Assignment 1

The econometric models simple moving-average model (MA) and simple autoregressive models (AR) will be discussed in this analysis, and model selection will be made.

MA(1) MODEL

The general form of an MA(1) model is : $y(t)= c_0 + a_t - \theta_1 a_{t-1}$, where c_0 is a constant and $\{a_t\}$ is a white noise series.

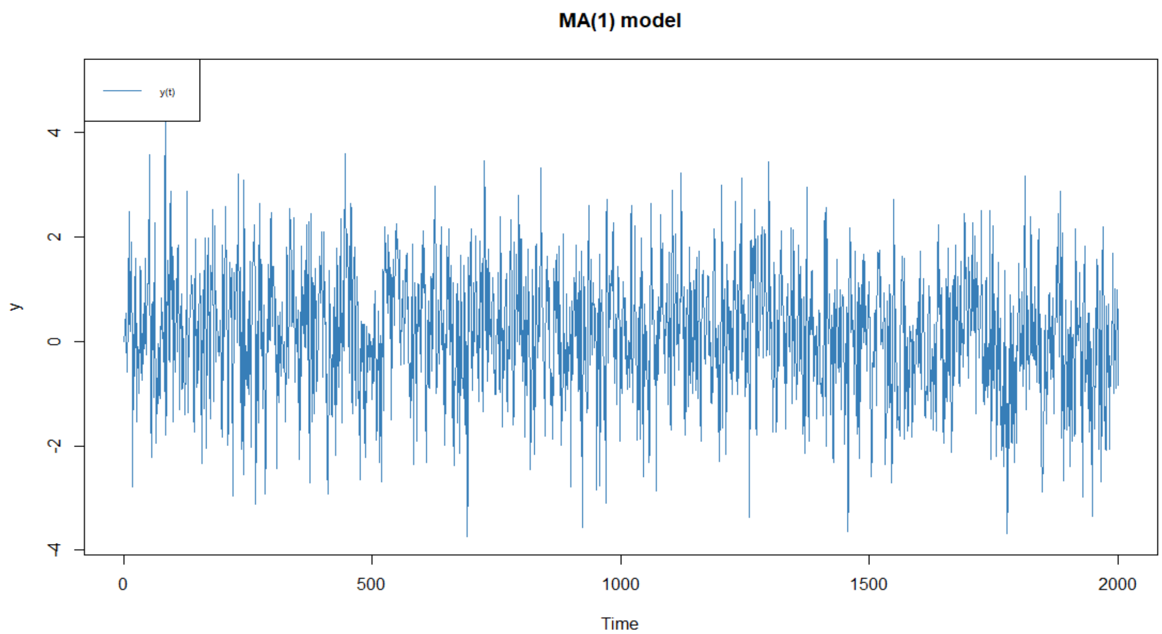

Figure1.1 Plot of 200 data points simulated from the MA(1) model y(t)=0.1+u(t)+0.7u(t-1), with u(t)~N(0,1)

Autocorrelation Function

We assume for simplicity that c0 = 0 for an MA(1) model. Multiplying the model by r_{t-l}, we have

$$r_{t-l}= r_{t-l}\, a_t - \theta_1 r_{t-l} a_{t-l}.$$

Taking expectation, we obtain

$$\gamma_1 = -\theta_1\sigma_a^2 , \quad \text{and} \quad \gamma_l=0 \quad , \text{for } l>1 .$$

Using the prior result and the fact that $Var(rt) = (1 + \theta_1^2)\sigma_a^2$, we have

$$\rho_0 = 1 , \qquad \rho_1 = -\theta_1 /(1 + \theta_1^2) , \qquad \rho_l=0, \quad \text{for } l>2.$$

Thus, for an MA(1) model, the lag-1 ACF is not zero, but all higher order ACFs are zero. In other words, the ACF of an MA(1) model cuts off at lag 1.

