



2018 Italian Stata Users Group meeting

Bologna | 15 November

I Portici Hotel



Finding data embedded in text files:
using `fileread()` and basic string functions to
extract spatial coordinates from google map or
counts in preformatted documents



Giovanni Capelli

*Dipartimento di Scienze Umane, Sociali e della Salute (SUSS)
Università degli Studi di Cassino e del Lazio Meridionale*

The strL format

- ▶ From Stata 13 on, Stata supports a new string data type
 - *long string* → *strL*
 - Up to two billion characters
 - String functions work within the long string
 - *To search and extract specific numerical or categorical data*
 - using `strpos()` and `substr()` string functions
 - Can contain entire files
 - *In plain text (ASCII) but also binary objects*
 - Multiple files can be uploaded at once using the programming function `fileread()`



Two problems to be solved

- ▶ #1 A database of addresses
 - *To be geocoded*
 - Finding out Longitude and Latitude of each address
- ▶ #2 A word document
 - *containing individual scores*
 - needs an anonymous version for public disclosure
- ▶ *Both can find a solution through a combination of `fileread()` and application of `strpos()` and `substr()` on Long Strings*



#1 Geocoding addresses

- ▶ In 2011, A. Ozimek and D. Miles published on the *Stata Journal* a paper on geocoding by Stata
 - *The Stata Journal (2011) 11, Number 1, pp. 106–119, «Stata utilities for geocoding and generating travel time and travel distance information»*
 - Presenting the command **geocode** (dm0053)
 - Which now can be downloaded in the version **geocode3**

help for **geocode3**

geocodes addresses using google maps or yahoo maps

geocode, **address**(varname) **city**(varname) **state**(varname) **zip**(varname) [**fulladdr**(varname) **yahoo both**]

Description

geocode uses Google Maps and Yahoo! maps api to geocode addresses and calculate latitude and longitude.



Troubles with geocode

- ▶ But... when trying to apply the geocode command to Italian addresses...
 - *The program enters an infinite loop:*

```
-----
- if "`addr'" != "" {
= if "VIA+GIACOMO+MATTEOTTI,+GALATONE,++000" != "" {
- noisily di as text "Google Geocoding `i' of `cnt'"
- noisily di as text "Google Geocoding 1 of 32"
Google Geocoding 1 of 32
- capture: copy "http://maps.google.com/maps/geo?q=`addr'&output=csv" `txtfile', replace
= capture: copy "http://maps.google.com/maps/geo?q=VIA+GIACOMO+MATTEOTTI,+GALATONE,++000&output=csv" /var/folders/k9/07
> 44q51954v817xvxvt5801r0000gn/T//S_01512.000002, replace
- while _rc == 2 | _rc==612 {
noi: di "Connection error, retrying observation #"`i'
capture: copy "http://maps.google.com/maps/geo?q=`addr'&output=csv" `txtfile', replace
}
- capture: insheet geocode geoscore latitude longitude using `txtfile', clear comma
= capture: insheet geocode geoscore latitude longitude using /var/folders/k9/0744q51954v817xvxvt5801r0000gn/T//S_01512.
> 000002, clear comma
- while _rc==601 {
- capture: insheet geocode geoscore latitude longitude using `txtfile', clear comma
= capture: insheet geocode geoscore latitude longitude using /var/folders/k9/0744q51954v817xvxvt5801r0000gn/T//S_01512.
> 000002, clear comma
- }
- while _rc==601 {
- capture: insheet geocode geoscore latitude longitude using `txtfile', clear comma
= capture: insheet geocode geoscore latitude longitude using /var/folders/k9/0744q51954v817xvxvt5801r0000gn/T//S_01512.
> 000002, clear comma
- }
-----
```

Finding a solution

- ▶ The **geocode** help itself suggests to find more information on codes at the webpage

