```
********* Calculate mean and covariance matrix of 4 factors;
data allreg;set allreg;
if p1=0 and p2=0 and p3=0 then rif0=meanret-meanlilv;
if p1=0 and p2=0 and p3=1 then rif1=meanret-meanlilv;
if p1=0 and p2=0 and p3=2 then rif2=meanret-meanlilv;
if p1=0 and p2=1 and p3=0 then rif3=meanret-meanlilv;
if p1=0 and p2=1 and p3=1 then rif4=meanret-meanlilv;
if p1=0 and p2=1 and p3=2 then rif5=meanret-meanlilv;
if p1=0 and p2=2 and p3=0 then rif6=meanret-meanlilv;
if p1=0 and p2=2 and p3=1 then rif7=meanret-meanlilv;
if p1=0 and p2=2 and p3=2 then rif8=meanret-meanlilv;
if p1=1 and p2=0 and p3=0 then rif9=meanret-meanlilv;
if p1=1 and p2=0 and p3=1 then rif10=meanret-meanlilv;
if p1=1 and p2=0 and p3=2 then rif11=meanret-meanlilv;
if p1=1 and p2=1 and p3=0 then rif12=meanret-meanlilv;
if p1=1 and p2=1 and p3=1 then rif13=meanret-meanlilv;
if p1=1 and p2=1 and p3=2 then rif14=meanret-meanlilv;
if p1=1 and p2=2 and p3=0 then rif15=meanret-meanlilv;
if p1=1 and p2=2 and p3=1 then rif16=meanret-meanlilv;
if p1=1 and p2=2 and p3=2 then rif17=meanret-meanlilv;
if p1=2 and p2=0 and p3=0 then rif18=meanret-meanlilv;
if p1=2 and p2=0 and p3=1 then rif19=meanret-meanlilv;
if p1=2 and p2=0 and p3=2 then rif20=meanret-meanlilv;
if p1=2 and p2=1 and p3=0 then rif21=meanret-meanlilv;
if p1=2 and p2=1 and p3=1 then rif22=meanret-meanlilv;
if p1=2 and p2=1 and p3=2 then rif23=meanret-meanlilv;
if p1=2 and p2=2 and p3=0 then rif24=meanret-meanlilv;
if p1=2 and p2=2 and p3=1 then rif25=meanret-meanlilv;
if p1=2 and p2=2 and p3=2 then rif26=meanret-meanlilv;

run;

data b;
yyyymm=199608;
data a;
%macro a(x);
data a;
set allreg;
where rif&x^=.;
keep rif&x yyyymm rmf smb hml liq;
run;

data b;
merge a b;
by yyyymm;
run;
%mend a;
%a(0);
%a(1);
%a(2);
%a(3);
%a(4);
%a(5);
%a(6);
%a(7);
%a(8);
%a(9);
```

```
%a(10);
%a(11);
%a(12);
%a(13);
%a(14);
%a(15);
%a(16);
%a(17);
%a(18);
%a(19);
%a(20);
%a(21);
%a(22);
%a(23);
%a(24);
%a(25);
%a(26);
run;
/*data allreg;set allreg;
drop meanret meanlilv;
run; */
ods output Cov=fcov;
ods output SimpleStats=fmean;

proc corr data=b cov vardef=n;
var rmf smb hml liq;
run;

data fcov; set fcov;  run;
data fmean; set fmean; run;


********* Use SUR procedure to calculate the residual covariance matrix;

proc syslin data=b sur vardef=n outest=parms;
rank0: model rif0=rmf smb hml liq/covb;
rank1: model rif1=rmf smb hml liq/covb;
rank2: model rif2=rmf smb hml liq/covb;
rank3: model rif3=rmf smb hml liq/covb;
rank4: model rif4=rmf smb hml liq/covb;
rank5: model rif5=rmf smb hml liq/covb;
rank6: model rif6=rmf smb hml liq/covb;
rank7: model rif7=rmf smb hml liq/covb;
rank8: model rif8=rmf smb hml liq/covb;
rank9: model rif9=rmf smb hml liq/covb;
rank10: model rif10=rmf smb hml liq/covb;
rank11: model rif11=rmf smb hml liq/covb;
rank12: model rif12=rmf smb hml liq/covb;
rank13: model rif13=rmf smb hml liq/covb;
rank14: model rif14=rmf smb hml liq/covb;
rank15: model rif15=rmf smb hml liq/covb;
rank16: model rif16=rmf smb hml liq/covb;
rank17: model rif17=rmf smb hml liq/covb;
rank18: model rif18=rmf smb hml liq/covb;
rank19: model rif19=rmf smb hml liq/covb;
rank20: model rif20=rmf smb hml liq/covb;
rank21: model rif21=rmf smb hml liq/covb;
```

```
rank22: model rif22=rmf smb hml liq/covb;
rank23: model rif23=rmf smb hml liq/covb;
rank24: model rif24=rmf smb hml liq/covb;
rank25: model rif25=rmf smb hml liq/covb;
rank26: model rif26=rmf smb hml liq/covb;

ods output InvCovResiduals=rinvcov;
ods select InvCovResiduals;
run;


data parms; set parms (keep=_TYPE_ _MODEL_ Intercept);
where _TYPE_='SUR';
proc print data=parms;
title 'INTERCEPTS for 4-FACTOR REGRESSION';
data rinvcov; set rinvcov;     run;
*** NOTE: rinvcov contains the inverse cov matrix of residuals;


********* Calculate GRS F-Statistics;
```

Many mistakes in the following codes

```
proc iml;

use fcov;
read b   var _num_ into O;
print O;

invO=inv(O); ***Inverse Omega;
print invO;

use fmean;
read b var _num_  into M; ***E(f);
print M;

tM=M`; ***Transpose E(f);
print tM;

tM_invO=tM*invO;
print tM_invO;

tM2_invO=tM_invO*M;
print tM2_invO;

Z=1+tM2_invO;
invZ=inv(Z);

use rinvcov;
read b var _num_ into invS;
print invS;

use parms;
read b var _num_ into A;
print A;

tA=A`;
```

```
print tA;

tA_invS=tA*invS;
print tA_invS;

tA2_invS=tA_invS*A;
print tA2_invS;

X=(T-N-K)/N; ***Need to plug in figure manually;
F1=X*invZ*tA2_invS;
print F1;
```