Data cleaning in Stata using internet search engines

Sergiy Radyakin

mailto:sradyakin@worldbank.org

Development Economics Research Group
The World Bank

October 22, 2009

Table of contents

Data cleaning

Data cleaning is often required before statistical processing of the following information is possible:

- geographic entities: countries, districts, counties, cities, etc.
- occupations/specializations, educational degrees
- products' names and brands, e.g. software products
- movie titles, music performers
- skills, talents, duties, diseases
- other open-ended questions

Data cleaning

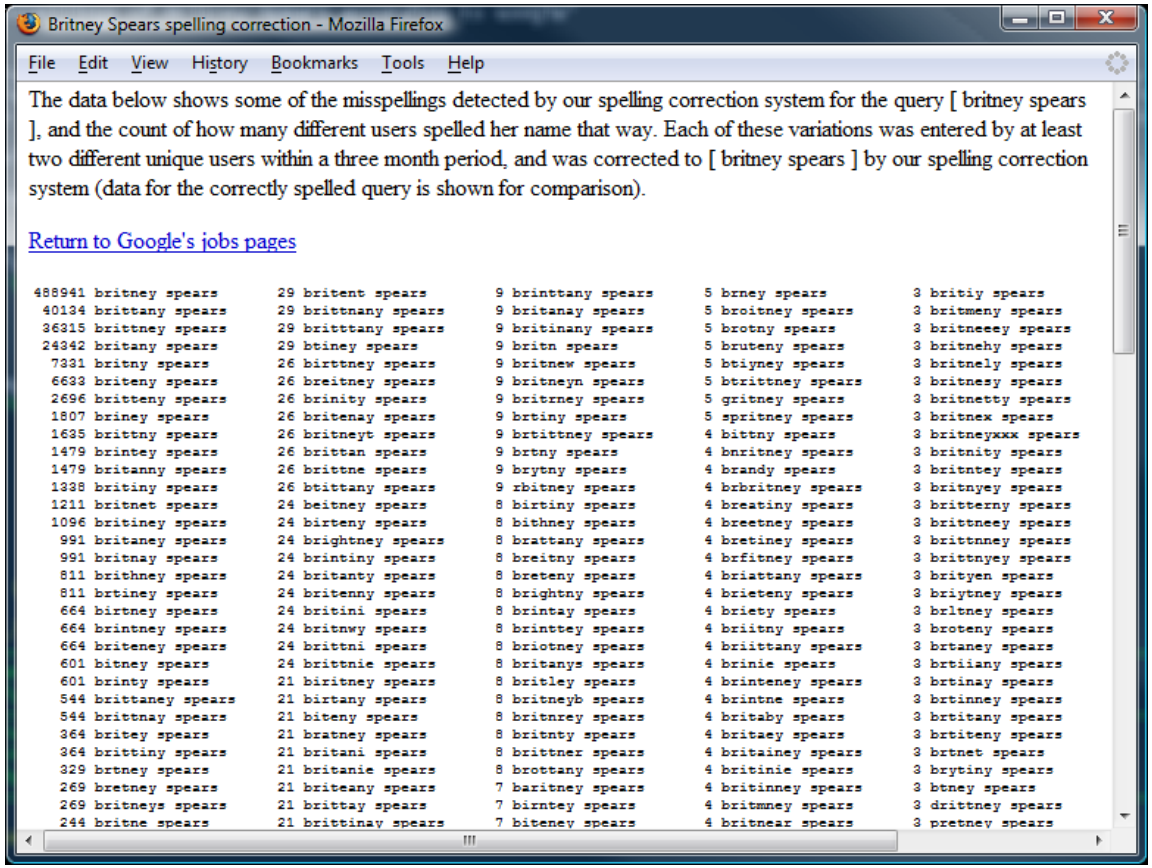
More and more people are asked to fill-in the questionnaires themselves, which increases the probability of an error.