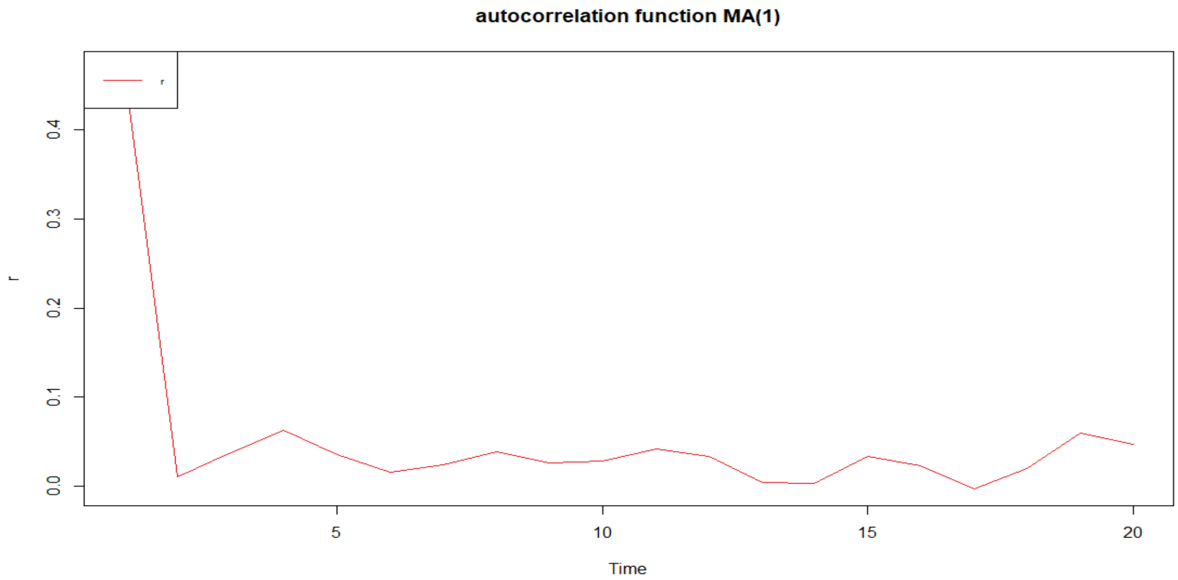

Figure1.2 . Plot of the autocorrelation function

Figure 1.2 shows that while there is a substantial autocorrelation in values at first, it quickly drops. This indicates that upcoming values are no longer influenced by prior ones, and that the recent past, with the exception of starting values, has no effect on the present.

AR(2) MODEL

The general form of an AR(2) model is: $y(t) = \varphi_0 + \varphi_1 r_{t-1} + \varphi_2 r_{t-2} + a_t$.

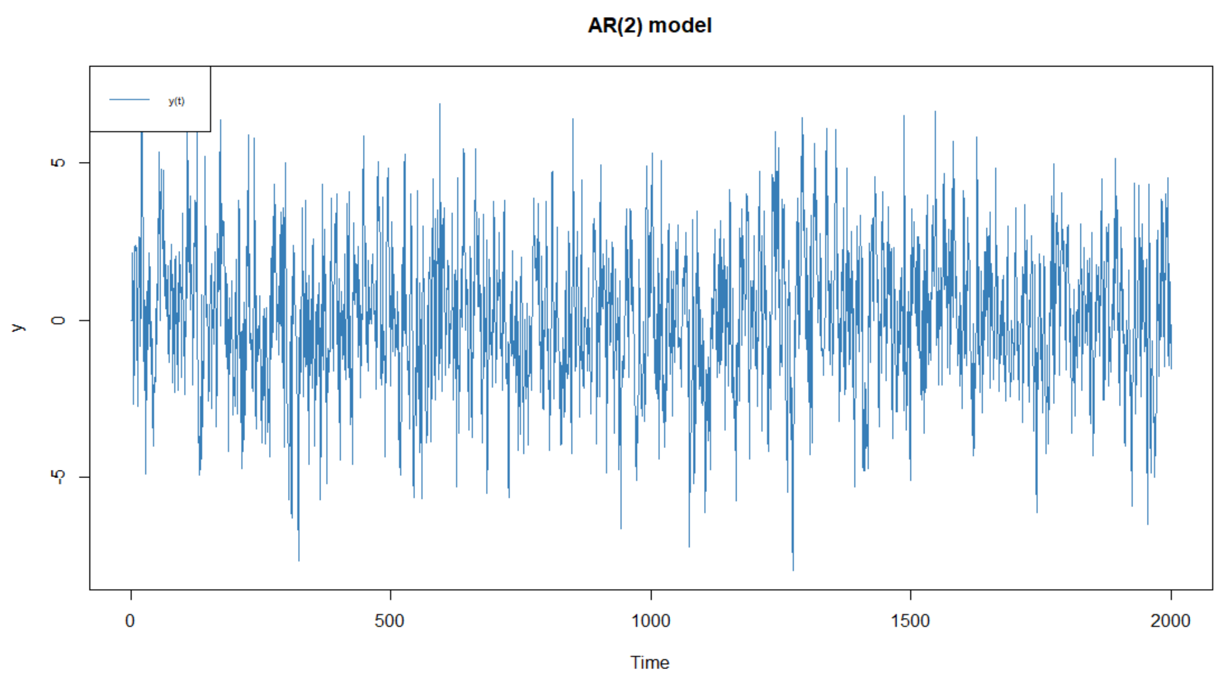

Figure1.3 Plot of 200 data points simulated from the AR(2) model y(t)=0.1+0.5y(t-1)+0.1y(t-2)+u(t), with u(t)~N(0,1)

