

DEST_PROV

Comando Stata per la conversione dei nomi
delle province italiane



Nicola Tommasi
C.I.D.E.
nicola.tommasi@univr.it

© Copyright by Nicola Tommasi

Without written permission of the author it is forbidden to reproduce or adapt in any form or by any means any part of this publication. Requests for obtaining the right to reproduce or utilize parts of this publication should be addressed to nicola.tommasi@univr.it or nicola.tommasi@gmail.com.

Materiale reperibile su GitHub https://github.com/NicolaTommasi8/dest_com.

Documento redatto in L^AT_EX 2 _{ε} .

Indice

1 Abstract	1
2 Introduzione	1
3 Installazione	1
4 Sintassi	2
4.1 Esempi	3
5 Note Finali	5

1 Abstract

`dest_prov` è un nuovo comando Stata che permette

- di convertire i nomi delle province italiane nei corrispettivi codici numerici Istat. Nella variabile stringa, le province possono essere indicate:
 - per nome
 - per sigla automobilistica
 - per codice NUTS3
 - per codice itter (Istat)
- di creare la variabile regione associata a ciascuna provincia
- di creare le variabili NUTS3, NUTS2 e NUTS1 specificando il riferimento alla versione (dalla 2003 alla 2024)
- di creare le variabili sulla ripartizione geografica (3 o 5 ripartizioni)
- di creare la targa automobilistica
- di creare o convertire la nomenclatura territoriale di Istat (itter)

2 Introduzione

In questo documento mostrerò il funzionamento del comando `dest_prov` con le sue varie opzioni. Per fare questo userò i dati forniti da Istat sul sito [SITUAS](#). Variabili stringa con i nomi delle provincie andrebbero sempre convertiti nei rispettivi codici numerici Istat. Non sempre nei database esiste anche la variabile numerica e allora in questi casi cosa fare? Serve un tedioso lavoro di conversione; ma perché farlo? Sostanzialmente per tre motivi:

- perché le variabili numeriche sono più efficienti delle variabili stringa.
- perché semplifica certe operazioni

Se, per esempio, devo prendere tutte le province venete è più semplice usare il comando `keep if inrange(cod_prov,23,29)` anziché dover elencare tutti i nomi delle province venete.

• perché consente di fare dei `merge` che altrimenti con variabili stringa sarebbero alquanto problematici.
`dest_prov` per la conversione del dato stringa a numerico, si basa sui dati forniti del sito SITUAS con riferimento al 31/12 di ciascun anno. Le informazioni sono state prese seguendo questo procedimento

1. Accedere al sito [SITUAS](#)
2. dal menù in alto cliccare su **Unità territoriali**
3. dal menù di sinistra cliccare su 'Unità amministrative'
4. sotto appare un ulteriore menù su cui cliccare 'Provincia/Uts'
5. Nei risultati di ricerca a destra individuare 'Elenco Province/Uts'
6. cliccare sul simbolo della tabella  (Visualizza report)
7. nel campo 'Dati al:' inserire la data rispetto alla quale si vuole l'elenco dei codici amministrativi, quindi cliccare su 'Applica'

3 Installazione

Dalla command bar di Stata digitate:

```
net install dest_com, from(https://raw.githubusercontent.com/NicolaTommasi8/dest\_prov/master/)
```

Questo comando installerà sul vostro PC quattro files:

- dest_prov.ado
- dest_prov.do
- dest_prov_lab.ado
- dest_prov.sthlp

Il comando richiede Stata dalla versione 13 o successive.

4 Sintassi

La sintassi del comando è la seguente:

```
dest_prov varname [if] [in], [time(var) generate(newvar) onlylab  
ignore tl(str) itter  
gregio(newvar) macro3(newvar) macro5(newvar)  
gnuts3(newvar) gnut2(newvar) gnut1(newvar)  
nuts(int) gitter(newvar) gtarga(newvar) ]
```

dove *varname* è il nome della variabile stringa con i nomi delle province, o i codici delle province se si vuole fare solo il label (opzione *onlylab*):

time(var) è il nome della variabile con l'anno rispetto al quale il nome della provincia va convertito in codice numerico. Se questa variabile non viene specificata, di default si assume che l'anno di riferimento sia quello attuale. Questa opzione è stata introdotta perché le province della Sardegna hanno un codice diverso a partire dal 2025 rispetto agli anni precedenti. Non è obbligatoria per compatibilità con le versioni precedenti del comando.

