Guida al comando eudbimport

Nicola Tommasi*

11 settembre 2022

*Centro Interdipartimentale di Documentazione Economica (C.I.D.E.)
email: nicola.tommasi@univr.it

Indice

1. Fonte dei dati

Tutti i database compatibili con il comando **eudbimport** sono reperibili a questo indirizzo che però a breve sarà sostituito da questo questo.
Per scaricare un database serve conoscere il suo nome, sul sito è indicato tra parentesi quadre come si vede in figura 1 dove sono elencati i database EI_BSCO_M, EI_BSCO_Q ed EI_BSIN_M_R2.

Figura 1 — Come reperire il nome di un database

È possibile avere l'elenco completo dei database. Cliccate sul link del nuovo sito e poi su DOWNLOADS (vedi figura 2). Selezionate Data, in Download operations selezionate Download full list of items e infine su Apply. Verrà prodotto e proposto per il download il file Full_Items_List_EN.txt. Questo file può essere usato per scaricare tutti i database del sito (sono circa 6900); il download è piuttosto lungo, 7-8 ore, e scarica circa 50GB di dati (QUESTO DATO è DA VERIFICARE).

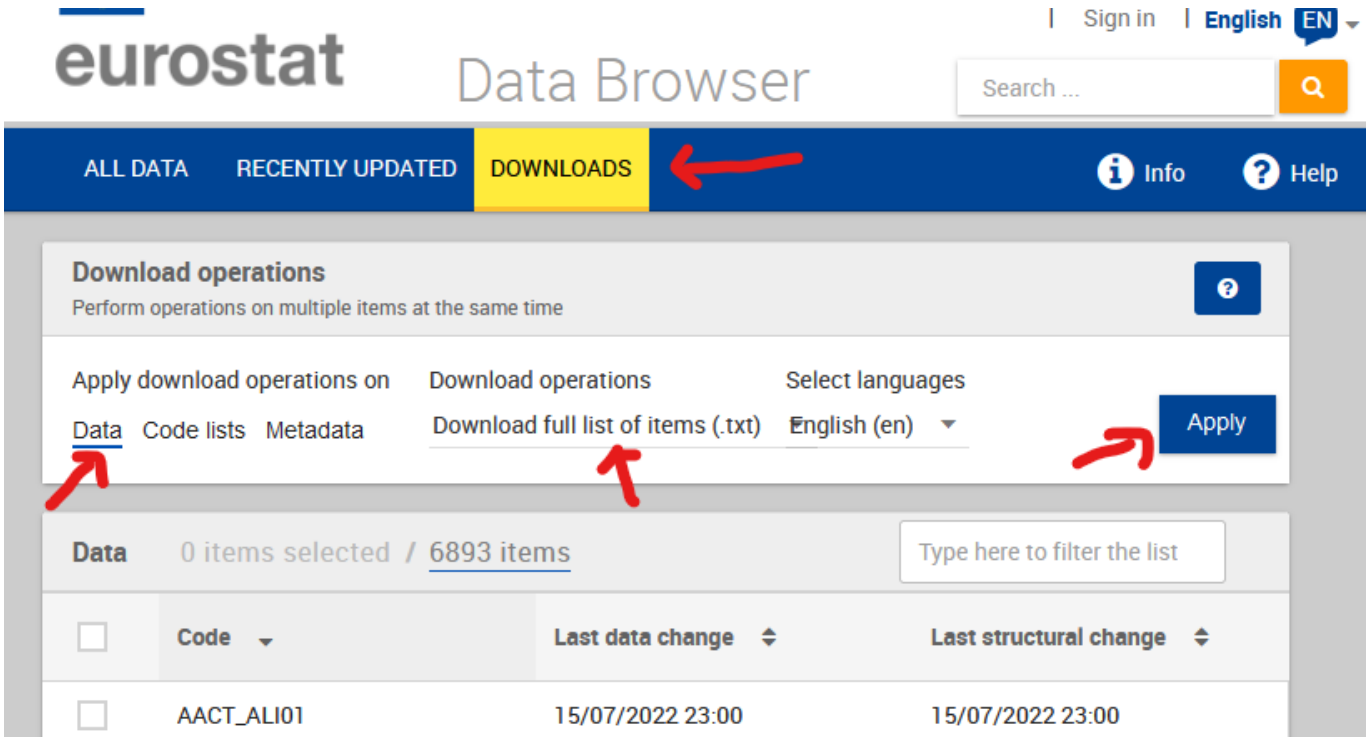

Figura 2 — Come reperire la lista dei database

In seguito verrà mostrato un do-file che a partire da questo file esegue il download completo dei database in esso elencati.

