

Geolocalizzazione in Stata

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

27 marzo 2023



1. Principali comandi Stata per la geolocalizzazione

1.1. georoute

Calcola la distanza stradale e il tempo di percorrenza tra 2 punti basandosi sulle api di HERE (<https://developer.here.com>). Inoltre, può fornire come output anche la latitudine e la longitudine di un indirizzo. Oltre al comando stesso, devono essere installati anche i comandi `insheetjson` e `libjson`. Dovrebbe avere un massimale di richieste piuttosto alto (250000 al mese)¹. Dal sito è possibile iscriversi al servizio ed avere un id e un code da inserire poi nel comando. Quelli riportati negli esempi successivi sono fittizi.

Questo comando calcola latitudine e longitudine dato un indirizzo:

```
georoute, herekey(<codice\alfanumerico\43\digits>) ///
  startad(indirizzo) endad(indirizzo\_fittizio) coordinates(p1 p2) replace timer pause
```

Questo calcola distanza stradale e tempo di percorrenza tra 2 punti

```
georoute, herekey(<codice\alfanumerico\43\digits>) ///
  startxy(p1_x p1_y) endxy(p2_x p2_y) km distance(road_dist) replace timer pause
```

1.2. geodist

Calcola la distanza lineare tra 2 punti dati latitudine e longitudine.

```
geodist p1_x p1_y p2_x p2_y, generate(linear_distance)
```

1.3. geocodeopen

Restituisce latitudine e longitudine usando le api di MapQuest (<https://developer.mapquest.com>). Ha un massimale di 15000 richieste per mese

```
geocodeopen, key("<key_code>") fulladdr(full_address) replace
```

Non pare molto preciso, però è abbastanza veloce.

1.4. geocode

Si basa sulle api di Google però non riesco a farlo funzionare. Il comando blocca Stata apparentemente senza fornire nessuna richiesta a Google.

```
geocode, fulladdr(indirizzo)
```

1. Tutti i comandi possono essere installati usando il comando `ssc inst <nome_comando>` nella command bar di Stata. Per visualizzare l'help invece digitare `help <nome_comando>`

2. Ottimizzazione del processo

Le richieste inviate ai server per avere la localizzazione geografica di un indirizzo o per calcolare la distanza stradale tra 2 punti sono in primo luogo esose in termini di tempi richiesti ed inoltre vanno a diminuire il plafond di richieste permesse su base mensile. La metodologia che viene di seguito descritta permette di ridurre di oltre metà il numero di richieste e di conseguenza anche i tempi. Questa metodologia vale solo nel caso si debba calcolare la matrice delle distanze tra n punti. Facciamo il caso che si debba costruire la matrice delle distanze tra dieci punti (luoghi), dei quali abbiamo già ottenuto le coordinate latitudine e longitudine. Le distanze da calcolare sarebbero $10^2 = 100$.

da \ a	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
P1	$D_{1,1}$									
P2		$D_{2,2}$								
P3			$D_{3,3}$							
P4				$D_{4,4}$						
P5					$D_{5,5}$					
P6						$D_{6,6}$				
P7							$D_{7,7}$			
P8								$D_{8,8}$		
P9									$D_{9,9}$	
P10										$D_{10,10}$

Però possiamo ridurre tale numero sapendo che:

- Le distanze $D_{i,j}$ con $i = j$ sono tutte nulle per cui è inutile sprecare una chiamata. Questo permette di risparmiare n chiamate
- Dato che la matrice delle distanze è simmetrica ($D_{i,j} = D_{j,i}$), è sufficiente popolare solo la matrice triangolare inferiore (quella in verde). La matrice triangolare superiore (quella in giallo) si ricostruisce in un secondo momento a partire da quella inferiore.

Questo permette di ridurre il numero delle chiamate da n^2 a $(n^2 - n)/2$. Nel nostro esempio da 100 a 45 o nel caso di $n=500$ si passerebbe da 250000 chiamate a 124750.