generate(newvar) crea la variabile numerica *newvar* con il codice Istat della provincia. Se non specificato verrà creata di default la variabile *cod_prov*

onlylab questa opzione indica di fare solo il label di *varname* e quindi serve se la variabile è già numerica

ignore *dest_prov* esegue due controlli. Il primo controlla che tutte le stringhe di *varname* siano riconosciute e quindi convertite in codice numerico, il secondo controllo verifica che tutti i codici numerici generati abbiano una label. Se uno dei due controlli non è verificato, l'esecuzione del comando viene interrotta e nessuna conversione viene eseguita. Specificando l'opzione *ignore*, *dest_prov* viene eseguito anche se uno dei controlli non viene superato.

tl(str) indica come deve essere fatto il label della variabile creata. Le specificazioni possibili sono **full** (default) che assegna il nome per esteso della provincia o **sigla** che assegna la sigla automobilistica

itter indica che la variabile stringa *varname* contiene il codice territoriale Istat della provincia (itter)¹

gregio(newvar) genera una variabile con i codici Istat delle regioni

macro3(newvar) genera una variabile con l'appartenenza a 3 macro regioni (Nord, Centro, Sud e Isole)

macro5(newvar) genera una variabile con l'appartenenza a 5 macro regioni (Nord-Ovest, Nord-Est, Centro, Sud, Isole)

gnuts3(newvar) genera una variabile (stringa) con i codici NUTS3

gnuts2(newvar) genera una variabile (stringa) con i codici NUTS2

1. Codice molto simile, ma non uguale al NUTS3

gnuts1(newvar) genera una variabile (stringa) con i codici NUTS1

nuts(int) le variabili gnut3(varname), gnut2(varname) e gnut1(varname) vengono create con la codifica nuts dell'anno indicato. I valori consentiti sono 2003, 2006, 2010, 2013, 2016, 2021 e 2024.

Se non specificato si fa riferimento alle versione 2024

gitter(newvar) genera una variabile (stringa) con i codici itter di Istat

gtarga(newvar) genera una variabile (stringa) con le sigle automobilistiche delle province

4.1 Esempi

Alcuni esempi commentati di utilizzo di `dest_prov`. Premesso che al momento l'opzione `time()` non è obbligatoria (ma forse lo sarà in futuro), negli esempi seguenti si preferisce sempre creare questa variabile.

```
. import excel using "data/Codici-statistici-e-denominazioni-al-31_12_2024.xlsx", firstrow ca
> se(lower) clear
(26 vars, 7,896 obs)

. rename denominazionedellunitàterritorio provincia

. destring codiceprovinciastorico1 codiceregione, replace
codiceprovinciastorico1: all characters numeric; replaced as int
codiceregione: all characters numeric; replaced as byte

. gen anno=2024

. dest_prov provincia, gen(cod_prov) time(anno) gregio(cod_reg) macro3(macro3) macro5(macro5)
> ///
> gnut3(nuts3) gnut2(nuts2) gnut1(nuts1) nuts(2021)
```

In questo primo esempio, dopo aver importato i dati, si fanno alcune operazioni di pulitura dei dati (rename e conversione in numeriche di variabili che servono per il controllo successivo). Poi si applica il comando per creare la variabile numerica `cod_prov` a partire dalla variabile stringa `provincia` con riferimento all'anno indicato in `anno` (2024). Inoltre si creano le variabili `cod_reg` con i codici regionali, `macro3` e `macro5` con i codici ripartizionali a 3 e a 5 macroregioni e, infine, le variabili stringa dei codici Nuts 3, 2 e 1 riferiti alla revisione Nuts del 2021.

```
. assert cod_prov==codiceprovinciastorico1

. assert cod_reg == codiceregione

. assert macro5 == codiceripartizionegeografica

. assert nuts3 == codicenuts32021

. assert nuts2 == codicenuts220213

. assert nuts1== codicenuts12021
```