Typical errors include:

- Typos: *Kazakjstan* - j instead of h, wrong key pressed
- Spell-as-you-hear, typical for foreign words: *Kazahstan* instead of *Kazakhstan*, *Yvette Gilber* - instead of *Yvette Guilbert*
- Recall errors: *"I don't remember what was that software that we worked in the 70s, BMDP? or BRDP?"*
- OCR errors during scanning and recognizing printed forms

How many ways to spell "Britney Spears"?

According to Google: 592 (about 23% of all searches within a 3 months period were spelled incorrectly)



Source: `http://labs.google.com/britney.html`

Actual dataset examples

photoshop	lotus notes	corel draw
photo shop	lotusnotes	carel draw
photo-shop	lotus-notes	orel draw
foto shop	lotusnotus	coreldraw
fotoshop	lotus notus	corldraw
foto-shop	lotus-notus	corel-draw
fhoto shop	lotes-notes	corel draw
fhoto-shop	lotesnotes	coral draw
fhotoshop	lotes notes	corral draw
potoshop	lotos	c.draw

Source: dataset from Kazakhstan containing self-reported knowledge of software by job-seekers.

Fragment of an ad-hoc data-cleaning program that calls a script to verify if any of the variants of spelling is present.

```
        recognize_skill oldDbTextField3, generate(msaccess) ///
          spelling("access" "acess" "acces")
        recognize_skill oldDbTextField3, generate(msproject) ///
          spelling("ms project" "msproject" "microsoft project" ", project,"
",project," ",project ," ", project ," "(project," ", project)")
        recognize_skill oldDbTextField3, generate(msoutlook) ///
          spelling("outlook" "utlook" "out look" "outllook" "out-look"
"Otlook-Exspress")
        recognize_skill oldDbTextField3, generate(buh1C) ///
          spelling("1c" "1-c" "1:c" "1 c")
        recognize_skill oldDbTextField3, generate(progcpp) ///
          spelling("c++" "c + +")
        recognize_skill oldDbTextField3, generate(autocad2) ///
          spelling("autocad" "auto cad" "auto-cad" "avto cad" "avtocad")
        recognize_skill oldDbTextField3, generate(coreldraw) ///
          spelling("corel draw" "carel draw" "orel draw" "coreldraw" "corldraw"
"corel-draw" "corel  draw" "coral draw" "corral draw" "c.draw")
        recognize_skill oldDbTextField3, generate(photoshop) ///
          spelling("photoshop" "photo shop" "photo-shop" "foto shop" "fotoshop"
"foto-shop" "fhoto shop" "fhoto-shop" "fhotoshop" "potoshop")
        recognize_skill oldDbTextField3, generate(foxpro) ///
          spelling("fox pro" "foxpro" "fox-pro")
        recognize_skill oldDbTextField3, generate(lotusnotes) ///
          spelling("lotus notes" "lotusnotes" "lotus-notes" "lotusnotus" "lotus
notus" "lotus-notus" "lotes-notes" "lotesnotes" "lotes notes" "lotos")
```

Typical ways of approaching the problem

Standard ways of approaching the problem:

- 1. Prevent the problem from appearing in the first place. Create codes for all possible answers (e.g. assign codes to countries or occupations) and let the respondent select those codes. Not always possible, may restrict the answers.
- 2. Mindless cleaning of the data: removing heading and trailing spaces, replacing multiple whitespaces, etc. Recommended to do before any more sophisticated cleaning. See "Stata Tip 64"
- 3. SOUNDEX-like algorithms of data cleaning: allow determining if two words "sound alike"; need a list of prototypes, against which to match.

