



RESEARCH DATA CENTER

accesso ai microdati per la ricerca economica

I bilanci bancari a livello individuale a fini statistici e altre informazioni

Esempi di utilizzo dati: Piattaforme Stata e R



BANCA D'ITALIA

EUROSISTEMA

Novembre 2023

Sommario

1.	Aspetti generali	3
2.	Esempi piattaforma Stata	3
	Esempio n. 1 – Gestione categorie delle banche	3
	Esempio n. 2 – Gestione di una fusione all’indietro.....	3
	Esempio n. 3 – Gestione dei gruppi bancari.....	4
	Esempio n. 4 – Gestione dei gruppi e delle fusioni.....	5
	Esempio n. 5 (disponibile solo in Stata) – Gli aggregati di bilancio usati in Banca d’Italia	7
3.	Esempi piattaforma R.....	15
	Esempio n. 1 – Gestione categorie delle banche	15
	Esempio n. 2 – Gestione di una fusione all’indietro.....	15
	Esempio n. 3 – Gestione dei gruppi bancari.....	16
	Esempio n. 4 – Gestione dei gruppi e delle fusioni.....	17

1. Aspetti generali

Gli esempi riportati nel documento sono predisposti nell'ottica di promuovere buone prassi per lo sfruttamento dei microdati sui bilanci bancari, prodotti e diffusi dalla Banca d'Italia.

Per ottenere più rapidamente i risultati delle proprie elaborazioni si suggerisce di limitare il numero di variabili utilizzate nelle elaborazioni. Si ricorda che non si possono memorizzare dataset permanenti.

Negli esempi che seguono vengono importati i file CSV contenenti sia i dati sui bilanci ([IBBaSS](#)) sia quelli sulle altre informazioni ([BANCI](#)), individuati dal menu dei dataset disponibili. Si prega vivamente di consultare la documentazione relativa a tali archivi, prima di utilizzare gli esempi sottostanti.

2. Esempi piattaforma Stata¹

I comandi Stata sono scritti in minuscolo: va ricordato che questo linguaggio è case-sensitive.

Esempio n. 1 – Gestione categorie delle banche

- Si vuole escludere una categoria di banca, nell'esempio le filiali di banche estere, dall'archivio.

```
clear all

/* Importo archivio con le categorie di banca */
import delimited "grucla.csv"
save grucla.dta, replace

/* Importo IBBaSS e associo all'identificativo della banca la categoria */
import delimited "ibbass.csv", clear
merge 1:1 idba_ps data using grucla.dta
keep if _merge == 3
drop _merge
tabulate cla_dim

/* Escludo le banche della categoria selezionata */
keep if cla_dim != "E"
tabulate cla_dim
```

Esempio n. 2 – Gestione di una fusione all'indietro

- Si vuole associare al codice identificativo della banca passiva quello dell'ultima banca attiva nelle operazioni di fusione e incorporazione nelle quali l'intermediario sia stato eventualmente coinvolto, lungo l'orizzonte dell'analisi, e sommare le due

¹ Stata è un marchio registrato della StataCorp LP, 4905 Lakeway Drive, College Station, TX 77845 USA.

serie. L'unica serie risultante simula che gli intermediari assoggettati a operazioni di concentrazione siano da sempre un'unica banca, correggendo la discontinuità.

```
clear all

/* Importo la tabella BANCII su fusioni e incorporazioni */
import delimited "fui.csv", varnames(1)
keep idba_ps cve22
save fui_22.dta, replace

/* Importo IBBaSS e associo fusioni e incorporazioni */
import delimited "ibbass.csv", clear
merge m:1 idba_ps using fui_22.dta
keep if _merge != 2
drop _merge

/* per tutte le voci aggrego gli importi sulla base di identificativi aggiornati */
quietly ds idba* data cve22, not
collapse (sum) `r(varlist)', by(cve22 data)

/* a seguito dell'aggiornamento la numerosità delle banche si è ridotta */
```

