RESEARCH DATA CENTER
REX - Remote Execution / Sistema di elaborazione a distanza

Individual bank balance sheets for statistical purposes (IBBaSS)

Other information

Examples: platforms ‘Stata’ and ‘R’

November 2023

Table of contents

1. General characteristics

The examples in the document are provided with a view to promoting good practices for the exploitation of the microdata on banks’ balance sheets produced and published by the Bank of Italy.

To obtain more rapid computing results, limit the number of variables used in the calculations. Please note that permanent datasets cannot be stored.

The examples below are constructed importing CSV files that contain both data on banks’ balance sheets (IBBaSS) and on other information (BANCI), selected through the menu of available datasets. It is strongly recommended that you read the documentation regarding these databases before using the examples below.

2. Stata[1] Examples

Stata commands are lower case. Please note that Stata is a case-sensitive programming language.

Example 1 – Bank classification

- We want to exclude a bank category from the database, e.g. Italian branches of foreign banks.

```
clear all
/* Loading BANCI data on groups */
import delimited "grucla.csv”
save grucla.dta, replace

/* Loading IBBaSS data and merge by bank id to add category */
import delimited "ibbass.csv", clear
merge 1:1 idba_ps data using grucla.dta
keep if _merge == 3
drop _merge
tabulate cla_dim

/* Drop Italian branches of foreign banks (category E)*/
keep if cla_dim != "E"
tabulate cla_dim
```

Example 2 – Mergers and acquisitions

- We want to associate the identification code of the merged bank with that of the last acquiring bank in any merger and acquisition operations involving the MFI throughout the time horizon of the analysis and add the two series together. The

[1] ‘Stata’ is a registered trademark of StataCorp LP, 4905 Lakeway Drive, College Station, TX 77845 USA.

resulting series simulates a scenario in which the merged MFIs have always been a single bank, thus correcting the break.

```
clear all
/* Loading BANCI data on merges and acquisitions */
import delimited “fui.csv", varnames(1)
keep idba_ps cve22
/* Temporary save */
save fui_22.dta, replace

/* Loading IBBaSS data and merge the two datasets */
import delimited “ibbass.csv", clear
merge m:1 idba_ps using fui_22.dta
keep if _merge != 2
drop _merge

/* Creating new id keeping also the old one */
rename idba_ps idba_ps_vero
rename cve22 idba_ps
replace idba_ps = idba_ps_vero if idba_ps == .

/* Grouping by new bank id all balance sheet items */
quietly ds idba_ps* data, not
collapse (sum) `r(varlist)', by(idba_ps data)


/* As a consequence of the grouping, the number of banks has shrank */
```

Example 3 – Banking groups

- We want to associate the microdata on balance sheets with the map of groups and aggregate the time series of banks belonging to the same group. The result is obtained by associating the banks contained in the IBBaSS dataset with their parent MFI at each date.

```
clear all
/* Loading BANCI data on groups */
import delimited “grucla.csv"
save grucla.dta, replace
```

```
/* Loading IBBaSS data and add group information */
import delimited "ibbass.csv", clear
merge 1:1 idba_ps data using grucla.dta
keep if _merge == 3
drop _merge

/* Grouping data by banking group id */
quietly ds idba* cla_dim* flag* data, not
collapse (sum) `r(varlist)', by(data idba_capogruppo_ps)
```

Example 4 – Banking groups with mergers and acquisitions

- We want to associate the microdata on balance sheets with the map of groups and aggregate the time series of banks belonging to the same group. As an alternative to the variable composition of banking groups[2] (see Example 3), we extract the data on any groups with which the MFI may be affiliated at the latest date reported in the map and 'project' it onto the previous dates. To obtain greater continuity in the time series relating to the groups, we can adjust parent MFIs for mergers and acquisitions.

```
clear all
/* Loading BANCI data on merges and acquisitions */
import delimited "fui.csv", varnames(1)
keep idba_ps cve22
save fui_22.dta, replace

/* Loading BANCI data on groups to obtain the latest parent MFI for each bank */
import delimited "grucla.csv", clear
preserve
collapse (max) data, by (idba_ps)
save last_capogruppo.dta, replace
restore
merge m:1 idba_ps data using last_capogruppo.dta
keep if _merge == 3
keep idba_ps idba_capogruppo_ps
rename idba_capogruppo_ps idba_capogruppo_last
```