- <http://code.google.com/apis/maps/documentation/geocoding/>

The screenshot shows the Google Maps Platform documentation page for the Geocoding API. The page is titled "Web Services > Geocoding API" and features a navigation menu with "Overview", "Products", "Pricing", and "Documentation". A search bar and a "Cerca" button are visible. The page includes a "GET STARTED" button and a "CONTACT SALES" button. A blue banner at the top of the content area states: "New pricing changes went into effect on July 16, 2018. For more information, check out the [Guide for Existing Users](#)." The main content area is titled "Get Started" and includes a star rating of four stars. The text describes the Geocoding API as a service that provides geocoding and reverse geocoding of addresses. A blue callout box contains a star icon and the text: "This service is also available as part of the client-side [Google Maps JavaScript API](#), or for server-side use with the [Java Client](#), [Python Client](#), [Go Client](#) and [Node.js Client](#) for [Google Maps Services](#)." The text continues: "Geocoding is the process of converting addresses (like a street address) into geographic coordinates (like latitude and longitude), which you can use to place markers on a map, or position the map." "Reverse geocoding is the process of converting geographic coordinates into a human-readable address." "You can also use the Geocoding API to find the address for a given [place ID](#)." The page also includes a "Sample request and response" section, which states: "You access the Geocoding API through an HTTP interface. Following are examples of geocoding and [reverse geocoding](#) requests." "Geocoding request and response (latitude/longitude lookup)". On the right side, there is a "Contenuti" section with a list of links: "Sample request and response", "Geocoding request and response (latitude/longitude lookup)", "Reverse geocoding request and response (address lookup)", "Start coding with our client libraries", "Authentication, quotas, pricing, and policies", "Activate the API and get an API key", "Quotas and pricing", "Policies", and "Learn more". On the left side, there is a "Get Started" section with links: "Developer Guide", "Get API Key", "Best Practices Geocoding Addresses", and "Geocoding FAQ". Below this is a "Web Services" section with links: "Best Practices" and "Client Libraries". Below that is a "Policies and Terms" section with links: "Usage and Billing", "Optimizing Quota Usage", "Policies", and "Terms of Service". At the bottom, there is an "Other Web Service APIs" section with links: "Directions API", "Distance Matrix API", "Elevation API", and "Geolocation API".



Sample request and response

You access the Geocoding API through an HTTP interface. Following are examples of geocoding and [reverse geocoding](#) requests.

Geocoding request and response (latitude/longitude lookup)

The following example requests the latitude and longitude of "1600 Amphitheatre Parkway, Mountain View, CA", and specifies that the output must be in JSON format.

```
https://maps.googleapis.com/maps/api/geocode/json?address=1600+Amphitheatre+Parkway,+Mountain+View,+CA
```

You can test this by entering the URL into your web browser (be sure to replace 'YOUR_API_KEY' with [your actual API key](#)). The response includes the latitude and longitude of the address.

View the [developer's guide](#) for more information about [building geocoding request URLs](#) and [available parameters](#) and [understanding the response](#).



```
{
  "results" : [
    {
      "address_components" : [
        {
          "long_name" : "14",
          "short_name" : "14",
          "types" : [ "street_number" ]
        },
        {
          "long_name" : "Via Guglielmö Röntgen",
          "short_name" : "Via Guglielmö Röntgen",
          "types" : [ "route" ]
        },
        {
          "long_name" : "Milano",
          "short_name" : "Milano",
          "types" : [ "locality", "political" ]
        },
        {
          "long_name" : "Milano",
          "short_name" : "Milano",
          "types" : [ "administrative_area_level_3", "political" ]
        },
        {
          "long_name" : "Città Metropolitana di Milano",
          "short_name" : "MI",
          "types" : [ "administrative_area_level_2", "political" ]
        },
        {
          "long_name" : "Lombardia",
          "short_name" : "Lombardia",
          "types" : [ "administrative_area_level_1", "political" ]
        },
        {
          "long_name" : "Italia",
          "short_name" : "IT",
          "types" : [ "country", "political" ]
        },
        {
          "long_name" : "20136",
          "short_name" : "20136",
          "types" : [ "postal_code" ]
        }
      ],
      "formatted_address" : "Via Guglielmö Röntgen, 14, 20136 Milano MI,
Italia",
      "geometry" : {
        "bounds" : {
          "northeast" : {
            "lat" : 45.450613,
            "lng" : 9.1871033
          },
          "southwest" : {
            "lat" : 45.450335,
            "lng" : 9.186801599999999
          }
        },
        "location" : {
          "lat" : 45.4504834,
          "lng" : 9.186947699999999
        },
        "location_type" : "ROOFTOP",
        "viewport" : {
          "northeast" : {
            "lat" : 45.4518229802915,
            "lng" : 9.188301430291503
          },
          "southwest" : {
            "lat" : 45.4491250197085,
            "lng" : 9.185603469708498
          }
        },
        "place_id" : "ChIJnQx6ugXEhkcRkKzVQm2ZREw",
        "types" : [ "premise" ]
      },
      "status" : "OK"
    }
  ]
}
```