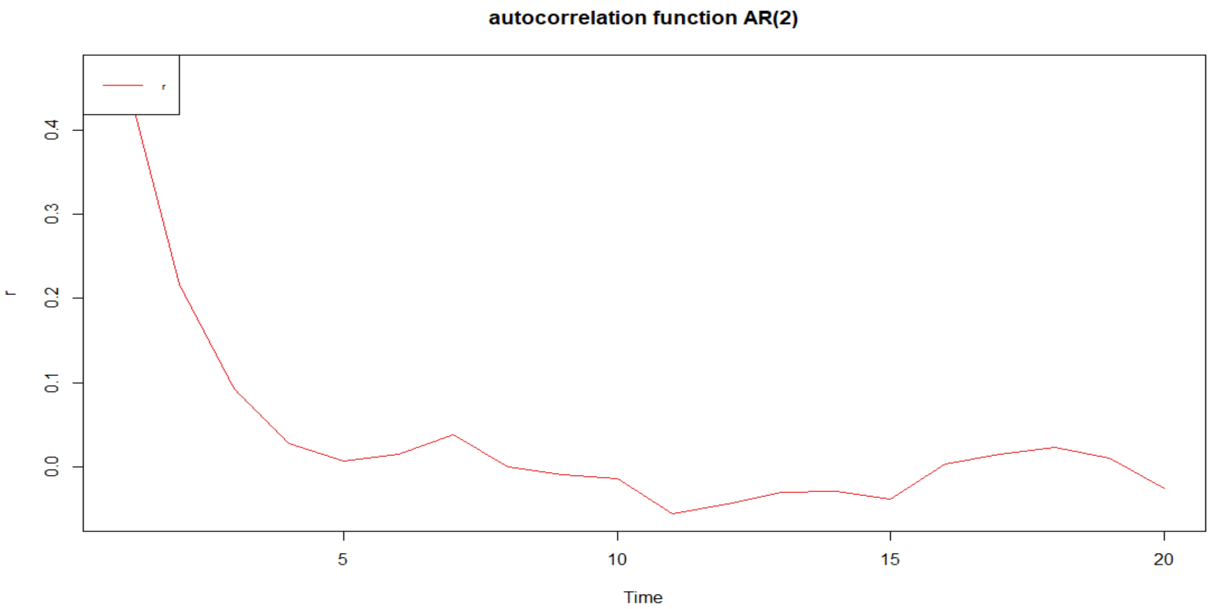

Figure1.4. Plot of the autocorrelation function

Figure 1.4 shows that while there is a high autocorrelation in values at first, it eventually falls, although not as steeply as before with a slightly better smoothness. This indicates that the present values cease to be affected by the prior ones, but at a slower rate, and that the recent past has a greater impact on the present than previously in MA(1) model . However, at some time, the autocorrelation approaches 0 and is no longer the present impacted by the past.

Partial Autocorrelation Function (PACF)

The PACF of a stationary time series is a function of its ACF and is a useful tool for determining the order p of an AR model. A simple, yet effective way to introduce PACF is to consider the following AR models in consecutive orders:

$$r_t = \varphi_{0,1} + \varphi_{1,1} r_{t-1} + e_{1t},$$

$$r_t = \varphi_{0,2} + \varphi_{1,2} r_{t-1} + \varphi_{2,2} r_{t-2} + e_{2t},$$

$$r_t = \varphi_{0,3} + \varphi_{1,3} r_{t-1} + \varphi_{2,3} r_{t-2} + \varphi_{3,3} r_{t-3} + e_{3t},$$

$$r_t = \varphi_{0,4} + \varphi_{1,4} r_{t-1} + \varphi_{2,4} r_{t-2} + \varphi_{3,4} r_{t-3} + \varphi_{4,4} r_{t-4} + e_{4t},$$

…

where $\varphi_{0,j}$, $\varphi_{i,j}$, and $\{e_{jt}\}$ are, respectively, the constant term, the coefficient of r_{t-i}, and the error term of an AR(j) model. These models are in the form of a multiple linear regression and can be estimated by the least squares method. As a matter of fact, they are arranged in a sequential order that enables us to apply the idea of partial F test in multiple linear regression analysis. The estimate $\varphi'_{1,1}$ of the first equation is called the lag-1 sample PACF of r_t . The estimate $\varphi'_{2,2}$ of the second equation is the lag-2 sample PACF of r_t . The estimate $\varphi'_{3,3}$of the third equation is the lag-3 sample PACF of r_t , and so on.