[2] Over time, parent MFIs may lose or acquire a controlling interest in the banks belonging to the group. This causes breaks in the time series at group level, making the series volatile.

```
save grucla_capogruppo_last, replace


/* Adding latest parent MFI information to IBBaSS dataset */
import delimited "ibbass.csv", clear
merge m:1 idba_ps using fui_22.dta
keep if _merge != 2
drop _merge
rename idba_ps idba_ps_vero
rename cve22 idba_ps
replace idba_ps = idba_ps_vero if idba_ps == .
merge m:1 idba_ps using grucla_capogruppo_last
keep if _merge != 2
drop _merge
replace idba_capogruppo_last = idba_ps if idba_capogruppo_last ==.


/* Grouping total main asset by latest parent MFI */
collapse (sum) tma_a_1_z5_0000_z01, by(idba_capogruppo_last data)
```

Example 5 (available only for Stata) – Balance sheet aggregates used by Bank of Italy

- We create aggregate balance sheet items based on definitions that are more in line with those used by the Bank of Italy in its official publications and Statistical Database.

```
import delimited “ibbass.csv”, clear


gen annome=int(data/100)

gen anno  =int(annome/100)

gen mese  = annome-anno*100

gen datam = ym(anno,mese)

drop anno annome mese data

format datam %tm


* ------------------------------------------------------------

*                ASSETS   (BSIB refers to the corresponding BDS table where series are included)

* ------------------------------------------------------------

* ------------------------------------------------

*  cash  - BSIB0100

* ------------------------------------------------

rename a10_x_1_z5_0000_z01 cash


* ------------------------------------------------
*  Loans by counterpart sector - BSIB0600
* --------------------------------------------------------------------------------------------------------------------------
* Italian General Government
rename a20_a_1_u6_2100_z01 loanggo_ita

* Italian non financial corporations
rename a20_a_1_u6_2240_z01 loannfc_ita

* Italian households
rename a20_a_1_u6_2250_z01 loanhho_ita

* [Italian] OFIs, non-MMF investment funds and ICPFs Euro area net of reverse repos with Domestic CCPs
gen temp = a20r_a_1_u6_2272_z01*-1
egen loanofi_ita = rowtotal(a20_a_1_u6_22f0_z01 temp)
drop temp

* Total domestic private loans (net of CCPs)
egen loantot_ita = rowtotal(loanggo_ita loannfc_ita loanhho_ita loanofi_ita)

* Italian bank loans - (Total IFM net of Bank of Italy)
gen temp = a20_a_1_u6_1100_z01*-1
```

```
egen loanban_ita = rowtotal(a20_a_1_u6_1000_z01 temp)

drop temp



* ----------------------------------------------------------------------------------------

*  Debts securities - BSIB0100

* ----------------------------------------------------------------------------------------

* Excluding banks debt securities repurchased by the issuer;

gen temp = 0

replace temp = l40p_a_1_z5_0000_z01*-1

egen tidban_ita = rowtotal(a30_a_1_u6_1000_z01 temp)

drop temp



* Italian General Government

rename a30_a_1_u6_2100_z01 tidggo_ita



* Italian private sector

rename a30_a_1_u6_2200_z01 tidpriv_ita



* Total domestic debt securities excluding Bank bonds

egen tidnba_ita = rowtotal(tidggo_ita tidpriv_ita)
```

```
* ----------------------------------------------------------------------------------------------------------
*  Claims on the Bank of Italy
* ----------------------------------------------------------------------------------------------------------
rename a20_a_1_u6_1100_z01 depeur_Bit


* ----------------------------------------------------------------------------------------------------------
*  External assets - BSIB0100
* ----------------------------------------------------------------------------------------------------------


egen   assets_ext   =   rowtotal(a20_a_1_u5_2100_z01   a20_a_1_u5_1000_z01   a20_a_1_u5_2240_z01   a20_a_1_u5_2250_z01
  a30_a_1_u5_2100_z01     a30_a_1_u5_1000_z01     a30_a_1_u5_2200_z01     a50_a_1_u5_1000_z01     a50_a_1_u5_2200_z01
  a20_a_1_u4_0000_z01 a30_a_1_u4_0000_z01 a50_a_1_u4_0000_z01 a20_a_1_u5_22f0_z01)


* ----------------------------------------------------------------------------------------------------------
*  total assets - BSIB0100
*                 but the series tma_: is less volatile and preferable
