-2.950	3.494	

```
R> #get rid of row counters, and center over decimal)
```

Generate coefficients, a dependent variable, and run a simple OLS regression:

```
R> X<-cbind(rep(1,n),x)
R> bvec=c(1,0.2)
R> y<-X %*% bvec+eps
R> k<-ncol(X)
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y);# compute OLS estimator
R>  e<-y-X%*%bols # Get residuals.
R> s2<-(t(e)%*%e)/(n-k) #get the regression error (estimated variance of "eps").
R> Vb<-s2[1,1]*solve((t(X))%*%X) # get the estimated variance-covariance matrix of bols
R> se=sqrt(diag(Vb)) # get the standard erros for your coefficients;
R> tval=bols/se # get your t-values.
```

Display results in a nice table:

```
R> tt<-data.frame(col1=c("constant","x"),
               col2=bvec,
               col3=bols,
               col4=se,
               col5=tval)
R> colnames(tt)<-c("variable","true value","estimate","s.e.","t")

R> ttx<- xtable(tt,caption="OLS output, Model 1 (independence)")
R> digits(ttx)<-3   #decimals to be shown for each column
R> print(ttx,include.rownames=FALSE,
 latex.environment="center", caption.placement="top",table.placement="!h")
R> #get rid of row counters, and center over decimal)
```

The estimated standard deviation of the regression error is 1.033.

TABLE 2. OLS output, Model 1 (independence)

variable	true value	estimate	s.e.	t
constant	1.000	1.005	0.051	19.656
x	0.200	0.215	0.023	9.460

2. Correlation of x and error

Now assume that the error term is correlated with x. We can use a "miniature regression model" to force such correlation.

```
R> n<-1000 #number of draws
R> eps<-0.5*x+rnorm(n,0,1)
```

Create a scatterplot to examine the relationship between x and eps:

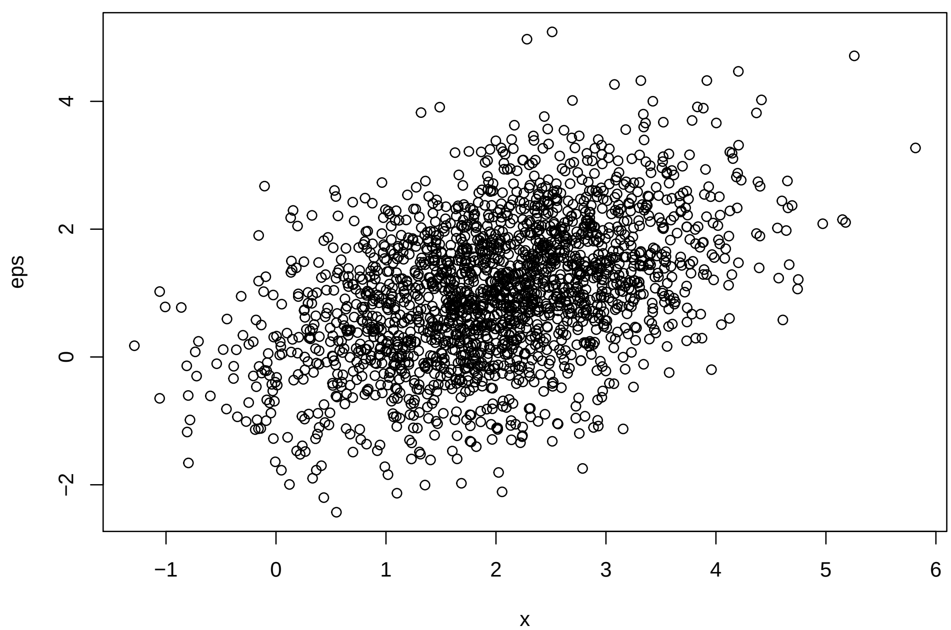

FIGURE 2. Scatterplot of x vs. epsilon

Sample statistics, including correlation:

```
R> tt<-data.frame(col1=c("x","eps"),
                col2=c(mean(x),mean(eps)),
                col3=c(sd(x),sd(eps)),
                col4=c(min(x),min(eps)),
                col5=c(max(x),max(eps)),
                col6=c(cor(x,eps)," "))
R> colnames(tt)<-c("variable","mean","std","min","max","corr")
```

```
R> ttx<- xtable(tt,caption="sample stats")
R> digits(ttx)<-3  #decimals to be shown for each column
R> print(ttx,include.rownames=FALSE,
 latex.environment="center", caption.placement="top",table.placement="!h")
```

Table 3. sample stats

variable	mean	std	min	max	corr
x	2.008	1.017	-1.288	5.815	0.444701064301412
eps	1.015	1.111	-2.430	5.087	

```
R> #get rid of row counters, and center over decimal)
```

Generate coefficients, a dependent variable, and run a simple OLS regression:

```
R> X<-cbind(rep(1,n),x)
R> bvec=c(1,0.2)
R> y<-X %*% bvec+eps
R> k<-ncol(X)
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y);# compute OLS estimator
R>  e<-y-X%*%bols # Get residuals.
R> s2<-(t(e)%*%e)/(n-k) #get the regression error (estimated variance of "eps").
R> Vb<-s2[1,1]*solve((t(X))%*%X) # get the estimated variance-covariance matrix of bols
R> se=sqrt(diag(Vb)) # get the standard erros for your coefficients;
R> tval=bols/se # get your t-values.
```

Display results in a nice table:

```
R> tt<-data.frame(col1=c("constant","x"),
                col2=bvec,
                col3=bols,
                col4=se,
                col5=tval)
R> colnames(tt)<-c("variable","true value","estimate","s.e.","t")

R> ttx<- xtable(tt,caption="OLS output, Model 2 (correlation)")
R> digits(ttx)<-3   #decimals to be shown for each column
R> print(ttx,include.rownames=FALSE,
 latex.environment="center", caption.placement="top",table.placement="!h")
```

Table 4. OLS output, Model 2 (correlation)

variable	true value	estimate	s.e.	t
constant	1.000	1.039	0.070	14.898
x	0.200	0.686	0.031	22.141

```
R> #get rid of row counters, and center over decimal)
```

The estimated standard deviation of the regression error is 1.409.
You can see that the coefficient on x is seriously off-target. This is the typical ”omitted variable

bias" that arises when regressors are correlated with the error term. Note that the constant term estimate is not affected by this bias, since the the underlying regressor (a column of ones) is - obviously - not correlated with x.

3. Illustration of the Law of Iterated Expectations

We want to show that $E(\epsilon_i) = E_{x_i}(E_{\epsilon_i}(\epsilon_i|x_i))$ See the Sweave file for comments to the `R` commands in the following chunk.

```
R> R<-1000
R> r<-1000
R> epsvec<-rep(0,R) #pre-allocate for computational speed
R> for (i in 1:R) {
     xi<-sample(x,1,replace=TRUE) #draw one xi from your x vector
     epsi<-rnorm(r,0.5*xi,1) #draw r error values from a normal with mean 0.5* xi.
     #This is consistent with the correlated model we created above.
     mepsi=mean(epsi) #take mean to approximate E(epsi|xi), the conditional expectation
     epsvec[i]<-mepsi #collect the conditional expectations, replacing the zeros in epsvec
 }
```

The unconditional expectation is 1.015. The iterated expectation is 1.009.

```
R> proc.time()-tic
   user  system elapsed
   0.50    0.02    0.52
```