Package 'manystates'

October 14, 2025

Title Many Data on State and State-Like Actors in the International
System

Version 1.0.2

Date 2025-10-08

Description Comprehensively identifying states and state-like actors is difficult.
This package provides data on states and state-like entities in the international system across time.
The package combines and cross-references several existing datasets consistent with the aims and functions of the manydata package.
It also includes functions for identifying state references in text, and for generating fictional state names.

URL <https://globalgov.github.io/manystates/>

BugReports <https://github.com/globalgov/manystates/issues>

LazyData true

License CC BY 4.0

Depends R (>= 3.5.0), manydata

Imports knitr, purrr, stringi

Suggests pointblank, messydates, testthat (>= 3.0.0), rmarkdown

Encoding UTF-8

RoxygenNote 7.3.3

Config/Needs/check covr, lintr, spelling

Config/Needs/website pkgdown

Config/testthat/parallel true

Config/testthat/edition 3

Config/testthat/start-first code_states

VignetteBuilder knitr

NeedsCompilation no

Author James Hollway [cre, aut, ctb] (IHEID, ORCID:
<https://orcid.org/0000-0002-8361-9647>),
Bernhard Bieri [ctb] (IHEID, ORCID:

<https://orcid.org/0000-0001-5943-9059>),
   Mylan Evrard [ctb] (IHEID, ORCID:
    <https://orcid.org/0009-0009-4404-1078>),
   Esther Peev [ctb] (IHEID, ORCID:
    <https://orcid.org/0000-0002-9678-2777>),
   Henrique Sposito [ctb] (IHEID, ORCID:
    <https://orcid.org/0000-0003-3420-6085>),
   Jael Tan [ctb] (IHEID, ORCID: <https://orcid.org/0000-0002-6234-9764>)

Contents

code_states	*Code stateIDs from text*

Description

This function allows for contemporary and historical countries or states to be identified in text. It uses a regular expression (regex) to search for a number of common names and alternative spellings for each entity. The function returns either the three-letter abbreviation (an extended version of ISO-3166 alpha-3), or the name of the state. The function can also return multiple matches, where more than one country is mentioned in the text. Currently, the function can identify 500 entities. Updates, bug reports, and suggestions welcome.

Usage

```
code_states(text, code = TRUE, max_count = 1)
```

Arguments

text	A vector of text to search for country names within.
code	Logical whether the function should return the three-letter abbreviation (an extended version of ISO-3166 alpha-3), or the name of the state. For the complete list of entities and their search terms, run the function without an argument (i.e. code_states()). Updates and suggestions welcome.
max_count	Integer how many countries to search for in each element of the vector. Where more than one country is matched, the countries are returned as a set, i.e. in the format "{AUS,NZL}". By default max_count = 1, which will just return the first match.

Value

A character vector of the same length as `text`, with either the three-letter abbreviation (an extended version of ISO-3166 alpha-3), or the name of the state, or `NA` where no match was found. If `max_count > 1`, multiple matches are returned as a set, i.e. in the format "{AUS,NZL}". If the function is run without an argument, it returns a data frame with the complete list of entities and their search terms.

Examples

```
code_states(c("I went to England",
  "I come from Venezuela",
  "Did you know there was a Lunda Empire?",
  "I like both Australia and New Zealand"))
code_states(c("I went to England",
  "I come from Venezuela",
  "Did you know there was a Lunda Empire?",
  "I like both Australia and New Zealand"), max_count = 2)
```

---

generate_states   *Generate fictional country names*

Description

This function generates a vector of fictional country names. While the generated names are designed to resemble real country names, the results will not match (at least not exactly) country names from the library provided. Please note that the function is still *experimental*.

The names are generated using a Markov chain approach based on syllable patterns found in a library of real country names. The function generate_states() uses the syllabise_states() function to split existing country names into syllable-like units, providing special attention to common patterns in country names such as "land", "stan", "burg", and others. A transition matrix is then built from these syllable units, allowing for the generation of new names that mimic the structure and length of real country names. Checks are included to ensure that the generated names are unique, do not match any existing country names, and avoid certain uncommon patterns such as ending on a preposition.

If no library of country names is provided, the function defaults to using a comprehensive list of country names from the {manystates} package. However, users can supply their own list of country names to customize the generation process.

This function can be useful for creating fictional datasets for testing, illustrative, or pedagogical purposes. For example, it can be used in classroom exercises that rely on invented country names, such as in-class simulations of international relations or negotiation, role-playing scenarios, or mock data analysis tasks. Using fictional country names helps avoid any unintended bias or preconceptions associated with real countries. Or they can be used in creative writing or game design. The names might inspire fictional settings or entities in stories, games, or other creative works. Each name could inspire a unique culture, conflict, or mythology. Writers could use them to kickstart short stories, while game designers might build entire maps or quests around them.

Usage

```
generate_states(n = 10, countries = NULL)

syllabise_states(word)

syllabize_states(word)
```

Arguments

`n`	Integer number of country names to generate from a library of fictional country names. Default is 10.
`countries`	Optional string vector of country names to use as a library for generating fictional names.
`word`	One or more words (character vector) to split into syllable-like units.

Value

String vector of fictional country names

Examples

```
generate_states(12)
syllabise_states("Afghanistan")
syllabise_states("Saint Pierre and Miquelon")
```

---

states   *States datacube*

Description

The `manystates::states` datacube is a list containing 3 datasets: ISD, GW, and GGO. It is a work-in-progress, so please let us know if you have any comments or suggestions.

