Problem Set 2

Solutions

1 Garch function

```
function [sumloglik,z,cond_var] = garch(par,y)

% NGARCH without conditional mean constant
% Inputs:
%   par: 4x1 vector of NGARCH parameters (omega;alpha;theta;beta)
%   y:   vector of returns

% Outputs:
%   sumloglik: (scalar) value of likelihood function (normal dist)
%   z: vector of standardized returns
%   cond_var: conditional VARIANCE

global cond_var
ret=y;
R=size(ret,1);
cond_var=NaN(R,1);
cond_var(1,1)=var(ret);
cond_var(2,1)=var(ret);

for i=3:R
    cond_var(i,1)=par(1,1)+par(2,1)*(ret(i-1,1)^2)+...
    +par(3,1)*(ret(i-1,1)^2)+par(4,1)*cond_var(i-1,1);
end
z=ret./sqrt(cond_var);

sumloglik=-sum(-0.5*log(2*pi)-0.5*log(cond_var)-0.5*(z.^2));
```

2 Exponential Smoothing function

```
function y = exp_smoothing(lambda,y)
global cond_var_es
ret=y;
```

```
R=size(ret,1);
C=size(ret,2);
global param_ng
cond_var_es=NaN(R,C);
cond_var_es(1,1)=var(ret);
for i=2:R
cond_var_es(i,1)=(1-lambda)*ret(i-1,1).^2+lambda*cond_var_es(i-1,1);
end
z=ret./sqrt(cond_var_es);

y=-sum(-0.5*log(2*pi)-0.5*l
```

3 Leverage Garch function

```
function [sumloglik,z,cond_varng] = ngarch(par,y)

% NGARCH without conditional mean constant
% Inputs:
%   par: 5x1 vector of NGARCH parameters (omega;alpha;theta;beta)
%   y:   vector of returns

% Outputs:
%   sumloglik: (scalar) value of likelihood function (normal dist)
%   z: vector of standardized returns
%   cond_var: conditional VARIANCE

global cond_varng
ret=y;
R=size(ret,1);
cond_varng=NaN(R,1);
cond_varng(1,1)=var(ret);
cond_varng(2,1)=var(ret);
for i=3:R
    cond_varng(i,1)=par(1,1)+par(2,1)*(ret(i-1,1)-...
    -par(3,1)*sqrt(cond_varng(i-1,1)))^2+par(4,1)*cond_varng(i-1,1)+...
    +par(5,1)*cond_varng(i-2,1);
end
z=ret./sqrt(cond_varng);

sumloglik=-sum(-0.5*log(2*pi)-0.5*log(cond_varng)-0.5*(z.^2));
```