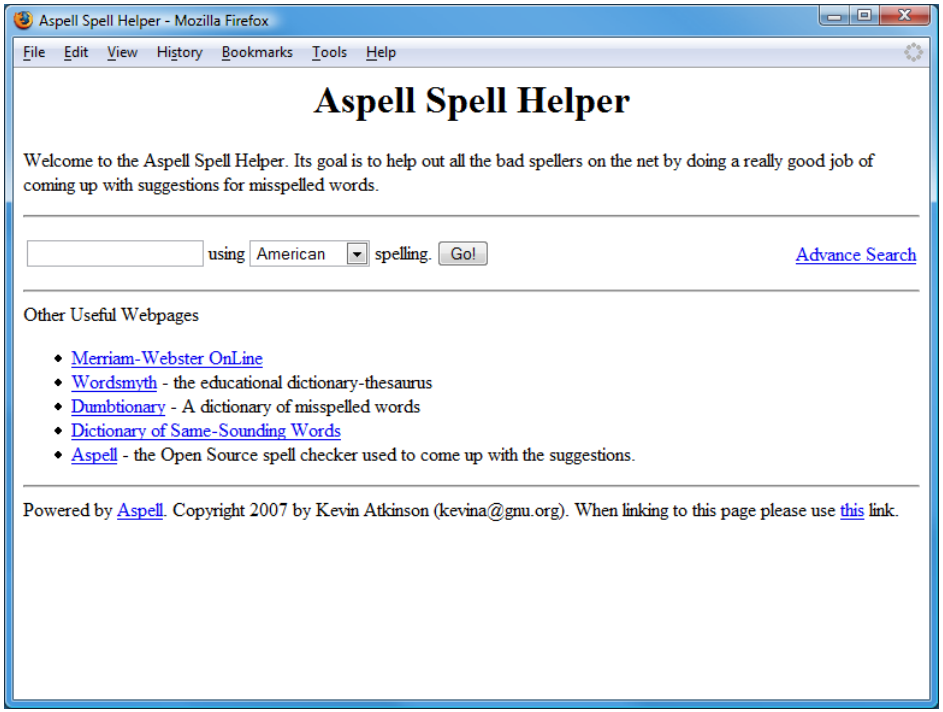
SOUNDEX performance

"Microsoft" and *"Macrosort"* have the same code M262 and hence according to this algorithm "sound alike".

- 4. Mindful cleaning the data - human operator reviews the responses and manually corrects each answer. Typically a long, tedious, and boring assignment: mistakes can be skipped when the operator is tired, new mistakes can be introduced.

We would like to minimize the load on the operator: pre-clean the data in some intelligent way and let the operator decide in ambiguous cases.

Online spelling correctors and search engines



A notable GNU-licensed specialized spelling suggestions system is ASpell (`http://www.aspell.net`)

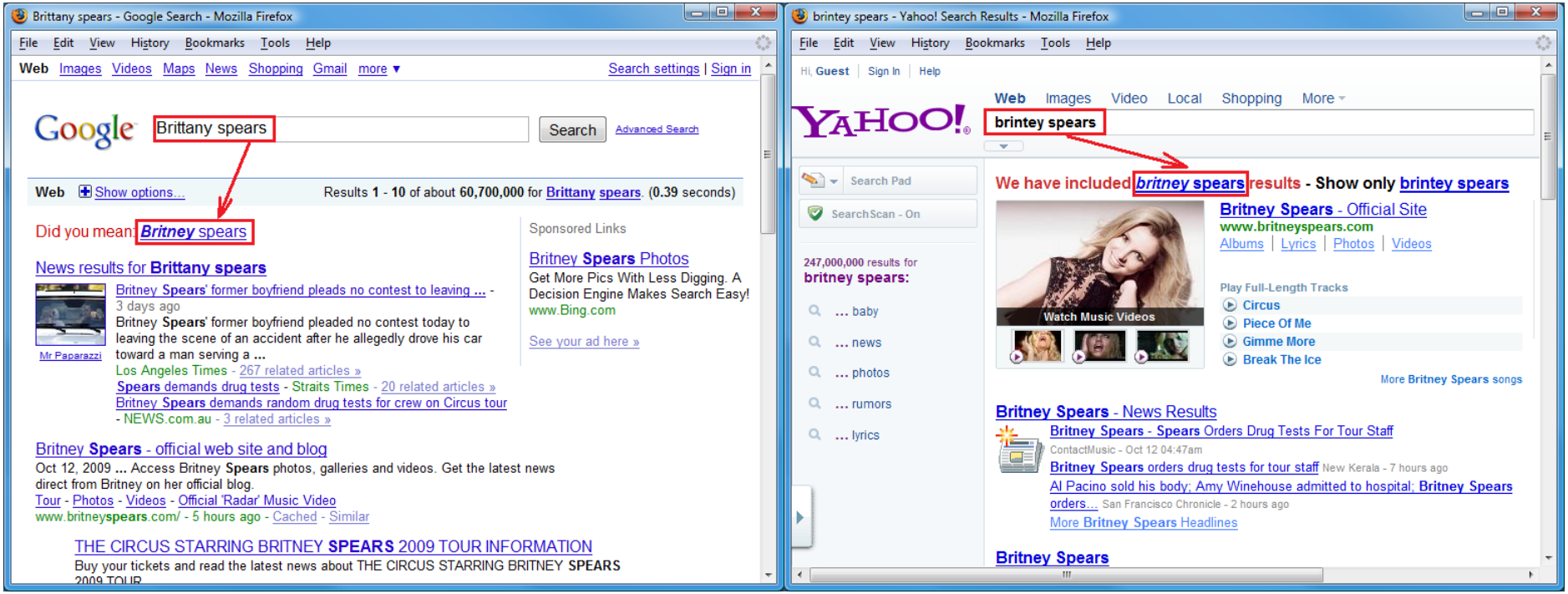
There is a number of websites that allow spell-checking online, most are oriented on humans and dictionary search. Typically allow:

- verifying the word is in the dictionary or not
- obtaining one or more spelling suggestions if the word is not in the dictionary
- obtaining a list of the related words/synonyms

Typically limited to the dictionary words, e.g. do not recognize *Photoshop* or *AutoCAD*.

Online spelling correctors and search engines

Users of the modern search engines enjoy the spelling correction/search suggestions features:



What would it take to correct our datasets in a similar fashion in Stata?

What is good about using Internet search engines?

Internet search engines have the following desired features helpful in data cleaning:

- Proper nouns: unlike many spelling correctors, search engines like Google and Yahoo can suggest common spellings for: places, names, brands, software titles, etc, which are often of interest in the open-ended questions.
- Continuous self-perfection: Internet search engines' databases are constantly updated, and their suggestions are automatically revised, as they discover new web pages in the Internet.
- Context and relevance: suggestions take into account the relationship between the words in the query, not just spelling.

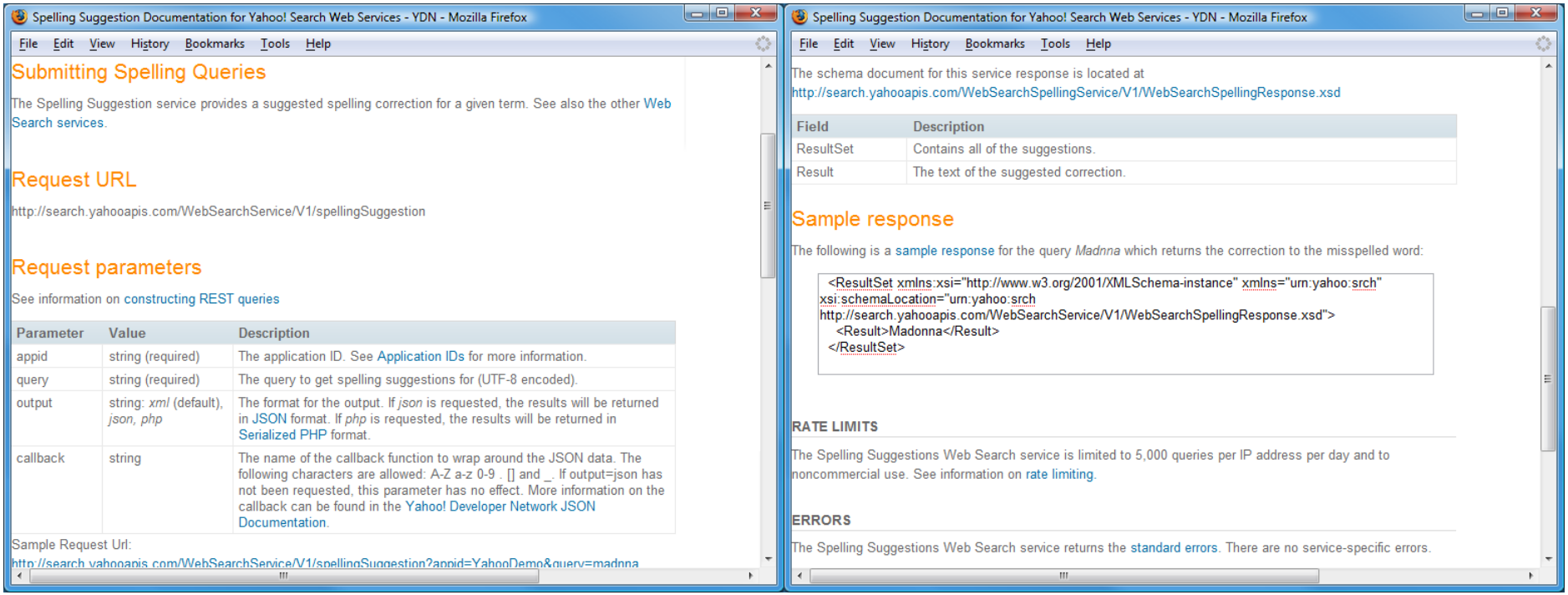
A word of caution

Internet-based data cleaning is subject to some limitations:

- not guaranteed to be reproducable: if you re-run your program next month, you can obtain different suggestions (because the search engine has changed the algorithm or renewed the database) or the web-service may not be available anymore
- not guaranteed to be 100% accurate: some spellings can be recognized as correct when they are not, some correct words may not be recognized as such
- data transmission: by definition you need to send your data to the remote system, which may be against the data license conditions
- increases the load on the site servicing the requests: you may need a permission to send automated queries - see the site use conditions for a particular search engine or web-service

Query

How to submit a query to the search engine from Stata and get it's response in a form undestandable by a Stata program?