Per fare il label delle variabili servono una serie di altri files presenti nella sezione Code lists (vedi sempre figura 2). Questi files non sono necessari se non siete interessati al label delle variabili. Se invece volete il label dovete selezionare l'item relativo alla variabile che vi interessa e fare il download. Gli item dei codes lists sono circa 600 e per scaricarli si possono selezionare solo a blocchi di 100. Anche in questo caso verrà mostrato e descritto un do-file che si occupa di caricare i files relativi e creare dei do-file con lo scopo di fare il label define delle variabili.
L'ispirazione vien da qui, dall'ottimo e ricchissimo sito di Asjad Naqvi.

2. Struttura dei databases

Tutti i database presenti sul sito hanno una struttura di base costante. Vediamone un esempio e famigliarizziamo con una serie di definizioni. Questo (figura 3) è il database MIGR_IMM3CTB. Nella parte superiore si vedono quelle che d'ora in poi chiameremo splitvars. In questo caso sono Geopolitical entity, Time,

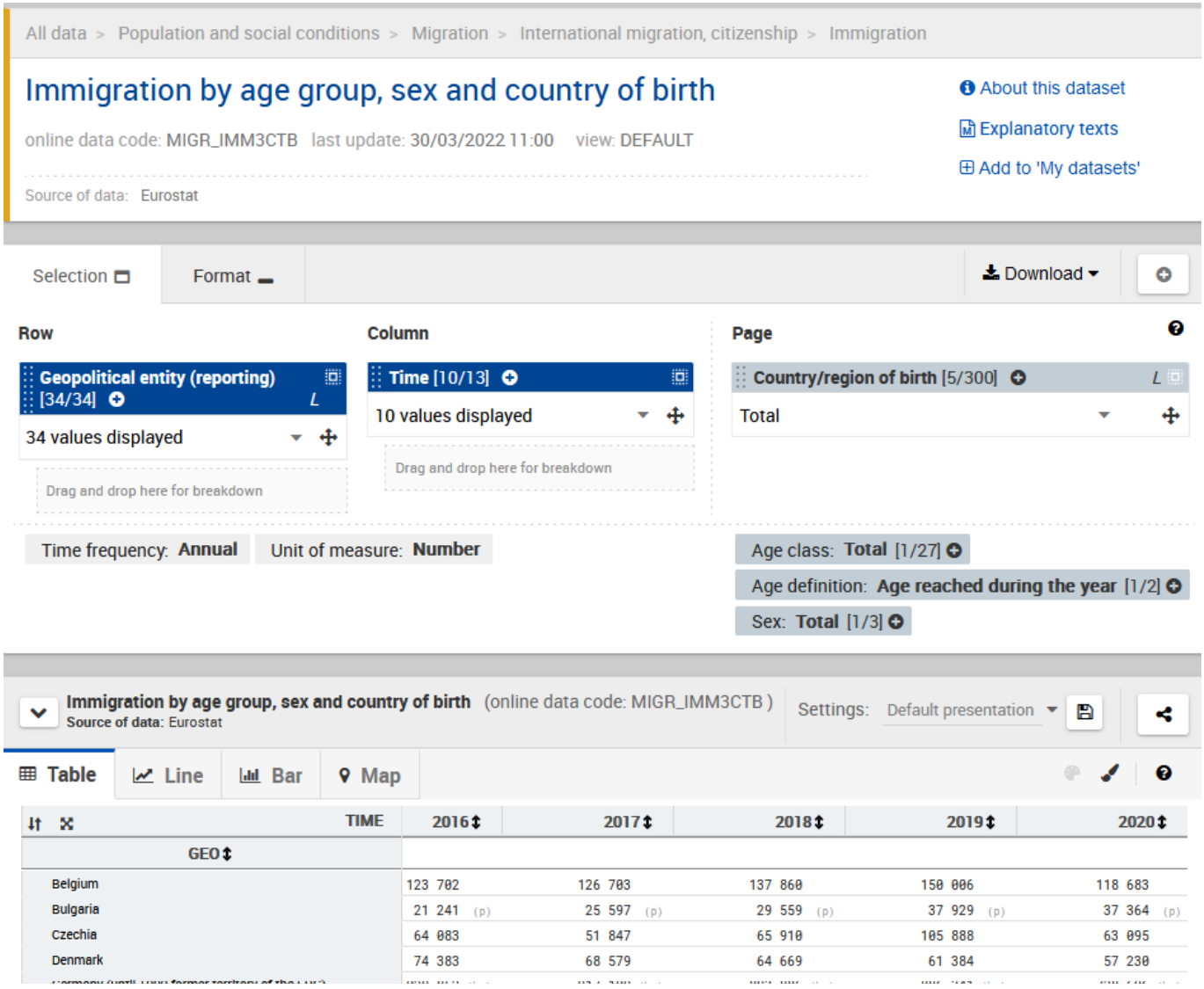

Figura 3 — La struttura dei database

Country/region of birth, Time frequency, Unit of measure, Age Class, Age definition e Sex. Nella parte inferiore si vedono i dati relativi a questa configurazione delle splitvars:

- sulle righe le 34 specificazioni di Geopolitical entity
- sulle colonne gli anni selezionati in Time
- il valore Total di Country/region of birth

- il valore Total di Age Class
- il valore Age reached during the year di Age definition
- il valore total di Sex
- la frequenza è annuale
- l'unità di misura è Number

Il simbolo + vicino a ciascuna splitvars indica che è possibile selezionare una diversa specificazione di quella variabile. Per esempio in Age class sono disponibili 27 diverse configurazioni delle classi di età (Total, Less than 5 years, From 5 to 9 years ...).
