Readme File

This readme file describes the data, programs, and steps necessary to replicate the findings in the paper. There are three sections:

a) Banking Data: this section describes the data.
b) Structural breaks: this section describes how estimation and testing for structural breaks happens, describing the Matlab codes.
c) IFE Regressions: this section describes how the regression results are obtained, after the breaks and their number have been estimated. Stata commands are presented.

1. Banking Data

The data were initially obtained from the Call Reports of the Federal Deposit Insurance Corporation. This is a balanced sample containing data on 3,557 commercial banks, observed from 2005Q3 to 2021Q3. In other words, N = 3, 557 banks that are observed over T = 64 quarters, for a total of 227, 648 bank-quarter observations. The variable names are:

Dependent Variables	
rdlntl	Log of total loans
rdlnre	Log of real estate loans
rdlnci	Log of commercial and industrial loans
Regressors	
mbshold	Mortgage-backed securities over total assets
schold	Total securities held minus MBS securities over total assets
lnta	Log of total assets
Eqta	Equity over total assets
Roa	Return on assets
Bankcash	Balance sheet cash flow over total assets
Depcost	Interest expense of deposits over total assets
Netincprc	Net income over total assets
Dinfl	The first difference of US inflation
Gdpgrowth	GDP US per capita, growth

The data appear in two formats. First, in a csv file so that they are widely accessible and can be imported in Stata, and second, in a Matlab format which can be used directly to replicate the Matlab results.

2. Structural Breaks

This section describes how to estimate the number and location of breaks.

2.1 Matlab Code Instructions

There are various imputs, some are parameters and some are variables. The variables need to be in specic form:

y is Tx1xN

x is TxpxN

w is TxqxN

db is Txq1

dnb is Txq2

These transformations can be done as follows.

Let ystar be a NTx1 variable, then use y=reshape(ystar,T,1,N)

Let var1, var2 and var 3 be a NTx1 variables, then use x=reshape([var1 var2 var3],T,N,3); x=permute(x, [1 3 2]);

Let cm1, cm2 be a NTx1 common factors where values repear accross units, then use dnb=reshape([var1 var2],T,N,2); dnb=permute(dnb, [1 3 2]);

Same for db

Files:

h1kcsd_gen_det.m

Runs the test for hypotesis 1 (F). Testing for breaks, when the number and location of breaks is known

h1ukcsd_gen_det.m

Runs the test for hypotesis 1 (supF). Testing for breaks, when the number

of breaks is known but their location unknown

h2csd_gen_det.m

Runs the test for hypotesis 2 (UDmax). Testing for breaks, when the number of breaks is unknwon (up to kmax)

h3pdcsd_gen_det.m

Runs the test for hypothesis 3 testing the null of b breaks vs the alternative of b+1 breaks.

h3repeat_gen_det.m

Runs the test for hypothesis 3 sequentially, starting from 0 breaks in the

null up to 6 breaks in the null (vs 7 under the alternative).

dpcsd_gen_det.m

Estimates the locations of breaks, given the number of breaks, for the

full structural change model

pcdpcsd_gen_det.m

Estimates the locations of breaks, given the number of breaks, for the

partial structural change model

breakCI_gen_det

Calculates confidence intervals for estimated break dates

Instructions per file:

h1kcsd_gen_det.m

Inputs: N number fo cross section units,T number of time series observations,

breaks vector containing the location of breaks i.e. [12 26] means breaks

at the 12th and 26th date in the sample,

p number of regressors which are not breaking,q number of regressors which are breaking,y dependent variable,

x vector of independent variables not breaking i.e. [var1 var2 var 3],

w vector of breaking independent variables i.e. [var4 var5],

db vector of breaking common factors i.e. [cm1 cm2],

dnb vector of non-breaking common factors i.e. [cm3],

phi dummy variable where 0 means no serial correlation and 1 means serial

correlation in the covariance matrix estimation.

