

# Uso di dest\_com

Nicola Tommasi

12 dicembre 2016

## Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
<b>2</b>	<b>Installazione</b>	<b>2</b>
<b>3</b>	<b>Duplicazioni nei nomi dei comuni</b>	<b>2</b>
<b>4</b>	<b>I Dati</b>	<b>3</b>
<b>5</b>	<b>Uso di dest_com</b>	<b>4</b>
<b>6</b>	<b>Esempio con Dati non in Formato utf-8</b>	<b>6</b>
<b>7</b>	<b>Modifica di dest_com.do e dest_com_lab.do</b>	<b>7</b>
<b>8</b>	<b>Variazioni Amministrative</b>	<b>7</b>
<b>9</b>	<b>Help del comando</b>	<b>11</b>

## Sommario

`dest_com` Converte una variabile stringa con i nomi dei comuni italiani in una variabile numerica con i corrispettivi codici ISTAT. Inoltre può creare anche la variabile numerica della regione e le variabili stringa dei codici NUTS3, NUTS2 e NUTS1 (versione 2010).

# 1 Introduzione

In questo documento mostrerò il funzionamento del comando `dest_com` con le sue varie opzioni. Per fare questo userò i dati più recenti forniti da ISTAT a questo [indirizzo](#). I dati sono presenti in fondo alla pagina, io ho usato questi. Il presente documento è stato redatto con il pacchetto `texdoc` di Ben Jann; potete trovarlo [qui](#). Variabili stringa con i nomi dei comuni italiani (o 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 (prendere tutti i comuni di Verona: `keep if inrange(cod_com,23001,23999)` ).
- perché consente di fare dei merge che altrimenti con variabili stringa sarebbero alquanto problematici.

# 2 Installazione

Dalla command bar di Stata digitate:

```
. net from https://sites.google.com/site/nicolatommasi/stata
```

---

```
https://sites.google.com/site/nicolatommasi/stata/  
Alcuni miei programmi
```

---

```
PACKAGES you could -net describe-:  
  dest_prov      Comando per convertire i nomi delle provincie italiane nel rispettivo codice I  
> STAT.  
  dest_com       Comando per convertire i nomi dei comuni italiani nel rispettivo codice ISTAT.  
  uniout         Comando per analizzare la presenza di valori outliers nel caso univariato.  
  lebp           ... coming soon.  
  cquads         ... coming soon.
```

---

quindi cliccate su `dest_com` e alla pagina successiva su ([click here to install](#)). Questa operazione installerà sul vostro PC quattro files:

- `dest_com.ado`
- `dest_com.do`
- `dest_com_lab.ado`
- `dest_com.sthlp`

Mostrerò più avanti in particolare la funzione dei files `dest_com.do` e `dest_com_lab.do` che in caso di necessità potrebbero anche essere modificati per esigenze personali. Il comando richiede Stata dalla versione 13 o successive, ma è caldamente consigliata almeno la versione 14 dato che viene usata la funzione `ustrfrom()` per consentire l'uso di dataset in codifica non UTF-8. In caso di utilizzo di Stata 13 alcuni comuni potrebbero non essere riconosciuti e per questi si renderà necessaria una codifica manuale.

# 3 Duplicazioni nei nomi dei comuni

Esistono comuni con uguale denominazione presenti in provincie diverse. In questi casi per assegnare il codice numerico corretto è indispensabile avere anche l'informazione relativa alla provincia da specificare

nell'opzione `mkc(varname)`. Se non è presente l'informazione relativa alla provincia, viene assegnato un valore di default secondo lo schema della tabella seguente:

Comune	Provincia 1	Provincia 2	Default
<b>Brione</b>	Brescia (17030)	Trento (22028)	17030
<b>Calliano</b>	Asti (5014)	Trento (22035)	5014
<b>Castro</b>	Bergamo (16065)	Lecce (75096)	16065
<b>Livio</b>	Como (13130)	Trento (22106)	13130
<b>Peglio</b>	Como (13178)	Pesaro Urbino (41041)	13178
<b>Samone</b>	Torino (1235)	Trento (22165)	1235
<b>Valverde</b>	Pavia (18170)	Catania (87052)	18170
<b>San Teodoro</b>	Messina (83090)	Olbia Tempio (104023)	83090