4 Matlab Code for Problem Set 2

```
load tsdata2

% rets
us_p=tsdata2(:,3);
ger_p=tsdata2(:,5);
uk_p=tsdata2(:,7);
rets_ger_l=NaN(size(us_p));
rets_us_l=NaN(size(us_p));
rets_uk_l=NaN(size(us_p));
rets_us_eu=NaN(size(us_p));
rets_uk_eu=NaN(size(us_p));
rets_ger_l(2:end,:)=log(ger_p(2:end,:))-log(ger_p(1:end-1,:));
rets_us_l(2:end,:)=log(us_p(2:end,:))-log(us_p(1:end-1,:));
rets_uk_l(2:end,:)=log(uk_p(2:end,:))-log(uk_p(1:end-1,:));

%FOREX
EUvsDOL=tsdata2(:,10);
DOLvsUKP=tsdata2(:,9);
EUvsUKP=EUvsDOL.*DOLvsUKP;
r_EUvsDOL=NaN(size(us_p));
r_EUvsDOL(2:end,:)  = log((tsdata2(2:end,10))./tsdata2(1:end-1,10));
r_DOLvsUKP=NaN(size(us_p));
r_DOLvsUKP(2:end,:)= log((tsdata2(2:end,9))./tsdata2(1:end-1,9));
r_EUvsUKP  = r_EUvsDOL + r_DOLvsUKP;
rets_us_eu(2:end,:)=rets_us_l(2:end,:)+r_EUvsDOL(2:end,:);
rets_uk_eu(2:end,:)=rets_uk_l(2:end,:)+r_EUvsUKP(2:end,:);

w=ones(3,1)./3;                                          % portfolio weights
port_ret=[rets_ger_l, rets_us_eu, rets_uk_eu]*w;         % portfolio = sum of weights*returns
% port_ret=tsdata2(:,3)                                    % portfolio = GER total return


% The first option is to use the sample variance.

ret=port_ret(2:end,:);
cond_var(1,1)=var(ret);

clearvars ret cond_var
% This is a valid measure only if the variance is constant. We
% know that this is not the case in high frequency data to
% realize this, think how would have such a measure performed
% in the assessment of volatility of returns during the financial
% crisis. One of the problem of the sample mean is that recent
% past and far distant past receive the same weight in the
% determination of volatility. exponential volatility is a
% technique that allows to give higher weight to more recent
% observation in the determination of volatilities
```

```
% Daily Variance
  % sigma(t)^2 = y(t)^2
ret=port_ret(2:end,1);
cond_var_base=ret.^2
cond_vol_base=sqrt(cond_var_base)
changevol_base =cond_vol_base(3:end,1)./cond_vol_base(2:end-1,1)-1

clearvars ret


% Exponential smoothing
  % sigma(t+1)^2 = (1-lambda)*y(t)^2 + lambda*sigma(t)^2

global cond_var_es
par_initial=[0.2];
[lambda,mle_es]=fminsearch('smoothing',par_initial,[],port_ret(2:end,1));
cond_vol_es=sqrt(cond_var_es);

t=1:size(cond_vol_es,1)
t=t'
plot(t(3:end,1),cond_vol_base(3:end,1),'.', t(3:end,1),cond_vol_es(3:end,1))
clear global


% Consider the following GARCH MODEL  estimation

 % sigma(t+1)^2 = omega + alpha*y(t)^2 + beta*sigma(t)^2
 % where, y(t)= port_ret;

% 1. Compute the parameters omega;alpha;theta;beta;
% 2. Forcast the volatility based on the parameters estimated from the whole sample;
% 3. Compute the parameters omega;alpha;theta;beta; based on the first half of the sample
% 4. Forcast the volatility based on the parameters estimated from the first half of the sample;

global cond_var
par_initial_ng(1:4,1)=[0.00005;0.1;0.1; 0.8]; % par_init=[omega;alpha;theta;beta]
[param_ng,mle_ng]=fminsearch('garch3',par_initial_ng,[],port_ret(2:end,:));
global param_ng
cond_vol_g=sqrt(cond_var);

clearvars param_ng
clear global


t=1:size(cond_vol_es,1)
t=t'
plot(t(3:end,1),cond_vol_base(3:end,1),'.', t(3:end,1),cond_vol_g(3:end,1))
```

```
% Consider the following GARCH WITH LEVERAGE MODEL NGARCH estimation

 % sigma(t+1)^2 = omega + alpha*(y(t)-theta*sigma(t))^2+beta*sigma(t)^2