https://maps.googleapis.com/maps/api/geocode/json?address=14+Via+roentgen+milano+ITALY&key=AlzaSyBU7B8Vl1ZbazXceeYqnuauo_XXXXXXXXXX

Keypoints

- ▶ The `https://` address string can be built
 - *Using the available elements of the address*
 - + the personal API key (the red and blue one...)
 - Which has to be released by Google Cloud Platform
 - *Latitude and Longitude come constantly after "sentinel text" such as "lat" and "long"*
 - Numerical Latitude and Longitude can be found and extracted searching the "sentinel text" by `strpos()` and `substr()`
 - If the json format file is imported in a `strL` variable



```

1 * crea stringa indirizzi per coordinate googlemap
2
3 if "`1'"==""{
4     local nation="ITALY"
5     local stub "_ita"
6 }
7 if "`1'"!="" {
8     local nazione="`1'"
9     local stub="_" + substr(`nazione',1,3)
10    tokenize "`nazione'", parse(" " ",")
11    local nation="`1'"
12    mac shift
13    while "`1'" != "" {
14        local nation="`nation'"+"+"+"`1'"
15        mac shift
16    }
17 }
18
19 capture drop indirizzo apigoogole test poslat poslng lngtxt
20 lattxt lat* lng*
21 gen indirizzo= ustrregexra(Indirizzo,`""',""s2)
22
23 set more off
24 capture log close
25
26 local n=_N
27
28 gen apigoogole=""
29
30 local i=1
31
32 while `i' <= `n' {
33     local indirizzo=indirizzo[`i']
34     tokenize "`indirizzo'", parse(" " ",")
35     local address="`1'"
36     mac shift
37     while "`1'" != "" {
38         local address="`address'"+"+"+"`1'"
39         mac shift
40     }
41     local cap=Cap[`i']
42     local comune=Comune[`i']
43     tokenize "`comune'", parse(" " ",")
44     local town="`1'"
45     mac shift
46     while "`1'" != "" {
47         local town="`town'"+"+"+"`1'"
48         mac shift

```

```

48     }
49
50     local provincia=Provincia[`i']
51     tokenize "`provincia'", parse(" " ",")
52     local region="`1'"
53     mac shift
54     while "`1'" != "" {
55         local region="`region'"+"+"+"`1'"
56         mac shift
57     }
58
59
60     local api=
61     "https://maps.googleapis.com/maps/api/geocode/json?address="+
62     "`address'"+"+"`cap'"+"+"`town'"+"+"`region'"+"+"`nation'"+"
63     "&key=AIzaSyBU7B8Vl1ZbazXceeYqnuauo_XXXXXXXXX"
64     replace apigoogole="`api'" in `i'/'i'
65     local i=`i'+1
66 }
67
68 generate strL googlepage = fileread(apigoogole)
69 gen poslat=strpos(googlepage,`"lat"'')+7
70 gen poslng=strpos(googlepage,`"lng"'')+7
71 gen lattxt=substr(googlepage,poslat,10)
72 gen lngtxt=substr(googlepage,poslng,10)
73 gen lat`stub'=real(trim(substr(googlepage,poslat,7)))
74 gen lng`stub'=real(trim(substr(googlepage,poslng,7)))
75
76
77