A search engine may or may not have a specialized API (*Application Programmer's Interface*) which describes the answer to the above question:



Requires obtaining special application ID. Use of the spelling corrections is subject to a 5000 queries per 24hrs limit. Describes the parameters necessary to submit a query: URL, query and output format.

Query

Communication with most of the search websites starts with a query, which describes what information we want to retrieve, restrictions on the search, output format:

Query-URL

`http://search.engine.com/search_command?query=our_search_term&parameter1=value1&parameter2=value2...`

Query element	Example (Google)	Example (Yahoo)
protocol	http://	http://
search_engine_site	www.google.com	search.yahooapis.com
search_command	search	WebSearchService/V1/spellingSuggestion
?	?	?
query=our_search_term	q=chicaga	query=chicaga
parameter1=value1	hl=en	&appid=xjC1kefV34...vf4a
&parameter2=value2	&source=hp	
&parameter3=value3	&aq=f	
&parameter4=value4	&oq=	
&parameter5=value5	&aqi=g10	

see also `http://en.wikipedia.org/wiki/Uniform_Resource_Locator`

Query

- 1. use the website in the regular (human-operated mode). notice where you submit the query, investigate how the query is formed. What parameters are submitted? which ones are required? which ones are optional? Decide which parameters and values you will submit in your program.

For example, Google

`http://www.google.com/search?hl=en&source=hp&q=chicaga&aq=f&oq=&aqi=g10`
E.g. here *hl = en* sets the English language for the search
q = chicaga is the query term
other options are irrelevant, but required - keep them

- 2. Submit your query from Stata using the -copy- command*; save results to a file (in almost all cases it is an HTML file) and investigate it with a text editor.
- 3. Compare the HTML source with the output of the web page on the screen. Identify where in the HTML code you see the suggestion. Identify the pattern/template of the server's response.
- 4. Write a small program, that given a search term submits it to the website in the determined query form, saves the response to a tempfile and reads it in, following the identified pattern and returns the suggestion recieved from the web site.