## 4 I Dati

Ora carico i dati relativi ai nomi e ai codici dei comuni italiani presenti nel file scaricato dai ISTAT. Ho scelto di usare nomi maiuscoli per queste variabili in modo da distinguerle da quelle che creerò in minuscolo con il comando `dest_com`.

```
. import excel using Elenco-codici-statistici-e-denominazioni-al-01_07_2016.xls, ///
> firstrow case(upper) clear
```

Ecco che variabili sono state caricate

```
. desc, fullnames
Contains data
  obs:      8,016
  vars:      25
  size:    1,683,360
```

---

variable name	storage type	display format	value label	variable label
CODICEREGIONE	str2	%9s		Codice Regione
CODICECITTÀMETROPOLITANA	str3	%9s		Codice Città Metropolitana
CODICEPROVINCIA1	str3	%9s		Codice Provincia (1)
PROGRESSIVODELCOMUNE2	str3	%9s		Progressivo del Comune (2)
CODICECOMUNEFORMATOALFANUMERI	str6	%9s		Codice Comune formato alfanumerico
DENOMINAZIONEINITALIANO	str34	%34s		Denominazione in italiano
DENOMINAZIONEINTEDESCO	str36	%36s		Denominazione in tedesco
CODICERIPARTIZIONEGEOGRAFICA	byte	%10.0g		Codice Ripartizione Geografica
RIPARTIZIONEGEOGRAFICA	str10	%10s		Ripartizione geografica
DENOMINAZIONEREGIONE	str29	%29s		Denominazione regione
DENOMINAZIONECITTÀMETROPOLITANA	str7	%9s		Denominazione Città metropolitana
DENOMINAZIONEPROVINCIA	str29	%29s		Denominazione provincia
FLAGCOMUNECAPOLUOGODIPROVINC	byte	%10.0g		Flag Comune capoluogo di provincia
SIGLAUTOMOBILISTICA	str2	%9s		Sigla automobilistica
CODICECOMUNEFORMATONUMERICO	long	%10.0g		Codice Comune formato numerico
CODICECOMUNENUMERICOCON107P				

	long	%10.0g	Codice Comune numerico con 107 province (dal 2006 al 2009)
CODICECOMUNENUMERICOCN103P	long	%10.0g	Codice Comune numerico con 103 province (dal 1995 al 2005)
CODICECATASTALEDELCOMUNE	str4	%9s	Codice Catastale del comune
POPOLAZIONELEGALE201109102	long	%10.0g	Popolazione legale 2011 (09/10/2011)
CODICENUTS12010	str3	%9s	Codice NUTS1 2010
CODICENUTS220103	str4	%9s	Codice NUTS2 2010 (3)
CODICENUTS32010	str5	%9s	Codice NUTS3 2010
CODICENUTS12006	str3	%9s	Codice NUTS1 2006
CODICENUTS220063	str4	%9s	Codice NUTS2 2006 (3)
CODICENUTS32006	str5	%9s	Codice NUTS3 2006

Sorted by:

Note: Dataset has changed since last saved.

Ora cancello quelle superflue

```
. drop CODICECITTÀMETROPOLITANA PROGRESSIVODELCOMUNE2 CODICECOMUNEFORMATOALFANUMERI ///
> CODICERIPARTIZIONE GEOGRAFICA RIPARTIZIONE GEOGRAFICA DENOMINAZIONE REGIONE ///
> DENOMINAZIONE CITTÀ METROPOLITANA FLAGCOMUNECAPOLUOGODIPROVINC ///
> CODICECATASTALEDELCOMUNE POPOLAZIONELEGALE201109102 ///
> CODICENUTS12006 CODICENUTS220063 CODICENUTS32006
```

e rinomino quelle di interesse

```
. rename (CODICEREGIONE CODICEPROVINCIA1 DENOMINAZIONEINITALIANO DENOMINAZIONEINTEDESCO ///
> DENOMINAZIONEPROVINCIA SIGLAUTOMOBILISTICA CODICECOMUNEFORMATONUMERICO ///
> CODICECOMUNENUMERICOCN107P CODICECOMUNENUMERICOCN103P CODICENUTS12010 ///
> CODICENUTS220103 CODICENUTS32010) ///
> (COD_REGIO COD_PROV DENOM_ITA DENOM_TED DENOM_PROV TARGA COD_COM ///
> COD_COM107 COD_COM103 NUTS1 NUTS2 NUTS3)
```

