1. Installation of R

For windows: <https://cran.r-project.org/bin/windows/base/>

For mac: <https://cran.r-project.org/bin/macosx/>

Install R-studio

<https://www.rstudio.com/products/rstudio/download/>

note: R-studio is using R. Make sure install R first then R-studio.

1. In your computer, set up a new folder (for example, a folder named as GMM in your C drive).

Then you should put our code script (gmm.r) into the folder as well as the datasets.

1. Open R-studio, click File, then click Open script, and choose our GMM code

(you can then also click New script, and copy the GMM code into a new one if you need to change some moment conditions and dataset).

1. Change the working directory in our code to your own file directory using the “setwd” function

(for example, our current one is setwd(“C:/GMM/”), you can change to setwd(“D:/Tom’s course/GMM/”) (the destination of your folder)). Put the data file “example1.csv” in your working folder.

1. Then install packages (simply by using the install.packages("sandwich") and install.packages("optimx") functions), and then call these two packages by using library(sandwich) and library(optimx). In the code, just simply delete”#” from line 2 to line 6 and run these lines. After all packages installed, enter back “#”.
2. Follow the instructions in the code to define settings from line 10 to line 40 including read in files. The data file name is “ example1.csv”. It has 13 variables, Date, vwretd, ewretd, d1-d10. vwretd: value-weighted return; ewretd: equal-weighted return; d1-d10: 10 portfolio return. The data is stored in the variable name “data1”. The risk free rate data is “example2.csv”. We read in as “data2”.
3. Define DIFSTAT. If DIFSTAT=1, unrestricted model and restricted model are both estimated. If DIFSTAT=0, only one restricted model is estimated.
4. Set up an initial guess for all of your Parameters. Set up iniPU and iniPR. If only one restricted model is estimated, set them equal.
5. From line 48 to line 51, set up moment conditions for restricted model: “PR” is a vector of all parameters: 10 alpha and 10 betas. Z matrix stores all moment condition residuals. g stores all moment condition means, Q is the objective error. This function “FUN2” returns all moment conditions mean, all residuals, and one objective error.
6. From line 67 to 70, set up moment conditions for unrestricted model. If no unrestricted model, this part doesn’t matter. “PU” is a vector of all parameters: 10 betas. Z matrix stores all moment condition residuals. g stores all moment condition means, Q is the objective error. This function “FUN1” returns all moment conditions mean, all residuals, and one objective error.
7. After all setting, select all code, run it.

**Example1: For CAPM, we should set following variables:**

NOBS=924; ###Number of observations in the input data set

N=20; ###Number of parameters to be estimated (unrestricted), if there's no unrestricted model, define it same to restrited model

NR=10; ###Number of parameters to be estimated (restricted)

M=20; ###Number of Moment conditions

DIFSTAT=1;

#Moment Conditions:

###objective function2: restricted model with NR parameters

FUN2=function(PR){ ###PR:Parameters for restricted model, for example, 10 betas

Z=matrix(0,nrow(data1),M);

for ( i in 1:10){ ### moment conditions for restricted model

Z[,i]=as.matrix(data1[,3+i]-data1$rf)-PR[i]\*(data1$vwretd-data1$rf);}

for ( i in 11:20){

Z[,i]=Z[,i-10]\*(data1$vwretd-data1$rf);}

g=colMeans(Z);

Q=0;

for (i in 1:N){

Q=Q+mean(Z[,i])^2\*WEGR[i]

}

result=list("residual"=Z,"moments"=g,"obj"=Q);

return(result);

}

###function1: unrestricted model with N parameters

FUN1=function(PU){ ###PU: Parameters for unrestricted model, for example, 10 alphas and 10 betas

Z=matrix(0,nrow(data1),M);

for ( i in 1:10){ ### moment conditions for unrestricted model

Z[,i]=as.matrix(data1[,3+i]-data1$rf)-PU[i]-PU[i+10]\*(data1$vwretd-data1$rf);}

for ( i in 11:20){

Z[,i]=Z[,i-10]\*(data1$vwretd-data1$rf);}

g=colMeans(Z);

Q=0;

for (i in 1:N){

Q=Q+mean(Z[,i])^2\*WEG[i]

}

result=list("residual"=Z,"moments"=g,"obj"=Q);

return(result);

}

**Example2: For Black CAPM:**

NOBS=924; ###Number of observations in the input data set

N=11; ###Number of parameters to be estimated (unrestricted), if there's no unrestricted model, define it same to restrited model

NR=11; ###Number of parameters to be estimated (restricted)

M=20; ###Number of Moment conditions

DIFSTAT=0;

#Moment Conditions:

###objective function2: restricted model with NR parameters

FUN2=function(PR){ ###PR:Parameters for restricted model, for example, 10 betas

Z=matrix(0,nrow(data1),M);

for ( i in 1:10){ ### moment conditions for restricted model

Z[,i]=as.matrix(data1[,3+i])-(1-PR[i])\*PR[11]-PR[i]\*data1$vwretd;}

for ( i in 11:20){

Z[,i]=Z[,i-10]\*data1$vwretd;}

g=colMeans(Z);

Q=0;

for (i in 1:N){

Q=Q+mean(Z[,i])^2\*WEGR[i]

}

result=list("residual"=Z,"moments"=g,"obj"=Q);

return(result);

}

No need to change FUN1

**Example3: For risk aversion coefficient**

NOBS=924; ###Number of observations in the input data set

N=1; ###Number of parameters to be estimated (unrestricted), if there's no unrestricted model, define it same to restrited model