Esempio n. 3 – Gestione dei gruppi bancari

- Si vuole associare ai microdati sui bilanci la mappa dei gruppi e aggregare le serie storiche delle banche appartenenti allo stesso gruppo. Tale risultato si ottiene associando alle banche presenti in IBBaSS la rispettiva capogruppo, valida in ciascuna data.

```
clear all

/* Importo archivio con la mappa dei gruppi (capogruppo=idba_ps banca se non
fa parte di gruppi) */
import delimited "grucla.csv"
save grucla.dta, replace

/* Importo IBBaSS e associo capogruppo */
import delimited "ibbass.csv", clear
merge 1:1 idba_ps data using grucla.dta
keep if _merge == 3
```

```
/* per tutte le voci aggrego gli importi sulla base della capogruppo (=ident banca
se non appartenenti a gruppi) */
quietly ds idba* cla_dim* flag* data, not
collapse (sum) `r(varlist)', by(data idba_capogruppo_ps)
```

Esempio n. 4 – Gestione dei gruppi e delle fusioni

- Si vuole associare ai microdati sui bilanci la mappa dei gruppi e aggregare le serie storiche delle banche appartenenti allo stesso gruppo. In alternativa alla composizione variabile dei gruppi² (cfr. esempio n. 3), si estrae l'eventuale appartenenza dell'intermediario a un gruppo bancario all'ultima data presente nella mappa e si "proietta" sulle date precedenti. Per ottenere serie storiche dei gruppi in maggiore continuità, si possono correggere per le fusioni e le incorporazioni le capogruppo.

```
clear all

/* Importo la tabella BANCII su fusioni e incorporazioni */
import delimited "fui.csv", varnames(1)
keep idba_ps cve22
save fui_22.dta, replace

/* Importo la tabella BANCII su gruppi e classificazioni per generare la colonna
della capogruppo più recente */
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
drop _merge
keep idba_ps idba_capogruppo_ps
rename idba_capogruppo_ps idba_capogruppo_last
save grucla_capogruppo_last.dta, replace
```

² Le capogruppo possono nel tempo perdere o acquisire il controllo delle rispettive componenti bancarie. Tale fenomeno determina discontinuità nelle serie a livello di gruppo, che si presentano volatili.

```
/* Associa la capogruppo più recente agli identificativi aggiornati */  
import delimited "ibbass.csv", clear  
merge m:1 idba_ps using fui_22.dta  
keep if _merge != 2  
drop _merge  
rename idba_ps idba_vero  
rename cve22 idba_ps  
replace idba_ps = idba_vero if idba_ps == ""  
merge m:1 idba_ps using grucla_capogruppo_last.dta  
keep if _merge != 2  
drop _merge  
replace idba_capogruppo_last = idba_ps if idba_capogruppo_last == ""  
  
/* aggrego gli importi del totale attivo sulla base della capogruppo più recente */  
collapse (sum) tma_a_1_z5_0000_z01, by(idba_capogruppo_last data)
```

Esempio n. 5 (disponibile solo in Stata) – Gli aggregati di bilancio usati in Banca d'Italia

- Costruisco le voci di bilancio aggregate, secondo definizioni maggiormente in linea con quelle utilizzate dalla Banca d'Italia nelle proprie pubblicazioni ufficiali e nella Base dati statistica.

```
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 = 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 serie 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)

* [Italian] 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 depooofi_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 depooofi_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_z01l20_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. Esempi piattaforma R³

Si ricorda che questo linguaggio è case-sensitive.

Tutti gli esempi presuppongono che su ogni riga ci sia un solo comando e che lo stesso comando possa essere esteso su più righe, se troppo lungo.

Esempio n. 1 – Gestione categorie delle banche

- Si vuole escludere una categoria di banca, nell'esempio le filiali di banche estere, dall'archivio.

```
# Importo archivio con le categorie di banca
grucla <- read.csv("grucla.csv", sep=";")

# Importo IBBaSS e associo all'identificativo della banca la categoria
ibbass <- read.csv("ibbass.csv", sep=";")
merged_data <- merge(ibbass, grucla, by = c("idba_ps", "data"), all.x=T)
table(merged_data$cla_dim)

# Escludo le banche della categoria selezionata
filtered_data <- merged_data[merged_data$cla_dim != "E",]
table(filtered_data$cla_dim)
```

Esempio n. 2 – Gestione di una fusione all'indietro

- Si vuole associare al codice identificativo della banca passiva quello dell'ultima banca attiva nelle operazioni di fusione e incorporazione nelle quali l'intermediario sia stato eventualmente coinvolto, lungo l'orizzonte dell'analisi, e sommare le due serie. L'unica serie risultante simula che gli intermediari assoggettati a operazioni di concentrazione siano da sempre un'unica banca, correggendo la discontinuità.