* ----------------------------------------------------------------------------------------------------------


rename tat_a_1_z5_0000_z01 assets_tot


egen temp1 = rowtotal(cash loantot_ita loanban_ita tidnba_ita tidban_ita depeur_Bit assets_ext a20r_a_1_u6_2272_z01)
```

```
gen temp = temp1*-1


* other assets

egen assets_oth = rowtotal(assets_tot temp)

drop temp*

* --------------------------------------------------------------------------------------------------------------------------------------

*                LIABILITIES   (BSIB refers to the corresponding BDS table where series are included)

* --------------------------------------------------------------------------------------------------------------------------------------

* ---------------------------------------------------------------------------------------------------------

*  Deposits by counterpart sector - BSIB0300

* ---------------------------------------------------------------------------------------------------------

* [Italian] General Government

egen depoggo_ita = rowtotal(l20_a_1_u6_2110_z01 l2c_m_1_u6_2120_z01 l2d_n_1_u6_2120_z01)


* Italian non financial corporations

egen deponfc_ita = rowtotal(l2c_m_1_u6_2240_z01 l2d_n_1_u6_2240_z01)


* Italian households

egen depohho_ita = rowtotal(l2c_m_1_u6_2250_z01 l2d_n_1_u6_2250_z01)


* [talian] OFIs, non-MMF investment funds and ICPFs Euro area net of repos with Domestic CCPs
* agreed maturities over 2y incuded in L2D_N_1_U2_2210_Z01 are mainly connected with transfers of claims
egen temp1 = rowtotal(l24_a_1_u6_2272_z01 l2c_m_1_u6_2120_z01 l2d_n_1_u6_2120_z01)
gen  temp  = temp1*-1
egen depoofi_ita = rowtotal(l2c_m_1_u6_22fg_z01 l2d_n_1_u6_22fg_z01 temp)
drop temp*

* Total domestic private deposits (net of CCPs)
egen depotot_ita = rowtotal(depoggo_ita deponfc_ita depohho_ita depoofi_ita)

* Italian bank deposits - (Total IFM net of Bank of Italy)
gen temp = l20_a_1_u6_1100_z01*-1
egen depoban_ita = rowtotal(l20_a_1_u6_1000_z01 temp)
drop temp

* repos (net of reverse) with CCPs
gen temp         = a20r_a_1_u6_2272_z01*-1
egen depoccg_net = rowtotal(l24_a_1_u6_2272_z01 temp)
drop temp
* ----------------------------------------------------------------------------------------
*  Debts securities issued - BSIB0500     includes   intra-group positions
* ----------------------------------------------------------------------------------------
* Excluding banks debt securities repurchased by the issuer
gen temp = l40p_a_1_z5_0000_z01*-1
egen bondissu = rowtotal(l40_a_1_z5_0000_z01 temp)
drop temp

* --------------------------------------------------------------------------------------------------------------------------------
*  Liabilities towards the Bank of Italy
* --------------------------------------------------------------------------------------------------------------------------------
rename l20_a_1_u6_1100_z01 loan_Bit

* --------------------------------------------------------------------------------------------------------------------------------
*  Capital & reserves
* --------------------------------------------------------------------------------------------------------------------------------
rename l60_x_1_z5_0000_z01 capris

* ------------------------------------------------------------------------------------------------------------------------------------------------------------------
*  External liabilities - BSIB0100
* ------------------------------------------------------------------------------------------------------------------------------------------------------------------
```

```
egen  liabilities_ext  =  rowtotal(l2c_m_1_u5_2240_z01  l2d_n_1_u5_2240_z01  l2c_m_1_u5_2250_z01  l2d_n_1_u5_2250_z01
 l20_a_1_u5_1000_z01    l20_a_1_u4_0000_z01    l20_a_1_u5_2110_z01    l2c_m_1_u5_2120_z01    l2d_n_1_u5_2120_z01
 l2c_m_1_u5_22fg_z01 l2d_n_1_u5_22fg_z01)


* -------------------------------------------------------------------------------------------------------------------------------
*  Other liabilities
* -------------------------------------------------------------------------------------------------------------------------------


egen temp1 = rowtotal(depotot_ita depoban_ita bondissu loan_Bit capris liabilities_ext l24_a_1_u6_2272_z01)
gen  temp  = temp1*-1
egen liabilities_oth = rowtotal(assets_tot temp)
drop temp*
```