Il comando **eudbimport** sostanzialmente permette di trasformare questo database in una time serie, cioè trasforma la variabile temporale che qui è rappresentata nelle colonne in una variabile e trasforma i valori di una delle splitvars nelle nuove variabili di colonna attraverso un reshape. Questa variabile è definita reshapevar e le sue specificazioni diventeranno nuove variabili.

3. Scaricare tutti i databasein local

Adesso mostro come scaricare in locale tutti i database del sito EUROSTAT. Questa operazione non è necessaria perchè come sarà mostrato in seguito il comando `eudbimport` prevede la possibilità di scaricare direttamente il file del database dal sito di EUROSTAT o, alternativamente, di importare il file precedentemente scaricato da una cartella del PC.
Per prima cosa dobbiamo scaricare il file Full_Items_List_EN.txt con l'elenco completo dei databases (circa 7000)[1]. È stato descritto prima come procurarsi questo file.

```
import delimited "Full_Items_List_EN.txt", clear varnames(1)

qui count
forvalues i=1/`r(N)' {
  local urltsv = datadownloadurltsv in `i'
  **local urlcsv = datadownloadurlcsv in `i'
  local filename = code  in `i'
  di "Download file `filename' - `i' of `r(N)'"
  copy "`urltsv'" `filename'.tsv, replace
  **copy "`urlcsv'" `filename'.csv, replace

  if "`c(os)'" == "Unix" shell 7zz a -t7z `filename'.7z `filename'.tsv  -mx9 -bb1
  else shell "$E7z" a -t7z `filename'.7z `filename'.tsv  -mx9 -bb1
  erase `filename'.tsv
}
```

Il codice è abbastanza semplice. Con `import delimited` importiamo in Stata il file txt, poi inizia un ciclo prendere certe informazioni da ciascuna riga del file importato. Nella local urltsv viene registrato l'url per accedere al database e nella local filename il suo nome. Poi il comando copy provvede a scaricare filename in locale con estensione .tsv. Le ultime 2 righe si occupano di comprimere filename.tsv.
Dato che alcuni database sono abbastanza grandi (EF_M_FARMANG.tsv ad esempio pesa circa 2.3GiB), è possibile zippare i singoli file tsv scaricati. Per adesso ho implementato solo 7zip come metodo di compressione ma a breve aggiungerò anche il classico zip.