```

#2 Anonymizing documents

- ▶ University of Cassino & SL curriculum management software produces reports on student's course evaluation questionnaires
 - *The main report is produced in Word Format, and contains individual evaluation scores in graphical and tabular format*
 - These "disclosed" versions are used by the Course Management Structures
 - But the University policy is to publish only anonymous data on the website
 - *How can graphics and total number of questionnaires be "extracted" from the files and rebuilt in a new file?*

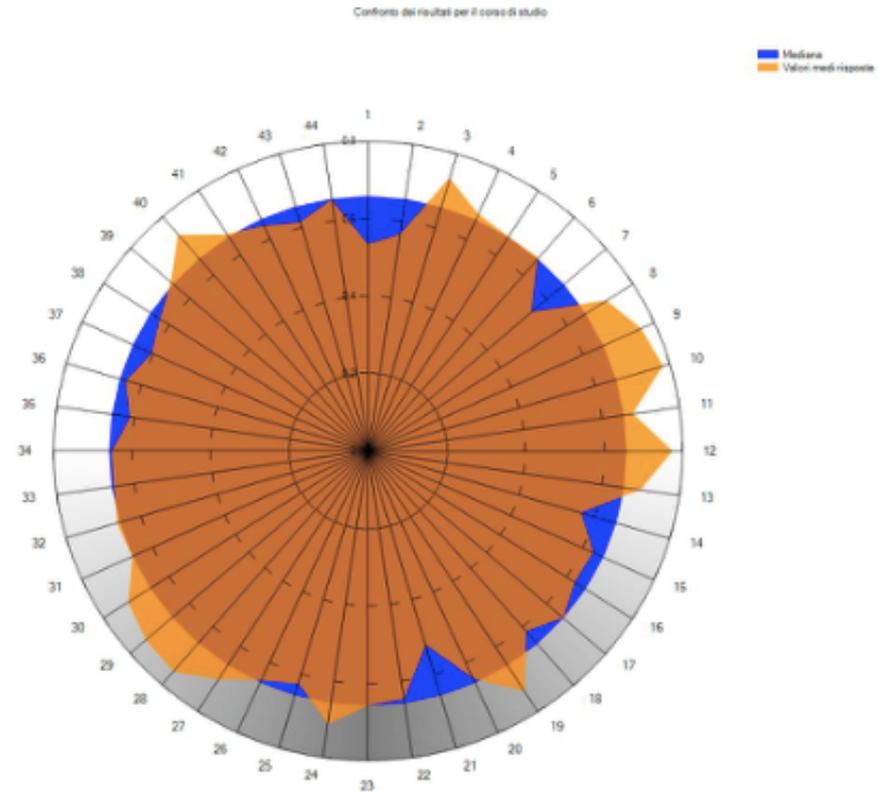


The Word original file format

Corsi di studio

[7132] Scienze Motorie L-22

Il grafico che segue è elaborato sulla base dell' **21.895** risposte nel contesto stabilito dai filtri impostati. Il valore mediano (rappresentato nel grafico in colore blu) calcolato sulla serie dei punteggi medi di ogni docente è pari a: **0,658**



#	Docente	Insegnamento	Questionari	Risposte	Media	+/- Mediana
1	ANASTASI DANIELA	[7LCG0090] C.I. ANALISI DEI DATI MOTORI E SPORTIVI	21	296	0,535	-0,123
2	ANASTASI DANIELA	[91485] C.I. Salute e attività motoria	62	850	0,567	-0,091



