# 软件操作和程序代码手册

**第三章 系统性风险的测度**

**1、DD的计算**

使用EXCEL 2013预处理数据，使用STATA 12.0合并数据（merge）然后导出数据供Matlab备用，使用Matlab 2015b迭代计算DD。STATA的数据处理代码见附录code\_for\_DD.do，Matlab的DD迭代算法见KMV.m，KMVfun.m和KMVOptSearch.m。得到季度的DD序列后，以日度的市值序列为基准，通过比例差值把将季度DD序列转换为日度DD序列。

**（1）code\_for\_DD.do**

/\*初始化合并KMVdata.xls\*/

insheet using KMVdata1.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-9,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

gen year=year(date)

gen month=month(date)

gen quarter=quarter(date)

order stockid stockname date year month quarter

sort stockid date year month quarter

egen id=group(stockid)

by id,sort:gen num=\_n

tsset id num

egen id\_stockid\_yq=group(stockid year quarter)

sort id num

by id:gen ret=ln(close/l.close)

sort stockid date year month quarter

by id\_stockid\_yq,sort:egen sigma=sd(ret)

gen sigma\_yq=sigma\*sqrt(62.5)

by id\_stockid\_yq,sort:egen stockvalue\_yq=mean(stockvalue)

save KMVdata1.dta,replace

insheet using KMVdata2.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-9,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

gen year=year(date)

gen month=month(date)

gen quarter=quarter(date)

order stockid stockname date year month quarter

sort stockid date year month quarter

replace sd=sd\*10000

replace ld=ld\*10000

gen dp=sd+0.5\*ld

save KMVdata2.dta,replace

insheet using KMVdata3.csv,clear

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

gen year=year(date)

gen month=month(date)

gen quarter=quarter(date)

sort year month quarter

egen id=group(year quarter)

by id,sort:egen rf\_year\_mean=mean(rf)

duplicates drop id,force

drop id date month

gen rf\_yq=rf\_year\_mean/4

keep year quarter rf\_yq

order year quarter rf\_yq

export excel using KMVdata3.xlsx,replace firstrow(variables)

save KMVdata3.dta,replace

//使用EXCEL补全季度rf数据

insheet using KMVdata3\_add.csv,clear

replace rf\_yq=rf\_yq/100

drop if year<1990

save KMVdata3.dta,replace

//合并进程

use KMVdata1,clear

sort stockid year quarter

merge m:1 stockid year quarter using KMVdata2.dta

keep if \_merge==3

drop \_merge

sort stockid date year month quarter

drop id num id\_stockid\_yq

merge m:1 year quarter using KMVdata3.dta

keep if \_merge==3

drop \_merge

sort stockid date year month quarter

save KMVdata.dta,replace

//将合并的数据准备给Matlab处理

use KMVdata.dta,clear

sort stockid date year month quarter

egen id\_stockid\_yq=group(stockid year quarter)

duplicates drop id\_stockid\_yq,force

drop date month close stockvalue ret sigma sd ld

save KMVdata\_for\_Matlab.dta,replace

export excel using KMVdata\_for\_Matlab.xls,replace firstrow(variables)

\*\* //Matlab算完DD后返回KMVresult文件// \*\*

//补充BV数据

insheet using KMVdata2-2.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-9,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

gen year=year(date)

gen month=month(date)

gen quarter=quarter(date)

order stockid stockname date year month quarter

sort stockid date year month quarter

replace bv=bv\*10000

sort stockid year quarter

egen id\_stockid\_yq=group(stockid year quarter)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year2 quarter2

sort stockid yearqt

save KMVdata2-2.dta,replace

//将返回KMVresult文件处理成季度的以及日度的DD初步文件

insheet using KMVresult.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-6,6)

sort stockid year quarter

egen id\_stockid\_yq=group(stockid year quarter)

save KMVdata\_quarterly.dta,replace

use KMVdata.dta,clear

sort stockid date year month quarter

egen id\_stockid\_yq=group(stockid year quarter)

by id\_stockid\_yq,sort:gen num=\_n

by id\_stockid\_yq,sort:gen nummax=\_N

by id\_stockid\_yq,sort:gen numdelta=num-nummax

merge m:1 year id\_stockid\_yq using KMVdata\_quarterly.dta

drop \_merge

sort stockid date year month quarter

replace av=. if numdelta!=0

replace dd=. if numdelta!=0

replace edf=. if numdelta!=0

tsset id\_stockid\_yq num

sort id\_stockid\_yq num

save KMVdata\_daily.dta,replace

export excel using KMVdata\_daily.xls,replace firstrow(variables)

//\*季度DD数据处理进程\*//

/\*处理季度的DD数据并保存为DD\_quarterly.dta\*/

use KMVdata\_quarterly.dta,clear

merge 1:1 stockid year quarter using KMVdata2-2.dta

keep if \_merge==3

drop \_merge

gen mvtbv=sv/bv

gen banktype=.

replace banktype=1 if stockid=="601398"|stockid=="601288"|stockid=="601988"|stockid=="601939"|stockid=="601328"

replace banktype=2 if stockid=="601818"|stockid=="000001"|stockid=="600015"|stockid=="600016"|stockid=="600000"|stockid=="601166"|stockid=="600036"|stockid=="601998"

replace banktype=3 if stockid=="601169"|stockid=="601009"|stockid=="002142"

drop yearqt

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year2 quarter2

sort stockid yearqt

sort banktype yearqt stockid

egen id\_banktype\_yqt=group(banktype yearqt)

by id\_banktype\_yqt,sort:egen add=mean(dd)

by id\_banktype\_yqt,sort:gen num=\_n

by id\_banktype\_yqt,sort:egen sum\_sv=sum(sv)

by id\_banktype\_yqt,sort:gen sv\_p=sv/sum\_sv

by id\_banktype\_yqt,sort:egen wdd=sum(sv\_p\*dd)

by id\_banktype\_yqt,sort:egen sum\_av=sum(av)

by id\_banktype\_yqt,sort:gen av\_p=av/sum\_av

sort yearqt stockid

egen id\_bankall\_yqt=group(yearqt)

by id\_bankall\_yqt,sort:egen add\_all=mean(dd)

by id\_bankall\_yqt,sort:gen num\_all=\_n

by id\_bankall\_yqt,sort:egen sum\_sv\_all=sum(sv)

by id\_bankall\_yqt,sort:gen sv\_p\_all=sv/sum\_sv\_all

by id\_bankall\_yqt,sort:egen wdd\_all=sum(sv\_p\_all\*dd)

by id\_bankall\_yqt,sort:egen sum\_av\_all=sum(av)

by id\_bankall\_yqt,sort:gen av\_p\_all=av/sum\_av\_all

save DD\_quarterly.dta,replace

\*\*\*\*duplicates drop id\_banktype\_yqt,force

\*\*\*\*sort id\_banktype\_yqt

//\*日度DD数据处理进程\*//

/\*先通过EXCEL插值处理日度的DD数据，然后再导入KMVdata\_daily\_add.csv并处理，最后保存为DD\_daily.dta\*/

insheet using KMVdata\_daily\_add.csv,clear

rename stockvalue sv

rename stockvalue\_yq sv\_yq

tostring stockid,replace

replace stockid=substr("000000"+stockid,-6,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order stockid date year month quarter

sort stockid date year month quarter

drop num

egen id=group(stockid)

by id,sort:gen num=\_n

tsset id num

by id:gen ret\_av=ln(av/l.av)

sort stockid date year month quarter

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year year2 quarter\*

sort stockid yearqt date

merge m:1 stockid yearqt using KMVdata2-2.dta

keep if \_merge==3

drop \_merge

gen mvtbv=sv/bv

gen banktype=.

replace banktype=1 if stockid=="601398"|stockid=="601288"|stockid=="601988"|stockid=="601939"|stockid=="601328"

replace banktype=2 if stockid=="601818"|stockid=="000001"|stockid=="600015"|stockid=="600016"|stockid=="600000"|stockid=="601166"|stockid=="600036"|stockid=="601998"

replace banktype=3 if stockid=="601169"|stockid=="601009"|stockid=="002142"

sort banktype date stockid

egen id\_banktype\_date=group(banktype date)

by id\_banktype\_date,sort:egen add=mean(dd)

by id\_banktype\_date,sort:gen num=\_n

by id\_banktype\_date,sort:egen sum\_sv=sum(sv)

by id\_banktype\_date,sort:gen sv\_p=sv/sum\_sv

by id\_banktype\_date,sort:egen wdd=sum(sv\_p\*dd)

by id\_banktype\_date,sort:egen sum\_av=sum(av)

by id\_banktype\_date,sort:gen av\_p=av/sum\_av

sort date stockid

egen id\_bankall\_date=group(date)

by id\_bankall\_date,sort:egen add\_all=mean(dd)

by id\_bankall\_date,sort:gen num\_all=\_n

by id\_bankall\_date,sort:egen sum\_sv\_all=sum(sv)

by id\_bankall\_date,sort:gen sv\_p\_all=sv/sum\_sv\_all

by id\_bankall\_date,sort:egen wdd\_all=sum(sv\_p\_all\*dd)

by id\_bankall\_date,sort:egen sum\_av\_all=sum(av)

by id\_bankall\_date,sort:gen av\_p\_all=av/sum\_av\_all

save DD\_daily.dta,replace

//给16个银行编号

use DD\_daily,clear

duplicates drop stockid,force

keep banktype stockid stockname

sort banktype stockid

gen stock\_serialnumber=\_n

save stock\_serialnumber.dta,replace

export excel using stock\_serialnumber.xls,replace firstrow(variables)

use DD\_daily,clear

merge m:1 stockid using stock\_serialnumber.dta

drop \_merge

save DD\_daily.dta,replace

use DD\_quarterly,clear

merge m:1 stockid using stock\_serialnumber.dta

drop \_merge

save DD\_quarterly.dta,replace

\*\* open do file: sigmaP\_solver.do //打开并执行sigmaP\_solver的DO文件

\*\* open do file: pdd\_generator.do //打开并执行pdd\_generator的DO文件

\*\*\*\*\*\*DD保存点\*\*\*\*\*\*

**（2）KMVfun.m**

%% KMVfun过程函数：生成d1、d2，生成两个非线性方程用于优化求解。

% KMV Main Function

% code by SCY 2015-3-31

%数据导入

clear

[ndata, text, alldata]=xlsread('KMVdata\_for\_Matlab.xls');

alldata(1,:)=[];

stockid=alldata(:,1);

stockname=alldata(:,2);

year=alldata(:,3);

quarter=alldata(:,4);

sigmaSV=alldata(:,5);

SV=alldata(:,6);

DP=alldata(:,7);

rf=alldata(:,8);

id=alldata(:,9);

%循环迭代

for i=1:size(id)

DPtemp=DP{i};

SVtemp=SV{i};

sigmaSVtemp=sigmaSV{i};

rftemp=rf{i};

[AVtemp,sigmaAVtemp]=KMVOptSearch(SVtemp,DPtemp,rftemp,sigmaSVtemp)

AV{i}=AVtemp;

sigmaAV{i}=sigmaAVtemp;

DDtemp=(AVtemp-DPtemp)/(AVtemp\*sigmaAVtemp);

DD{i}=DDtemp;

EDF{i}=normcdf(-DDtemp);

end

AV2=AV';

sigmaAV2=sigmaAV';

DD2=DD';

EDF2=EDF';

%结果输出

xlswrite('KMV迭代结果.xls', {'股票代码'},'Sheet1','A1');

xlswrite('KMV迭代结果.xls', {'股票名称'},'Sheet1','B1');

xlswrite('KMV迭代结果.xls', {'年份'},'Sheet1','C1');

xlswrite('KMV迭代结果.xls', {'季度'},'Sheet1','D1');

xlswrite('KMV迭代结果.xls', {'无风险利率'},'Sheet1','E1');

xlswrite('KMV迭代结果.xls', {'股权价值'},'Sheet1','F1');

xlswrite('KMV迭代结果.xls', {'股权价值波动率'},'Sheet1','G1');

xlswrite('KMV迭代结果.xls', {'资产价值'},'Sheet1','H1');

xlswrite('KMV迭代结果.xls', {'资产价值波动率'},'Sheet1','I1');

xlswrite('KMV迭代结果.xls', {'违约点DP'},'Sheet1','J1');

xlswrite('KMV迭代结果.xls', {'违约距离DD'},'Sheet1','K1');

xlswrite('KMV迭代结果.xls', {'违约概率EDF'},'Sheet1','L1');

xlswrite('KMV迭代结果.xls', stockid,'Sheet1','A2');

xlswrite('KMV迭代结果.xls', stockname,'Sheet1','B2');

xlswrite('KMV迭代结果.xls', year,'Sheet1','C2');

xlswrite('KMV迭代结果.xls', quarter,'Sheet1','D2');

xlswrite('KMV迭代结果.xls', rf,'Sheet1','E2');

xlswrite('KMV迭代结果.xls', SV,'Sheet1','F2');

xlswrite('KMV迭代结果.xls', sigmaSV,'Sheet1','G2');

xlswrite('KMV迭代结果.xls', AV2,'Sheet1','H2');

xlswrite('KMV迭代结果.xls', sigmaAV2,'Sheet1','I2');

xlswrite('KMV迭代结果.xls', DP,'Sheet1','J2');

xlswrite('KMV迭代结果.xls', DD2,'Sheet1','K2');

xlswrite('KMV迭代结果.xls', EDF2,'Sheet1','L2');

**（3）KMVOptSearch.m**

%% KMVOptSearch优化求解函数：使用fsolve命令，调用KMVfun.m，采用最小二乘法来求解非线性方程，输出银行资产价值VA和其波动率给主程序。

function [AV,sigmaAV]=KMVOptSearch(SV,DP,rf,sigmaSV)

% KMVOptSearch

% code by SCY 2015-3-31

SVtoDP=SV/DP;

x0=[1,1];%搜索初始点

result=fsolve(@(x) KMVfun(SVtoDP,rf,sigmaSV,x),x0);

AV=result(1)\*SV;

sigmaAV=result(2);

%AV代表资产价值，sigmaAV代表资产价值的波动率

**（4）KMV.m**

%% KMV主程序：包含数据输入模块，迭代模块（调用优化求解程序KMVOptSearch.m），数据输出模块（根据迭代求解得到的资产价值VA和其波动率，计算DD，并输出到EXCEL文件）。

% KMV Main Function

% code by SCY 2015-3-31

%数据导入

clear

[ndata, text, alldata]=xlsread('KMVdata\_for\_Matlab.xls');

alldata(1,:)=[];

stockid=alldata(:,1);

stockname=alldata(:,2);

year=alldata(:,3);

quarter=alldata(:,4);

sigmaSV=alldata(:,5);

SV=alldata(:,6);

DP=alldata(:,7);

rf=alldata(:,8);

id=alldata(:,9);

%循环迭代

for i=1:size(id)

DPtemp=DP{i};

SVtemp=SV{i};

sigmaSVtemp=sigmaSV{i};

rftemp=rf{i};

[AVtemp,sigmaAVtemp]=KMVOptSearch(SVtemp,DPtemp,rftemp,sigmaSVtemp)