 % where, y(t)= port_ret;

% 1. Compute the parameters omega;alpha;theta;beta;
% 2. Forcast the volatility based on the parameters estimated from the whole sample;

opts = optimset('DerivativeCheck','off','Display','off','TolX',1e-6,...
'TolFun',1e-6,'Diagnostics','off','MaxIter',1000,'LargeScale','on');
% seting options for minimizer

global cond_varng
par_initial_ng(1:5,1)=[0.00005;0.1;0.2;0.8;0.4]; % par_init=[omega;alpha;theta;beta]
[param_ng,mle_ng]=fminsearch('ngarch3',par_initial_ng,[],port_ret(2:end,:));
global param_ng
cond_vol_ng=sqrt(cond_varng);


t=1:size(cond_vol_es,1)
t=t'
plot(t(3:end,1),cond_vol_base(3:end,1),'.', t(3:end,1),cond_vol_ng(3:end,1))



clearvars param_ng
clear global



% Consider the following GARCH MODEL  estimation

 % sigma(t+1)^2 = omega + alpha*y(t)^2+beta*sigma(t)^2
 % where, y(t)= port_ret;

% 1. Compute the parameters omega;alpha;theta;beta; based on the first half of the sample
% 2. Forcast the volatility based on the parameters estimated from the first half of the sample;


k=round(size(port_ret,1)/2)
R=size(port_ret(2:end,:),1);

global cond_var
par_initial_ng(1:4,1)=[0.00005;0.1;0.1;0.8]; % par_init=[omega;alpha;theta;beta]
[param_ng,mle_ng]=fminsearch('garch3',par_initial_ng,[],port_ret(2:k,:));
global param_ng
```

```
ret=port_ret(2:R,:)

for i=k:R
    cond_var(i,1)=param_ng(1,1)+param_ng(2,1)*(ret(i-1,1)^2)+...
    +param_ng(3,1)*(ret(i-2,1)^2)+param_ng(4,1)*cond_var(i-1,1);
end

cond_vol_g1=sqrt(cond_var);

plot(t(k:end,1),cond_vol_base(k:end,1),'.', t(k:end,1),cond_vol_g1(k:end,1))



clearvars param_ng ret
clear global


% Consider the following GARCH WITH LEVERAGE MODEL NGARCH estimation

 % sigma(t+1)^2 = omega + alpha*(y(t)-theta*sigma(t))^2+beta*sigma(t)^2
 % where, y(t)= port_ret;

% 1. Compute the parameters omega;alpha;theta;beta; based on the first half of the sample
% 2. Forcast the volatility based on the parameters estimated from the first half of the sample;

k=round(size(port_ret,1)/2)
R=size(port_ret(2:end,:),1);


global cond_varng
par_initial_ng(1:5,1)=[0.00005;0.1;0.2;0.8;0.4]; % par_init=[omega;alpha;theta;beta]
[param_ng,mle_ng]=fminsearch('ngarch3',par_initial_ng,[],port_ret(2:end,:));
global param_ng

ret=port_ret(2:R,:)

for i=k:R
    cond_varng(i,1)=param_ng(1,1)+...
    +param_ng(2,1)*(ret(i-1,1)-param_ng(3,1)*sqrt(cond_varng(i-1,1)))^2+...
    +param_ng(4,1)*cond_varng(i-1,1)+param_ng(5,1)*cond_varng(i-2,1);
end

cond_vol_ng1=sqrt(cond_varng);

plot(t(k:end,1),cond_vol_base(k:end,1),'.', t(k:end,1),cond_vol_ng1(k:end,1))
clear global
```

```
changevol_ng1 =cond_vol_ng1(3:end,1)./cond_vol_ng1(2:end-1,1)-1
changevol_ng =cond_vol_ng(3:end,1)./cond_vol_ng(2:end-1,1)-1
changevol_g =cond_vol_g(3:end,1)./cond_vol_g(2:end-1,1)-1
changevol_g1 =cond_vol_g1(3:end,1)./cond_vol_g1(2:end-1,1)-1


plot(t(k:end-2,1),changevol_base(k:end,1),'.', t(k:end-2,1),changevol_ng1(k:end,1))
plot(t(k:end-2,1),changevol_base(k:end,1),'.', t(k:end-2,1),changevol_ng(k:end,1))
plot(t(k:end-2,1),changevol_base(k:end,1),'.', t(k:end-2,1),changevol_g1(k:end,1))
plot(t(k:end-2,1),changevol_base(k:end,1),'.', t(k:end-2,1),changevol_g(k:end,1))


corr(changevol_base,changevol_ng1)
corr(changevol_base,changevol_ng)
corr(changevol_base,changevol_g)
corr(changevol_base,changevol_g1)
```