# Script mod4s3b: Serial Correlation, Hotel Application

Instructor: Klaus Moeltner

March 28, 2013

## Load and Prepare Data

This example uses 96 months of observations on water use by one of the hotels from our "water" data set.

```
R> data<- read.table('c:/Klaus/AAEC5126/R/data/hotel.txt', sep="\t", header=FALSE)
R> #
R> #assign variable names
R> names(data)[1]<-"period"
R> names(data)[2]<-"con"
R> names(data)[3]<-"dayhrs"
R> names(data)[4]<-"ethat"
R> names(data)[5]<-"cldg"
R> names(data)[6]<-"htdg"
R> names(data)[7]<-"avtemp"
R> #
R> save(data, file = "c:/Klaus/AAEC5126/R/data/hotel.rda")
R> attach(data)
```

Variable definitions:

```
% Contents of data
%%%%%%%%%%%%%%%%%%%%%
% Variable Obs Mean Std. Dev. Min Max
%
% 1 period      1 through 96; 1= jan 1993, 96=dec 2000
% 2 con         average monthly consumption in 1000 gallons for all hotels in the Reno area
% 3 dayhrs      hrs of daylight that month
% 4 ethat     estimated monthly evapotransporation
% 5 cldg     colling degree days
% 6 htdg     heating degree days
% 7 avtemp     monthly average of average daily temperature
```

## Simple OLS

```
R> # Define variables
R> n<-nrow(data)
```

```
R> y<-log(1000*con)#log of monthly water consumption
R> X<-cbind(rep(1,n),dayhrs,ethat,cldg,htdg,avtemp)
R> k<-ncol(X)
R> #
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y)# compute OLS estimator
R> e<-y-X%*%bols # Get residuals.
R> SSR<-(t(e)%*%e)#sum of squared residuals - should be minimized
R> s2<-(t(e)%*%e)/(n-k) #get the regression error (estimated variance of "eps").
R> s2ols<-s2 #for Hausman test below
R> Vb<-s2[1,1]*solve((t(X))%*%X) # get the estimated VCOV matrix of bols
R> se=sqrt(diag(Vb)) # get the standard erros for your coefficients;
R> tval=bols/se # get your t-values.
R> #
R> tt<-data.frame(col1=c("constant","dayhrs","ethat","cldg","htdg","avtemp"),
                 col2=bols,
                 col3=se,
                 col4=tval)
R> colnames(tt)<-c("variable","estimate","s.e.","t")
```

Table 1: OLS output

| variable | estimate | s.e. | t |
|---|---|---|---|
| constant | 12.078 | 0.176 | 68.774 |
| dayhrs | -0.002 | 0.001 | -2.649 |
| ethat | -0.004 | 0.022 | -0.168 |
| cldg | -0.000 | 0.000 | -1.148 |
| htdg | -0.000 | 0.000 | -0.487 |
| avtemp | 0.016 | 0.002 | 8.770 |

## Residual Plot

## Robust OLS

We'll use the Newey-West (1987) procedure as shown in the lecture notes. The tricky part is composing the $S_1$ matrix.

```
R> L<-ceiling(n^(1/4)); #rounds upwards to nearest integer;
R> # this would be the generic choice
R> H<-matrix(0,k,k)
R> for (j in 1:L) {
     t<-j+1
     G<-matrix(0,k,k)
     for (i in t:n)  {
         m<-(1-(j/(L+1)))*e[i]*e[i-j]*
```