Ora pulisco delle osservazioni missing caricate dal file excel e converto in numerica la variabile COD\_PROV che viene letta come stringa

```
. drop if COD_PROV=="
(18 observations deleted)
. destring COD_PROV COD_REGIO, replace
COD_PROV: all characters numeric; replaced as int
COD_REGIO: all characters numeric; replaced as byte
```

## 5 Uso di dest\_com

Adesso supponiamo di avere la variabile DENOM\_ITA con la denominazione dei comuni italiani e di volerla trasformare in una variabile numerica con i codici numerici ufficiali ISTAT. L'opzione `time(varname)` è obbligatoria per cui prima creo la variabile `anno` con valore 2016

```
. gen anno=2016
. dest_com DENOM_ITA, time(anno) mkc(COD_PROV) gen(cod_com)
L'attribuzione del codice numerico sembra andata a buon fine, ma non ci può essere la certezza al 10
> 0%
Usare i dati con attenzione!
```

a questo punto verifichiamo che la variabile `cod_com` corrisponda alla variabile `COD_COM`

```
. assert cod_com==COD_COM
```

Ok, le due variabili coincidono. Vediamo cosa succede se cambiamo l'anno di riferimento al 2014:

```
. replace anno=2014
(7,998 real changes made)
. dest_com DENOM_ITA, time(anno) mkc(COD_PROV) gen(cod_com2014)
L'attribuzione del codice numerico sembra andata a buon fine, ma non ci può essere la certezza al 10
> 0%
Usare i dati con attenzione!
. compare cod_com cod_com2014
```

	count	minimum	difference average	maximum
cod_com=cod_-2014	7996			
cod_com>cod_-2014	2	56	65.5	75
jointly defined	7998	0	.0163791	75
total	7998			

ci sono 2 comuni con codice diverso:

```
. list DENOM_ITA TARGA COD_PROV cod_com cod_com2 if cod_com!=cod_com2, nolab noobs
```

DENOM_ITA	TARGA	COD_PROV	cod_com	cod-2014
Lessona	BI	96	96085	96029
Campiglia Cervo	BI	96	96086	96011

```
. drop cod_com2 cod_com
. replace anno=2016
(7,998 real changes made)
```

Adesso aggiungiamo le opzioni per creare il codice numerico delle provincie (**gprov**), delle regioni (**gregio**) e i codici alfanumerici nuts3 (**gnuts3**), nuts2 (**gnuts2**) e nuts1 (**gnuts1**).

```
. dest_com DENOM_ITA, time(anno) mkc(COD_PROV) gen(cod_com) gprov(cod_prov) gregio(cod_regio) ///
> gnuts3(nuts3) gnuts2(nuts2) gnuts1(nuts1)
L'attribuzione del codice numerico sembra andata a buon fine, ma non ci può essere la certezza al 10
> 0%
Usare i dati con attenzione!
```

e come fatto in precedenza verificiamo la correttezza della conversione

```
. compare cod_prov COD_PROV
```

	count	minimum	difference average	maximum
cod_prov=COD_PROV	7998			
jointly defined	7998	0	0	0
total	7998			

```
. compare cod_regio COD_REGIO
```

	count	minimum	difference average	maximum
cod_regio=COD_REGIO	7998			
jointly defined	7998	0	0	0
total	7998			

```
. compare nuts3 NUTS3
```

	count
nuts3=NUTS3	7998
jointly defined	7998

```

total                7998
. compare nuts2 NUTS2
count
nuts2=NUTS2          7998
jointly defined      7998
total                7998
. compare nuts1 NUTS1
count
nuts1=NUTS1          7998
jointly defined      7998
total                7998