h1ukcsd_gen_det.m

Same as above plus: tr trimming parameter 0.05, 0.1, 0.15

h2csd_gen_det.m

Same as above plus: M is maximum number of breaks

h3pdcsd_gen_det.m

Same as above plus: b is the number of breaks under the null (to be tested against b+1) under the alternative

h3repeat_gen_det.m

Same as above

breakCI_gen_det

Same as above

dpcsd_gen_det.m

Same as above plus: b is the number of breaks, no need to use this as it is called by the above commands where necessary

pcdpcsd_gen_det.m

Same as above plus: b is the number of breaks, no need to use this as it is called by the above commands where necessary

Replication code:

```
warning("off")
leqta=leqta/100;
lroa=lroa/1000;
lnetincprc=lnetincprc/100;
lbankcash=lbankcash/100;
lloandep=lloandep/100;
ldepcost=ldepcost/100;
lmbshold=lmbshold/100;
lschold=lschold/100;
lhighmbsv=lhighmbsv/100;
lhighscv=lhighscv/100;
```

```
x=reshape([llnta leqta lroa lbankcash ldepcost lnetincprc],T,N,6);
x=permute(x, [1 3 2]);
dnb=reshape([ldinfl lgdpgrowth],T,N,2);
dnb=permute(dnb, [1 3 2]);
dnb=dnb(:,:,1);
db=ones(T,1);
w=reshape([lmbshold lschold],T,N,2);
w=permute(w, [1 3 2]);
p=6;
q=2;

y=reshape(rdlnre,T,1,N);
[UDmax,UDmaxcrit]=h2csd_gen_det(N,T,y,x,w,db,dnb,p,q,10,0.05,0)

y=reshape(rdlnci,T,1,N);
[UDmax,UDmaxcrit]=h2csd_gen_det(N,T,y,x,w,db,dnb,p,q,10,0.05,0)

y=reshape(rdlntl,T,1,N);
[UDmax,UDmaxcrit]=h2csd_gen_det(N,T,y,x,w,db,dnb,p,q,10,0.05,0)

y=reshape(rdlnre,T,1,N);
[supF,finalbreaks]=h1ucsd_gen_det(N,T,7,p,q,y,x,w,db,dnb,0.05,0)

y=reshape(rdlnci,T,1,N);
[supF,finalbreaks]=h1ucsd_gen_det(N,T,7,p,q,y,x,w,db,dnb,0.05,0)

y=reshape(rdlntl,T,1,N);
[supF,finalbreaks]=h1ucsd_gen_det(N,T,7,p,q,y,x,w,db,dnb,0.05,0)
```