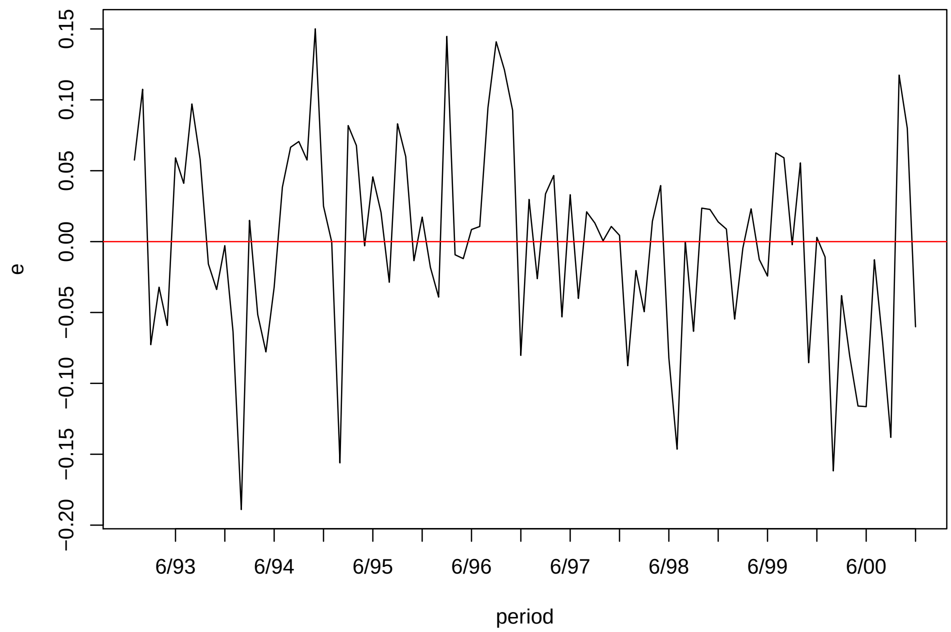


Figure 1: OLS residual plots

```
        (t(X[i,,drop=FALSE])%*% X[i-j,]+t(X[i-j,,drop=FALSE]) %*% X[i,])
        #drop=FALSE forces the transpose to be a column vector
        G<-G+m
     }
     H<-H+G
 }
R> e<-as.vector(e)
R> S1<-(t(X) %*% diag(e^2) %*% X)+H
R> Vb<-solve((t(X))%*%X) %*% S1 %*% solve((t(X))%*%X)
R> se=sqrt(diag(Vb))
R> tval=bols/se
R> tt<-data.frame(col1=c("constant","dayhrs","ethat","cldg","htdg","avtemp"),
                col2=bols,
                col3=se,
                col4=tval)
R> colnames(tt)<-c("variable","estimate","s.e.","t")
```

Table 2: Robust OLS output

| variable | estimate | s.e. | t |
|---|---|---|---|
| constant | 12.078 | 0.196 | 61.699 |
| dayhrs | -0.002 | 0.001 | -2.321 |
| ethat | -0.004 | 0.025 | -0.146 |
| cldg | -0.000 | 0.000 | -0.978 |
| htdg | -0.000 | 0.000 | -1.231 |
| avtemp | 0.016 | 0.002 | 8.455 |

## Testing for AR(1) Serial Correlation

We first plot, then regress the OLS residuals against their lag-1 neighbors.