```

## 6 Esempio con Dati non in Formato utf-8

Se avete dei dati con codifica diversa da utf-8, può succedere questo:

```

. clear
. unicode analyze test_noutf.dta
File summary (before starting):
  1 file(s) specified
  1 file(s) to be examined ...
File test_noutf.dta (Stata dataset)
  3 str# variables need translation
-----
File needs translation. Use unicode translate on this file.
File test_noutf.dta needs translation
File summary:
  1 file(s) need translation
.
. use test_noutf.dta, clear
. desc, short
Contains data from test_noutf.dta
obs:      8,054
vars:      3
size:     885,940
Sorted by:
3 Dec 2016 16:53

```

Infatti le lettere accentate non vengono riconosciute e rappresentate con dei caratteri "strani"

```

. list Comune in 7
+-----+
| Comune |
+-----+
7.  CIRI[U+FFFD]
+-----+

. list Comune in 34
+-----+
| Comune |
+-----+
34. MONDOV[U+FFFD]
+-----+

```

Se usate Stata 14 (o successivo) non avrete problemi

```

. gen anno=2014

```

```

. dest_prov Provincia
.
. dest_com Comune, time(anno) mkc(cod_prov) gen(cod_com) ignore
L'attribuzione del codice numerico sembra andata a buon fine, ma non ci può essere la certezza al 10
> 0%
Usare i dati con attenzione!

```

Se usate Stata 13 questi comuni non verranno codificati con i ripetitivi codici ISTAT e dovrete procedere ad una ricodifica manuale del tipo

```

. replace cod_com=1086 if strmatch(Comune,"CIRI*") & cod_com==.
(0 real changes made)
. replace cod_com=4130 if strmatch(Comune,"MONDOV*") & cod_com==.
(0 real changes made)

```

## 7 Modifica di dest\_com.do e dest\_com\_lab.do

dest\_com.do contiene le regole per assegnare alla stringa del nome di ciascun comune il corrispettivo codice numerico, dest\_com\_lab.do invece il label dei valori numerici. Vediamo dove sono posizionati questi 2 files:

```

. sysdir
  STATA: C:\Program Files (x86)\Stata14\
  BASE: C:\Program Files (x86)\Stata14\ado\base\
  SITE: C:\Program Files (x86)\Stata14\ado\site\
  PLUS: c:\ado\plus\
  PERSONAL: c:\ado\personal\
  OLDPLACE: c:\ado\

```

li troverete nella cartella d contenuta all'interno della cartella indicata alla riga PLUS: del precedente comando. Quindi nel mio caso nella cartella c:\ado\plus\d.

## 8 Variazioni Amministrative

Il comando dest\_com tiene conto delle seguenti variazioni amministrative:

### 2016

Provincia	Nuovo Comune	Comuni Soppresi	Motivo
Bologna	Alto Reno Terme	Granaglione Porretta Terme	Fusione
Parma	Polesine Zibello	Polesine Parmense Zibello	Fusione
Reggio Emilia	Ventasso	Busana Collagna Ligonchio Ramiseto	Fusione
Rimini	Montescudo-Monte Colombo	Monte Colombo Montescudo	Fusione

Brescia	Bienno	Bienno Prestine	Incorporazione
Pavia	Corteolona e Genzone	Corteolona Genzone	Fusione
Biella	Campiglia Cervo	Campiglia Cervo Quittengo San Paolo Cervo	Fusione
Biella	Lessona	Crosa Lessona	Fusione
Verbano-Cusio-Ossola	Borgomezzavalle	Seppiana Viganella	Fusione
Trento	Altavalle	Faver Grauno Grumes Valda	Fusione
Trento	Altopiano della Vigolana	Bosentino Centa San Nicolò Vattaro Vigolo Vattaro	Fusione
Trento	Amblar-Don	Amblar Don	Fusione
Trento	Borgo Chiese	Brione Cimego Condino	Fusione
Trento	Borgo Lares	Bolbeno Zuclo	Fusione
Trento	Castel Ivano	Spera Strigno Villa Agnedo Ivano-Fracena	Fusione
Trento	Cembra Lisignago	Cembra Lisignago	Fusione
Trento	Contà	Cunevo Flavon Terres	Fusione
Trento	Dimaro Folgarida	Dimaro Monclassico	Fusione
Trento	Madruzzo	Calavino Lasino	Fusione
Trento	Pieve di Bono-Prezzo	Pieve di Bono	Fusione