3. Regressions

First, import the csv into Stata. The following commands can be entered in Stata to retrieve the regression results of Table 4.

```
replace eqta=eqta/100
replace roa=roa/1000
replace netincprc=netincprc/100
replace bankcash=bankcash/100
replace loandep=loandep/100
replace demdep=demdep/100
replace depcost=depcost/100
replace mbshold=mbshold/100
replace schold=schold/100

drop if qtime==39

gen qe1=0
replace qe1=1 if qtime>=45
gen qe2=0
replace qe2=1 if qtime>=53
gen qe3=0
replace qe3=1 if qtime>=63
gen qe4=0
replace qe4=1 if qtime>=71
gen qe5=0
replace qe5=1 if qtime>=76
gen qe6=0
replace qe6=1 if qtime>=85
gen qe7=0
replace qe7=1 if qtime>=97
```

```
gen mbsholdqe1=mbshold*qe1

gen mbsholdqe2=mbshold*qe2

gen mbsholdqe3=mbshold*qe3

gen mbsholdqe4=mbshold*qe4

gen mbsholdqe5=mbshold*qe5

gen mbsholdqe6=mbshold*qe6

gen mbsholdqe7=mbshold*qe7


gen scholdqe1=schold*qe1

gen scholdqe2=schold*qe2

gen scholdqe3=schold*qe3

gen scholdqe4=schold*qe4

gen scholdqe5=schold*qe5

gen scholdqe6=schold*qe6

gen scholdqe7=schold*qe7


eststo: xtdcce2fast rdlntl l.mbshold l.mbsholdqe1 l.mbsholdqe2 l.mbsholdqe3 l.mbsholdqe4
l.mbsholdqe5 l.mbsholdqe6 l.mbsholdqe7 l.schold l.scholdqe1 l.scholdqe2 l.scholdqe3
l.scholdqe4 l.scholdqe5 l.scholdqe6 l.scholdqe7 l.lnta l.eqta l.roa l.bankcash l.depcost
l.netincprc, crosssectional(l.mbshold l.mbsholdqe1 l.mbsholdqe2 l.mbsholdqe3 l.mbsholdqe4
l.mbsholdqe5 l.mbsholdqe6 l.mbsholdqe7 l.schold l.scholdqe1 l.scholdqe2 l.scholdqe3
l.scholdqe4 l.scholdqe5 l.scholdqe6 l.scholdqe7 l.lnta l.eqta l.roa l.bankcash l.depcost
l.netincprc l.qe1 l.qe2 l.qe3 l.qe4 l.qe5 l.qe6 l.qe7 l.gdpgrowth l.dinfl) pooled pooledvce(nw)


eststo: xtdcce2fast rdlnre l.mbshold l.mbsholdqe1 l.mbsholdqe2 l.mbsholdqe3 l.mbsholdqe4
l.mbsholdqe5 l.mbsholdqe6 l.mbsholdqe7 l.schold l.scholdqe1 l.scholdqe2 l.scholdqe3
l.scholdqe4 l.scholdqe5 l.scholdqe6 l.scholdqe7 l.lnta l.eqta l.roa l.bankcash l.depcost
l.netincprc, crosssectional(l.mbshold l.mbsholdqe1 l.mbsholdqe2 l.mbsholdqe3 l.mbsholdqe4
l.mbsholdqe5 l.mbsholdqe6 l.mbsholdqe7 l.schold l.scholdqe1 l.scholdqe2 l.scholdqe3
l.scholdqe4 l.scholdqe5 l.scholdqe6 l.scholdqe7 l.lnta l.eqta l.roa l.bankcash l.depcost
l.netincprc l.qe1 l.qe2 l.qe3 l.qe4 l.qe5 l.qe6 l.qe7 l.gdpgrowth l.dinfl) pooled pooledvce(nw)


drop qe1-scholdqe7


gen qe1=0
```

```
replace qe1=1 if qtime>=45
gen qe2=0
replace qe2=1 if qtime>=53
gen qe3=0
replace qe3=1 if qtime>=63
gen qe4=0
replace qe4=1 if qtime>=71
gen qe5=0
replace qe5=1 if qtime>=76
gen qe6=0
replace qe6=1 if qtime>=84
gen qe7=0
replace qe7=1 if qtime>=97

gen mbsholdqe1=mbshold*qe1
gen mbsholdqe2=mbshold*qe2
gen mbsholdqe3=mbshold*qe3
gen mbsholdqe4=mbshold*qe4
gen mbsholdqe5=mbshold*qe5
gen mbsholdqe6=mbshold*qe6
gen mbsholdqe7=mbshold*qe7

gen scholdqe1=schold*qe1
gen scholdqe2=schold*qe2
gen scholdqe3=schold*qe3
gen scholdqe4=schold*qe4
gen scholdqe5=schold*qe5
gen scholdqe6=schold*qe6
gen scholdqe7=schold*qe7
```

eststo: xtdcce2fast rdlnci l.mbshold l.mbsholdqe1 l.mbsholdqe2 l.mbsholdqe3 l.mbsholdqe4 l.mbsholdqe5 l.mbsholdqe6 l.mbsholdqe7 l.schold l.scholdqe1 l.scholdqe2 l.scholdqe3 l.scholdqe4 l.scholdqe5 l.scholdqe6 l.scholdqe7 l.lnta l.eqta l.roa l.bankcash l.depcost l.netincprc, crosssectional(l.mbshold l.mbsholdqe1 l.mbsholdqe2 l.mbsholdqe3 l.mbsholdqe4 l.mbsholdqe5 l.mbsholdqe6 l.mbsholdqe7 l.schold l.scholdqe1 l.scholdqe2 l.scholdqe3 l.scholdqe4 l.scholdqe5 l.scholdqe6 l.scholdqe7 l.lnta l.eqta l.roa l.bankcash l.depcost l.netincprc l.qe1 l.qe2 l.qe3 l.qe4 l.qe5 l.qe6 l.qe7 l.gdpgrowth l.dinfl) pooled pooledvce(nw)