Usage

```
states
```

Format

ISD: A dataset with 499 observations and 36 variables: stateID, StateName, Begin, End, State-NameAlt, Latitude, Longitude, StartType, EndType, cowID, cowNR, ISD_Category, Region, Start_Am, EStart_Am, Declare, DecDate, Population, ..., VioEnd, and VioEnd_Am.

HUGGO: A dataset with observations and variables: .

GW: A dataset with 216 observations and 7 variables: stateID, StateName, Begin, End, State-NameAlt, cowID, and cowNR.

For more information and references to each of the datasets used, please use the `manydata::call_sources()` and `manydata::compare_dimensions()` functions.

Details

```
#> $ISD
#> ------------------------------------------------------------
#> |   Variable   |  Class   |  Obs  |  Missing  |  Miss %  |
#> ------------------------------------------------------------
#> |stateID       |character|   499|         0|        0|
#> |StateName     |character|   499|         0|        0|
#> |Begin         |mdate    |   282|       217|    43.49|
#> |End           |mdate    |   499|         0|        0|
#> |StateNameAlt  |character|   210|       289|    57.92|
#> |Latitude      |character|   343|       156|    31.26|
#> |Longitude     |character|   343|       156|    31.26|
#> |StartType     |numeric  |   337|       162|    32.46|
#> |EndType       |numeric  |   296|       203|    40.68|
#> |cowID         |character|   499|         0|        0|
#> |cowNR         |numeric  |   499|         0|        0|
#> |ISD_Category  |numeric  |   497|         2|      0.4|
#> |Region        |numeric  |   499|         0|        0|
#> |Start_Am      |numeric  |   499|         0|        0|
#> |EStart_Am     |numeric  |   232|       267|    53.51|
#> |Declare       |numeric  |   315|       184|    36.87|
#> |DecDate       |character|    72|       427|    85.57|
#> |Population    |character|   319|       180|    36.07|
#> |PopDate       |numeric  |   316|       183|    36.67|
#> |PopAm         |numeric  |   339|       160|    32.06|
#> |PopulationHigh|numeric  |   139|       360|    72.14|
#> |PopulationLow |numeric  |   108|       391|    78.36|
#> |StartType_Am  |numeric  |   339|       160|    32.06|
#> |StartSettle   |numeric  |   320|       179|    35.87|
#> |End_Am        |numeric  |   499|         0|        0|
#> |EndType_Am    |numeric  |   294|       205|    41.08|
#> |EndSettle     |numeric  |   282|       217|    43.49|
#> |Sovereignty_Am|numeric  |   499|         0|        0|
#> |EuroDip       |numeric  |   331|       168|    33.67|
#> |Borders       |numeric  |   332|       167|    33.47|
#> |Borders_Am    |numeric  |   342|       157|    31.46|
#> |Capital       |character|   284|       215|    43.09|
#> |VioStart      |numeric  |   318|       181|    36.27|
#> |VioStart_Am   |numeric  |   327|       172|    34.47|
#> |VioEnd        |numeric  |   292|       207|    41.48|
#> |VioEnd_Am     |numeric  |   297|       202|    40.48|
#> ------------------------------------------------------------
#>
#>
#> $GW
#> ---------------------------------------------------------
#> |  Variable  |  Class  |  Obs  |  Missing  |  Miss %  |
#> ---------------------------------------------------------
```

```
#> |stateID     |character|    216|          0|          0|
#> |StateName   |character|    216|          0|          0|
#> |Begin       |mdate    |    216|          0|          0|
#> |End         |mdate    |    216|          0|          0|
#> |StateNameAlt|character|     18|        198|      91.67|
#> |cowID       |character|    216|          0|          0|
#> |cowNR       |character|    216|          0|          0|
#> ----------------------------------------------------------
#>
#>
#> $GGO
#> ----------------------------------------------------------
#> |  Variable  |  Class  |  Obs  |  Missing  |  Miss %  |
#> ----------------------------------------------------------
#> |stateID     |character|    409|          0|          0|
#> |StateName   |character|    409|          0|          0|
#> |Capital     |character|    409|          0|          0|
#> |Begin       |mdate    |    409|          0|          0|
#> |End         |mdate    |    409|          0|          0|
#> |Latitude    |numeric  |    409|          0|          0|
#> |Longitude   |numeric  |    409|          0|          0|
#> |Region      |character|    409|          0|          0|
#> |StateNameAlt|character|     61|        348|      85.09|
#> |CapitalAlt  |character|      7|        402|      98.29|
#> |Coder       |character|    409|          0|          0|
#> |Comments    |character|     72|        337|       82.4|
#> |Source      |character|    136|        273|      66.75|
#> ----------------------------------------------------------
```

Mapping

GGO	*GW*	*ISD*
stateID		
Begin	Start	Start
End	Finish	Finish
StateName	Name of State	State.Name
cowID	Cow ID	COW.ID
cowNR	Cow NR.	COW.Nr

Source

- Griffiths, Ryan D., and Charles R. Butcher. 'Introducing the international system(s) dataset (ISD), 1816-2011'. International Interactions 39.5 (2013), pp. 748-768.

- Gleditsch, Kristian S., and Michael D. Ward. 'Interstate system membership: A revised list of the independent states since 1816'. International Interactions 25.4 (1999), pp. 393-413.

- Hollway, James, and Jael Tan. 'Global governance observations on states and state-like entities' (2025)

Index