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.

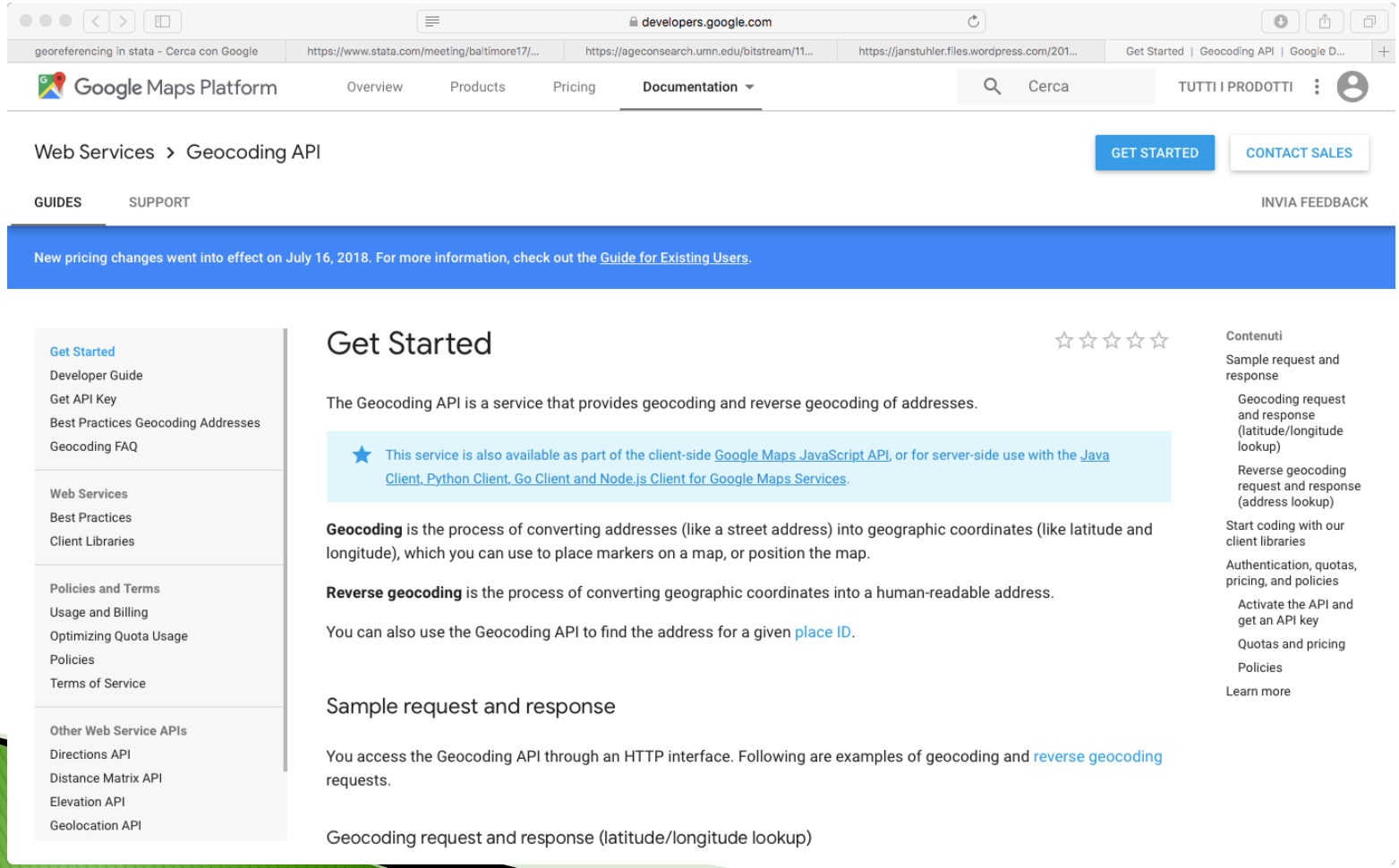
Troubles with geocode

- 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
> 44q51954v8l7xvxvt5801r0000gn/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/k9/0744q51954v8l7xvxvt5801r0000gn/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/k9/0744q51954v8l7xvxvt5801r0000gn/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/k9/0744q51954v8l7xvxvt5801r0000gn/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/*



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,+C
```

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=AIzaSyBU7B8Vl1ZbazXceeYqnuauo_XXXXXXXXX

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*

```
* crea stringa indirizzi per coordinate googlemap

if "`1'"==""{
    local nation="ITALY"
    local stub "_ita"
    }
if "`1'"!="" {
    local nazione="`1'"
        local stub="_"+substr(`nazione',1,3)
        tokenize "`nazione'", parse(" " ",")
        local nation="`1'"
        mac shift
        while "`1'" != "" {
        local nation="`nation'"+"+"+"`1'"
        mac shift
        }