AV{i}=AVtemp;

sigmaAV{i}=sigmaAVtemp;

DDtemp=(AVtemp-DPtemp)/(AVtemp\*sigmaAVtemp);

DD{i}=DDtemp;

EDF{i}=normcdf(-DDtemp);

end

AV2=AV';

sigmaAV2=sigmaAV';

DD2=DD';

EDF2=EDF';

%结果输出

xlswrite('KMV迭代结果.xls', {'股票代码'},'Sheet1','A1');

xlswrite('KMV迭代结果.xls', {'股票名称'},'Sheet1','B1');

xlswrite('KMV迭代结果.xls', {'年份'},'Sheet1','C1');

xlswrite('KMV迭代结果.xls', {'季度'},'Sheet1','D1');

xlswrite('KMV迭代结果.xls', {'无风险利率'},'Sheet1','E1');

xlswrite('KMV迭代结果.xls', {'股权价值'},'Sheet1','F1');

xlswrite('KMV迭代结果.xls', {'股权价值波动率'},'Sheet1','G1');

xlswrite('KMV迭代结果.xls', {'资产价值'},'Sheet1','H1');

xlswrite('KMV迭代结果.xls', {'资产价值波动率'},'Sheet1','I1');

xlswrite('KMV迭代结果.xls', {'违约点DP'},'Sheet1','J1');

xlswrite('KMV迭代结果.xls', {'违约距离DD'},'Sheet1','K1');

xlswrite('KMV迭代结果.xls', {'违约概率EDF'},'Sheet1','L1');

xlswrite('KMV迭代结果.xls', stockid,'Sheet1','A2');

xlswrite('KMV迭代结果.xls', stockname,'Sheet1','B2');

xlswrite('KMV迭代结果.xls', year,'Sheet1','C2');

xlswrite('KMV迭代结果.xls', quarter,'Sheet1','D2');

xlswrite('KMV迭代结果.xls', rf,'Sheet1','E2');

xlswrite('KMV迭代结果.xls', SV,'Sheet1','F2');

xlswrite('KMV迭代结果.xls', sigmaSV,'Sheet1','G2');

xlswrite('KMV迭代结果.xls', AV2,'Sheet1','H2');

xlswrite('KMV迭代结果.xls', sigmaAV2,'Sheet1','I2');

xlswrite('KMV迭代结果.xls', DP,'Sheet1','J2');

xlswrite('KMV迭代结果.xls', DD2,'Sheet1','K2');

xlswrite('KMV迭代结果.xls', EDF2,'Sheet1','L2');

**2、PDD的计算**

银行资产组合收益率的波动率：的求解过程非常复杂，也是PDD求解最关键的地方，本文编写了专门的代码程序用于计算，编程软件为STATA 12.0，计算过程见sigmaP\_solver.do。PDD的计算过程见pdd\_generator.do。求解具体的编程思路是：

STEP1 先将16个上市银行进行编号，然后将银行资产收益率的面板数据进行reshape转换，变为按时间同列的16个时间序列数据。

STEP2 设置双重循环（i=1/15，j=2/16，同时递增，步长为1），并按季节分组，使用statsby命令将16个上市银行的资产收益率两两交叉求解得到两个银行间的资产收益率协方差序列cov\_i\_j（季度数据），在内循环中将文件命名为covdata\_i\_j.dta，分散输出到各个文件中。

STEP3 设置单层循环（i=1/16），并按季节分组，使用statsby命令将16个上市银行的资产收益率进行自我交叉，求得单个银行的资产收益率方差序列cov\_i\_i，在循环中将文件命名为covdata\_i\_i.dta，分散输出到各个文件中。

STEP4 合并（merge）各资产收益率序列的协方差和方差文件，保存为COV\_result.dta。

STEP5 将银行资产价值权数的面板数据进行reshape转换，变为按时间同列的16个时间序列数据。设置两个算子：

算子一：2\*银行i的资产价值权重\*银行j的资产价值权重\*银行i和银行j的资产收益协方差cov\_i\_j

算子二：银行i的资产价值权重\*银行i的资产价值权重\*银行i的资产收益方差cov\_i\_i

STEP6 设置双重循环，对算子一进行加总，得到sigmap\_part1；设置单层循环，对算子二进行加总，得到sigmap\_part2。最后将两个算子的和加总再开方，得到sigmaP。完毕。

**（1）sigmaP\_solver.do**

//生产COV数据，用于进一步的协方差计算

use DD\_daily,clear

keep date stock\_serialnumber ret\_av

sort date stock\_serialnumber

reshape wide ret\_av,i(date) j(stock\_serialnumber)

save ret\_av\_by\_stock,replace

use DD\_daily,clear

sort banktype yearqt stockid date

egen id\_banktype\_yqt=group(banktype yearqt)

keep date banktype yearqt id\_banktype\_yqt stock\_serialnumber stockid ret\_av

sort date

merge m:1 date using ret\_av\_by\_stock.dta

drop \_merge

sort banktype yearqt date

drop banktype yearqt stock\_serialnumber stockid

drop date ret\_av

save COV\_compute.dta,replace

//协方差矩阵计算过程

use COV\_compute,clear

set more off

forvalues i=1/15 {

global t=`i'+1

forvalues j=$t/16 {

statsby,by(id\_banktype\_yqt) clear: cor ret\_av`i' ret\_av`j',c

rename cov\_12 cov\_`i'\_`j'

keep id\_banktype\_yqt cov\_`i'\_`j'

save covdata\_`i'\_`j'.dta,replace

use COV\_compute,clear

}

}

use COV\_compute,clear

set more off

forvalues i=1/16 {

statsby,by(id\_banktype\_yqt) clear: cor ret\_av`i' ret\_av`i',c

rename cov\_12 cov\_`i'\_`i'

keep id\_banktype\_yqt cov\_`i'\_`i'

save covdata\_`i'\_`i'.dta,replace

use COV\_compute,clear

}

use COV\_compute,clear

keep id\_banktype\_yqt

duplicates drop id\_banktype\_yqt,force

forvalues i=1/15 {

global t=`i'+1

forvalues j=$t/16 {

merge 1:1 id\_banktype\_yqt using covdata\_`i'\_`j'.dta

drop \_merge

save COV\_result.dta,replace

}

}

forvalues i=1/16 {

merge 1:1 id\_banktype\_yqt using covdata\_`i'\_`i'.dta

drop \_merge

save COV\_result.dta,replace

}

//生成sigmaP

use DD\_quarterly,clear

keep id\_banktype\_yqt stock\_serialnumber av\_p

sort id\_banktype\_yqt stock\_serialnumber

reshape wide av\_p,i(id\_banktype\_yqt) j(stock\_serialnumber)

save av\_p\_by\_stock,replace

merge 1:1 id\_banktype\_yqt using COV\_result.dta

drop \_merge

save sigmaP.dta,replace

forvalues i=1/15 {

global t=`i'+1

forvalues j=$t/16 {

gen part1\_`i'\_`j'=2\*av\_p`i'\*av\_p`j'\*cov\_`i'\_`j'

}

}

forvalues i=1/16 {

gen part2\_`i'\_`i'=av\_p`i'\*av\_p`i'\*cov\_`i'\_`i'

}

egen part1=rowtotal(part1\_\*)

egen part2=rowtotal(part2\_\*)

gen sigmap=sqrt(part1+part2)

keep id\_banktype\_yqt sigmap

save sigmaP.dta,replace

**（2）pdd\_generator.do**

//DD\_daily的PDD计算

use DD\_daily,clear

sort banktype yearqt stockid

egen id\_banktype\_yqt=group(banktype yearqt)

merge m:1 id\_banktype\_yqt using sigmaP.dta

drop \_merge

sort banktype date stockid

by id\_banktype\_date,sort:egen pdd\_dp=sum(dp)

by id\_banktype\_date,sort:egen pdd\_av=sum(av)

gen pdd=(ln(pdd\_av/pdd\_dp)+(rf-sigmap^2))/sigmap

sort date stockid

by id\_bankall\_date,sort:egen pdd\_dp\_all=sum(dp)

by id\_bankall\_date,sort:egen pdd\_av\_all=sum(av)

gen pdd\_all=(ln(pdd\_av\_all/pdd\_dp\_all)+(rf-sigmap^2))/sigmap

save DD\_daily,replace

//DD\_quarterly的PDD计算

use DD\_quarterly,clear

sort banktype yearqt stockid

merge m:1 id\_banktype\_yqt using sigmaP.dta

drop \_merge

sort banktype yearqt stockid

by id\_banktype\_yqt,sort:egen pdd\_dp=sum(dp)

by id\_banktype\_yqt,sort:egen pdd\_av=sum(av)

gen pdd=(ln(pdd\_av/pdd\_dp)+(rf-sigmap^2))/sigmap

sort yearqt stockid

by id\_bankall\_yqt,sort:egen pdd\_dp\_all=sum(dp)

by id\_bankall\_yqt,sort:egen pdd\_av\_all=sum(av)

gen pdd\_all=(ln(pdd\_av\_all/pdd\_dp\_all)+(rf-sigmap^2))/sigmap

save DD\_quarterly,replace

**3、LRSQ的计算**

具体的编程思路是：

STEP1 由银行i在季度t的每天d的违约距离计算每天的变化。

STEP2 通过单层循环的方式，对于银行i，生成除i以外的其他银行的违约距离变化的加总值，再除以其他银行的个数（注意如果某个其他银行k的数据缺失的话，它将不被包括在内）。

STEP3 运用双重循环的方式，对于银行i和季度t，参照式（3.7）运行一个简单的OLS回归，提取出季度的R平方：。将所有的R平方值输出到矩阵中保存。

STEP 4 将矩阵转换为16个序列。通过单层循环的方式，对各个序列进行Logistic转换替代，然后将这16个序列再reshape成面板数据。

**（1）lrsq\_generator.do**

//初始化利率宏观变量

insheet using rate.csv,clear

destring hs300,replace force

replace hs300=. if hs300==-1

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year quarter year2 quarter2

drop if yearqt==".0."

sort yearqt date

gen id=\_n

tsset id

gen lag\_term=l.term

gen lag\_def=l.def

gen lag\_hs300=l.hs300

gen lag\_rate=l.rate

drop if id==1

drop id

save rate,replace

//生成16个deltadd变量

use DD\_daily,clear

sort stockid date

drop id

egen id=group(stockid)

by id,sort:gen time=\_n

tsset id time

gen deltadd=d.dd

keep date stock\_serialnumber deltadd

sort date stock\_serialnumber

reshape wide deltadd,i(date) j(stock\_serialnumber)

save deltadd\_by\_stock,replace

//merge利率宏观变量

use deltadd\_by\_stock,clear

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year quarter year2 quarter2

order date yearqt

egen sumdeltadd=rowtotal(deltadd\*)

sort yearqt date

merge 1:1 yearqt date using rate.dta

drop if \_merge==2

drop \_merge

egen nonmisscount=rownonmiss(deltadd\*)

drop if nonmisscount==0

save deltadd\_by\_stock,replace

//计算rsquare

use deltadd\_by\_stock,clear

forvalues i=1/16 {

gen sumdeltadd\_except`i'=sumdeltadd-deltadd`i'

replace sumdeltadd\_except`i'=sumdeltadd\_except`i'/(nonmisscount-1)

egen nonmisscount\_`i'=rownonmiss(deltadd`i' sumdeltadd\_except`i')

}

sort yearqt

egen id=group(yearqt)

egen idmax=max(id)

global idmax=idmax

matrix rsquare=J($idmax,16,.)

set more off

forvalues i=1/16 {

forvalues j=1/$idmax {

egen tempcount=mean(nonmisscount\_`i') if id==`j'

egen count=mean(tempcount)

local count=count

drop tempcount count

if `count'==2 {

reg deltadd`i' sumdeltadd\_except`i' if id==`j'

matrix rsquare[`j',`i']=e(r2)

}

}

}

matrix list rsquare

drop \*

svmat double rsquare

forvalues i=1/16 {

replace rsquare`i'=ln(rsquare`i'/(1-rsquare`i'))

}

gen id=\_n

reshape long rsquare,i(id) j(stock\_serialnumber)

sort id stock\_serialnumber

save rsquare,replace

**4、△CoVaR的计算**

具体的编程思路是：

STEP1 由银行i在季度t的每天d的违约距离计算每天的变化，进行横截面加总得到。

STEP2 将日度的宏观市场向量数据TERM、DEF、HS300和RATE全部滞后一期备用。

STEP3 运用双重循环的方式，对于银行i和季度t，参照式（9）运行一个q分位回归（q=10%），提取出q分位的贝塔系数：。将所有的q分位的贝塔系数值输出到矩阵中保存，将矩阵转换为16个序列，保存为数据文件。

STEP 4 通过单层循环的方式，对16个序列（注意：这里开始使用季度的序列）生成q分位数的（q=10%）和中位数的，并计算它们的差值。

STEP 5 将前面的16个序列合并（merge）进来，通过单层循环的方式，乘以，得到16个序列，然后将这16个序列再reshape成面板数据。

**（1）deltacovar\_generator.do**

//初始化利率宏观变量

insheet using rate.csv,clear

destring hs300,replace force

replace hs300=. if hs300==-1

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year quarter year2 quarter2

drop if yearqt==".0."

sort yearqt date

gen id=\_n

tsset id

gen lag\_term=l.term

gen lag\_def=l.def

gen lag\_hs300=l.hs300

gen lag\_rate=l.rate

drop if id==1

drop id

save rate,replace

//生成16个deltadd变量

use DD\_daily,clear

sort stockid date

drop id

egen id=group(stockid)

by id,sort:gen time=\_n

tsset id time

gen deltadd=d.dd

keep date stock\_serialnumber deltadd

sort date stock\_serialnumber

reshape wide deltadd,i(date) j(stock\_serialnumber)

save deltadd\_by\_stock,replace

//merge利率宏观变量

use deltadd\_by\_stock,clear

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year quarter year2 quarter2

order date yearqt

egen sumdeltadd=rowtotal(deltadd\*)

sort yearqt date

merge 1:1 yearqt date using rate.dta

drop if \_merge==2

drop \_merge

egen nonmisscount=rownonmiss(deltadd\*)

drop if nonmisscount==0

save deltadd\_by\_stock,replace

//计算betasystem，采取的分位数点为0.1

use deltadd\_by\_stock,clear

forvalues i=1/16 {

egen corenonmisscount\_`i'=rownonmiss(deltadd`i' sumdeltadd)

egen nonmisscount\_`i'=rownonmiss(deltadd`i' sumdeltadd term def hs300 rate)

}

sort yearqt

egen id=group(yearqt)

egen idmax=max(id)

global idmax=idmax

global percentile=50

matrix beta=J($idmax,16,.)

matrix temp=J(6,1,.)

set more off

forvalues i=1/16 {

forvalues j=1/$idmax {

egen tempcount=mean(nonmisscount\_`i') if id==`j'

egen count=mean(tempcount)

local count=count

egen tempcorecount=mean(corenonmisscount\_`i') if id==`j'

egen corecount=mean(tempcorecount)

local corecount=corecount

drop tempcount tempcorecount count corecount

if `corecount'==2 & `count'!=2{

qreg sumdeltadd deltadd`i' lag\_term lag\_def lag\_hs300 lag\_rate if id==`j',q(0.5)

matrix temp=e(b)

matrix beta[`j',`i']=temp[1,1]