Di seguito un esempio di costruzione della matrice delle distanze. Nel dataset sono presenti 3 variabili: il nome della località, la sua latitudine ($p1_x$) e la sua longitudine ($p1_y$). Viene creata la variabile id che numera in senso crescente ciascuna osservazione e poi viene aperto un ciclo `forvalues` che opera su ciascuna osservazione. Dalla osservazione i -esima prende il valore $p1_x$ e $p1_y$ e costruisce due variabili (costanti) con tali valori ($xc_id'i$ e $yc_id'i$). Poi per tutte le osservazioni successive alla i -esima viene calcolata la distanza stradale in chilometri con il comando `georoute`. In questa maniera vengono calcolati i soli valori della matrice triangolare inferiore. Le distanze lineari (`geodist`) invece vengono calcolate tutte in quanto non servono chiamate api ai servers. Con i due `replace` vengono poste a zero le distanze sulla diagonale principale. Talvolta possono presentarsi dei problemi per cui le chiamate api al server potrebbero interrompersi. Per non perdere i dati già acquisiti ogni cinquanta variabili viene salvato il database. Alla fine, ci sono due cicli `forvalues` annidati che prendono i dati dalla matrice diagonale inferiore e li ricopiano nei corrispettivi della matrice diagonale superiore.

```

gen id = _n
summarize id
forvalues i=1/`r(max)' {
    local xc=p1_x in `i'
    local yc=p1_y in `i'
    gen xc_id`i' = `xc'
    gen yc_id`i' = `yc'

    georoute if id>`i', hereid(<hereid>) herecode(<herecode>) ///
        startxy(p1_x p1_y) endxy(xc_id`i' yc_id`i') ///
        km distance(road_d`i') replace timer pause

    geodist p1_x p1_y xc_id`i' yc_id`i', generate(linear_d`i')
    replace linear_d`i' = 0 in `i'
    replace road_d`i' = 0 in `i'
    drop ?c_id`i' travel_time georoute_diagnostic
    assert linear_d`i' <= road_d`i', rc0 /*obviously*/

    if mod(`i',50)==1 {
        compress
        save data/step2, replace
    }
}

summ id
local max=`r(max)'
forvalues r=1(1)`max' {
    forvalues c=2(1)`max' {
        summarize road_d`r' if id==`c', meanonly
        replace road_d`c'=`r(mean)' if id==`r'
    }
}

```

L'operazione di portare i dati dalla matrice diagonale inferiore alla matrice diagonale superiore si può fare anche utilizzando Mata. Nello script che segue simulo una matrice delle distanze sulla falsariga della tabella precedente

```

use temp, clear
list, sep(0) noobs

```

```

+-----+
| place  p1  p2  p3  p4  p5  p6  p7  p8  p9  p10 |
+-----+
|      1   0   .   .   .   .   .   .   .   .   . |
|      2   1   0   .   .   .   .   .   .   .   . |
|      3   2  10   0   .   .   .   .   .   .   . |
|      4   3  11  18   0   .   .   .   .   .   . |
|      5   4  12  19  25   0   .   .   .   .   . |
|      6   5  13  20  26  31   0   .   .   .   . |
|      7   6  14  21  27  32  36   0   .   .   . |
|      8   7  15  22  28  33  37  40   0   .   . |
|      9   8  16  23  29  34  38  41  43   0   . |
|     10   9  17  24  30  35  39  42  44  45   0 |
+-----+

```

```

**METHOD #1

```

```

summ place
local max=`r(max)'
forvalues r=1(1)`max' {
  forvalues c=2(1)`max' {
    qui summarize p`r' if place==`c', meanonly
    qui replace p`c'=`r(mean)' if place==`r'
  }
}
list, sep(0) noobs

```

place	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
1	0	1	2	3	4	5	6	7	8	9
2	1	0	10	11	12	13	14	15	16	17
3	2	10	0	18	19	20	21	22	23	24
4	3	11	18	0	25	26	27	28	29	30
5	4	12	19	25	0	31	32	33	34	35
6	5	13	20	26	31	0	36	37	38	39
7	6	14	21	27	32	36	0	40	41	42
8	7	15	22	28	33	37	40	0	43	44
9	8	16	23	29	34	38	41	43	0	45
10	9	17	24	30	35	39	42	44	45	0

```

**METHOD #2 (Mata)
use temp, clear
putmata X=(p1-p10), replace
mata: X=mm_cond(X==., 0, X)
mata: X=X:+X'
mata: X

list, sep(0) noobs
getmata (p*)=X, replace
list, sep(0) noobs

```

place	p1	p2	p3	p4	p5	p6	p7	p8	p9	p10
1	0	1	2	3	4	5	6	7	8	9
2	1	0	10	11	12	13	14	15	16	17
3	2	10	0	18	19	20	21	22	23	24
4	3	11	18	0	25	26	27	28	29	30
5	4	12	19	25	0	31	32	33	34	35
6	5	13	20	26	31	0	36	37	38	39
7	6	14	21	27	32	36	0	40	41	42
8	7	15	22	28	33	37	40	0	43	44
9	8	16	23	29	34	38	41	43	0	45
10	9	17	24	30	35	39	42	44	45	0

Method #1 è lo stesso discusso prima, Method #2 è la versione Mata della stessa procedura. Con putmata importo in Mata i dati da Stata, con la funzione mm_cond² converto i dati missing della matrice diagonale superiore in zeri. A questo punto sommo elemento per elemento la

2. mm_cond è una funzione del pacchetto moremata. Per installarlo digitare nella command bar ”ssc inst moremata”.

matrice X con la sua trasposta; infine riporto i dati della matrice X in Stata, sostituendoli nelle variabili p1-p10.

Last revision: 27 marzo 2023