Use of the -copy- command

A common misunderstanding regarding the -copy- command is that it *"can only read information from Internet, but cannot write"*. This is not true.
While it cannot copy a file to a specified site, it can call the site engine and request it doing something useful, for example, altering a web page or sending an email.
The limits are defined by the range of actions a site can undertake in response to the commands sent from the Internet.

Constructing the query URL

URL may not contain whitespaces and some other special characters, which need to be replaced before the query is submitted.
Use the -subinstr()- extended macro substitution to replace the characters that may not occur in the query. For example for Google:

```
local search_term `"`:subinstr local anything " " "+", all'"'
```

Find the most reliable pattern

If API is not provided and the Stata-implemented command relies on parsing the HTML output, try to find the most reliable pattern of determining where the results are in the output. Search engines periodically change the templates, which they use. This typically requires revising the ado-code and changing a few "magic numbers".

Yahoo and Google

This is what the two Stata commands presented here are doing:

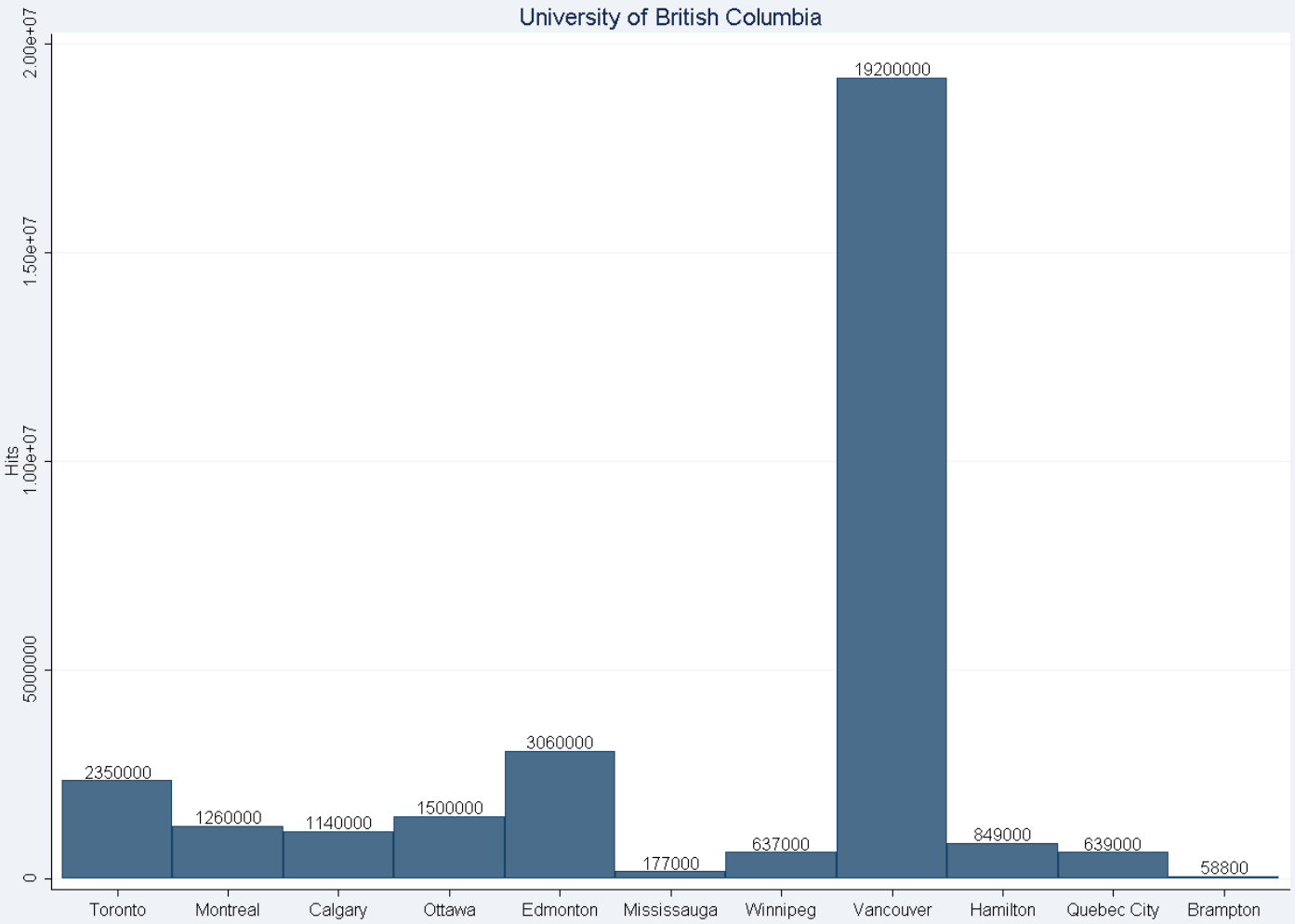
- `-yahoo-` searches for a particular search term using Yahoo API and returns the correction if suggested by Yahoo or the search term itself. Yahoo server returns the search results in a well-formed XML format.
- `-google-` searches for a particular search term using Google (not using API) and returns the correction if suggested by Google or the search term itself (plus number of hits if no correction is suggested, or 0 if the search term not found). Google returns the results in the HTML format.
- Both commands return the spelling suggestion in the r()-saved results, so it is easy to derive other user-written commands on their basis. For example, `-google_clean-` cleans a string variable by repeatedly asking Google for suggestions for each observation
- `-google-` also returns the "hits" score, which allows comparing the relevance of the search word to other words or selecting a proper variant in case of several possible suggestions. It can also be used for missing values imputations, as in `-google_compare-`.