}

}

}

matrix list beta

drop \*

svmat double beta

gen id=\_n

save betapercentile,replace

//计算deltacovar

use deltadd\_by\_stock,clear

sort yearqt

egen id=group(yearqt)

forvalues i=1/16 {

by id,sort:egen p10\_deltadd`i'=pctile(deltadd`i'),p(10)

by id,sort:egen p50\_deltadd`i'=pctile(deltadd`i'),p(50)

by id,sort:gen p1050\_deltadd`i'=p10\_deltadd`i'-p50\_deltadd`i'

}

duplicates drop id,force

merge 1:1 id using betapercentile

drop \_merge

keep yearqt id p1050\_deltadd\* beta\*

forvalues i=1/16 {

gen deltacovar`i'=beta`i'\*p1050\_deltadd`i'

}

keep id deltacovar\*

reshape long deltacovar,i(id) j(stock\_serialnumber)

sort id stock\_serialnumber

save deltacovar,replace

**5、静态MES的计算**

具体的思路是：

STEP1 将日度银行股价数据处理成日度银行股票回报数据。

STEP2 按季将日度银行股票回报数据进行排序，找出2007年、2009年、2010年每年4个季度的季度内最差的5%的日度银行股票回报，求得每季的MES，对这些季度MES按年进行相关性分析。

STEP3 将2009年1月至2010年12月作为MES测算区间，找出最差的5%的日度银行股票回报，求得MES。

STEP 4 将2008年1月至2008年12月的日度银行股票实际收益率取均值，求得SES；并将此区间内的季度银行杠杆率取均值，求得Leverage。

STEP 5 将MES对SES做散点图和回归分析，观察回归分析的拟合效果。

**（1）static\_mes.txt**

/\*初始化银行股价数据\*/

insheet using bankstock.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-9,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

gen year=year(date)

gen month=month(date)

gen quarter=quarter(date)

order stockid stockname date year quarter month

gen banktype=.

replace banktype=1 if stockid=="601398"|stockid=="601288"|stockid=="601988"|stockid=="601939"|stockid=="601328"

replace banktype=2 if stockid=="601818"|stockid=="000001"|stockid=="600015"|stockid=="600016"|stockid=="600000"|stockid=="601166"|stockid=="600036"|stockid=="601998"

replace banktype=3 if stockid=="601169"|stockid=="601009"|stockid=="002142"

gen return=(close-lag\_close)/lag\_close

sort stockid date

egen id\_stockid\_date=group(stockid date)

duplicates drop id\_stockid\_date,force

merge m:1 stockid using stock\_serialnumber.dta

keep if \_merge==3

drop \_merge

sort stockid date

save bankdata,replace

/\*季度MES求解\*/

use bankdata,clear

sort stockid year quarter date

egen id=group(stockid)

egen id\_stockid\_yqt=group(stockid year quarter)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year2 quarter2

by id\_stockid\_yqt,sort:cumul return,gen(quantile\_quarter)

gen mes\_window\_quarter=1 if quantile\_quarter<=0.05

by id\_stockid\_yqt,sort:egen mes\_temp=mean(return) if mes\_window\_quarter==1

by id\_stockid\_yqt,sort:egen mes\_quarter\_2=mean(mes\_temp)

gen mes\_quarter=-mes\_quarter\_2

drop mes\_temp mes\_quarter\_2

save mes\_quarter,replace

/\*MES季度数据导出\*/

use mes\_quarter,clear

keep stockid yearqt mes\_quarter id\_stockid\_yqt

duplicates drop id\_stockid\_yqt,force

merge m:1 stockid using stock\_serialnumber.dta

keep if \_merge==3

drop \_merge

keep yearqt mes\_quarter stock\_serialnumber

order yearqt stock\_serialnumber mes\_quarter

sort yearqt stock\_serialnumber mes\_quarter

reshape wide mes\_quarter,i(yearqt) j(stock\_serialnumber)

sort yearqt

destring yearqt,gen(yearqt2)

gen year=int(yearqt2/100)

keep if year>=2007 & year<=2010

save mes\_quarter\_by\_stock,replace

export excel using MES\_quarterly\_bystock.xls,replace firstrow(variables)

/\*MES求解\*/

use mes\_quarter,clear

gen mes\_window=1 if year>=2009 & year<=2010

keep if mes\_window==1

by id,sort:cumul return,gen(quantile)

gen mes\_window\_target=1 if quantile<=0.05

by id,sort:egen mes\_temp=mean(return) if mes\_window\_target==1

by id,sort:egen mes\_2=mean(mes\_temp)

gen mes=-mes\_2

drop mes\_temp mes\_2

keep stockid mes

duplicates drop stockid,force

sort stockid

save mes,replace

use mes\_quarter,clear

sort stockid date

merge m:1 stockid using mes.dta

keep if \_merge==3

drop \_merge

sort stockid yearqt date

save mes,replace

/\*初始化杠杆率\*/

use nii2,clear

gen liability=asset-equity

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year2 quarter2

sort stockid yearqt

merge m:1 stockid using stock\_serialnumber.dta

keep if \_merge==3

drop \_merge

save leverage,replace

/\*SES求解\*/

use mes,clear

merge m:1 stockid yearqt using leverage.dta

keep if \_merge==3

drop \_merge

gen leverage\_original=1/eta

gen ses\_window=1 if year==2008

by id,sort:egen ses\_temp=mean(return) if ses\_window==1

by id,sort:egen ses=mean(ses\_temp)

by id,sort:egen leverage\_temp=mean(leverage\_original) if ses\_window==1

by id,sort:egen leverage=mean(leverage\_temp)

drop ses\_temp leverage\_temp

save mes,replace

/\*回归分析\*/

use mes,clear

keep stockid yearqt mes ses leverage stock\_serialnumber

duplicates drop stockid,force

drop if ses==.

drop yearqt

sort stock\_serialnumber

export excel using SES\_MES\_static.xls,replace firstrow(variables)

sum ses mes leverage

reg ses mes,rob

est store OLS1

reg ses leverage,rob

est store OLS2

reg ses mes leverage,rob

est store OLS3

outreg2 [OLS1 OLS2 OLS3] using reg\_mes.doc,bdec(4) replace word

**6、动态MES的计算**

具体的思路是：

STEP1 将日度银行股价数据和日度市场指数数据处理成日度银行股票收益数据和日度市场收益数据。

STEP2 使用STATA依次对16家银行收益数据和市场收益数据进行去势处理（去势后的序列=序列取值-序列均值）。得到零均值的收益序列后，然后进行单变量TGARCH(1,1,1)过程（即ARCH(1) GARCH(1) TARCH(1)），得到16家银行和市场指数它们各自的方差和残差序列，然后对残差序列进行标准化（标准化残差=残差/标准差）。

STEP3 使用STATA提取单个银行i和市场指数的标准化残差，分散保存到16个文件中（每个文件中只含两个变量：银行i的标准化残差和市场指数的标准化残差）供Matlab调用：tarch\_zts\_i.xls。

STEP4 安装USCD GARCH工具箱到Matlab中，然后在Matlab中调用dcc-mvgarch函数，进行DCC-GARCH(1,1,1,1)过程（即ARCH(1) GARCH(1) DCC(1,1)），得到市场指数和各家银行i的动态时变波动率序列和，以及各家银行i和市场指数的动态时变协方差序列，按照公式求得动态相关系数序列。将DCCGARCH的运算结果分散保存到16个文件中供STATA调用。

STEP5 使用STATA计算独立同分布的银行收益残差和市场收益残差：，对样本期的市场收益（demean值）进行排序，找出下5%分位数作为临界值C，计算（3.28）式的右手条件值，从市场收益残差序列中找出满足条件的行：，然后依照式子（3.28）求MES。

STEP6 上一步计算出来的MES只是一个散点式的MES（仅针对满足条件的行进行了求解）。对空缺的MES，按照时间序列，以为基础，进行向上逐步滚动式比例填充（由于Matlab对矩阵的运算更快速，这里使用Matlab设置三重循环：i=不同的银行，n=找到已存在MES值的行，j and k=对不存在MES值的行进行向上逐步滚动式比例填充）。将填充完毕的MES返回到16个文件中：tarch\_zts\_i\_fill.xls。

STEP7 使用STATA合并这16个文件，得到16个银行的动态时变波动率序列和边际期望损失序列。

STEP8 根据（3.22）式，由边际期望损失序列、法定资本充足率cap和银行负债以及股权市场价值求得系统性期望损失。

**（1）dynamic\_mes.txt**

/\*银行股价数据分股导出（STATA预处理）\*/

use bankdata,clear

keep date stock\_serialnumber return

sort date stock\_serialnumber

reshape wide return,i(date) j(stock\_serialnumber)

sort date

save dailyreturn\_by\_stock,replace

export excel using bankstock\_return.xls,replace firstrow(variables)

/\*初始化市场指数数据并导出（STATA预处理）\*/

insheet using stockindex.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-9,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

gen year=year(date)

gen month=month(date)

gen quarter=quarter(date)

order stockid stockname date year quarter month

gen market\_return=(close-lag\_close)/lag\_close

keep date market\_return

drop if date==.

save indexdata,replace

export excel using stockindex\_return.xls,replace firstrow(variables)

/\*收益率数据合并（STATA处理）\*/

use dailyreturn\_by\_stock,clear

merge 1:1 date using indexdata.dta

keep if \_merge==3

drop \_merge

save bankreturn,replace

/\*TGARCH过程-提取标准化残差（STATA处理）\*/

use dailyreturn\_by\_stock,clear

sort date

tsset date

forvalues i=1/16 {

egen return\_mean`i'=mean(return`i')

gen return\_demean`i'=return`i'-return\_mean`i'

arch return\_demean`i', arch(1) garch(1) tarch(1)

predict ht`i',variance

predict et`i',residual

gen zt`i'=et`i'/sqrt(ht`i')

}

save tarch\_by\_stock,replace

use indexdata,clear

sort date

tsset date

egen mreturn\_mean=mean(market\_return)

gen mreturn\_demean=market\_return-mreturn\_mean

arch mreturn\_demean, arch(1) garch(1) tarch(1)

predict hmt,variance

predict emt,residual

gen zmt=emt/sqrt(hmt)

save tarch\_index,replace

use tarch\_by\_stock,clear

merge 1:1 date using tarch\_index.dta

keep if \_merge==3

drop \_merge

save tarch\_results,replace

use tarch\_results,clear

keep date zt\* zmt

save tarch\_zts,replace

export excel using tarch\_zts.xls,replace firstrow(variables)

/\*TGARCH过程-提取标准化残差-分散文件（STATA处理）\*/

use tarch\_zts,clear

forvalues i=1/16 {

sort date

tsset date

keep date zt`i' zmt

drop if zt`i'==.

drop if zmt==.

export excel using tarch\_zts\_`i'.xls,replace firstrow(variables)

use tarch\_zts,clear

}

%%DCC-GARCH过程（Matlab处理）

clear

for i=1:1:16

dccP=1;

dccQ=1;

archP=1;

garchQ=1;

eval(['zts=xlsread(''tarch\_zts\_',num2str(i),'.xls'',''Sheet1'',''B2:C10000'');']);

data(:,1)=zts(:,1);

data(:,2)=zts(:,2);

N=size(data);

rowscount=N(1);

[parameters, loglikelihood, Ht, Qt, likelihoods, stdresid, stderrors, A,B, jointscores]=dcc\_mvgarch(data,dccP,dccQ,archP,garchQ);

hii\_temp=Ht(1,1,:);

hii=reshape(hii\_temp,rowscount,1);

him\_temp=Ht(1,2,:);

him=reshape(him\_temp,rowscount,1);

hmm\_temp=Ht(2,2,:);

hmm=reshape(hmm\_temp,rowscount,1);

pim=him./(sqrt(hii).\*sqrt(hmm));

eval(['xlswrite(''tarch\_zts\_',num2str(i),'.xls'', {''hii''} ,''Sheet1'',''D1'');']);

eval(['xlswrite(''tarch\_zts\_',num2str(i),'.xls'', {''him''} ,''Sheet1'',''E1'');']);

eval(['xlswrite(''tarch\_zts\_',num2str(i),'.xls'', {''hmm''} ,''Sheet1'',''F1'');']);

eval(['xlswrite(''tarch\_zts\_',num2str(i),'.xls'', {''pim''} ,''Sheet1'',''G1'');']);

eval(['xlswrite(''tarch\_zts\_',num2str(i),'.xls'', hii,''Sheet1'',''D2'');']);

eval(['xlswrite(''tarch\_zts\_',num2str(i),'.xls'', him,''Sheet1'',''E2'');']);

eval(['xlswrite(''tarch\_zts\_',num2str(i),'.xls'', hmm,''Sheet1'',''F2'');']);

eval(['xlswrite(''tarch\_zts\_',num2str(i),'.xls'', pim,''Sheet1'',''G2'');']);

clearvars -EXCEPT i

end

/\*独立残差计算过程（STATA处理）\*/

forvalues i=1/16 {

import excel tarch\_zts\_`i'.xls,firstrow clear

format date %td

sort date

rename zt`i' zit\_`i'

rename zmt zmt\_`i'

rename hii hii\_`i'

rename him him\_`i'

rename hmm hmm\_`i'

rename pim pim\_`i'

save tarch\_zts\_`i',replace

use tarch\_results,clear

keep date return\_demean`i' mreturn\_demean

sort date

merge 1:1 date using tarch\_zts\_`i'.dta

keep if \_merge==3

drop \_merge

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year quarter year2 quarter2

sort yearqt date

egen id\_yqt=group(yearqt)

gen emt\_`i'=mreturn\_demean/sqrt(hmm\_`i')

gen eit\_`i'=return\_demean`i'/sqrt(hii\_`i')

gen xit\_`i'=(eit\_`i'-pim\_`i'\*emt\_`i')/sqrt(1-pim\_`i'^2)

save tarch\_zts\_`i',replace

use tarch\_zts\_`i',clear

sort date

gen id=\_n

tsset id

gen sigmahii\_`i'=sqrt(hii\_`i')

gen lag\_sigmahii\_`i'=l.sigmahii\_`i'

egen critical\_`i'=pctile(mreturn\_demean),p(5)

gen condition\_`i'=.

gen right\_hand\_`i'=critical\_`i'/sqrt(hmm\_`i')

replace condition\_`i'=1 if emt\_`i'<right\_hand\_`i'

gen mes\_`i'=-lag\_sigmahii\_`i'\*eit\_`i' if condition\_`i'==1

save tarch\_zts\_`i',replace

export excel using tarch\_zts\_`i'\_fill.xls,replace firstrow(variables)