		Prezzo	
Trento	Porte di Rendena	Darè Vigo Rendena Villa Rendena	Fusione
Trento	Premiero San Martino di Castrozza	Fiera di Primiero Siror Tonadico Transacqua	Fusione
Trento	Sella Giudicarie	Bondo Breguzzo Lardaro Roncone	Fusione
Trento	Tre Ville	Montagne Preore Ragoli	Fusione
Trento	Vallelaghi	Padergnone Terlago Vezzano	Fusione
Trento	Ville d'Anaunia	Nanno Tassullo Tuenno	Fusione
Belluno	Alpago	Farra d'Alpago Pieve d'Alpago Puos d'Alpago	Fusione
Belluno	Val di Zoldo	forno di Zoldo Zoldo Alto	Fusione

Il comune di Capaccio ha modificato il nome in Capaccio Paestum

## 2015

Provincia	Nuovo Comune	Comuni Soppresi	Motivo
Pordenone	Valvasone Arzene	Arzene Valvasone	Fusione
Lecco	La Valletta Brianza	Perego Rovagnate	Fusione
Sondrio	Gordona	Gordona Menarola	Incorporazione
Lucca	Silano Giuncugnano	Giuncugnano Silano	Fusione

Trento	Predaia	Coredo Smarano Taio Tres Vervò	Fusione
--------	---------	--	---------

Lucca	San Lorenzo Dorsino	Dorsino San Lorenzo in Banale	Fusione
-------	---------------------	----------------------------------	---------

Lucca	Valdaone	Bersone Daone Praso	Fusione
-------	----------	---------------------------	---------

Il comune di Castellinaldo ha modificato il nome in Castellinaldo d'Alba

## 2014

<b>Provincia</b>	<b>Nuovo Comune</b>	<b>Comuni Soppresi</b>	<b>Motivo</b>
Bologna	Valsamoggia	Bazzano Castello di Serravalle Crespellano Monteveglia Savigno	Fusione
Ferrara	Fiscaglia	Massa Fiscaglia Migliarino Migliaro	Fusione
Parma	Sissa Trecasali	Sissa Trecasali	Fusione
Rimini	Poggio Torriana	Poggio Berni Torriana	Fusione
Udine	Rivignano Teor	Rivignano Teor	Fusione
Bergamo	Sant'Omobono Terme	Sant'Omobono Terme Sant'Omobono Terme	Fusione
Bergamo	Val Brembilla	Brembilla Gerosa	Fusione
Como	Bellagio	Bellagio Civenna	Fusione
Como	Colverde	Drezzo Gironico Parè	Fusione
Como	Tremezzina	Lenno Mezzegra	Fusione

		Ossuccio Tremezzo	
Lecco	Verderio	Verderio Inferiore Verderio Superiore	Fusione
Mantova	Borgo Virgilio	Borgoforte Virgilio	Fusione
Pavia	Cornale e Bastida	Bastida de' Dossi Cornale	Fusione
Varese	Maccagno con Pino e Vaddasca	Maccagno Pino sulla Sponda del Lago Maggiore Vaddasca	Fusione
Ancona	Trecastelli	Castel Colonna Monterado Ripe	Fusione
Pesaro-Urbino	Vallefoglia	Colbordolo Sant'Angelo in Lizzola	Fusione
Arezzo	Castelfranco Piandiscò	Castelfranco di Sopra Pian di Sco	Fusione
Arezzo	Pratovecchio Stia	Pratovecchio Stia	Fusione
Firenze	Figline e Incisa Valdarno	Figline Valdarno Incisa in Val d'Arno	Fusione
Firenze	Scarperia e San Piero	San Piero a Sieve Scarperia	Fusione
Lucca	Fabbriche di Vergemoli	Fabbriche di Vallico Vergemoli	Fusione
Pisa	Casciana Terme Lari	Casciana Terme Lari	Fusione
Pisa	Crespina Lorenzana	Crespina Lorenzana	Fusione
Belluno	Longarone	Castellavazzo Longarone	Fusione

Il comune di Brenzone cambia la sua denominazione in Brenzone sul Garda.