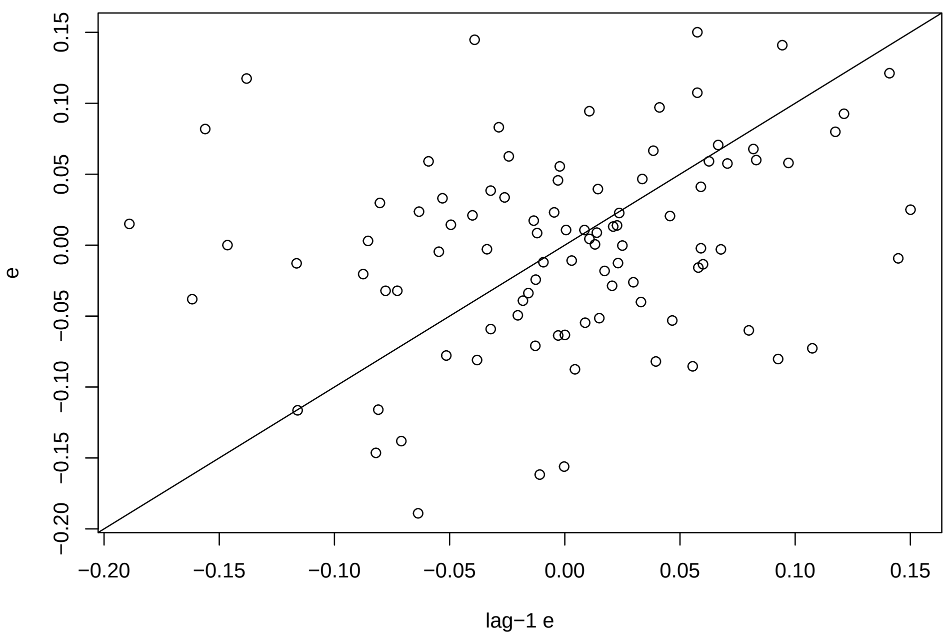


Figure 2: residuals vs. lag-1 residuals

```
R> n<-length(ecurr)  #can't use nrow() for a vector
R> y<-ecurr
R> X<-elag
R> k<-1
```

```
R> #
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y)# compute OLS estimator
R> e<-y-X%*%bols # Get residuals.
R> SSR<-(t(e)%*%e)#sum of squared residuals - should be minimized
R> s2<-(t(e)%*%e)/(n-k) #get the regression error (estimated variance of "eps").
R> s2ols<-s2 #for Hausman test below
R> Vb<-s2[1,1]*solve((t(X))%*%X) # get the estimated VCOV matrix of bols
R> se=sqrt(diag(Vb)) # get the standard erros for your coefficients;
R> tval=bols/se # get your t-values.
R> #
R> tt<-data.frame(col1=c("lag-1 e"),
                 col2=bols,
                 col3=se,
                 col4=tval)
R> colnames(tt)<-c("variable","estimate","s.e.","t")
```

Table 3: Residual vs. lagged residual plot

| variable | estimate | s.e. | t |
|---|---|---|---|
| lag-1 e | 0.234 | 0.100 | 2.327 |

# 1 Breusch-Godfrey Multipier Test for AR(1)

```
R> #re-run original OLS and capture residuals
R> n<-nrow(data)
R> y<-log(1000*con)#log of monthly water consumption
R> X<-cbind(rep(1,n),dayhrs,ethat,cldg,htdg,avtemp)
R> k<-ncol(X)
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y)# compute OLS estimator
R> e<-y-X%*%bols # Get residuals.
R> elag<-e[1:(n-1)]
R> #
R> e0lag<-c(0,elag) # fill first position with 0 */
R> Xo=cbind(X, e0lag) #augment X with a column of lagged residuals
R> #
R> LM<-n*((t(e) %*% Xo %*% solve(t(Xo) %*% Xo) %*% t(Xo) %*% e)/(t(e) %*% e))
R> pval=1-pchisq(LM,1)
```

The BG-statistic for this test is 5.988. The degrees of freedom for the test are 1. The corresponding p-value is 0.014.

# 2 Durbin-Watson Test

```
R> ecurr<-e[2:n]
R> elag<-e[1:(n-1)]
```

```
R> d<-(t(ecurr-elag) %*% (ecurr-elag))/(t(e) %*% e)
```

The DW-statistic for this test is 1.521. The sample size is 96. The column space of X is 6.

# 3 Prais-Winsten FGLS

```
R> # Step 1: Get a consistent estimate of rho:
R> rho<-solve(t(elag) %*% elag) %*% t(elag) %*% ecurr #OLS solution for our
R> # "e vs. e-lag 1 regression model above
R> #
R> #Step 2: compose the correlation matrix R
R> R<-matrix(0,n,n)
R> up<-seq(1,(n-1),1)
R> down<-seq((n-1),1,-1)
R> int<- c(rho^(down), 1, rho^(up)) #1 by 2*(n-1)+1
R> for (i in 1:n){
 R[i,]<-int[(n-(i-1)):(length(int)-(i-1))]
 }
R> #
R> #Step 3: compute FGLS estimator
R> bgls<-solve((t(X)) %*% solve(R) %*% X) %*% (t(X) %*% solve(R) %*% y)
R> #
R> #Step 4: compute a consistent estimate of sig(eps)
R> e<-y-X%*%bgls
R> sige<-(1/n)*t(e) %*% solve(R) %*% (e)
R> #
R> #Step 5: Compute consistent variance-covariance matrix for b_fgls
R> Om<-sige[1,1]*R
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","dayhrs","ethat","cldg","htdg","avtemp"),
                col2=bgls,
                col3=se,
                col4=tval)
R> colnames(ttgls)<-c("variable","estimate","s.e.","t")
```

Table 4: FGLS output

| variable | estimate | s.e. | t |
|---|---|---|---|
| constant | 12.112 | 0.176 | 68.720 |
| dayhrs | -0.002 | 0.001 | -2.423 |
| ethat | 0.003 | 0.022 | 0.138 |
| cldg | -0.000 | 0.000 | -1.177 |
| htdg | 0.000 | 0.000 | 0.235 |
| avtemp | 0.015 | 0.002 | 7.954 |

```
R> proc.time()-tic
   user  system elapsed 
   0.25    0.08    0.33 
```