Some examples

Search word	Google result	Yahoo result
chicaga	chicago	chicago
firefiter	firefighter	firefighter
Microsoft Exel	Microsoft Excel	Microsoft Excel
Germania	Germania	Germania
softvare	software	software
Al Bukerke	Albukerke	Al Bukerke
Albukerque	Albuquerque	Albuquerque
Nashional	National	National
Geografic	Geografic	Geographic
Ciciety	Ciciety	Ciciety
Sosiety	Society	Society
Washengton	Washington	Washington
Washengtone	Washengtone	Washington
Vashingtown	Vashingtown	Washington
reserch dpt	research dept	research dpt
Originall manufactur	Original manufacturer	Original manufacturer
Ukraina	Ukraina	Ukraine
Kharkiv National University	Kharkiv National University	Kharkov National University
Sergey Radyakin	Sergiy Radyakin	Sergey Radyakin
Time	**5.31sec**	**2.30sec**

Note: measured times do not measure respective web sites/search engines performance, they measure performance of the corresponding Stata commands in their current implementation. Highlighted words yield different suggestions.

Some examples



Data imputation with a search engine.

Here is how we could try to guess where a particular university is located if the respondent didn't specify the city.

The bar chart shows the number of hits as reported by the Google search engine for the searches of a combination of *"University of British Columbia"* and major Canadian cities.

Most hits are reported for the true location.

Literature

- Herrin, Jeph, and Eva Poen, 2008 "Stata tip 64: Cleaning up user-entered string variables", The Stata Journal, Vol. 8, No. 3, pp.444-445
- Dornfest, Rael, Paul Bausch, and Tara Calishain "Google Hacks", 3rd Ed., O'Reilly Media, ISBN 0-596-52706-3, 2006
- `http://www.stata.com/statalist/archive/2005-05/msg00288.html`
- `http://www.stata.com/statalist/archive/2008-08/msg00467.html`
- `http://www.stata.com/statalist/archive/2009-10/msg00279.html`

Legal

It is explicitly forbidden by the author to store this presentation at any document-harvesting web sites, such as, but not limited to:

- `http://www.docstoc.com`
- `http://www.sunum.org`
- `http://www.toodoc.com`