## 9 Help del comando

```
. type C:\ado\plus\d\dest_com.sthlp
help dest_com
```

---

## Version

3.1 aug2016

## Description

`dest_com` converte i nomi dei comuni italiani nel rispettivo codice numerico ISTAT.

## Syntax

```
dest_com varname [if] [in] [, generate(varname) mkc(varname) ignore onlylab time(varname)
gprov(varname) gregio(name) gnuts3(name) gnuts2(name) gnuts1(name) ]
```

dove `varname` è la variabili stringa con i nomi dei comuni.

options	Description
<code>generate(varname)</code>	crea la variabile numerica <code>varname</code> con il codice ISTAT del comune. Se non specificato verrà creata di default la variabile <code>cod_com</code> .
<code>mkc(varname)</code>	variabile numerica con i codici ISTAT delle provincie. Serve per ricolmare i casi di omonimia (vedi Remarks).
<code>ignore</code>	<code>dest_com</code> esegue due controlli. Il primo controlla che tutte le stringhe di <code>varname</code> 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 <code>ignore</code> , <code>dest_com</code> viene eseguito anche se uno dei controlli non viene superato.
<code>onlylab</code>	questa opzione serve se la variabile è già numerica e l'operazione da compiere è solo il label dei suoi valori.
<code>time(varname)</code>	alcuni comuni a partire dal 2016 hanno mantenuto lo stesso nome ma hanno cambiato il codice <code>istat</code> a seguito di unione con un altro/i comune/i. Per assegnare il codice ISTAT corretto è necessario specificare la variabile con l'anno a cui si riferiscono i dati. Vedi Remarks per ulteriori chiarimenti. Se questa variabile non viene specificata, di default si assume che l'anno di riferimento sia il 2015.
<code>gprov(varname)</code>	genera una variabile con i codici ISTAT delle province.
<code>gregio(varname)</code>	genera una variabile con i codici ISTAT delle regioni.
<code>gnuts3(varname)</code>	genera una variabile (stringa) con i codici NUTS3 2010.
<code>gnuts2(varname)</code>	genera una variabile (stringa) con i codici NUTS2 2010.
<code>gnuts1(varname)</code>	genera una variabile (stringa) con i codici NUTS1 2010.

## Examples

```
codifica la variabile stringa comune nella variabile numerica com_num:
. dest_com comune, gen(com_num)
```

```
come la precedente, ma i casi di omonimia vengono risolti con l'ausilio della variabile
prov_num:
. dest_com comune, gen(com_num) mkc(prov_num)
```

## Saved results

Nessun risultato salvato

## Remarks

Esistono comuni con uguale denominazione presenti in provincie diverse. Per risolvere questi casi deve esistere una variabile con il codice della provincia e questa deve essere indicata nell'opzione `mkc(varname)`. Se `mkc(varname)` non viene specificata viene assegnato il codice di default.

Questi sono i casi di omonimia:

			default
Brione	Brescia (17030)	Trento (22028)	17030
Calliano	Asti (5014)	Trento (22035)	5014
Castro	Bergamo (16065)	Lecce (75096)	16065
Livio	Como (13130)	Trento (22106)	13130
Peglio	Como (13178)	Pesaro Urbino (41041)	13178
Samone	Torino (1235)	Trento (22165)	1235
Valverde	Pavia (18170)	Catania (87052)	18170
San Teodoro	Messina (83090)	Olbia Tempio (104023)	83090

A partire dal 2016 alcuni comuni hanno mantenuto lo stesso nome ma hanno cambiato il codice `istat` a seguito di unione con un altro/i comune/i. Questi sono i casi: Campiglia Cervo ha codice 96011 fino al 2015, assume codice 96086 a partire dal 2016. Lessona ha codice 96029 fino al 2015, assume codice 96085 a partire dal 2016.

References

ISTAT Pagina di riferimento

Author

Nicola Tommasi  
Centro Interdipartimentale di Documentazione Economica (C.I.D.E.)  
University of Verona, Italy  
nicola.tommasi@univr.it

Help: dest\_prov