}

%%MES填充和写入过程（Matlab处理）

clear

for i=1:16

eval(['filldata=xlsread(''tarch\_zts\_',num2str(i),'\_fill.xls'',''Sheet1'',''B2:U10000'');']);

rows= find(filldata(:,18)==1)

for j=1:rows(1)-1

k=rows(1)-j;

filldata(k,20)=filldata(k+1,20)\*filldata(k+1,16)/filldata(k+1,15);

end

for n=2:size(rows)

for j=rows(n-1)+1:rows(n)-1

k=rows(n-1)+rows(n)-j;

filldata(k,20)=filldata(k+1,20)\*filldata(k+1,16)/filldata(k+1,15);

end

end

eval(['xlswrite(''tarch\_zts\_',num2str(i),'\_fill.xls'', filldata(:,20),''Sheet1'',''U2'');']);

clearvars -EXCEPT i

end

/\*MES文件合并生成MES数据集（STATA处理）\*/

clear

forvalues i=1/16 {

import excel tarch\_zts\_`i'\_fill.xls,firstrow clear

format date %td

sort date

drop yearqt id\_yqt id mreturn\_demean

save tarch\_zts\_`i'\_ok,replace

}

use tarch\_zts\_6\_ok,clear

merge 1:1 date using tarch\_zts\_1\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_2\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_3\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_4\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_5\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_7\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_8\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_9\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_10\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_11\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_12\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_13\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_14\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_15\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_16\_ok.dta

drop \_merge

merge 1:1 date using tarch\_zts\_1\_ok.dta

drop \_merge

save tarch\_zts\_ok,replace

use tarch\_zts\_ok,clear

keep date hii\_\* pim\_\* mes\_\*

order date hii\_1 hii\_2 hii\_3 hii\_4 hii\_5 hii\_6 hii\_7 hii\_8 hii\_9 hii\_10 hii\_11 hii\_12 hii\_13 hii\_14 hii\_15 hii\_16 pim\_1 pim\_2 pim\_3 pim\_4 pim\_5 pim\_6 pim\_7 pim\_8 pim\_9 pim\_10 pim\_11 pim\_12 pim\_13 pim\_14 pim\_15 pim\_16 mes\_1 mes\_2 mes\_3 mes\_4 mes\_5 mes\_6 mes\_7 mes\_8 mes\_9 mes\_10 mes\_11 mes\_12 mes\_13 mes\_14 mes\_15 mes\_16

forvalues i=1/16 {

rename hii\_`i' hii`i'

rename pim\_`i' pim`i'

rename mes\_`i' mes`i'

}

sort date

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year quarter year2 quarter2

order date yearqt

save mes\_ok,replace

use mes\_ok,clear

sort yearqt date

reshape long hii pim mes,i(date) j(stock\_serialnumber)

sort stock\_serialnumber yearqt date

order stock\_serialnumber yearqt date

replace hii=-hii if hii<0

replace pim=-pim if pim<0

replace mes=-mes if mes<0

save mes\_dynamic,replace

/\*导入杠杆率数据（STATA处理）\*/

use leverage,clear

gen ln\_e=ln(equity)

gen ln\_d=ln(liability)

gen leverage\_dynamic=1/eta

sort stock\_serialnumber yearqt date

keep stockid stockname banktype stock\_serialnumber yearqt ln\_e ln\_d leverage\_dynamic

order stockid stockname banktype stock\_serialnumber yearqt ln\_e ln\_d leverage\_dynamic

save leverage\_for\_merge,replace

/\*生成MES和SES的数据集（STATA处理）\*/

use mes\_dynamic,clear

merge m:1 stock\_serialnumber yearqt using leverage\_for\_merge.dta

keep if \_merge==3

drop \_merge

order stockid stockname yearqt stock\_serialnumber

gen cap=0.08

gen ses=cap\*ln\_d-(1-cap)\*ln\_e\*(1-mes)

gen srisk=abs(ses)

drop if ses==.

save ses\_dynamic,replace

**7、描述性统计（每一种系统性风险测度的描述性统计图表生成）**

**（1）descriptive\_stat.txt**

/\*日度DD数据描述性统计\*/

use DD\_daily,clear

sum dd

bysort banktype:sum dd

sort banktype stock\_serialnumber date

export excel using DD\_daily.xls,replace firstrow(variables)

use DD\_daily,clear

keep date stock\_serialnumber dd

sort date stock\_serialnumber

reshape wide dd,i(date) j(stock\_serialnumber)

sort date

save dailydd\_by\_stock,replace

export excel using DD\_daily\_bystock.xls,replace firstrow(variables)

use stock\_serialnumber,clear

use DD\_daily,clear

sort banktype date stockid

duplicates drop id\_banktype\_date,force

keep date stock\_serialnumber dd add add\_all banktype

keep date banktype add add\_all

sort date banktype

reshape wide add add\_all,i(date) j(banktype)

rename add\_all2 add\_all

drop add\_all1 add\_all3

order date add1 add2 add3 add\_all

sort date

save dailydd\_by\_banktype,replace

use dailydd\_by\_stock,clear

merge 1:1 date using dailydd\_by\_banktype.dta

drop \_merge

save dailydd\_by\_stockandbanktype,replace

export excel using DD\_daily\_bystockandbanktype.xls,replace firstrow(variables)

/\*季度DD数据描述性统计\*/

use DD\_quarterly,clear

sum dd

bysort banktype:sum dd

sort banktype stock\_serialnumber date

export excel using DD\_quarterly.xls,replace firstrow(variables)

use DD\_quarterly,clear

keep date stock\_serialnumber dd

sort date stock\_serialnumber

reshape wide dd,i(date) j(stock\_serialnumber)

sort date

save quarterlydd\_by\_stock,replace

export excel using DD\_quarterly\_bystock.xls,replace firstrow(variables)

use stock\_serialnumber,clear

use DD\_quarterly,clear

sort banktype yearqt stockid

duplicates drop id\_banktype\_yqt,force

keep date stock\_serialnumber dd add add\_all banktype

keep banktype date add add\_all

sort date banktype

reshape wide add add\_all,i(date) j(banktype)

rename add\_all2 add\_all

drop add\_all1 add\_all3

order date add1 add2 add3 add\_all

sort date

save quarterlydd\_by\_banktype,replace

use quarterlydd\_by\_stock,clear

merge 1:1 date using quarterlydd\_by\_banktype.dta

drop \_merge

save quarterlydd\_by\_stockandbanktype,replace

export excel using DD\_quarterly\_bystockandbanktype.xls,replace firstrow(variables)

/\*季度LRSQ数据描述性统计\*/

use DD\_quarterly.dta,clear

sort yearqt

egen id=group(yearqt)

sort id stock\_serialnumber

merge 1:1 id stock\_serialnumber using rsquare

keep if \_merge==3

drop \_merge

by id\_banktype\_yqt,sort:egen rsquare\_banktype=mean(rsquare)

by id\_bankall\_yqt,sort:egen rsquare\_all=mean(rsquare)

save rsquare\_for\_ds,replace

sum rsquare

bysort banktype:sum rsquare

sort banktype stock\_serialnumber date

use rsquare\_for\_ds,clear

keep date stock\_serialnumber rsquare

sort date stock\_serialnumber

reshape wide rsquare,i(date) j(stock\_serialnumber)

sort date

save quarterlyrsquare\_by\_stock,replace

export excel using LRSQ\_quarterly\_bystock.xls,replace firstrow(variables)

use stock\_serialnumber,clear

use rsquare\_for\_ds,clear

sort banktype yearqt stockid

duplicates drop id\_banktype\_yqt,force

keep date stock\_serialnumber rsquare rsquare\_banktype rsquare\_all banktype

keep date banktype rsquare\_banktype rsquare\_all

sort date banktype

reshape wide rsquare\_banktype rsquare\_all,i(date) j(banktype)

rename rsquare\_all2 rsquare\_all

drop rsquare\_all1 rsquare\_all3

order date rsquare\_banktype1 rsquare\_banktype2 rsquare\_banktype3 rsquare\_all

sort date

save quarterlyrsquare\_by\_banktype,replace

use quarterlyrsquare\_by\_stock,clear

merge 1:1 date using quarterlyrsquare\_by\_banktype.dta

drop \_merge

save quarterlyrsquare\_by\_stockandbanktype,replace

export excel using LRSQ\_quarterly\_bystockandbanktype.xls,replace firstrow(variables)

/\*季度DeltaCoVaR数据描述性统计\*/

use DD\_quarterly.dta,clear

sort yearqt

egen id=group(yearqt)

sort id stock\_serialnumber

merge 1:1 id stock\_serialnumber using deltacovar

keep if \_merge==3

drop \_merge

by id\_banktype\_yqt,sort:egen deltacovar\_banktype=mean(deltacovar)

by id\_bankall\_yqt,sort:egen deltacovar\_all=mean(deltacovar)

save deltacovar\_for\_ds,replace

sum deltacovar

bysort banktype:sum deltacovar

sort banktype stock\_serialnumber date

use deltacovar\_for\_ds,clear

keep date stock\_serialnumber deltacovar

sort date stock\_serialnumber

reshape wide deltacovar,i(date) j(stock\_serialnumber)

sort date

save quarterlydeltacovar\_by\_stock,replace

export excel using DeltaCoVaR\_quarterly\_bystock.xls,replace firstrow(variables)

use stock\_serialnumber,clear

use deltacovar\_for\_ds,clear

sort banktype yearqt stockid

duplicates drop id\_banktype\_yqt,force

keep date stock\_serialnumber deltacovar deltacovar\_banktype deltacovar\_all banktype

keep date banktype deltacovar\_banktype deltacovar\_all

sort date banktype

reshape wide deltacovar\_banktype deltacovar\_all,i(date) j(banktype)

rename deltacovar\_all2 deltacovar\_all

drop deltacovar\_all1 deltacovar\_all3

order date deltacovar\_banktype1 deltacovar\_banktype2 deltacovar\_banktype3 deltacovar\_all

sort date

save quarterlydeltacovar\_by\_banktype,replace

use quarterlydeltacovar\_by\_stock,clear

merge 1:1 date using quarterlydeltacovar\_by\_banktype.dta

drop \_merge

save quarterlydeltacovar\_by\_stockandbanktype,replace

export excel using DeltaCoVaR\_quarterly\_bystockandbanktype.xls,replace firstrow(variables)

/\*日度MES数据描述性统计\*/

%%hii

use ses\_dynamic,clear

keep date stock\_serialnumber hii

sort date stock\_serialnumber

reshape wide hii,i(date) j(stock\_serialnumber)

sort date

save dailyhii\_by\_stock,replace

export excel using hii\_daily\_bystock.xls,replace firstrow(variables)

use ses\_dynamic,clear

sort banktype date stockid

egen id\_banktype\_date=group(banktype date)

by id\_banktype\_date,sort:egen hii\_banktype=mean(hii)

sort date stockid

egen id\_date=group(date)

by id\_date,sort:egen hii\_all=mean(hii)

duplicates drop id\_banktype\_date,force

keep date banktype hii\_banktype hii\_all

sort date banktype

reshape wide hii\_banktype hii\_all,i(date) j(banktype)

rename hii\_all2 hii\_all

drop hii\_all1 hii\_all3

order date hii\_banktype1 hii\_banktype2 hii\_banktype3 hii\_all

sort date

save dailyhii\_by\_banktype,replace

use dailyhii\_by\_stock,clear

merge 1:1 date using dailyhii\_by\_banktype.dta

drop \_merge

save dailyhii\_by\_stockandbanktype,replace

export excel using hii\_daily\_bystockandbanktype.xls,replace firstrow(variables)

%%pim

use ses\_dynamic,clear

keep date stock\_serialnumber pim

sort date stock\_serialnumber

reshape wide pim,i(date) j(stock\_serialnumber)

sort date

save dailypim\_by\_stock,replace

export excel using pim\_daily\_bystock.xls,replace firstrow(variables)

use ses\_dynamic,clear

sort banktype date stockid

egen id\_banktype\_date=group(banktype date)

by id\_banktype\_date,sort:egen pim\_banktype=mean(pim)

sort date stockid

egen id\_date=group(date)

by id\_date,sort:egen pim\_all=mean(pim)

duplicates drop id\_banktype\_date,force

keep date banktype pim\_banktype pim\_all

sort date banktype

reshape wide pim\_banktype pim\_all,i(date) j(banktype)

rename pim\_all2 pim\_all

drop pim\_all1 pim\_all3

order date pim\_banktype1 pim\_banktype2 pim\_banktype3 pim\_all

sort date

save dailypim\_by\_banktype,replace

use dailypim\_by\_stock,clear

merge 1:1 date using dailypim\_by\_banktype.dta

drop \_merge

save dailypim\_by\_stockandbanktype,replace

export excel using pim\_daily\_bystockandbanktype.xls,replace firstrow(variables)

%%mes

use ses\_dynamic,clear

keep date stock\_serialnumber mes

sort date stock\_serialnumber

reshape wide mes,i(date) j(stock\_serialnumber)

sort date

save dailymes\_by\_stock,replace

export excel using mes\_daily\_bystock.xls,replace firstrow(variables)

use ses\_dynamic,clear

sort banktype date stockid

egen id\_banktype\_date=group(banktype date)

by id\_banktype\_date,sort:egen mes\_banktype=mean(mes)

sort date stockid

egen id\_date=group(date)

by id\_date,sort:egen mes\_all=mean(mes)

duplicates drop id\_banktype\_date,force

keep date banktype mes\_banktype mes\_all

sort date banktype

reshape wide mes\_banktype mes\_all,i(date) j(banktype)

rename mes\_all2 mes\_all

drop mes\_all1 mes\_all3

order date mes\_banktype1 mes\_banktype2 mes\_banktype3 mes\_all

sort date

save dailymes\_by\_banktype,replace

use dailymes\_by\_stock,clear

merge 1:1 date using dailymes\_by\_banktype.dta

drop \_merge

save dailymes\_by\_stockandbanktype,replace

export excel using mes\_daily\_bystockandbanktype.xls,replace firstrow(variables)

%%ses

use ses\_dynamic,clear

keep date stock\_serialnumber ses

sort date stock\_serialnumber

reshape wide ses,i(date) j(stock\_serialnumber)

sort date

save dailyses\_by\_stock,replace

export excel using ses\_daily\_bystock.xls,replace firstrow(variables)

use ses\_dynamic,clear

sort banktype date stockid

egen id\_banktype\_date=group(banktype date)

by id\_banktype\_date,sort:egen ses\_banktype=mean(ses)

sort date stockid

egen id\_date=group(date)

by id\_date,sort:egen ses\_all=mean(ses)

duplicates drop id\_banktype\_date,force

keep date banktype ses\_banktype ses\_all

sort date banktype

reshape wide ses\_banktype ses\_all,i(date) j(banktype)

rename ses\_all2 ses\_all

drop ses\_all1 ses\_all3

order date ses\_banktype1 ses\_banktype2 ses\_banktype3 ses\_all

sort date

save dailyses\_by\_banktype,replace

use dailyses\_by\_stock,clear

merge 1:1 date using dailyses\_by\_banktype.dta

drop \_merge

save dailyses\_by\_stockandbanktype,replace

export excel using ses\_daily\_bystockandbanktype.xls,replace firstrow(variables)

