The country converter coco - a Python package for converting country names between different classification schemes

Konstantin Stadler[1]

[1]Industrial Ecology Programme, NTNU Trondheim, Norway.

12 July 2017

Paper DOI: http://dx.doi.org/10.21105/joss.00332
Software Repository: https://github.com/konstantinstadler/country_converter
Software Archive: http://dx.doi.org/10.5281/zenodo.838248

Summary

Gathering the data basis for regional to global models and scenarios in many scientific fields regularly requires the parsing of multiple data sources. In particular for data related to economic (national accounts, trade statistics, etc.) and environmental (climate, ecosystem descriptions, etc.) variables, these data are mostly categorised per country. There is, however, no single standard of how to name or specify individual countries. As a consequence, parsing routines need to be adopted for each used data source.

Of the existing standards, ISO 3166-1 (see ("ISO 3166-1" 2017)) defines a two and a three letter code in addition to a numerical classification. The UN uses its own numerical classification scheme, mostly based on the ISO 3166-1 numerical code. To further complicate the matter, instead of using one of the existing standards, many databases use unstandardised country names to classify countries. For example, South Korea might be referred to as "Korea", "Korea, Republic of", "Republic of Korea" or "Korea, Rep." and must be distinguished from the various names of North Korea: "Democratic People's Republic of Korea", "Dem. People's Rep. of Korea", "Korea, Dem. Rep.".

In a research project relying on multiple data sources, one usually decides on one standard name or abbreviation to be used throughout the project and then provide conversion routines for matching all encountered country names and abbreviations into the master classification.

The country converter (coco - a Python3 programme) aims to automate this step by providing a fully tested conversion set between the different country abbreviations standards and unstandardised country names.

The basis of coco is a table of regular expression which match all English versions of country names encountered by the author. The included tests match the regular expression against all so far encountered names, checking for unique matching between regular expression and country names and vice versa.

The table of regular expression and linked country names and classification can be extended by passing additional tables to account for databases using erroneous country codes or names.

In addition to the one to one matching, coco includes regional country classifications based on continent, UN region, OECD membership (per year), UN membership (per year), EU membership (per year) as well as classification of the Rest of the World countries (Stadler, Steen-Olsen, and Wood (2014)) in the Multi Regional Input Output Databases EXIOBASE (Wood et al. (2014)) and WIOD (Timmer et al. (2012)).

Coco can be used in Python and also provides a command line interface. Examples of how to use coco in Matlab(c) are provided in the Readme. An accompanying IPython notebook provides some instruction for advanced usage.

References

"ISO 3166-1." 2017. *Wikipedia*. https://en.wikipedia.org/w/index.php?title=ISO_3166-1&oldid=785413591.

Stadler, Konstantin, Kjartan Steen-Olsen, and Richard Wood. 2014. "The 'Rest of the World' – Estimating the Economic Structure of Missing Regions in Global Multi-Regional Input-Output Tables." *Economic Systems Research* 26 (3): 303–26. doi:10.1080/09535314.2014.936831.

Timmer, Marcel, Abdul A. Erumban, Reitze Gouma, Bart Los, Umed Temurshoev, Gaaitzen J. de Vries, Iñaki Arto, et al. 2012. "The World Input-Output Database (WIOD)." *Working Paper Number: 10*. http://www.wiod.org/publications/papers/wiod10.pdf.

Wood, Richard, Konstantin Stadler, Tatyana Bulavskaya, Stephan Lutter, Stefan Giljum, Arjan de Koning, Jeroen Kuenen, et al. 2014. "Global Sustainability Accounting—Developing EXIOBASE for Multi-Regional Footprint Analysis." *Sustainability* 7 (1): 138–63. doi:10.3390/su7010138.