Package 'countrycode'

May 30, 2023

Type Package

Title Convert Country Names and Country Codes

Version 1.5.0

Description Standardize country names, convert them into one of 40 different coding schemes, convert between coding schemes, and assign region descriptors.

License GPL-3

URL https://vincentarelbundock.github.io/countrycode/

BugReports https://github.com/vincentarelbundock/countrycode/issues

Depends R (>= 2.10)

Suggests eurostat, testthat, tibble, utf8

Encoding UTF-8

LazyData yes

LazyLoad yes

RoxygenNote 7.2.3.9000

NeedsCompilation no

Author Vincent Arel-Bundock [aut, cre] (<https://orcid.org/0000-0003-2042-7063>), CJ Yetman [ctb] (<https://orcid.org/0000-0001-5099-9500>), Nils Enevoldsen [ctb] (<https://orcid.org/0000-0001-7195-4117>), Samuel Meichtry [ctb] (<https://orcid.org/0000-0003-2165-791X>)

Maintainer Vincent Arel-Bundock <vincent.arel-bundock@umontreal.ca>

Repository CRAN

Date/Publication 2023-05-30 14:40:06 UTC

R topics documented:

`countrycode-package` *Convert Country Codes or Country Names*

Description

Convert country codes or country names

Details

The `countrycode` function can convert to and from several different country coding schemes. It uses regular expressions to convert country names (e.g. Sri Lanka) into any of those coding schemes, or into standardized country names in several languages. It can create variables with the name of the continent and/or several regional groupings to which each country belongs.

Type ?codelist to get a list of available origin and destination codes.

Author(s)

Vincent Arel-Bundock <vincent.arel-bundock@umontreal.ca>

References

\url{http://arelbundock.com}
\url{https://github.com/vincentarelbundock/countrycode}

`cldr_examples` *List of CLDR country name codes and associated examples*

Description

- Code: CLDR code
- Example: French Southern Territories in different languages

Usage

```
cldr_examples
```

Format

data frame

codelist *Country Code Translation Data Frame (Cross-Sectional)*

Description

A data frame used internally by the `countrycode()` function. `countrycode` can use any valid code as destination, but only some codes can be used as origin.

Format

A data frame with codes as columns.

Details

Origin and Destination:

- `cctld`: IANA country code top-level domain
- `country.name`: country name (English)
- `country.name.de`: country name (German)
- `country.name.fr`: country name (French)
- `country.name.it`: country name (Italian)
- `cowc`: Correlates of War character
- `cown`: Correlates of War numeric
- `dhs`: Demographic and Health Surveys Program
- `ecb`: European Central Bank
- `eurostat`: Eurostat
- `fao`: Food and Agriculture Organization of the United Nations numerical code
- `fips`: FIPS 10-4 (Federal Information Processing Standard)
- `gaul`: Global Administrative Unit Layers
- `genc2c`: GENC 2-letter code
- `genc3c`: GENC 3-letter code
- `genc3n`: GENC numeric code
- `gwc`: Gleditsch & Ward character
- `gwn`: Gleditsch & Ward numeric
- `imf`: International Monetary Fund
- `ioc`: International Olympic Committee
- `iso2c`: ISO-2 character
- `iso3c`: ISO-3 character
- `iso3n`: ISO-3 numeric
- `p5n`: Polity V numeric country code
- `p5c`: Polity V character country code
- `p4n`: Polity IV numeric country code
- `p4c`: Polity IV character country code

- un: United Nations M49 numeric codes
- unicode.symbol: Region subtag (often displayed as emoji flag)
- unhcr: United Nations High Commissioner for Refugees
- unpd: United Nations Procurement Division
- vdem: Varieties of Democracy (V-Dem version 8, April 2018)
- wb: World Bank (very similar but not identical to iso3c)
- wvs: World Values Survey numeric code

Destination only:

- cldr.*: 600+ country name variants from the UNICODE CLDR project (e.g., "cldr.short.en"). Inspect the cldr_examples data.frame for a full list of available country names and examples.
- ar5: IPCC's regional mapping used both in the Fifth Assessment Report (AR5) and for the Reference Concentration Pathways (RCP)
- continent: Continent as defined in the World Bank Development Indicators
- cow.name: Correlates of War country name
- currency: ISO 4217 currency name
- eurocontrol_pru: European Organisation for the Safety of Air Navigation
- eurocontrol_statfor: European Organisation for the Safety of Air Navigation
- eu28: Member states of the European Union (as of December 2015), without special territories
- icao.region: International Civil Aviation Organization region
- iso.name.en: ISO English short name
- iso.name.fr: ISO French short name
- iso4217c: ISO 4217 currency alphabetic code
- iso4217n: ISO 4217 currency numeric code
- p4.name: Polity IV country name
- region: 7 Regions as defined in the World Bank Development Indicators
- region23: 23 Regions as used to be in the World Bank Development Indicators (legacy)
- un.name.ar: United Nations Arabic country name
- un.name.en: United Nations English country name
- un.name.es: United Nations Spanish country name
- un.name.fr: United Nations French country name
- un.name.ru: United Nations Russian country name
- un.name.zh: United Nations Chinese country name
- un.region.name: United Nations region name
- un.region.code: United Nations region code
- un.regionintermediate.name: United Nations intermediate region name
- un.regionintermediate.code: United Nations intermediate region code
- un.regionsub.name: United Nations sub-region name
- un.regionsub.code: United Nations sub-region code
- unhcr.region: United Nations High Commissioner for Refugees region name
- wvs.name: World Values Survey numeric code country name

Note

The Correlates of War (cow) and Polity 4 (p4) project produce codes in country year format. Some countries go through political transitions that justify changing codes over time. When building a purely cross-sectional conversion dictionary, this forces us to make arbitrary choices with respect to some entities (e.g., Western Germany, Vietnam, Serbia). `countrycode` includes a reconciled dataset in panel format, `codelist_panel`. Instead of converting code, we recommend that users dealing with panel data "left-merge" their data into this panel dictionary.

`codelist_panel` *Country Code Translation Data Frame (Country-Year Panel)*

Description

A panel of country-year observations with various codes

Usage

```
codelist_panel
```

Format

data frame with codes as columns

`countrycode` *Convert Country Codes*

Description

Converts long country names into one of many different coding schemes. Translates from one scheme to another. Converts country name or coding scheme to the official short English country name. Creates a new variable with the name of the continent or region to which each country belongs.

Usage

```
countrycode(
  sourcevar,
  origin,
  destination,
  warn = TRUE,
  nomatch = NA,
  custom_dict = NULL,
  custom_match = NULL,
  origin_regex = NULL
)
```

Arguments

sourcevar	Vector which contains the codes or country names to be converted (character or factor)
origin	A string which identifies the coding scheme of origin (e.g., "iso3c"). See codelist for a list of available codes.
destination	A string or vector of strings which identify the coding scheme of destination (e.g., "iso3c" or c("cowc", "iso3c")). See codelist for a list of available codes. When users supply a vector of destination codes, they are used sequentially to fill in missing values not covered by the previous destination code in the vector.
warn	Prints unique elements from sourcevar for which no match was found
nomatch	When countrycode fails to find a match for the code of origin, it fills-in the destination vector with nomatch. The default behavior is to fill non-matching codes with NA. If nomatch = NULL, countrycode tries to use the origin vector to fill-in missing values in the destination vector. nomatch must be either NULL, of length 1, or of the same length as sourcevar.
custom_dict	A data frame which supplies a new dictionary to replace the built-in country code dictionary. Each column contains a different code and must include no duplicates. The data frame format should resemble codelist. Users can pre-assign attributes to this custom dictionary to affect behavior (see Examples section): • "origin.regex" attribute: a character vector with the names of columns containing regular expressions. • "origin.valid" attribute: a character vector with the names of columns that are accepted as valid origin codes.
custom_match	A named vector which supplies custom origin and destination matches that will supercede any matching default result. The name of each element will be used as the origin code, and the value of each element will be used as the destination code.
origin_regex	NULL or Logical: When using a custom dictionary, if TRUE then the origin codes will be matched as regex, if FALSE they will be matched exactly. When NULL, countrycode will behave as TRUE if the origin name is in the custom_dictionary's origin_regex attribute, and FALSE otherwise. See examples section below.

Note

For a complete description of available country codes and languages, please see the documentation for the codelist conversion dictionary.