**第四章 系统性风险传染研究**

**1、银行间双边风险敞口矩阵X的计算**

首先根据年报数据获取16家银行在2007-2014年的银行间资产（=存放同业+拆出资金）和银行间负债（=同业存放+拆入资金）数据，将这些原始数据进行归一化处理后得到比重数据（=变量/变量总和），然后根据归一化的银行间资产和银行间负债，根据以上的计算原理和原始数据，用Lingo11.0软件编程，使用RAS算法求解交叉熵最小化得到银行间双边风险敞口矩阵X。Lingo代码见lingo.txt，Lingo代码运行结果中的银行间双边风险敞口矩阵X见lingo\_result.txt。

**（1）lingo.txt**

model:!16家银行;

sets:

A/1..16/:ai;

B/1..16/:bj;

links(A,B):X,X0;

!X1=X\_\*;

endsets

data:

ai=0.189757705,0.067117304,0.151589721,0.099730425,0.198893443,0.021846399,0.031809255,0.015836073,0.048777834,0.034871932,0.029429022,0.033512078,0.031405482,0.006451513,0.005250189,0.033721615;

bj=0.078975528,0.092359349,0.115108758,0.090227067,0.147211328,0.029838527,0.061668146,0.024452362,0.066685334,0.059231874,0.100899182,0.040676739,0.052941807,0.007531111,0.006969665,0.025223225;

enddata

!目标函数;

[obj]min=@sum(links(i,j)|i#ne#j:X(i,j)\*(@log(X(i,j))-@log(X0(i,j))));

aa=@sum(A(i):ai(i));

bb=@sum(b(j):bj(j));

!约束条件;

@for(A(i):@sum(B(j):X(i,j))=X(i,i)+ai(i););

@for(B(j):@sum(A(i):X(i,j))=X(j,j)+bj(j););

@for(links(i,j):X0(i,j)=@if(i#EQ#j,0,ai(i)\*bj(j)));

end

**2、银行间风险传染仿真**

按照不同的同业拆借损失率水平LGD，运用Matlab 2015b对整个银行体系进行模拟，在每一个数值的情况下按照传染机制的定义和判定条件的银行倒闭传染风险进行判断，确定银行风险传染的轮数、每一轮中倒闭银行的数量及倒闭银行的总数、受影响的银行资产比重，来测量银行风险传染性。Matlab的银行倒闭传染函数BankDownFun.m中设定了银行倒闭风险传染的判定条件，输出了每一轮中倒闭银行的数量及倒闭银行的总数，Matlab的主程序riskcontagion.m则输出了不同LGD情境下的银行ii倒闭引起的银行风险传染的轮数、倒闭银行的总数、受影响的银行资产比重。riskcontagion.m仅展示2014年的银行系统性风险传染仿真过程，针对2007-2014年全年份的银行系统性风险传染仿真过程可以对riskcontagion.m设立一个针对年份的参数，建立外层循环从而实现一般化，具体过程参见riskcontagion\_general.m。

**（1）BankDownFun.m**

function[BankDownTotalNumOutput BankDownNumOutput]=BankDownFun(X,Deta,LGD,BankNum,BankDownNum,InterBankDebtSum,CoreCapital,RiskWeightedAsset)

X\_data=X.\*InterBankDebtSum; %将归一化X变成实际数值，其单位为元

CoreCapital\_T=CoreCapital';

RiskWeightedAsset\_T=RiskWeightedAsset';

BankDownNum\_temp=BankDownNum+eye(BankNum); %假设自身银行倒闭

for ii=1:BankNum %ii银行最先倒闭

Down\_temp=BankDownNum\_temp(ii,:); %在该轮数下，ii银行已经造成其他银行倒闭的情况(包括自身)

for jj=1:BankNum %判断jj银行倒闭情况

if 0==Down\_temp(jj) %若jj银行还没有倒闭

if (CoreCapital\_T(jj)-sum(LGD.\*X\_data(jj,:).\*Down\_temp))/(RiskWeightedAsset\_T(jj)-Deta.\*sum(X\_data(jj,:).\*Down\_temp))<0.06 %银行倒闭的判定规则

BankDownNum\_temp(ii,jj)=1; %在该轮数下，ii银行已经造成jj银行倒闭

end

end

end

end

BankDownNumOutput=BankDownNum\_temp-eye(BankNum); %当前违约率和当前轮次下，银行ii倒闭所引起其他银行倒闭的情况（不包含银行ii自身）

BankDownTotalNumOutput=sum(BankDownNumOutput'); %当前违约率和当前轮次下，银行ii倒闭所引起其他银行倒闭的总数目

End

**（2）riskcontagion.m**

clear

X=xlsread('lingo.xlsx','2014','A1:P16'); %银行间双边风险敞口矩阵(Lingo Result)

BankNum=16; %银行数量

Deta=0.25; %银行同业间拆借资产的风险权重系数

LGDList=[5 10 15 20 40 50 60 65 70 75 80 90 100]/100; %同业拆借资金的违约损失率

LGDLen=length(LGDList);

Num=10; %传染轮数最大值设定

LunCi=zeros(BankNum,LGDLen); %在违约率下，每家银行倒闭所引起的传染轮次（初始化）

BankDownLunCi=zeros(BankNum,LGDLen); %在违约率下，每家银行倒闭所引起的银行倒闭总数（初始化）

BiLi=zeros(BankNum,LGDLen); %在违约率下，每家银行倒闭所引起的受影响银行资产比重（初始化）

TotalAsset=xlsread('asset.xlsx','Sheet1','A1:P1'); %银行总资产

Liability\_interbank=xlsread('liability\_interbank.xlsx','Sheet1','A1:P1'); %银行间负债

InterBankDebtSum=sum(Liability\_interbank); %所有银行间负债之和

CoreCapital=xlsread('corecapital.xlsx','Sheet1','A1:P1'); %核心资本

RiskWeightedAsset=xlsread('rwa.xlsx','Sheet1','A1:P1'); %风险加权资产总额

BankDownResult2014=cell(size(LGDList,2),Num); %银行倒闭情况结果矩阵初始化

BankDownTotalResult2014=cell(size(LGDList,2),Num); %银行倒闭数量结果矩阵初始化

for ll=1:LGDLen %最外层循环，循环变量ll=LGD

LGD=LGDList(ll);

BankDownTotalNum=zeros(Num,BankNum);

BankDownNum=zeros(BankNum,BankNum);

for kk=1:Num %里层循环，循环变量kk=轮次

[BankDownTotalNum(kk,:) BankDownNum] = BankDownFun(X,Deta,LGD,BankNum,BankDownNum,InterBankDebtSum,CoreCapital,RiskWeightedAsset); %ii银行最先倒闭,在kk轮次下,银行倒闭的情况

%输出参数1：当前违约率和当前轮次下，银行ii倒闭所引起其他银行倒闭的总数目；输出参数2：当前违约率和当前轮次下，银行ii倒闭所引起其他银行倒闭的情况（不包含银行ii自身）

BankDownResult2014{ll,kk}=BankDownNum; %数据保存：当前违约率和当前轮次下，银行ii倒闭所引起其他银行倒闭的情况（不包含银行ii自身）

BankDownTotalResult2014{ll,kk}=BankDownTotalNum(kk,:); %数据保存：当前违约率和当前轮次下，银行ii倒闭所引起其他银行倒闭的总数目

%统计当前违约率和所有轮次下，各家银行倒闭所引起的银行倒闭总数的最大值（10轮中的最大值）（覆盖式算法）

for mm=1:BankNum %最里层循环，循环变量mm=银行mm

if BankDownLunCi(mm,ll)<BankDownTotalNum(kk,mm)

BankDownLunCi(mm,ll)=BankDownTotalNum(kk,mm);

LunCi(mm,ll)=kk; %条件满足时，把10轮中造成最大倒闭数量的轮次数输出

BiLi(mm,ll)=sum(TotalAsset.\*BankDownNum(mm,:))/(sum(TotalAsset)-TotalAsset(mm)); %条件满足时，把10轮中造成最大倒闭数量的受影响资产比重输出

%分子=资产列\*已倒闭的银行序号=已倒闭的资产量；分母=sum（资产列）-银行ii自身资产

end

end

end

end

%%结果输出模块

xlswrite('银行风险传染2014.xls', {'银行ii倒闭引起的倒闭银行总数'},'Sheet1','A1');

xlswrite('银行风险传染2014.xls', {'农业银行'},'Sheet1','A2');

xlswrite('银行风险传染2014.xls', {'交通银行'},'Sheet1','A3');

xlswrite('银行风险传染2014.xls', {'工商银行'},'Sheet1','A4');

xlswrite('银行风险传染2014.xls', {'建设银行'},'Sheet1','A5');

xlswrite('银行风险传染2014.xls', {'中国银行'},'Sheet1','A6');

xlswrite('银行风险传染2014.xls', {'平安银行'},'Sheet1','A7');

xlswrite('银行风险传染2014.xls', {'浦发银行'},'Sheet1','A8');

xlswrite('银行风险传染2014.xls', {'华夏银行'},'Sheet1','A9');

xlswrite('银行风险传染2014.xls', {'民生银行'},'Sheet1','A10');

xlswrite('银行风险传染2014.xls', {'招商银行'},'Sheet1','A11');

xlswrite('银行风险传染2014.xls', {'兴业银行'},'Sheet1','A12');

xlswrite('银行风险传染2014.xls', {'光大银行'},'Sheet1','A13');

xlswrite('银行风险传染2014.xls', {'中信银行'},'Sheet1','A14');

xlswrite('银行风险传染2014.xls', {'宁波银行'},'Sheet1','A15');

xlswrite('银行风险传染2014.xls', {'南京银行'},'Sheet1','A16');

xlswrite('银行风险传染2014.xls', {'北京银行'},'Sheet1','A17');

xlswrite('银行风险传染2014.xls', LGDList,'Sheet1','B1');

xlswrite('银行风险传染2014.xls', BankDownLunCi,'Sheet1','B2');

xlswrite('银行风险传染2014.xls', {'银行ii倒闭引起的风险传染轮数'},'Sheet2','A1');

xlswrite('银行风险传染2014.xls', {'农业银行'},'Sheet2','A2');

xlswrite('银行风险传染2014.xls', {'交通银行'},'Sheet2','A3');

xlswrite('银行风险传染2014.xls', {'工商银行'},'Sheet2','A4');

xlswrite('银行风险传染2014.xls', {'建设银行'},'Sheet2','A5');

xlswrite('银行风险传染2014.xls', {'中国银行'},'Sheet2','A6');

xlswrite('银行风险传染2014.xls', {'平安银行'},'Sheet2','A7');

xlswrite('银行风险传染2014.xls', {'浦发银行'},'Sheet2','A8');

xlswrite('银行风险传染2014.xls', {'华夏银行'},'Sheet2','A9');

xlswrite('银行风险传染2014.xls', {'民生银行'},'Sheet2','A10');

xlswrite('银行风险传染2014.xls', {'招商银行'},'Sheet2','A11');

xlswrite('银行风险传染2014.xls', {'兴业银行'},'Sheet2','A12');

xlswrite('银行风险传染2014.xls', {'光大银行'},'Sheet2','A13');

xlswrite('银行风险传染2014.xls', {'中信银行'},'Sheet2','A14');

xlswrite('银行风险传染2014.xls', {'宁波银行'},'Sheet2','A15');

xlswrite('银行风险传染2014.xls', {'南京银行'},'Sheet2','A16');

xlswrite('银行风险传染2014.xls', {'北京银行'},'Sheet2','A17');

xlswrite('银行风险传染2014.xls', LGDList,'Sheet2','B1');

xlswrite('银行风险传染2014.xls', LunCi,'Sheet2','B2');

xlswrite('银行风险传染2014.xls', {'银行ii倒闭引起的受影响资产比重'},'Sheet3','A1');

xlswrite('银行风险传染2014.xls', {'农业银行'},'Sheet3','A2');

xlswrite('银行风险传染2014.xls', {'交通银行'},'Sheet3','A3');

xlswrite('银行风险传染2014.xls', {'工商银行'},'Sheet3','A4');

xlswrite('银行风险传染2014.xls', {'建设银行'},'Sheet3','A5');

xlswrite('银行风险传染2014.xls', {'中国银行'},'Sheet3','A6');

xlswrite('银行风险传染2014.xls', {'平安银行'},'Sheet3','A7');

xlswrite('银行风险传染2014.xls', {'浦发银行'},'Sheet3','A8');

xlswrite('银行风险传染2014.xls', {'华夏银行'},'Sheet3','A9');

xlswrite('银行风险传染2014.xls', {'民生银行'},'Sheet3','A10');

xlswrite('银行风险传染2014.xls', {'招商银行'},'Sheet3','A11');

xlswrite('银行风险传染2014.xls', {'兴业银行'},'Sheet3','A12');

xlswrite('银行风险传染2014.xls', {'光大银行'},'Sheet3','A13');

xlswrite('银行风险传染2014.xls', {'中信银行'},'Sheet3','A14');

xlswrite('银行风险传染2014.xls', {'宁波银行'},'Sheet3','A15');

xlswrite('银行风险传染2014.xls', {'南京银行'},'Sheet3','A16');

xlswrite('银行风险传染2014.xls', {'北京银行'},'Sheet3','A17');

xlswrite('银行风险传染2014.xls', LGDList,'Sheet3','B1');

xlswrite('银行风险传染2014.xls', BiLi,'Sheet3','B2');

**第五章 系统重要性银行识别**

**1、基于SES衍生的SRISK相对系统重要性银行识别**

**（1）srisk.txt**

/\*生成SRISK排序数据集（STATA处理）\*/

use ses\_dynamic,clear

sort stockid yearqt date

egen id\_yqt=group(stockid yearqt)

by id\_yqt,sort:egen srisk\_quarterly=mean(srisk)

duplicates drop id\_yqt,force

drop id\_yqt

sort yearqt stockid

egen id\_yqt=group(yearqt)

by id\_yqt,sort:egen srisk\_sum=sum(srisk\_quarterly)

by id\_yqt,sort:gen srisk\_p=100\*srisk\_quarterly/srisk\_sum

gsort yearqt -srisk\_p

by id\_yqt,sort:gen rank=\_n

save srisk\_p,replace

/\*导出SRISK排序数据集（STATA处理，给EXCEL备用）\*/

use srisk\_p,clear

keep stockid stockname yearqt stock\_serialnumber banktype id\_yqt srisk\_p rank

destring yearqt,gen(yearqt2)

gen year=int(yearqt2/100)

keep if year>=2007

drop if yearqt=="200701"

keep if rank>=1 & rank<=10

sort yearqt rank

keep yearqt stockname srisk\_p rank stock\_serialnumber

reshape wide stock\_serialnumber stockname srisk\_p,i(yearqt) j(rank)

sort yearqt

save srisk\_results,replace

export excel using srisk\_results.xls,replace firstrow(variables)

/\*导出SRISK排序数据集（STATA处理，给EXCEL备用）\*/

use srisk\_p,clear

keep stockid stockname yearqt stock\_serialnumber banktype id\_yqt srisk\_p rank

destring yearqt,gen(yearqt2)

gen year=int(yearqt2/100)

keep if year>=2007

drop if yearqt=="200701"

sort yearqt stock\_serialnumber

keep yearqt stockname srisk\_p rank stock\_serialnumber

reshape wide stockname rank srisk\_p,i(yearqt) j(stock\_serialnumber)

sort yearqt

save srisk\_results\_2,replace

export excel using srisk\_results2.xls,replace firstrow(variables)