Coefficient	*Standard Deviation*
0.1707	0.0447
0.4684	0.0198

Table 1.AR(1)

Coefficient	*Standard Deviation*
0.1714	0.0449
0.4702	0.0224
-0.0039	0.0224

Table 2.AR(2)

Coefficient	*Standard Deviation*
0.1724	0.0451
0.4702	0.0224
0.0005	0.0248
-0.0092	0.0224

Table 3.AR(3)

The estimated values of the constant and the corresponding coefficients of the relevant model, and the standard deviations are shown in Table 1,2,3. It is easy to choose a model from these estimates since estimates 3 and 4 in table 3 are nearly zero, indicating that it is not AR(3) model but it is AR(2) model .

Information Criteria

There are several information criteria available to determine the order p of an AR process. All of them are likelihood based. For example, the well-known Akaike information criterion (AIC) (Akaike, 1973) is defined as

$$AIC = (-2/T)\ln(\text{likelihood}) + 2T \times (\text{number of parameters}),$$

where the likelihood function is evaluated at the maximum likelihood estimates and T is the sample size. For a Gaussian AR(l) model, AIC reduces to

$$AIC(l) = \ln(\sigma_l^2) + 2l/T,$$

where σ_l^2 is the maximum likelihood estimate of σ_a^2, which is the variance of a_t ,and T is the sample size. The first term of the AIC measures the goodness of fit of the AR(l) model to the data whereas the second term is called the penalty function of the criterion because it penalizes a candidate model by the number of parameters used. Different penalty functions result in different information criteria.

Model	*AIC*
AR(1)	8407.037
AR(2)	8403.827
AR(3)	11500.39

Table 4.Aic Estimation

Table 4 shows the estimated AIC values for the models AR (1), AR(2), and AR (3). The best model is the one with the lowest AIC value. One can see that the model AR(2) is the most appropriate model here as before.