    }
capture drop indirizzo apigoogle test poslat poslng lngtxt
lattxt lat* lng*
gen indirizzo= ustrregexra(Indirizzo,`"""',"'"s2)

set more off
capture log close

local n=_N

gen apigoogle=""

local i=1

while `i' <= `n' {
    local indirizzo=indirizzo[`i']
    tokenize "`indirizzo'", parse(" " ",")
        local address="`1'"
        mac shift
        while "`1'" != "" {
        local address="`address'"+"+"+"`1'"
        mac shift
        }
    local cap=Cap[`i']
    local comune=Comune[`i']
    tokenize "`comune'", parse(" " ",")
        local town="`1'"
        mac shift
        while "`1'" != "" {
        local town="`town'"+"+"+"`1'"
        mac shift
        }

    local provincia=Provincia[`i']
        tokenize "`provincia'", parse(" " ",")
        local region="`1'"
        mac shift
        while "`1'" != "" {
        local region="`region'"+"+"+"`1'"
        mac shift
        }


    local api=
"https://maps.googleapis.com/maps/api/geocode/json?address="+
"`address'"+"+`cap'"+"+`town'"+"+`region'"+"+`nation'"+
"&key=AIzaSyBU7B8Vl1ZbazXceeYqnuauo_XXXXXXXXX"
    replace apigoogle="`api'" in `i'/`i'
    local i=`i'+1
}

generate strL googlepage = fileread(apigoogle)
gen poslat=strpos(googlepage,`"lat""')+7
gen poslng=strpos(googlepage,`"lng""')+7
gen lattxt=substr(googlepage,poslat,10)
gen lngtxt=substr(googlepage,poslng,10)
gen lat`stub'=real(trim(substr(googlepage,poslat,7)))
gen lng`stub'=real(trim(substr(googlepage,poslng,7)))
```

#2 Anonymizing documents

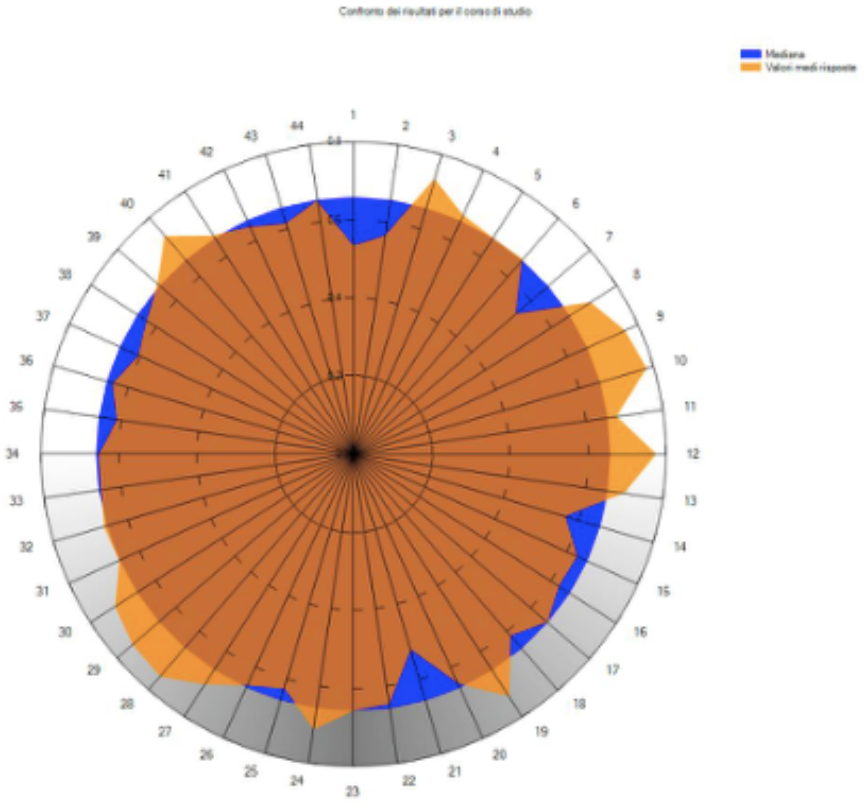
- University of Cassino & SL curriculum management software produces reports on student's course evaluation questionnaires
  - *The main report is produced in Word Format, and contains individual evaluation scores in graphical and tabular format*
    - These "disclosed" versions are used by the Course 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 delle 21.895 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 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

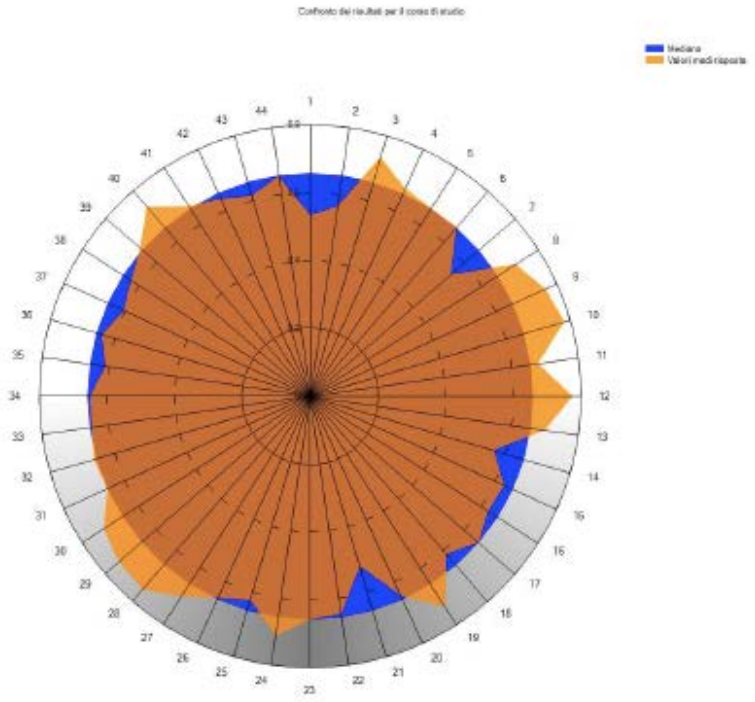
19 settembre 2018

Corso di Studio: [7132] Scienze Motorie L-22
Fonte dati: GOMP Universita' di Cassino, Rilevazioni AA 2017/18

Indice

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