NR=1; ###Number of parameters to be estimated (restricted)

M=10; ###Number of Moment conditions

DIFSTAT=0;

####add in the ratios

ratiom<-(1+data1$vwretd)/(1+data2$bid/1200);

ratio1<-(1+data1$d1)/(1+data2$bid/1200);

ratio2<-(1+data1$d2)/(1+data2$bid/1200);

ratio3<-(1+data1$d3)/(1+data2$bid/1200);

ratio4<-(1+data1$d4)/(1+data2$bid/1200);

ratio5<-(1+data1$d5)/(1+data2$bid/1200);

ratio6<-(1+data1$d6)/(1+data2$bid/1200);

ratio7<-(1+data1$d7)/(1+data2$bid/1200);

ratio8<-(1+data1$d8)/(1+data2$bid/1200);

ratio9<-(1+data1$d9)/(1+data2$bid/1200);

ratio10<-(1+data1$d10)/(1+data2$bid/1200);

data1<-cbind(data1,ratiom,data1,ratio1,ratio2,ratio3,ratio4,ratio5,ratio6,ratio7,ratio8,ratio9,ratio10);

###Moment Conditions

FUN2=function(PR){ ###PR:Parameters for restricted model, for example, 10 betas

Z=matrix(0,nrow(data1),M); ### moment conditions for restricted model

Z[,1]<-(data1$ratio1-1)\*(data1$ratiom^PR)

Z[,2]<-(data1$ratio2-1)\*(data1$ratiom^PR)

Z[,3]<-(data1$ratio3-1)\*(data1$ratiom^PR)

Z[,4]<-(data1$ratio4-1)\*(data1$ratiom^PR)

Z[,5]<-(data1$ratio5-1)\*(data1$ratiom^PR)

Z[,6]<-(data1$ratio6-1)\*(data1$ratiom^PR)

Z[,7]<-(data1$ratio7-1)\*(data1$ratiom^PR)

Z[,8]<-(data1$ratio8-1)\*(data1$ratiom^PR)

Z[,9]<-(data1$ratio9-1)\*(data1$ratiom^PR)

Z[,10]<-(data1$ratio10-1)\*(data1$ratiom^PR)

g=colMeans(Z);

Q=0;

for (i in 1:N){

Q=Q+mean(Z[,i])^2\*WEGR[i]

}

result=list("residual"=Z,"moments"=g,"obj"=Q);

return(result);

}

No need to change FUN1

**Assignment2:**

**Q1:**

**Read in consumption data :**

data1=read\_csv("example1.csv");

###example1: value-we return, equal-we return and 10 deciles return

###variables: vwretd: value-weighted return; ewretd: equal-weighted return; d1-d10: 10 portfolio return.

data2=read\_csv("example2.csv"); ## risk-free rate

data1$rf=data2$bid/1200;

data1$ratiom=(1+data1$vwretd)/(1+data2$bid/1200);

data3=read\_csv("a2q1.csv");

data3=data3 %>% mutate(deltac=c/lag(c)); ##consumption

data1=subset(data1,Date>19590100);

data3=subset(data3,month<2003);

data1$deltac=data3$deltac;

data1=na.omit(data1);

NOBS=nrow(data1);

**Moment conditions:**

Z=matrix(0,nrow(data1),M);

for ( i in 1:10){ ### moment conditions for restricted model

Z[,i]=as.matrix(PR[1]\*(data1[,3+i]-data1$rf+1)\*data1$deltac^(-PR[2])-1);}

**Q2:**

**Read in consumption data :**

data1=read\_csv("example1.csv");

###example1: value-we return, equal-we return and 10 deciles return

###variables: vwretd: value-weighted return; ewretd: equal-weighted return; d1-d10: 10 portfolio return.

data2=read\_csv("example2.csv"); ## risk-free rate

data1$rf=data2$bid/1200;

data1$ratiom=(1+data1$vwretd)/(1+data2$bid/1200);

data3=read\_csv("a2q1.csv");

data3=data3 %>% mutate(deltac=c/lag(c)); ##consumption

data1=subset(data1,Date>19590100);

data3=subset(data3,month<2003);

data1$deltac=data3$deltac;

data1=data1 %>% mutate(lagdeltac=lag(deltac));

data1=data1 %>% mutate(lagrm=lag(vwretd)-lag(rf));

data1=na.omit(data1);

NOBS=nrow(data1);

**Moment conditions:**

Z=matrix(0,nrow(data1),M);

for ( i in 1:5){ ### moment conditions for restricted model

Z[,i]=as.matrix((PR[1]\*(data1[,2+2\*i]-data1$rf+1)\*data1$deltac^(-PR[2])-1)\*data1$lagdeltac);}

for ( i in 1:5){

Z[,i+5]=as.matrix((PR[1]\*(data1[,2+2\*i]-data1$rf+1)\*data1$deltac^(-PR[2])-1)\*data1$lagrm);}

**Q3:**

**Moment Conditions:**

Z=matrix(0,nrow(data1),M);

for ( i in 1:10){ ### moment conditions for restricted model

Z[,i]=as.matrix(data1[,3+i]-data1$rf-PR[i]\*(data1$vwretd-data1$rf));}

for ( i in 1:10){

Z[,i+10]=as.matrix(PR[i]\*((data1$vwretd-data1$rf)^2-PR[11]^2)-(data1$vwretd-data1$rf-PR[11])\*(data1[,3+i]-data1$rf));}

Z[,21]=as.matrix(data1$vwretd-data1$rf-PR[11]);