**2、基于熵值指标法的系统重要性银行识别**

**（1）dsibs.txt（包含熵值法过程核心函数entropy\_method.m）**

/\*初始化D-SIBs数据（STATA预处理）\*/

insheet using dsibs\_1.csv,clear

sort stockname date

save dsibs\_1,replace

insheet using dsibs\_2.csv,clear

sort stockname date

save dsibs\_2,replace

insheet using dsibs\_3.csv,clear

sort stockname date

save dsibs\_3,replace

use dsibs\_1,clear

merge 1:1 stockname date using dsibs\_2.dta

drop \_merge

merge 1:1 stockname date using dsibs\_3.dta

drop \_merge

merge m:1 stockname using stock\_serialnumber.dta

drop \_merge

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

gen year=year(date)

order stockid stockname stock\_serialnumber banktype year date total\_asset total\_liability asset\_in\_system liability\_in\_system ii nii oi held\_for\_trading available\_for\_sale deriv\_finasset deriv\_finliability ind\_deposit

sort stock\_serialnumber year

save dsibs,replace

use dsibs,clear

drop revenue corp\_deposit ind\_loan corp\_loan

save dsibs\_for\_matlab,replace

/\*分年导出的D-SIBs数据（STATA处理）\*/

forvalues i=1/8 {

use dsibs\_for\_matlab,clear

global k=`i'+2006

keep if year==$k

export excel using dsibs\_$k.xls,replace firstrow(variables)

}

%%熵值法过程（Matlab处理）

%%编写熵值法函数，命名为entropy\_method.m

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [socre,weight]=entropy\_method(data)

[id,indexnum]=size(data);

[mapdata,range\_settings]=mapminmax(data');

range\_settings.ymin=0.002; %调整归一化后的最小值

range\_settings.ymax=0.996; %调整归一化后的最大值

range\_settings.yrange=range\_settings.ymax-range\_settings.ymin; %调整归一化后的极差

data2=mapminmax(data',range\_settings); %使用调整后的极差设定，重新对原数据进行归一化

normaldata=data2'; %归一化后的数据

%% 计算第j个指标下，第i个记录占该指标的比重p(i,j)

for i=1:id

for j=1:indexnum

p(i,j)=normaldata(i,j)/sum(normaldata(:,j));

end

end

%% 计算第j个指标的熵值e(j)

k=1/log(id);

for j=1:indexnum

e(j)=-k\*sum(p(:,j).\*log(p(:,j)));

end

d=ones(1,indexnum)-e; % 计算信息熵冗余度

weight=d./sum(d); % 求权值w

score=weight\*p'; % 求综合得分

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%使用熵值法批量处理D-SIBs数据

clear

for i=1:8

k=i+2006;

eval(['data=xlsread(''dsibs\_',num2str(k),'.xls'',''Sheet1'',''G2:R17'');']);

[score(i,:),weight(i,:)]=entropy\_method(data);

year(i)=k;

end

score=score';

weight=weight';

xlswrite('D-SIBs\_result.xls', {'D-SIBs指标权重'},'Sheet1','A1');

xlswrite('D-SIBs\_result.xls', {'总资产'},'Sheet1','A2');

xlswrite('D-SIBs\_result.xls', {'总负债'},'Sheet1','A3');

xlswrite('D-SIBs\_result.xls', {'金融系统内资产'},'Sheet1','A4');

xlswrite('D-SIBs\_result.xls', {'金融系统内负债'},'Sheet1','A5');

xlswrite('D-SIBs\_result.xls', {'利息净收入'},'Sheet1','A6');

xlswrite('D-SIBs\_result.xls', {'手续费及佣金净收入'},'Sheet1','A7');

xlswrite('D-SIBs\_result.xls', {'其他经营净收益'},'Sheet1','A8');

xlswrite('D-SIBs\_result.xls', {'交易性金融资产'},'Sheet1','A9');

xlswrite('D-SIBs\_result.xls', {'可供出售金融资产'},'Sheet1','A10');

xlswrite('D-SIBs\_result.xls', {'衍生金融资产'},'Sheet1','A11');

xlswrite('D-SIBs\_result.xls', {'衍生金融负债'},'Sheet1','A12');

xlswrite('D-SIBs\_result.xls', {'居民储蓄存款'},'Sheet1','A13');

xlswrite('D-SIBs\_result.xls', year,'Sheet1','B1');

xlswrite('D-SIBs\_result.xls', weight,'Sheet1','B2');

xlswrite('D-SIBs\_result.xls', {'D-SIBs银行得分'},'Sheet2','A1');

xlswrite('D-SIBs\_result.xls', {'农业银行'},'Sheet2','A2');

xlswrite('D-SIBs\_result.xls', {'交通银行'},'Sheet2','A3');

xlswrite('D-SIBs\_result.xls', {'工商银行'},'Sheet2','A4');

xlswrite('D-SIBs\_result.xls', {'建设银行'},'Sheet2','A5');

xlswrite('D-SIBs\_result.xls', {'中国银行'},'Sheet2','A6');

xlswrite('D-SIBs\_result.xls', {'平安银行'},'Sheet2','A7');

xlswrite('D-SIBs\_result.xls', {'浦发银行'},'Sheet2','A8');

xlswrite('D-SIBs\_result.xls', {'华夏银行'},'Sheet2','A9');

xlswrite('D-SIBs\_result.xls', {'民生银行'},'Sheet2','A10');

xlswrite('D-SIBs\_result.xls', {'招商银行'},'Sheet2','A11');

xlswrite('D-SIBs\_result.xls', {'兴业银行'},'Sheet2','A12');

xlswrite('D-SIBs\_result.xls', {'光大银行'},'Sheet2','A13');

xlswrite('D-SIBs\_result.xls', {'中信银行'},'Sheet2','A14');

xlswrite('D-SIBs\_result.xls', {'宁波银行'},'Sheet2','A15');

xlswrite('D-SIBs\_result.xls', {'南京银行'},'Sheet2','A16');

xlswrite('D-SIBs\_result.xls', {'北京银行'},'Sheet2','A17');

xlswrite('D-SIBs\_result.xls', year,'Sheet2','B1');

xlswrite('D-SIBs\_result.xls', score,'Sheet2','B2');

**第六章 系统性风险的影响因素**

**1、银行竞争**

**（1）code\_for\_comp.txt**

/\*COMP数据合并\*/

//四个COMP数据初始化

insheet using comp1.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-9,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

destring revenue,replace ignore(",") force

destring interestincome,replace ignore(",") force

destring interestexpense,replace ignore(",") force

destring noninterestexpense,replace ignore(",") force

destring adminexpense,replace ignore(",") force

gen tc=interestexpense+noninterestexpense+adminexpense

label var tc "银行总成本"

order stockid stockname date

sort stockid date

save comp1.dta,replace

insheet using comp2.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-9,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order stockid stockname date

sort stockid date

save comp2.dta,replace

insheet using comp3.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-9,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order stockid stockname date

sort stockid date

save comp3.dta,replace

insheet using comp4.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-9,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order stockid stockname date

sort stockid date

save comp4.dta,replace

insheet using comp5.csv,clear

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year2 quarter2

drop date year quarter

order yearqt

sort yearqt

save comp5.dta,replace

insheet using comp6.csv,clear

tostring date,replace

gen date2=date(date,"YM")

format date2 %td

drop date

rename date2 date

label var date "日期"

gen year=year(date)

gen month=month(date)

gen quarter=quarter(date)

keep if month==3|month==6|month==9|month==12

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year2 quarter2

drop date year month quarter

order yearqt

sort yearqt

save comp6.dta,replace

//补全COMP3数据

use comp2,clear

merge 1:1 stockid date using comp3.dta

drop \_merge

export excel using comp3\_add.xlsx,replace firstrow(variables)

insheet using comp3\_add.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-6,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order stockid date

sort stockid date

save comp3\_add.dta,replace

//合并前四个数据

use comp2,clear

merge 1:1 stockid date using comp1.dta

drop \_merge

sort stockid date

merge 1:1 stockid date using comp3\_add.dta

drop \_merge

sort stockid date

merge 1:1 stockid date using comp4.dta

drop \_merge

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year quarter year2 quarter2

order stockid stockname yearqt date

sort stockid yearqt

rename asset ta

label var ta "银行总资产"

gen w1=interestexpense/deposit

label var w1 "存款价格"

gen w2=(noninterestexpense+adminexpense)/ta

label var w2 "劳动力价格"

order stockid stockname yearqt date tc ta w1 w2

sort stockid

merge m:1 stockid using stock\_serialnumber

drop \_merge

gen year=year(date)

save comp,replace

//计算Lerner指数

use comp,clear

sort stockid yearqt

gen lntc=ln(tc)

gen lnta=ln(ta)

gen lnta2=[(ln(ta))^2]

gen lnw1=ln(w1)

gen lnw2=ln(w2)

gen lntalnw1=ln(ta)\*ln(w1)

gen lntalnw2=ln(ta)\*ln(w2)

gen lnw12=[(ln(w1))^2]

gen lnw22=[(ln(w2))^2]

gen lnw1lnw2=ln(w1)\*ln(w2)

gen lnw2lnw1=ln(w2)\*ln(w1)

set more off

cons def 1 lnw1+lnw2=1

cons def 2 lntalnw1+lntalnw2=0

cons def 3 lnw12+lnw1lnw2=0

cons def 4 lnw22+lnw2lnw1=0

cons def 5 lnw1lnw2=lnw2lnw1

xi:cnsreg lntc lnta lnta2 lnw1 lnw2 lntalnw1 lntalnw2 lnw12 lnw1lnw2 lnw22 lnw2lnw1 i.year i.stock\_serialnumber,c(1-5)

outreg2 using tc\_solve.doc,bdec(4) replace word

matrix temp=e(b)

forvalues i=1/6 {

gen beta`i'=temp[1,`i']

}

gen mc=(tc/ta)\*(beta1+2\*beta2\*lnta+beta5\*lnw1+beta6\*lnw2)

gen p=revenue/ta

gen lerner=(p-mc)/p

save lerner,replace

use lerner,clear

keep stockid stockname yearqt stock\_serialnumber lerner

sort stockid yearqt

save lerner\_for\_combine,replace

//计算CR4和HHI

use comp,clear

sort stockid yearqt

merge m:1 yearqt using comp5.dta

drop if \_merge==2

drop \_merge

merge m:1 yearqt using comp6.dta

keep if \_merge==3

drop \_merge

gen ta\_p=ta/industry\_asset

gen deposit\_p=deposit/industry\_deposit

gen loan\_p=loan/industry\_loan

gen ta\_2p=ta\_p^2

gen deposit\_2p=deposit\_p^2

gen loan\_2p=loan\_p^2

drop if stockname==""

keep yearqt stock\_serialnumber ta\_p deposit\_p loan\_p ta\_2p deposit\_2p loan\_2p

save concentration.dta,replace

use concentration,clear

sort yearqt stock\_serialnumber

reshape wide ta\_p deposit\_p loan\_p ta\_2p deposit\_2p loan\_2p,i(yearqt) j(stock\_serialnumber)

order yearqt ta\_p\* deposit\_p\* loan\_p\*

egen acr4=rowtotal(ta\_p1 ta\_p3 ta\_p4 ta\_p5),missing

egen dcr4=rowtotal(deposit\_p1 deposit\_p3 deposit\_p4 deposit\_p5),missing

egen lcr4=rowtotal(loan\_p1 loan\_p3 loan\_p4 loan\_p5),missing

egen ahhi=rowtotal(ta\_2p\*),missing

egen dhhi=rowtotal(deposit\_2p\*),missing

egen lhhi=rowtotal(loan\_2p\*),missing

keep yearqt \*cr4 \*hhi

sort yearqt

save concentration.dta,replace

**2、货币政策**

**（1）code\_for\_mp.txt**

//初始化宏观变量

insheet using inflation.csv,clear

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order date

sort date

save inflation,replace

insheet using trade.csv,clear

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order date

sort date

save trade,replace

insheet using m2.csv,clear

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order date

sort date

save trade,replace

insheet using rr.csv,clear

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order date

sort date

save trade,replace

insheet using interbankrate.csv,clear

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order date

sort date

save trade,replace

//初始化GDP数据并采用HP滤波法得到trend和cycle

insheet using gdp.csv,clear

tostring date,replace

gen date2=date(date,"YMD")

gen year2=year(date2)

gen quarter2=quarter(date2)

gen quarter=yq(year2,quarter2)

format date2 %td

format quarter %tq

drop date year2 quarter2

rename date2 date

label var date "日期"

order date quarter

tsset quarter

gen lngdp=ln(gdp)

tsfilter hp gdp\_hp\_cycle=gdp,smooth(1600) trend(gdp\_hp\_trend)

tsfilter hp lngdp\_hp\_cycle=lngdp,smooth(1600) trend(lngdp\_hp\_trend)

save gdp,replace

//

//合并宏观变量

use gdp,clear

merge 1:1 date using inflation.dta

drop \_merge

merge 1:1 date using trade.dta

drop \_merge

merge 1:1 date using m2.dta

drop \_merge

merge 1:1 date using rr.dta

drop \_merge

merge 1:1 date using interbankrate.dta

drop if \_merge==2

drop \_merge

save macro\_temp,replace

//补全银行间拆借利率季度数据

use macro\_temp,clear

keep date interbank\_rate

gsort -date

export excel using interbankrate\_yearqt\_add.xlsx,replace firstrow(variables)

insheet using interbankrate\_yearqt\_add.csv,clear

tostring date,replace

drop if date=="."