1. 6893 il 10 settembre 2022

3.1. Come usare 7zip in Stata

Il sito del programma è https://www.7-zip.org/ dove trovate la versione per i diversi sistemi operativi. Installate quella adatta al vostro.
Se il vostro sistema operativo è Windows:
Installare il programma 7zip
Nella cartella di installazione di Stata creare il file profile.do o se esiste già aggiungere questa riga:
global E7z "C:\Program Files\7-Zip\7z.exe"
Il percorso riportato è quello di installazione del programma, modificarlo di conseguenza se lo avete installato in una directory diversa.
In Linux:
A seconda della vostra distribuzione installate il pacchetto 7zip-full (si chiama così ??). Poi non serve fare più nulla, in linux non serve intervenire nel file profile.do
In MacOS:
boooo!!!

4. Costruzione dei do-file per gli items

Nel capitolo 1 è stato descritto come scaricare dal sito di EUROSTAT i file relativi ai Codes list. Ora vediamo come sono strutturati questi files. Sono dei files di testo con estensione .tsv e il nome è ESTAT_-<CODELIST>_en.tsv[2] dove CODELIST è il nome della variabile a cui fanno riferimento i code list. Per esempio ESTAT_ACCIDENT_en.tsv riporta i codici relativi alla variabile accident (Accident). Questo è il suo contenuto:

```
TOTAL        Total
SRS         Serious accidents
SRS_F        Serious accidents - women
SRS_M        Serious accidents - men
FATAL        Fatal accidents
COLLIS        Collisions of trains, including collisions with obstacles within the clearance gauge
COLLIS_X_LVLCROS        Collisions (excluding level-crossing accidents)
DERAIL        Derailments of trains
DGD         Accidents involving transport of dangerous goods
NDGD         Accidents not involving transport of dangerous goods
DGD_RL        Accidents in which dangerous goods are released
DGD_NRL        Accidents in which dangerous goods are not released
LVLCROS        Level crossing accidents
RSTK_MOT        Accidents to persons caused by rolling stock in motion
RD_TRF        Road traffic
HOM_SCH        Accident at home / school / leisure
HOM_LEIS        Home and leisure
HOM         Home
LEIS         Leisure
RSTK_FIRE        Fires in rolling stock
FAT_NT        Fatalities in injury accidents on national territory (all operators)
ACC_NT        Injury accidents on national territory (all operators)
FAT_NC        Fatalities in injury accidents where a national company was involved (worldwide)
ACC_NC        Injury accidents where a national company was involved (worldwide)
OTH         Others
UNK         Unknown
```

Lo possiamo vedere come un database con due variabili. La prima variabile contiene i valori che può assumere la variabile accident, la seconda le relative descrizioni. Le due variabili sono separati da tabulazione. Ma perchè questi files sono utili? Perchè se accident sarà la variabile scelta per il reshape (opzione `reshapevar()`), le sue specifiche saranno le variabili create con il reshape e le descrizioni potranno essere usate per fare il label delle variabili. Questa operazione viene svolta dal file db_items.do, che viene di seguito riportato e commentato:

2. _en finale è per la versione inglese del file, le altre possibili sono fr e de.

```
clear all
set more off

capture mkdir items
capture mkdir dic
**cd items

local itemslist : dir "items" files "*.tsv", respectcase
local nitems : word count `itemslist'
di `nitems'

foreach f of local itemslist {
  local item : subinstr local f "ESTAT_" ""
  local item : subinstr local item "_en.tsv" ""
  di "`item'"
  local item = lower("`item'")

  import delimited "items/`f'", clear encoding(UTF-8) stringcols(_all) delimiter(tab)
  **devono essere lette solo 2 variabili
  qui describe
  assert r(k)==2
  duplicates report v1
  assert r(unique_value)==r(N)
  *! regola per avere nomi compatibili
  di "`f'"
  if "`f'" == "ESTAT_ICD10_en.tsv" {
    replace v1="C54__C55" if v1=="C54-C55"
    replace v1="F00__F03" if v1=="F00-F03"
    replace v1="G40__G41" if v1=="G40-G41"
  }
  if "`f'" == "ESTAT_LCSTRUCT_en.tsv" replace v1="D12__D4_MD5" if v1=="D12-D4_MD5"
  if "`f'" == "ESTAT_NACE_R1_en.tsv" {
    replace v1="C__E" if v1=="C-E"
    replace v1="L__Q" if v1=="L-Q"
  }
  if "`f'" == "ESTAT_NACE_R2_en.tsv" {
    replace v1="B06__B09" if v1=="B06-B09"
    replace v1="O__U" if v1=="O-U"
  }

  **forse è un errore la presenza di _2000W01 dato che sono date
  if "`f'" == "ESTAT_TIME_en.tsv" drop if v1=="_2000W01"

  if "`f'" == "ESTAT_UNIT_en.tsv" replace v1="MIO__EUR__NSA" if v1=="MIO-EUR-NSA"

  replace v1 = ustrtoname(v1,1)

  duplicates report v1
  assert r(unique_value)==r(N)

  gen labelvar = "cap label var " + v1 + `" ""' + v2 + `"""'

  if "`item'"=="variable" local item VARIABLE
  outfile labelvar using "dic/labvar_`item'.do", replace noquote
}

exit
```