The extraction and rebuilding procedure

1. Save the Word file in: a) Plain text version (to be processed for the «numbers»); b) html version (to extract the radar plots)
2. Upload in a single Stata file all the txt files for each study curriculum using fileread() → [counter_radar.do](#)
3. Extract the number of questionnaires and the average value for each question in each curriculum using strpos() and substr() → [counter_radar.do](#)
4. Rebuilt LaTeX files for each line of the Stata file, combining standard text + the extracted numbers + the jpg images of the radar plots saved for the html version → [LaTeX_izza.do](#)



Questionario Allegato IX - Scheda 1 CASSINO - STUDENTI FREQUENTANTI

Corso di Studio: L-22

19 settembre 2018

Corso di Studio: [7132] Scienze Motorie L-22

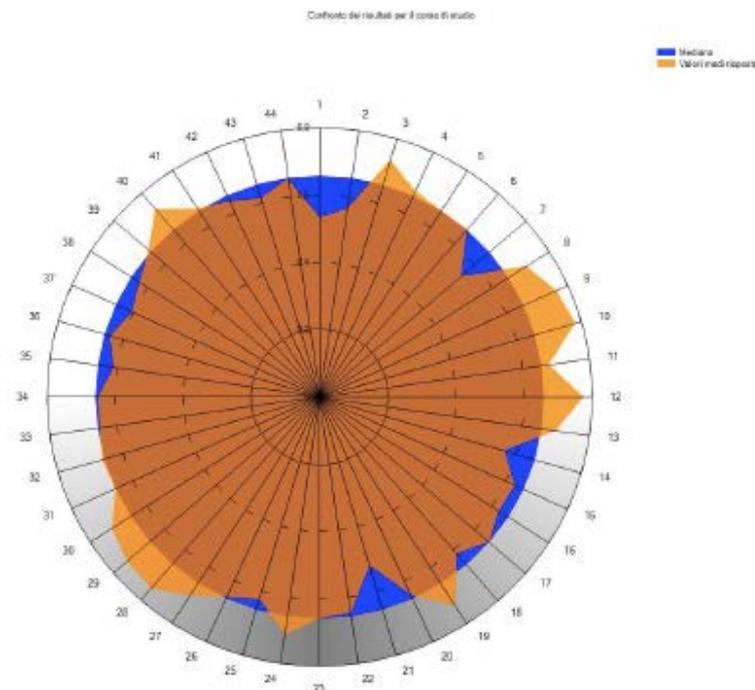
Fonte dati: GOMP Università' di Cassino, Rilevazioni AA 2017/18

Indice

- | | | |
|----|--|----|
| 1 | Complessivo del Corso di Studio | 2 |
| 2 | Le conoscenze preliminari possedute sono risultate sufficienti per la comprensione degli argomenti previsti nel programma d'esame? | 3 |
| 3 | Il carico di studio dell'insegnamento e' proporzionato ai crediti assegnati? | 4 |
| 4 | Il materiale didattico (indicato e disponibile) e' adeguato per lo studio della materia? | 5 |
| 5 | Le modalita' di esame sono state definite in modo chiaro? | 6 |
| 6 | Gli orari di svolgimento di lezioni, esercitazioni e altre eventuali attivita' didattiche sono rispettati? | 7 |
| 7 | Il docente stimola / motiva l'interesse verso la disciplina? | 8 |
| 8 | Il docente espone gli argomenti in modo chiaro? | 9 |
| 9 | Le attivita' didattiche integrative (esercitazioni, tutorati, laboratori, etc...) sono utili all'apprendimento della materia? | 10 |
| 10 | L'insegnamento e' stato svolto in maniera coerente con quanto dichiarato sul sito Web del corso di studio? | 11 |
| 11 | Il docente e' reperibile per chiarimenti e spiegazioni? | 12 |

1 Complessivo del Corso di Studio

Il grafico che segue e' elaborato sulla base delle 21895 risposte nel contesto stabilito dai filtri impostati. Il valore mediano (visualizzato nel grafico in colore blu) calcolato sulla serie dei punteggi medi di ogni docente e' pari a: 0.658



**The LaTeX/PDF
final anonymous
version**