Panel data (i.e., country-year) can pose particular problems when converting codes. For instance, some countries like Vietnam or Serbia go through political transitions that justify changing codes over time. This can pose problems when using codes from organizations like CoW or Polity IV, which produce codes in country-year format. Instead of converting codes using countrycode(), we recommend that users use the codelist_panel data.frame as a base into which they can merge their other data. This data.frame includes most relevant code, and is already "reconciled" to ensure

that each political unit is only represented by one row in any given year. From there, it is just a matter of using `merge()` to combine different datasets which use different codes.

Examples

```
library(countrycode)

# ISO to Correlates of War
countrycode(c('USA', 'DZA'), origin = 'iso3c', destination = 'cown')

# English to ISO
countrycode('Albania', origin = 'country.name', destination = 'iso3c')

# German to French
countrycode('Albanien', origin = 'country.name.de', destination = 'iso.name.fr')

# Using custom_match to supercede default codes
countrycode(c('United States', 'Algeria'), 'country.name', 'iso3c')
countrycode(c('United States', 'Algeria'), 'country.name', 'iso3c',
            custom_match = c('Algeria' = 'ALG'))

## Not run:
 # Download the dictionary of US states from Github
 state_dict <- "https://bit.ly/2ToSrFv"
 state_dict <- read.csv(state_dict)

 # The "state.regex" column includes regular expressions, so we set an attribute.
 attr(state_dict, "origin_regex") <- "state.regex"
 countrycode(c('AL', 'AK'), 'abbreviation', 'state',
             custom_dict = state_dict)
 countrycode(c('Alabama', 'North Dakota'), 'state.regex', 'state',
             custom_dict = state_dict)

## End(Not run)
```

`countryname` *Convert country names in any language to another name or code*

Description

Converts long country names in any language to one of many different country code schemes or country names. `countryname` does 2 passes on the data. First, it tries to detect variations of country names in many languages extracted from the Unicode Common Locale Data Repository. Second, it applies `countrycode`'s English regexes to try to match the remaining cases. Because it does two passes, `countryname` can sometimes produce ambiguous results, e.g., Saint Martin vs. Saint Martin (French Part). Users who need a "safer" option can use: `countrycode(x, "country.name", "country.name")` Note that the function works with non-ASCII characters. Please see the Github page for examples.

Usage

```
countryname(sourcevar, destination = "cldr.short.en", warn = TRUE)
```

Arguments

sourcevar	Vector which contains the codes or country names to be converted (character or factor)
destination	Coding scheme of destination (string such as "iso3c" enclosed in quotes ""): type ?codelist for a list of available codes.
warn	Prints unique elements from sourcevar for which no match was found

Examples

```
## Not run:
x <- c('Afaganisitani', 'Barbadas', 'Sverige', 'UK')
countryname(x)
countryname(x, destination = 'iso3c')

## End(Not run)
```

---

countryname_dict *A dataframe of alternative country names in many languages. Used internally by the* countryname *function.*

Description

A dataframe of alternative country names in many languages. Used internally by the countryname function.

Format

dataframe

---

get_dictionary *Get Custom Dictionaries*

Description

Download a custom dictionary to use in the custom_dict argument of countrycode()

Usage

```
get_dictionary(dictionary = NULL)
```

Arguments

dictionary	A character string that specifies the dictionary to be retrieved. It must be one of "global_burden_of_disease", "ch_cantons", "us_states", "exiobase3", "gtap10". If NULL, the function will print the list of available dictionaries. Default is NULL.

Value

If a valid dictionary is specified, the function will return that dictionary as a data.frame. If an invalid dictionary or no dictionary is specified, the function will stop and throw an error message.

Examples

```
## Not run:
cd <- get_dictionary("us_states")
countrycode::countrycode(c("MO", "MN"), origin = "state.abb", "state.name", custom_dict = cd)

## End(Not run)
```

---

guess_field *Guess the code/name of a vector*

Description

Users sometimes do not know what kind of code or field their data contain. This function tries to guess by comparing the similarity between a user-supplied vector and all the codes included in the countrycode dictionary.

Usage

```
guess_field(codes, min_similarity = 80)
```

Arguments

codes	a vector of country codes or country names
min_similarity	the function returns all field names where over than min_similarity% of codes are shared between the supplied vector and the countrycode dictionary.

Examples

```
# Guess ISO codes
guess_field(c('DZA', 'CAN', 'DEU'))

# Guess country names
guess_field(c('Guinea','Iran','Russia','North Korea',rep('Ivory Coast',50),'Scotland'))
```

Index