Una volta scaricati tutti i files relativi ai code list nella cartella items, si crea una lista di questi files da passasre ad un ciclo. Nel ciclo, dal nome del file si si isola il nome della variabile a cui fa riferimento (local item), si importa il contenuto del file (import delimited) e si verifica che non sisiano duplicati. Questo controllo è importante altrimenti poi ci sarebbero due o più variabili con lo stesso nome. Le specifiche assunte dalle diverse variabili non sempre sono compatibili con le regole relative ai nomi delle variabili in Stata. Il grosso di queste incompatibilità viene risolto dalla funzione `ustrtoname()` che opera tre tipi di modifiche

- aggiunge _ (underscore) se una specifica inizia con un numero. Si vedano ad esempio le specifiche di item_newa.
- sostituisce il carattere - con _ (underscore) se una specifica contiene tale carattere. Si vedano ad esempio le specifiche di icd10.
- tronca la specifica al trentaduesimo carattere nel caso fosse più lunga

rimangono pochi casi da risolvere manualmente, dovuti principalmente al fatto che le modifiche prodotte da `ustrtoname()` generano dei duplicati. Ad esempio in icd10 esistono sia la specifica C54-C55 che C54_C55. `ustrtoname()` converte la prima in C54_C55 creando un duplicato. In questi casi si interviene aggiungento un secondo _ in modo che C54-C55 diventi C54__C55. Anche il comando eudbimport esegue le stesse operazioni.
A questo punto si crea la variabile labelvar, ovvero una stringa che contiene il comando label variabile relativo alla variabile. Poi con `outfile` questa variabile viene esportato il un do-file. Questo per esempio è il risultato realtivo alla variabile accomsize:

```
cap label var TOTAL "Total"
cap label var LT10 "Less than 10 bedplaces"
cap label var GE10 "10 bedplaces or more"
```

5. Il file eudbimport_labvar.do

Contiene il label di tutte le variabili presenti nei vari db. Non è strettamente necessario, il label può essere fatto anche manualmente dopo aver importato il db. Se manca qualche label, può essere aggiunto editando il do-file e aggiungendo una riga di comando con lo schema `capture label var <varname> "descripion"`.

6. Regole per rinominare le variabili

La variabile scelta nell'opzione `reshapevar()` può non essere compatibile con le regole che Stata impone per i nomi delle variabili. I motivi sono essenzialmente due, più un terzo che si verifica solo una volta:

- nomi che iniziano con un numero
- nomi che contengono il carattere "-"
- nomi che corrispondo a reserved word di Stata[3]

Per i primi due casi viene usata la funzione `ustrtoname(s,1)` converte tutti i caratteri non ammessi in Stata con l'_, che aggiunge _ se un nome inizia con un carattere numerico e tronca il nome a 32 caratteri. L'utilizzo di `ustrtoname(s,1)` è stato modificato nei seguenti casi per evitare casi di nomi duplicati in seguito alla modifica apportata dalla funzione

- nell'item ICD10
  - C54-C55 convertito in C54__C55
  - F00-F03 convertito in F00__F03
  - G40-G41 convertito in G40__G41
- nell'item LCSTRUCT
  - D12-D4_MD5 convertito in D12__D4_MD5
- nell'item NACE_R1
  - C-E convertito in C__E
  - L-Q convertito in L__Q