3 Examples from ‘R’[3]

Please note that ‘R’ is a case-sensitive programming language.

All the examples assume that there is only one command per line and that the same command can extend over multiple lines if it is too long.

Example 1 – Bank classification

- We want to exclude a bank category from the database, e.g. Italian branches of foreign banks.

```
# Loading BANCI data on groups
grucla <- read.csv("grucla.csv", sep=";")

# Loading IBBaSS data and merge by bank id to add category
ibbass <- read.csv("ibbass.csv", sep=";")
merged_data <- merge(ibbass, grucla, by = c("idba_ps", "data"), all.x=T)
table(merged_data$cla_dim)

# Drop Italian branches of foreign banks (category E)
filtered_data <- merged_data[merged_data$cla_dim != "E",]
table(filtered_data$cla_dim)
```

Example 2 – Mergers and acquisitions

- We want to associate the identification code of the merged bank with that of the last acquiring bank in any merger and acquisition operations involving the MFI throughout the time horizon of the analysis and add the two series together. The resulting series simulates a scenario in which the merged MFIs have always been a single bank, thus correcting the break.

```
# Loading useful packages
library(dplyr)

# Loading BANCI data on merges and acquisitions
fui <- read.csv("fui.csv", sep=";")
fui <- fui[, c("idba_ps", "cve22")]

# Loading IBBaSS data and merge the two datasets
ibbass <- read.csv("ibbass.csv", sep=";")
merged_data <- merge(ibbass, fui, by = "idba_ps", all.x = T)
```

[3] ‘R’ is an open source software environment for statistical data analysis; for more information on the programming language, visit the websitehttp://cran.r-project.org/ .

```
# Creating new id keeping also the old one
merged_data <- merged_data %>%
 mutate(idba_vero = idba_ps,
            idba_ps = ifelse(is.na(cve22), idba_vero, cve22))

# Grouping by new bank id all balance sheet items
data_new <- merged_data %>%
 select(-idba_vero, -cve22) %>%
 group_by(idba_ps, data) %>%
 summarise_all(sum, na.rm=T)

# As a consequence of the grouping, the number of banks has shrank
```

Example 3 – Banking groups

- We want to associate the microdata on balance sheets with the map of groups and aggregate the time series of banks belonging to the same group. The result is obtained by associating the banks contained in the IBBaSS dataset with their parent MFI at each date.

```
# Loading useful packages
library(dplyr)

# Loading BANCI data on groups
grucla <- read.csv("grucla.csv", sep=";")

# Loading IBBaSS data and adding group information
ibbass <- read.csv("ibbass.csv", sep=";")
merged_data <- merge(ibbass, grucla, by = c("idba_ps", "data"), all.x = T)
merged_data <- merged_data %>%
 mutate(idba_capogruppo_ps = ifelse(is.na(idba_capogruppo_ps), idba_ps,
  idba_capogruppo_ps))
```

```
# Grouping data by banking group id
data_new <- merged_data %>%
 select(-idba_ps, -cla_dim, -flag_bcc) %>%
 group_by(idba_capogruppo_ps, data) %>%
 summarise_all(sum,.na.rm=T)
```

Example 4 – Banking groups with mergers and acquisitions

- We want to associate the microdata on balance sheets with the map of groups and aggregate the time series of banks belonging to the same group. As an alternative to the variable composition of banking groups[4] (see Example 3), we extract the data on any groups with which the MFI may be affiliated at the latest date reported in the map and 'project' it onto the previous dates. To obtain greater continuity in the time series relating to the groups, we can adjust parent MFIs for mergers and acquisitions.

```
# Loading useful packages
library(dplyr)


# Loading BANCI data on merges and acquisitions
fui <- read.csv("fui.csv", sep=";")
fui_22 <- fui[, c("idba_ps", "cve22")]


# Loading BANCI data on groups to obtain the latest parent MFI for each bank
grucla <- read.csv("grucla.csv", sep=";")
tmp <- arrange(grucla, idba_ps, -data)
last_capogruppo <- subset(mutate(group_by(tmp, idba_ps),
                          capo = 1:length(data)),
                          capo==1)[, c("idba_ps","idba_capogruppo_ps")]
last_capogruppo <- rename(last_capogruppo,
                          cg_last = idba_capogruppo_ps)
grucla_new <- merge(grucla, last_capogruppo, by.all = idba_ps)
```

[4] Over time, parent MFIs may lose or acquire a controlling interest in the banks belonging to the group. This causes breaks in the time series at group level, making the series volatile.

```
# Add latest parent MFI information to IBBaSS dataset
ibbass <- read.csv("ibbass.csv", sep=";")
data_new <- merge(ibbass, fui_22, by = "idba_ps", all.x = T) %>%
 rename(idba_vero=idba_ps, idba_ps=cve22) %>%
 mutate(idba_ps = ifelse(is.na(idba_ps), idba_vero, idba_ps))

data_all <- merge(data_new, grucla_new, by.all = idba_ps, all.x = T)

# Grouping total main asset by latest parent MFI
tma_cg <- data_all %>%
 select(tma_a_1_z5_0000_z01, cg_last, data) %>%
 group_by(cg_last, data) %>%
 summarise(tma_a_1_z5_0000_z01 = sum(tma_a_1_z5_0000_z01, na.rm=T))
```