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order date

sort date

save interbankrate\_yearqt\_add,replace

//继续合并宏观变量

use macro\_temp,clear

drop interbank\_rate

merge 1:1 date using interbankrate\_yearqt\_add.dta

drop \_merge

gen trade\_p=trade/gdp

save macro,replace

//计算泰勒规则下的IGAP

use macro,clear

global rate\_equilibrium=0.01

global inflation\_target=0.02

global inflation\_gap\_weight=0.05

global gdp\_gap\_weight=0.05

gen rate\_target=$rate\_equilibrium+inflation+$inflation\_gap\_weight\*(inflation-$inflation\_target)+$gdp\_gap\_weight\*lngdp\_hp\_cycle

gen igap=interbank\_rate-inflation-rate\_target

drop inflation trade m2 dm2

drop if rr==.

order date quarter igap rr m2\_growth interbank\_rate

gen cycle=1 if lngdp\_hp\_cycle>0

replace cycle=0 if lngdp\_hp\_cycle<0

sort date

rename quarter yearquarter

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop year quarter year2 quarter2 yearquarter date

order yearqt

sort yearqt

save monetarypolicy,replace

**3、面板合成**

**（1）code\_for\_combine.txt**

//初始化银行监管指标

insheet using bankregulation.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-9,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

sort stockid date

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop date year year2 quarter\*

order stockid stockname yearqt

label var provision\_coverage "拨备覆盖率"

label var loanlossreserve\_loan "拨贷比"

gen npl\_rate=loanlossreserve\_loan/provision\_coverage

gen npl=npl\_rate\*loan

gen provision=loanlossreserve\_loan\*loan

label var npl\_rate "不良贷款率"

label var npl "不良贷款"

label var provision "贷款损失准备"

sort stockid yearqt

save regulation,replace

//以DD\_quarterly为基准，开始不断合并变量

use DD\_quarterly.dta,clear

sort yearqt

egen id=group(yearqt)

sort id stock\_serialnumber

merge 1:1 id stock\_serialnumber using rsquare

keep if \_merge==3

drop \_merge

sort id stock\_serialnumber

merge 1:1 id stock\_serialnumber using deltacovar

keep if \_merge==3

drop \_merge

sort stockid yearqt

merge 1:1 stockid yearqt using lerner\_for\_combine

keep if \_merge==3

drop \_merge

sort stockid yearqt

merge 1:1 stockid yearqt using nii.dta

keep if \_merge==3

drop \_merge

drop loan deposit ltd

sort yearqt

merge m:1 yearqt using monetarypolicy.dta

keep if \_merge==3

drop \_merge

sort yearqt

merge m:1 yearqt using concentration.dta

keep if \_merge==3

drop \_merge

sort stockid yearqt

merge 1:1 stockid yearqt using regulation.dta

drop if \_merge==2

save paneldata\_temp,replace

//补全银行监管指标数据

use paneldata\_temp,clear

keep stockid date loan provision\_coverage loanlossreserve\_loan netcapital car core1\_netcapital core1car rwa

gsort stockid -date

export excel using regulation\_for\_add.xlsx,replace firstrow(variables)

insheet using regulation\_add.csv,clear

tostring stockid,replace

replace stockid=substr("000000"+stockid,-6,6)

tostring date,replace

gen date2=date(date,"YMD")

format date2 %td

drop date

rename date2 date

label var date "日期"

order date

sort date

gen year=year(date)

gen quarter=quarter(date)

tostring year,gen(year2)

tostring quarter,gen(quarter2)

replace quarter2=substr("00"+quarter2,-2,2)

gen yearqt=year2+quarter2

drop date year year2 quarter\*

order stockid yearqt

gen npl\_rate=loanlossreserve\_loan/provision\_coverage

gen npl=npl\_rate\*loan

gen provision=loanlossreserve\_loan\*loan

gen regulation=1/car-1/0.08

replace regulation=0 if car>=0.08

label var provision\_coverage "拨备覆盖率"

label var loanlossreserve\_loan "拨贷比"

label var npl\_rate "不良贷款率"

label var npl "不良贷款"

label var provision "贷款损失准备"

label var regulation "资本监管压力"

save regulation\_add,replace

//继续合并面板数据

use paneldata\_temp,clear

drop loan provision\_coverage loanlossreserve\_loan netcapital car core1\_netcapital core1car rwa npl\_rate npl provision

drop \_merge

sort stockid yearqt

merge 1:1 stockid yearqt using regulation\_add.dta

keep if \_merge==3

drop \_merge

gen yearquarter=yq(year,quarter)

format yearquarter %tq

order stockid stockname year quarter yearquarter yearqt

rename npl npl\_amount

rename npl\_rate npl

rename rwa rwa\_amount

gen rwa=rwa\_amount/asset

save paneldata,replace

//描述性统计

use paneldata,clear

foreach v of varlist add wdd pdd rsquare deltacovar lerner acr4 ahhi rr m2\_growth interbank\_rate igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle {

rename `v' `v'2

winsor `v'2, gen(`v') p(0.01)

drop `v'2

}

set more off

sum pdd rsquare deltacovar lerner acr4 ahhi rr m2\_growth interbank\_rate igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle

sum pdd rsquare deltacovar lerner acr4 ahhi rr m2\_growth interbank\_rate igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle,detail

sort year yearqt stockid

by year,sort:egen yearcount=count(year)

duplicates drop year,force

keep year yearcount

pwcorr pdd rsquare deltacovar lerner acr4 ahhi rr m2\_growth interbank\_rate igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle,sig

sort yearqt

by yearqt,sort:egen pdd\_mean=mean(pdd)

by yearqt,sort:egen lrsq\_mean=mean(rsquare)

by yearqt,sort:egen deltacovar\_mean=mean(deltacovar)

by yearqt,sort:egen lerner\_mean=mean(lerner)

by yearqt,sort:egen cr4\_mean=mean(acr4)

by yearqt,sort:egen hhi\_mean=mean(ahhi)

duplicates drop yearqt,force

keep year quarter yearquarter yearqt pdd\_mean lrsq\_mean deltacovar\_mean lerner\_mean cr4\_mean hhi\_mean rr m2\_growth interbank\_rate igap

line pdd\_mean yearquarter||line lerner\_mean yearquarter,yaxis(2)

line lrsq\_mean yearquarter||line lerner\_mean yearquarter,yaxis(2)

line deltacovar\_mean yearquarter||line lerner\_mean yearquarter,yaxis(2)

line rr yearquarter||line m2\_growth yearquarter||line interbank\_rate yearquarter||line igap yearquarter||line lerner\_mean yearquarter,yaxis(2)

line svi\_avg dif,xtitle("Day") ytitle("Cross-Section of Average SVI") xlabel(-15(5)15) xline(0)

**4、回归分析**

**（1）code\_for\_xtreg.txt**

//回归分析

use paneldata,clear

sort stockid yearqt

gen Dummy2007=(year>2007)

gen banktype2=(banktype==2)

gen banktype3=(banktype==3)

gen negativecycle=1 if cycle==0

replace negativecycle=0 if cycle==1

foreach v of varlist add wdd pdd rsquare deltacovar lerner acr4 ahhi rr m2\_growth interbank\_rate igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle {

rename `v' `v'2

winsor `v'2, gen(`v') p(0.01)

drop `v'2

}

drop id num

egen id=group(stockid)

by id,sort:gen num=\_n

tsset id num

gen lerner\_acr4=lerner\*acr4

gen lerner\_ahhi=lerner\*ahhi

gen lerner\_rr=lerner\*rr

gen ahhi\_igap=ahhi\*igap

gen ahhi\_rr=ahhi\*rr

gen igap\_rr=igap\*rr

gen igap\_m2growth=igap\*m2\_growth

gen igap\_interbankrate=igap\*interbank\_rate

gen lerner\_igap=lerner\*igap

/\*竞争对系统性风险的影响\*/

//PDD与竞争

xi:reg pdd l.pdd l2.pdd lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg pdd l.pdd l2.pdd lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 pdd l.pdd l2.pdd lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 pdd l.pdd l2.pdd lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) twostep robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using pdd\_lerner.doc,bdec(4) replace word

//Rsquare与竞争

xi:reg rsquare l.rsquare l2.rsquare lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg rsquare l.rsquare l2.rsquare lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 rsquare l.rsquare l2.rsquare lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 rsquare l.rsquare l2.rsquare lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) iv(i.id i.year) robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using rsquare\_lerner.doc,bdec(4) replace word

//DeltaCoVaR与竞争

xi:reg deltacovar l.deltacovar l2.deltacovar lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg deltacovar l.deltacovar l2.deltacovar lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 deltacovar l.deltacovar l2.deltacovar lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 deltacovar l.deltacovar l2.deltacovar lerner ahhi lerner\_ahhi lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) iv(i.id i.year) twostep robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using deltacovar\_lerner.doc,bdec(4) replace word

/\*货币政策对系统性风险的影响\*/

//PDD与货币政策

xi:reg pdd l.pdd l2.pdd igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg pdd l.pdd l2.pdd igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 pdd l.pdd l2.pdd igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(igap, laglimits(0 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 pdd l.pdd l2.pdd igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(igap, laglimits(0 .)) iv(i.id i.year) robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using pdd\_igap.doc,bdec(4) replace word

//Rsquare与货币政策

xi:reg rsquare l.rsquare l2.rsquare igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg rsquare l.rsquare l2.rsquare igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 rsquare l.rsquare l2.rsquare igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(igap, laglimits(0 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 rsquare l.rsquare l2.rsquare igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(igap, laglimits(0 .)) robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using rsquare\_igap.doc,bdec(4) replace word

//DeltaCoVaR与货币政策

xi:reg deltacovar l.deltacovar l2.deltacovar igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg deltacovar l.deltacovar l2.deltacovar igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 deltacovar l.deltacovar l2.deltacovar igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(igap, laglimits(0 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 deltacovar l.deltacovar l2.deltacovar igap rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(igap, laglimits(0 .)) robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using deltacovar\_igap.doc,bdec(4) replace word

/\*竞争、货币政策对系统性风险的影响\*/

//PDD与所有变量

xi:reg pdd l.pdd l2.pdd lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg pdd l.pdd l2.pdd lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 pdd l.pdd l2.pdd lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 pdd l.pdd l2.pdd lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using pdd\_lerner\_igap.doc,bdec(4) replace word

//Rsquare与所有变量

xi:reg rsquare l.rsquare l2.rsquare lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg rsquare l.rsquare l2.rsquare lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 rsquare l.rsquare l2.rsquare lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 rsquare l.rsquare l2.rsquare lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using rsquare\_lerner\_igap.doc,bdec(4) replace word

//DeltaCoVaR与所有变量

xi:reg deltacovar l.deltacovar l2.deltacovar lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg deltacovar l.deltacovar l2.deltacovar lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 deltacovar l.deltacovar l2.deltacovar lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 deltacovar l.deltacovar l2.deltacovar lerner ahhi igap rr lerner\_ahhi lerner\_igap lerner\_rr ahhi\_igap ahhi\_rr igap\_rr lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using deltacovar\_lerner\_igap.doc,bdec(4) replace word

/\*竞争、货币政策对系统性风险的影响(约化)\*/

//PDD、Lerner和IGAP

xi:reg pdd l.pdd l2.pdd lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 pdd l.pdd l2.pdd lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 pdd l.pdd l2.pdd lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using pdd\_lerner\_igap\_reduced.doc,bdec(4) replace word

//Rsquare、Lerner和IGAP

xi:reg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 rsquare l.rsquare l2.rsquare lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 rsquare l.rsquare l2.rsquare lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using rsquare\_lerner\_igap\_reduced.doc,bdec(4) replace word

//Deltacovar、Lerner和IGAP

xi:reg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,rob

est store OLS

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE

xi:xtabond2 deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) noleveleq robust small

est store DIFGMM

xi:xtabond2 deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle, gmm(lerner, laglimits(1 .)) gmm(igap, laglimits(0 .)) iv(i.id i.year) robust small

est store SYSGMM

outreg2 [OLS FE DIFGMM SYSGMM] using deltacovar\_lerner\_igap\_reduced.doc,bdec(4) replace word

/\*稳健性检验\*/

//不同银行分类对系统性风险的影响

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.lerner##banktype2 c.lerner##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE1

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.igap##banktype2 c.igap##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE2

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.lerner\_igap##banktype2 c.lerner\_igap##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE3

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.lerner##banktype2 c.lerner##banktype3 c.igap##banktype2 c.igap##banktype3 c.lerner\_igap##banktype2 c.lerner\_igap##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE4

outreg2 [FE1 FE2 FE3 FE4] using pdd\_banktype.doc,bdec(4) replace word

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.lerner##banktype2 c.lerner##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE1

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.igap##banktype2 c.igap##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE2

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.lerner\_igap##banktype2 c.lerner\_igap##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE3

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.lerner##banktype2 c.lerner##banktype3 c.igap##banktype2 c.igap##banktype3 c.lerner\_igap##banktype2 c.lerner\_igap##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE4

outreg2 [FE1 FE2 FE3 FE4] using rsquare\_banktype.doc,bdec(4) replace word

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.lerner##banktype2 c.lerner##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE1

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.igap##banktype2 c.igap##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE2

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.lerner\_igap##banktype2 c.lerner\_igap##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE3

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.lerner##banktype2 c.lerner##banktype3 c.igap##banktype2 c.igap##banktype3 c.lerner\_igap##banktype2 c.lerner\_igap##banktype3 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE4

outreg2 [FE1 FE2 FE3 FE4] using deltacovar\_banktype.doc,bdec(4) replace word

//金融危机对系统性风险的影响

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.lerner##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE1

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.igap##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE2

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.lerner\_igap##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE3

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.lerner##Dummy2007 c.igap##Dummy2007 c.lerner\_igap##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE4

outreg2 [FE1 FE2 FE3 FE4] using pdd\_crisis.doc,bdec(4) replace word

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.lerner##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE1

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.igap##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE2

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.lerner\_igap##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE3

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.lerner##Dummy2007 c.igap##Dummy2007 c.lerner\_igap##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE4

outreg2 [FE1 FE2 FE3 FE4] using rsquare\_crisis.doc,bdec(4) replace word

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.lerner##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE1

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.igap##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE2

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.lerner\_igap##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE3

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.lerner##Dummy2007 c.igap##Dummy2007 c.lerner\_igap##Dummy2007 lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE4

outreg2 [FE1 FE2 FE3 FE4] using deltacovar\_crisis.doc,bdec(4) replace word

//监管压力对系统性风险的影响

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.lerner##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE1

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.igap##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE2

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.lerner\_igap##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE3

xtreg pdd l.pdd l2.pdd lerner igap lerner\_igap c.lerner##c.regulation c.igap##c.regulation c.lerner\_igap##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE4

outreg2 [FE1 FE2 FE3 FE4] using pdd\_regulation.doc,bdec(4) replace word

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.lerner##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE1

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.igap##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE2

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.lerner\_igap##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE3

xtreg rsquare l.rsquare l2.rsquare lerner igap lerner\_igap c.lerner##c.regulation c.igap##c.regulation c.lerner\_igap##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE4

outreg2 [FE1 FE2 FE3 FE4] using rsquare\_regulation.doc,bdec(4) replace word

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.lerner##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE1

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.igap##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE2

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.lerner\_igap##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE3

xtreg deltacovar l.deltacovar l2.deltacovar lerner igap lerner\_igap c.lerner##c.regulation c.igap##c.regulation c.lerner\_igap##c.regulation lnta mvtbv provision\_coverage npl rwa nii regulation gdp\_growth trade\_p cycle i.id i.year,fe

est store FE4

outreg2 [FE1 FE2 FE3 FE4] using deltacovar\_regulation.doc,bdec(4) replace word

**第七章 基于DSGE模型的宏观审慎政策和货币政策的协调**

**1、Dynare工具箱的MOD文件**

**（1）macrodata\_dynare.mod**

//////////////////////////////////////////////////////////////////

////Macro-prudential, Monetary Policy and Bank Systematic Risk////

/////////////////////////////////////////////////////////////////

// This code introduces bank sector in Christiano et al. (2005) model. As for financial frictions, we introduces CAR shock as macro-prudential requirement and bank default loss as bank systematic risk transmission.