3. Fino ad ora ho trovato un solo caso che riguarda il nome variable e che è stato convertito in VARIABLE nel database ...DA TROVARE

- nell'item NACE_R2
  - B06-B09 convertito in B06__B09
  - O-U convertito in O__U
- nell'item NA_ITEM
  - D2_D5_D91_D61_M_D611V_D612_M_M_D613V_D614_M_D995 convertito in D2_D5_D91_D61_-M_D611V_D612_M_M_D
  - D2_D5_D91_D61_M_D612_M_D614_M_D995 convertito in D2_D5_D91_D61_M_D612_M_D614_-M_D9
- nell'item UNIT
  - MIO-EUR-NSA convertito in MIO__EUR__NSA

Questa è da verificare:
forse è un errore la presenza di _2000W01 dato che sono date in ESTAT_TIME_en.tsv drop if v1=="_-2000W01"

6.1. Le frequenze presenti nei db

Tipo	Formato	Etichetta	Stata compliant
Dati giornalieri	####-##-## (YYYY-MM-DD)	D	YYYYMMDD
Dati settimanali	####-W## (YYYY-WW)	W	YYYYwWW
Dati mensili	####-M## (YYYY-MM)	M	YYYYmMM
Dati trimestrali	####-Q# (YYYY-Q)	Q	YYYYQQ
Dati semestrali	####-S# (YYYY-S)	S	YYYYhH
Dati annuali	#### (YYYY)	A	YYYY

dati settimanali da così 2015-W01 a così 2015w1

Settimanali e semestrali sono da verificare
Regole per rendere le frequenze compatibili con i formati datetime e quindi per convertirli in formato numerico
Regole per rendere le frequenze compatibili con i nomi delle variabili in Stata, nel caso di database con frequenze multiple. Questa operazione è da fare per poter fare il reshape.

6.2. Trattamento delle frequenze

Si presentano i seguenti casi:

1. Database con frequenza unica. La frequenza viene convertita in formato numerico Stata compliant
2. Database con frequenze diverse, per esempio con dati mensili, trimestrali e annuali. In questo caso ci sono 2 possibili scenari:
   (a) tramite l'opzione `select()` si seleziona una sola frequenza e si ricade nel caso 1.
   (b) si tengono le frequenze diverse. In questo caso la variabile `freq` rimane stringa e nel database finale rimangono dati con frequenze diverse.

Nel caso 2.(b) il modo di scrivere le frequenze segue lo schema seguente:[4]
Frequenza giornaliera
Frequenza settimanale

4. Questa operazione è necessaria per poter eseguire il reshape dei dati

Frequenza mensile: Y
Frequenza trimestrale
Frequenza semestrale
Frequenza annuale: Y####

7. Sintassi di eudbimport

```
eudbimport DBNAME, reshapevar(varname) [rawdata(string) outdata(string)
                  download select(string asis) timeselect(string asis)
                  compress(string) decompress(string) /*undocumented*/
                  nodestring /*undocumented*/
                  debug /*undocumented*/ ]
```

dove:

`DBNAME` è il nome del database come riportato sul sito EUROSTAT e va indicato in maiuscolo.

`reshapevar()` è la variabile usata nel reshape e le cui specifiche diventano le nuove variabili.

`rawdata()` è il percorso dove verrà scaricato il file del database se si usa l'opzione `download` o dove trovare il file DBNAME se scaricato manualmente. Il percorso va specificato tra virgolette e con / finale. Se non viene specificato il comando cercherà il file DBANME nella directory di lavoro corrente.

`outdata()` è il percorso dove verrà salvato il file del database. Il percorso va specificato tra virgolette e con / finale. Se non viene specificato il comando salverà il file DBANME nella directory di lavoro corrente.

`download` specifica che DBNAME deve essere scaricato dal sito di EUROSTAT.

`select()` specifica un sottoinsieme di osservazioni di DBNAME che devono essere importate. Si possono usare tutti i comandi di Stata per selezionare osservazioni (`keep`, `drop`...).

`timeselect()` specifica l'intervallo temporale da importare.

`nosave` specifica che il databse importato non venga salvato.

Serve gtools (link), ma se manca viene installato automaticamente

splitvars			**date**	**reshapevar**			
splitvar 1	splitvar 2	splitvar3	timevar	item 1	item 2	item 3	item n