BIBLIOGRAPHY

Analysis Of Financial Time Series, 2nd Edition , by Ruey S. Tsay

The code used for the analysis is as follows

```
# an MA1 model: v(t=)  c + u(t) + theta1*u(t=1), u(t)=N(0,1)
yy=matrix(rep(0,2000),2000)
c=0.1
theta1=0.7 # theta2=0.6
e_i_lag1=0
for  (i  in  2:2000)

{

  e_i=rnorm(1, mean=0, sd=1)

  yy[i]=c+e_i+ theta1*e_i_lag1
  e_i_lag1 = e_i
}
plot(yy,type='l',col="#377eb8",main="MA(1) model",xlab="Time",ylab="y")
legend("topleft",lty=1,col=c("#377eb8"),
       legend=c("y(t)"),cex = 0.6)
```

```
rho=matrix( rep(0,20)   ,   20   )
for  (l in 1:20)
{
  w=autocorr_1(yy,l)
  ff=unlist(w)
  rho[l]=ff[1]
}


plot(rho,type='l',col="#e41a1c",main="autocorrelation function MA(1)",x
legend("topleft",lty=1,col=c("#e41a1c"),
       legend=c("r"),cex = 0.6)
#an AR2 model: v(t)= c + phi"v(t=1) +u(t), u(t)  N(0,sigma)
yy=matrix(rep(0,2000),2000)
c=0.1
phi=0.5

phi2= 0.1
```

```
for  (i  in  3:2000)
{
  yy[i]=c+phi*yy[i-1]+rnorm(1,mean=0,sd=2)

}
plot(yy,type='l')
plot(yy,type='l',col="#377eb8",main="AR(2) model",xlab="Time",ylab="y")
legend("topleft",lty=1,col=c("#377eb8"),
       legend=c("y(t)"),cex = 0.6)
for (l in 1:20)
{
  w=autocorr_1(yy,l)
  ff=unlist(w)
  rho[l]=ff[1]
}
plot(rho,type='l',col="#e41a1c",main="autocorrelation function AR(2)",xlab="Time
legend("topleft",lty=1,col=c("#e41a1c"),
       legend=c("r"),cex = 0.6)
```

```
# estimate an ARE , PACF
T=length(yy)
y=matrix( yy[2:T],T-1)
xo=matrix(yy[1:T-1],T-1)
c=matrix( rep(1,T-1),T-1)
y1=y
x1=cbind(c,xo)
q1=ols_1(y1,x1)
q1
y2=matrix(yy[3:T],T-2)
xo1=matrix(yy[2:(T-1)],T-2)
xo2=matrix(yy[1:(T-2)],T-2)
c=matrix( rep(1,T-2),T-2)
x2=cbind(c,xo1,xo2)
q2=ols_1(y2,x2)
q2
```

```
y3=matrix(yy[4:T],T-3)
xo1=matrix(yy[3:(T-1)],T-3)
xo2=matrix(yy[2:(T-2)],T-3)
xo3=matrix(yy[1:(T-3)],T-3)
c=matrix( rep(1,T-3),T-3)
x3=cbind(c,xo1,xo2,xo3)
q3=ols_1(y3,x3)
q3
```

```
theta0=c(0.5,0.9,5)
test=likel_ar1(theta0, yy)
low=c(0,0,0)
h=c(1,0.99,50)
theta_opt<-optim(theta0,likel_ar1, gr=NULL,y=yy,
                 method=  c( "L-BFGS-B"),
                 lower=low, upper= h,
                 hessian=  TRUE)

log_likel_1=  -theta_opt$value
aic=-(2/length(T)*log_likel_1)+2/length(T)*3
aic
theta0=c(0.5,0.9,0.1,0.9,var(yy))
test=likel_ar2(theta0, yy)
low=c(0,0,0)
h=c(1,0.97,0.97,5)
theta_opt<-optim(theta0,likel_ar2, gr=NULL,y=yy,
                 method=  c( "L-BFGS-B"),
```

```
                  hessian= TRUE)

log_likel_2=-theta_opt$value
aic=-(2/length(T)*log_likel_2)+2/length(T)*3
aic
theta0<- c(0.5,0.9,0.1,0.8,0.4,var(yy))
test<- likel_ar3(theta0,yy)
low<-c(0,0,0)
h<- c(1,0.97,0.97,50)
theta_opt<- optim(theta0,likel_ar3,gr=NULL,y=yy,
                  method = c('L-BFGS-B'),
                  lower = low, upper = h,
                  hessian = T)
log_likel_3<- -theta_opt$value
aic=-(2/length(T)*log_likel_3)+2/length(T)*3
aic
```

```
ols_1 <- function(y,x) {

  # Inverse matrix of x'x
  z1=solve(t(x)%*%x)
  b=z1%*%(crossprod(x,y))

  e = y-x%*%b

  T =length(e)


  k=ncol(x)

  ss=c( crossprod(e,e)/(T-k))
  s1=z1*ss
  std=matrix( sqrt( diag(s1)), k )

  w=list(b,std)
  return(w)
}
```

```
log_norm_pdf1=function(x,m,ss){
  #normal pdf
  f1=(-1/2)*log(ss);
  f2=-((x-m)^2)/(2*ss);

  f=f1+f2;

  return(f)
}
```

```
likel_ar1=function(theta, y){

  T=length(y);
  lik=0;
  for  (l in  2:T)  {
    lik=lik+log(norm_pdf1( y[l],theta[1]+theta[2]*y[l-1], theta[3] ))
  }

  lik1=-lik;
  return(lik1)
}
```

```
autocorr_1<- function(y,l) {

  T=length(y)
  mm=mean(y)
  yy=y-matrix(rep(mm,T),T)


  v=matrix( yy[c((l+1):T)]  , T-l)
  v_l=matrix(yy[1:(T-l)],T-l)

  g_l= crossprod(v,v_l)/(T-1)
  g_0=var(y)

  rho_l=g_l/g_0

  w=list(rho_l,g_l)

  return(w)
```

```
likel_ar2<- function(theta,y){
  T<-length(y);
  lik<-0;
  for (l in 3:T) {
    lik<-lik +log(norm_pdf1(y[l],theta[1]+theta[2]*y[l-1]+theta[3]*y[l-2],theta[4]))

  }
  lik2=-lik;
  return(lik2)
}
```

```
likel_ar3<- function(theta,y){
  T<-length(y);
  lik<-0;
  for (l in 4:T) {
    lik<-lik +log(norm_pdf1(y[l],theta[1]+theta[2]*y[l-1]+theta[3]*y[l-2]+theta[4]*y

  }
  lik3=-lik;
  return(lik3)
}
```