// The code runs on the last version of Dynare (Dynare 4.4.0)

// Code written by SCY (shenciyou@qq.com)

// I. ENDOGENOUS VARIABLES

var y m c pi i w n rl rd rk k q u b u\_w u\_p z a g v s;

// II. EXOGENOUS VARIABLES

varexo e\_b e\_w e\_p e\_z e\_a e\_g e\_v e\_s;

// III. PARAMETERS

parameters beta sigma phi nu delta alpha theta gamma omega phi\_i phi\_u xi\_p xi\_w phi\_y phi\_pi rl\_bar rd\_bar rk\_bar rho\_b rho\_w rho\_p rho\_z rho\_a rho\_g rho\_v rho\_s;

// 1. Fixed parameters

beta=0.98868; //消费贴现因子

delta=0.035; //资本折旧率

alpha=0.5; //资本产出弹性

theta=0.08; //巴塞尔协议资本充足率要求

gamma=4; //资本监管周期参数

omega=0.38; //稳态时的投资产出比

phi\_y=-0.2; //货币政策对产出的反应系数

phi\_pi=-0.2; //货币政策对通胀的反应系数

rl\_bar=1.0173; //稳态时的贷款价格

rd\_bar=1.01145; //稳态时的存款价格

rk\_bar=1.084575; //稳态时的资本租赁价格

// 2.Estimated parameters initialisation

sigma=2.75; //消费跨期替代弹性的倒数

phi=0.34052; //劳动跨期替代弹性的倒数

nu=3; //货币需求对利率弹性的倒数

phi\_i=0.148; //投资调整成本参数

phi\_u=0.169; //资本利用成本参数

xi\_p=0.7121; //价格粘性

xi\_w=0.5962; //工资粘性

rho\_b=0.5; //消费需求冲击自相关系数

rho\_w=0.5; //工资加成冲击自相关系数

rho\_p=0.5; //价格加成冲击自相关系数

rho\_z=0.5; //技术冲击自相关系数

rho\_a=0.5; //投资边际效率冲击自相关系数

rho\_g=0.5; //货币政策冲击自相关系数

rho\_v=0.5; //违约冲击自相关系数

rho\_s=0.5; //资本充足率冲击自相关系数

initval;

y=0;

m=0;

c=0;

pi=0;

i=0;

w=0;

n=0;

rl=0;

rd=0;

rk=0;

k=0;

q=0;

u=0;

b=0;

u\_w=0;

u\_p=0;

z=0;

a=0;

g=0;

v=0;

s=0;

e\_b=0;

e\_w=0;

e\_p=0;

e\_z=0;

e\_a=0;

e\_g=0;

e\_v=0;

e\_s=0;

end;

// 3. Derived from steady state

steady;

check;

resid;

// IV. MODEL

model;

// 1. FOC and Constraint Equation

m=(sigma/nu)\*c-(beta/(nu\*(1-beta)))\*rd; //货币需求方程

c=c(+1)+(1/sigma)\*pi(+1)-(1/sigma)\*rd-(1/sigma)\*(b(+1)-b); //消费方程

q=beta\*(1-delta)\*q(+1)-rd+pi(+1)+(1-(beta\*(1-delta)))\*rk(+1); //资本方程

rk=(1/phi\_u)\*u; //资本租赁方程

w=(1/(1+beta))\*(pi(-1)+w(-1))-pi+(beta/(1+beta))\*(w(+1)+pi(+1))-(((1-(beta\*xi\_w))\*(1-xi\_w))/((1+beta)\*xi\_w))\*(w-u\_w-phi\*n-sigma\*c); //工资方程

y=z+alpha\*(k+u)+(1-alpha)\*n; //中间品生产方程

u+k+rk=w+rl+n; //中间品成本方程

pi=(1/(1+beta))\*pi(-1)+(beta/(1+beta))\*pi(+1)+(((1-(beta\*xi\_p))\*(1-xi\_p))/(1+beta))\*(alpha\*rk+(1-alpha)\*(w+rl)-z+u\_p); //通胀方程

k=(1-delta)\*k(-1)+delta\*i(-1); //资本积累方程

i=(1/(1+beta))\*i(-1)+(beta/(1+beta))\*i(+1)+(phi\_i/(1+beta))\*(q-rl)+(beta/(1+beta))\*a(+1)-(1/(1+beta))\*a; //投资方程

rl\_bar\*rl-rk\_bar\*theta\*rk+(rd\_bar\*theta-rd\_bar)\*rd+(rl\_bar+rd\_bar\*theta\*gamma-rk\_bar\*theta\*gamma-1)\*v+(rd\_bar\*theta)\*s=0; //商业银行方程

// 2. Equilibrium

y=(1-omega)\*c+omega\*i; //IS方程

m-m(-1)=g-pi; //LM方程

// 3. AR(1) shocks

b=rho\_b\*b(-1)+e\_b;

u\_w=rho\_w\*u\_w(-1)+e\_w;

u\_p=rho\_p\*u\_p(-1)+e\_p;

z=rho\_z\*z(-1)+e\_z;

a=rho\_a\*a(-1)+e\_a;

g=rho\_g\*g(-1)+phi\_y\*(y-y(-1))+phi\_pi\*pi+e\_g;

v=rho\_v\*v(-1)+e\_v;

s=rho\_s\*s(-1)+e\_s;

// 4. END

end;

// V. SHOCKS

shocks;

var e\_b;

stderr 1;

var e\_z;

stderr 1;

var e\_g;

stderr 1;

var e\_p;

stderr 1;

var e\_w;

stderr 1;

var e\_a;

stderr 1;

var e\_s;

stderr 1;

var e\_v;

stderr 1;

end;

// VI. ESTIMATION

estimated\_params;

stderr e\_b,inv\_gamma\_pdf,0.1,2;

stderr e\_z,inv\_gamma\_pdf,0.1,2;

stderr e\_g,inv\_gamma\_pdf,0.1,2;

stderr e\_p,inv\_gamma\_pdf,0.1,2;

stderr e\_w,inv\_gamma\_pdf,0.1,2;

stderr e\_a,inv\_gamma\_pdf,0.1,2;

stderr e\_s,inv\_gamma\_pdf,0.1,2;

stderr e\_v,inv\_gamma\_pdf,0.1,2;

rho\_b,beta\_pdf,0.85,0.1;

rho\_z,beta\_pdf,0.85,0.1;

rho\_g,beta\_pdf,0.85,0.1;

rho\_p,beta\_pdf,0.85,0.1;

rho\_w,beta\_pdf,0.85,0.1;

rho\_a,beta\_pdf,0.85,0.1;

rho\_s,beta\_pdf,0.85,0.1;

rho\_v,beta\_pdf,0.85,0.1;

xi\_p,beta\_pdf,0.75,0.1;

xi\_w,beta\_pdf,0.75,0.1;

sigma,gamma\_pdf,2.5,0.5;

phi,gamma\_pdf,0.75,0.5;

nu,gamma\_pdf,3,0.5;

phi\_i,normal\_pdf,0.15,0.025;

phi\_u,normal\_pdf,0.15,0.025;

end;

varobs y m c pi i w n;

estimation(bayesian\_irf,irf=40,optim=('MaxIter',200),datafile='macrodata.mat',mode\_compute=1,first\_obs=1,nobs=79,presample=4,lik\_init=2,prefilter=0,mh\_replic=20000,mh\_nblocks=2,mh\_jscale=0.20,mh\_drop=0.2) y m c pi i w n rl rd rk k q u b u\_w u\_p z a g v s;

stoch\_simul(irf=40, conditional\_variance\_decomposition=[1, 10, 40]) y m c pi i w n;

**2、Matlab对Dynare文件的调用**

**（1）dynarecall.m**

addpath c:\dynare\4.x.y\matlab

dynare macrodata\_dynare.mod

**3、Matlab数据预处理模块**

**（1）macrodata\_input.m**

%%Matlab数据预处理模块

clear

clc

tic

%% For the model with Macro-prudential, Monetary Policy and Bank Systematic Risk

data=xlsread('macrodata.xlsx','data','J2:P81'); % Estimation up to 2015Q4

%data=xlsread('macrodata.xlsx','data','J2:P47'); % Estimationup to 2007Q2

gdp=data(:, 1);

investment=data(:,2);

m2=data(:,3);

labor=data(:,4);

consumption=data(:,5);

wage=data(:,6);

cpi=data(:,7);

y2=hpfilter(gdp,4);

c2=hpfilter(consumption,4);

i2=hpfilter(investment,4);

w2=hpfilter(wage,4);

[~,y]=hpfilter(log(y2),Inf);

[~,c]=hpfilter(log(c2),Inf);

[~,i]=hpfilter(log(i2),Inf);

[~,w]=hpfilter(log(w2),Inf);

%%[~,y]=hpfilter(log(gdp),Inf);

%%[~,c]=hpfilter(log(consumption),Inf);

%%[~,i]=hpfilter(log(investment),Inf);

%%[~,w]=hpfilter(log(wage),Inf);

[~,m]=hpfilter(log(m2),Inf);

[~,pi]=hpfilter(log(cpi),Inf);

[~,n]=hpfilter(log(labor),Inf);

clear gdp investment m2 labor consumption wage cpi;

clear data y2 c2 i2 w2

%结果输出

xlswrite('datainput.xls', y,'Sheet1','A1');

xlswrite('datainput.xls', m,'Sheet1','B1');

xlswrite('datainput.xls', c,'Sheet1','C1');

xlswrite('datainput.xls', pi,'Sheet1','D1');

xlswrite('datainput.xls', i,'Sheet1','E1');

xlswrite('datainput.xls', w,'Sheet1','F1');

xlswrite('datainput.xls', n,'Sheet1','G1');

**4、Matlab冲击光滑估计及脉冲响应值输出模块**

**（1）macrodata\_output.m**

%%Matlab数据提取模块

%%提取Smoothed Shocks

load('macrodata\_dynare\_results.mat', 'oo\_');

c=struct2cell(oo\_.SmoothedShocks);

e\_b=cell2mat(c(1,1));

e\_w=cell2mat(c(2,1));

e\_p=cell2mat(c(3,1));

e\_z=cell2mat(c(4,1));

e\_a=cell2mat(c(5,1));

e\_g=cell2mat(c(6,1));

e\_v=cell2mat(c(7,1));

e\_s=cell2mat(c(8,1));

%%结果输出

xlswrite('smoothed\_shocks.xls', e\_b,'Sheet1','A1');

xlswrite('smoothed\_shocks.xls', e\_w,'Sheet1','B1');

xlswrite('smoothed\_shocks.xls', e\_p,'Sheet1','C1');

xlswrite('smoothed\_shocks.xls', e\_z,'Sheet1','D1');

xlswrite('smoothed\_shocks.xls', e\_a,'Sheet1','E1');

xlswrite('smoothed\_shocks.xls', e\_g,'Sheet1','F1');

xlswrite('smoothed\_shocks.xls', e\_v,'Sheet1','G1');

xlswrite('smoothed\_shocks.xls', e\_s,'Sheet1','H1');

%%MatlabIRF结果输出

xlswrite('irfresult.xls', y\_e\_b,'Sheet1','A1');

xlswrite('irfresult.xls', m\_e\_b,'Sheet1','B1');

xlswrite('irfresult.xls', c\_e\_b,'Sheet1','C1');

xlswrite('irfresult.xls', pi\_e\_b,'Sheet1','D1');

xlswrite('irfresult.xls', i\_e\_b,'Sheet1','E1');

xlswrite('irfresult.xls', w\_e\_b,'Sheet1','F1');

xlswrite('irfresult.xls', n\_e\_b,'Sheet1','G1');

xlswrite('irfresult.xls', y\_e\_z,'Sheet1','H1');

xlswrite('irfresult.xls', m\_e\_z,'Sheet1','I1');

xlswrite('irfresult.xls', c\_e\_z,'Sheet1','J1');

xlswrite('irfresult.xls', pi\_e\_z,'Sheet1','K1');

xlswrite('irfresult.xls', i\_e\_z,'Sheet1','L1');

xlswrite('irfresult.xls', w\_e\_z,'Sheet1','M1');

xlswrite('irfresult.xls', n\_e\_z,'Sheet1','N1');

xlswrite('irfresult.xls', y\_e\_g,'Sheet1','O1');

xlswrite('irfresult.xls', m\_e\_g,'Sheet1','P1');

xlswrite('irfresult.xls', c\_e\_g,'Sheet1','Q1');

xlswrite('irfresult.xls', pi\_e\_g,'Sheet1','R1');

xlswrite('irfresult.xls', i\_e\_g,'Sheet1','S1');

xlswrite('irfresult.xls', w\_e\_g,'Sheet1','T1');

xlswrite('irfresult.xls', n\_e\_g,'Sheet1','U1');

xlswrite('irfresult.xls', y\_e\_p,'Sheet1','V1');

xlswrite('irfresult.xls', m\_e\_p,'Sheet1','W1');

xlswrite('irfresult.xls', c\_e\_p,'Sheet1','X1');

xlswrite('irfresult.xls', pi\_e\_p,'Sheet1','Y1');

xlswrite('irfresult.xls', i\_e\_p,'Sheet1','Z1');

xlswrite('irfresult.xls', w\_e\_p,'Sheet1','AA1');

xlswrite('irfresult.xls', n\_e\_p,'Sheet1','AB1');

xlswrite('irfresult.xls', y\_e\_w,'Sheet1','AC1');

xlswrite('irfresult.xls', m\_e\_w,'Sheet1','AD1');

xlswrite('irfresult.xls', c\_e\_w,'Sheet1','AE1');

xlswrite('irfresult.xls', pi\_e\_w,'Sheet1','AF1');

xlswrite('irfresult.xls', i\_e\_w,'Sheet1','AG1');

xlswrite('irfresult.xls', w\_e\_w,'Sheet1','AH1');

xlswrite('irfresult.xls', n\_e\_w,'Sheet1','AI1');

xlswrite('irfresult.xls', y\_e\_a,'Sheet1','AJ1');

xlswrite('irfresult.xls', m\_e\_a,'Sheet1','AK1');

xlswrite('irfresult.xls', c\_e\_a,'Sheet1','AL1');

xlswrite('irfresult.xls', pi\_e\_a,'Sheet1','AM1');

xlswrite('irfresult.xls', i\_e\_a,'Sheet1','AN1');

xlswrite('irfresult.xls', w\_e\_a,'Sheet1','AO1');

xlswrite('irfresult.xls', n\_e\_a,'Sheet1','AP1');

xlswrite('irfresult.xls', y\_e\_s,'Sheet1','AQ1');

xlswrite('irfresult.xls', m\_e\_s,'Sheet1','AR1');

xlswrite('irfresult.xls', c\_e\_s,'Sheet1','AS1');

xlswrite('irfresult.xls', pi\_e\_s,'Sheet1','AT1');

xlswrite('irfresult.xls', i\_e\_s,'Sheet1','AU1');

xlswrite('irfresult.xls', w\_e\_s,'Sheet1','AV1');

xlswrite('irfresult.xls', n\_e\_s,'Sheet1','AW1');

xlswrite('irfresult.xls', y\_e\_v,'Sheet1','AX1');

xlswrite('irfresult.xls', m\_e\_v,'Sheet1','AY1');

xlswrite('irfresult.xls', c\_e\_v,'Sheet1','AZ1');

xlswrite('irfresult.xls', pi\_e\_v,'Sheet1','BA1');

xlswrite('irfresult.xls', i\_e\_v,'Sheet1','BB1');

xlswrite('irfresult.xls', w\_e\_v,'Sheet1','BC1');

xlswrite('irfresult.xls', n\_e\_v,'Sheet1','BD1');