Dato che queste variabili sono già presenti nel dataset, con una serie di `assert` si verifica che le variabili create siano corrispondenti a quelle già presenti.

```
. drop cod_prov cod_reg-nuts1

. dest_prov provincia, gen(cod_prov) time(anno) gnut3(nuts3) gnut2(nuts2) gnut1(nuts1) ///
> nuts(2024) gitter(itter) gtarga(targa)
```

```

. assert nuts3 == codicenuts32024

. assert nuts2 == codicenuts220243

. assert nuts1== codicenuts12024

. assert targa==siglaautomobilistica

```

Sempre sugli stessi dati si riapplica il comando per creare, questa volta, i Nuts 3, 2 e 1 riferiti alla revisione Nuts del 2024, variabile con i codici territoriali dell'Istat (`itter`) e la variabile con la sigla automobilistica della provincia (`targa`)

Ora si impostano degli altri dati riferiti al 2023 e si fanno gli stessi controlli

```

. import delimited "data/Province31-12-2023.csv", clear
(encoding automatically selected: UTF-8)
(12 vars, 107 obs)

. gen anno=2023

. dest_prov provinciauts, gen(cod_prov) time(anno) gregio(cod_reg) gtarga(targa) ///
> gnuts3(nuts3) nuts(2021)

. assert cod_prov==codiceprovinciastorico

. assert cod_reg == codiceregione

. assert targa==siglaautomobilistica

. assert nuts3==codicenuts32021

```

Il codice della province può includere anche quelle delle unità territoriali speciali

```

. assert cod_prov==codiceprovinciauts, rc0
14 contradictions in 107 observations
assertion is false

. list provincia cod_prov codiceprovinciauts if cod_prov!=codiceprovinciauts, sep(0) noobs no
> lab

+-----+
| provinciauts   cod_prov   codice~s |
|-----|
|       Torino      1      201 |
|       Milano     15      215 |
|       Venezia    27      227 |
|       Genova     10      210 |
|       Bologna    37      237 |
|       Firenze    48      248 |
|       Roma       58      258 |
|       Napoli     63      263 |
|       Bari       72      272 |
| Reggio Calabria 80      280 |
|       Palermo    82      282 |
|       Messina    83      283 |
|       Catania    87      287 |
|       Cagliari   92      292 |
+-----+

```

In questo caso si vede che `dest_com` assegna il codice provinciale, non quello delle città metropolitane².

2. In pratica, il codice delle città metropolitane è quello della relativa provincia al 2004 a cui viene sommato il valore 200.

5 Note Finali

La variabile `varname` con i nomi delle province non può essere nel formato strL, ma deve essere nel formato str#, altrimenti il comando termina con il seguente errore

```
. dest_prov nuts_name, gen(cod_prov)
key variable _000004 is strL
    The key variables (on which observations are matched) can be str#, but they cannot be strLs.
r(106);

end of do-file

r(106);
```

In questi casi si può risolvere con il comando `compress` o con il comando `recast str# varname, force` dove str# sarà il numero di digits della variabile `varname`. Ecco un esempio:

```
. desc

Contains data
Observations:          137
Variables:             4
-----
Variable      Storage   Display   Value
      name        type     format   label      Variable label
-----
cntr_code     str2      %9s       CNTR_CODE
nuts_id       str5      %9s       NUTS_ID
name_latn    str70     %70s      NAME_LATN
nuts_name    strL      %80s      NUTS_NAME
-----
```

In prima battuta si può provare con `compress`, se il problema non si risolve, la variabile `nuts_name` può essere convertita così:

```
. recast str70 nuts_name, force

. desc

Contains data
Observations:          137
Variables:             4
-----
Variable      Storage   Display   Value
      name        type     format   label      Variable label
-----
cntr_code     str2      %9s       CNTR_CODE
nuts_id       str5      %9s       NUTS_ID
name_latn    str70     %70s      NAME_LATN
nuts_name    str70     %70s      NUTS_NAME
-----
```

Riferimenti bibliografici

[1] Codici delle unità amministrative

[2] SITUAS