```
# Carico i pacchetti utili
library(dplyr)

# Importo la tabella BANCII su fusioni e incorporazioni
fui <- read.csv("fui.csv", sep=";")
fui <- fui[, c("idba_ps", "cve22")]

# Importo IBBaSS e associo fusioni e incorporazioni
```

³ R è un ambiente opensource per l'analisi statistica dei dati; se si desiderano ulteriori informazioni sul linguaggio, si consiglia di visitare il sito <http://cran.r-project.org/>.

```

ibbass <- read.csv("ibbass.csv", sep=";")
merged_data <- merge(ibbass, fui, by = "idba_ps", all.x = T)

# Definisco il nuovo identificativo, conservo anche quello originario
merged_data <- merged_data %>%
  mutate(idba_vero = idba_ps,
         idba_ps = ifelse(is.na(cve22), idba_vero, cve22))

# Per tutte le voci aggrego gli importi sulla base degli identificativi aggiornati
data_new <- merged_data %>%
  select(-idba_vero, -cve22) %>%
  group_by(idba_ps, data) %>%
  summarise_all(sum, na.rm=T)

# A seguito dell'aggiornamento la numerosità delle banche si è ridotta

```

Esempio n. 3 – Gestione dei gruppi bancari

- Si vuole associare ai microdati sui bilanci la mappa dei gruppi e aggregare le serie storiche delle banche appartenenti allo stesso gruppo. Tale risultato si ottiene associando alle banche presenti in IBBaSS la rispettiva capogruppo, valida in ciascuna data.

```

# Carico i pacchetti utili
library(dplyr)

# Importo archivio con la mappa dei gruppi (capogruppo=ident banca se non fa
  parte di gruppi)
grucla <- read.csv("grucla.csv", sep=";")

# Importo IBBaSS e associo capogruppo
ibbass <- read.csv("ibbass.csv", sep=";")
merged_data <- merge(ibbass, grucla, by = c("idba_ps", "data"), all.x = T) %>%
  mutate(idba_capogruppo_ps = ifelse(is.na(idba_capogruppo_ps), idba_ps,
    idba_capogruppo_ps))

```

```
# Per tutte le voci aggrego gli importi sulla base della capogruppo
data_new <- merged_data %>%
  select(-idba_ps, -cla_dim, -flag_bcc) %>%
  group_by(idba_capogruppo_ps, data) %>%
  summarise_all(sum, na.rm=T)
```

Esempio n. 4 – Gestione dei gruppi e delle fusioni

- Si vuole associare ai microdati sui bilanci la mappa dei gruppi e aggregare le serie storiche delle banche appartenenti allo stesso gruppo. In alternativa alla composizione variabile dei gruppi⁴ (cfr. esempio n. 3), si estrae l'eventuale appartenenza dell'intermediario a un gruppo bancario all'ultima data presente nella mappa e si "proietta" sulle date precedenti. Per ottenere serie storiche dei gruppi in maggiore continuità, si possono correggere per le fusioni e le incorporazioni le capogruppo.

```
# Carico i pacchetti utili
library(dplyr)

# Importo la tabella BANCI su fusioni e incorporazioni
fui <- read.csv("fui.csv", sep=";")
fui_22 <- fui[, c("idba_ps", "cve22")]

# Importo la tabella BANCI su gruppi e classificazioni per generare la colonna della
  capogruppo più recente
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)

# Associa la capogruppo più recente agli identificativi aggiornati
ibbass <- read.csv("ibbass.csv", sep=";")
data_new <- merge(ibbass, fui_22, by.all = 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))
```

⁴ Le capogruppo possono nel tempo perdere o acquisire il controllo delle rispettive componenti bancarie. Tale fenomeno determina discontinuità nelle serie a livello di gruppo, che si presentano volatili.

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

# Aggrego gli importi del totale attivo sulla base della capogruppo più recente
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))
```