

Package: necountries (via r-universe)

November 20, 2024

Version 0.1-2

Date 2024-11-20

Title Countries of the World

Depends R (>= 4.0.0)

Imports Rdpack, tibble, ggplot2, sf, dplyr, ggrepel, rlang, classInt,
stringr, magrittr, stringi

Description Based on Natural Earth
<<https://www.naturalearthdata.com/>>, a subset of countries can
easily be selected with their administrative boundaries, joined
with an external data frame and plotted as a thematic map.

License GPL (>=2)

Encoding UTF-8

URL <https://www.R-project.org>

RoxygenNote 7.3.1

Roxygen list(markdown = TRUE)

Author Yves Croissant [aut, cre]
(<https://orcid.org/0000-0002-4857-7736>>)

Maintainer Yves Croissant <yves.croissant@univ-reunion.fr>

LazyData true

NeedsCompilation no

Suggests knitr, tidyr, purrr, quarto

VignetteBuilder quarto

RdMacros Rdpack

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libicu-dev
libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://ycroissant.r-universe.dev>

RemoteUrl <https://github.com/ycroissant/necountries>

RemoteRef HEAD

RemoteSha b58a029f3de928aabe83ecfe8839bb5a6a8c5b91

Contents

countries	2
dplyr.methods	4
labels.countries	5
ne_countries	5
ne_towns	6
plot.countries	7
slave_trade	8
sp_solow	9
towns	10
utm	11
Index	12

countries	<i>Countries from naturalearth</i>
-----------	------------------------------------

Description

Select a set of countries; talking about countries, we mean either sovereign countries, parts of countries and dependencies, each of these categories being on each own row. A single or a set of countries can be obtained by indicating a set of names of, either countries, regions or subregions

Usage

```
countries(
  name = NA,
  part = FALSE,
  dependency = FALSE,
  indeterminate = FALSE,
  exclude = NULL,
  include = NULL,
  utm = FALSE,
  crs = NULL,
  towns = FALSE,
  capital = FALSE,
  lang = NULL,
  extend = 1,
  shift = FALSE,
  coastlines = TRUE
)
```

Arguments

name	a character vector that contains one or several countries, regions or subregions (mixing the two of the three categories will result as an error),
------	--

part	should the parts of the countries be included (eg Azsores for Portugal or Alaska for the United States of America),
dependency	should the dependencies of the countries be included (eg Greenland and the Faroe Islands for Denmark),
indeterminate	should the indeterminate territories be included,
exclude	an optional set of countries that should be excluded from the request,
include	an optional set of countries that should be included
utm	if TRUE, the geometry is transformed using the relevant utm projection, if an integer, the geometry is transformed using the utm projection for the zone indicated,
crs	a CRS
towns	if TRUE, a tibble containing the cities of the countries selected is returned as a "towns" attribute,
capital	if TRUE the tibble containing the cities of the countries selected will contain the capitals, whatever their size,
lang	the language for countries and towns, one of "en", "fr", "es", "it"
extend	a number ≥ 1 , extend the bounding box so that the background is larger than the initial bounding box and can be transformed correctly if utm transformation is required
shift	a boolean, if TRUE, <code>st_shift_longitude</code> is used
coastlines	a boolean, TRUE to get the background coastines

Value

an object of class `countries` which inherits from `sf` with the following columns:

- `id` the two letters identifier of the country,
- `type` either "main" (the main part of a sovereign country, the whole country for most of them) - `country` the name of the entity,
- `sovereign` the sovereign country the entity belongs to,
- `capital` the name of the capital of the country (NA for parts and dependencies) - `subregion` the name of the subregion (United Nations' definition)
- `pop` the population of the entity,
- `gdp` currently undocumented
- `wbreg` the name of the region (World Bank's definition)
- `region` the name of the region (United Nations' definition) Two attributes "type" and "towns"

Examples

```
countries("Western Europe")
```

Description

countries' objects inherits from sf, when a verb of dplyr is used, the returned object is of class sf; these methods return a countries object.

Usage

```
check_join(x, y, by = NULL, side = c("right", "both", "left"))

## S3 method for class 'countries'
select(.data, ...)

## S3 method for class 'countries'
left_join(
  x,
  y,
  by = NULL,
  copy = FALSE,
  suffix = c(".x", ".y"),
  ...,
  keep = NULL
)
```

Arguments

x, y, by, copy, suffix, keep	see <code>dplyr::left_join</code>
side	for the <code>check_join</code> function
.data	see <code>dplyr::select</code>
...	further arguments

Value

for the `select` and the `left_join` method, a data frame

labels.countries	<i>Compute a unique sf to optimize the position of labels</i>
------------------	---

Description

Displaying labels on a map is complicated because of serious overlapping problems. Labels for different kinds of entities can be put in a unique sf

Usage

```
## S3 method for class 'countries'
labels(object, ..., var)
```

Arguments

object	a countries object',
...	further arguments (currently unused),
var	a character vector indicating the entities that should be labeled, it can be country, capital and towns

Value

a sf containing:

- name the names of the entities,
- type the type of the entity (either country, capitalandtowns')
- point the coordinate of the points (obtained using st_point_on_surface for countries)

ne_countries	<i>Countries of natureearth</i>
--------------	---------------------------------

Description

A sf containing 299 countries (in a large sense), either the main parts of sovereign countries, parts or dependencies

Format

a sf containing

- iso2: two letters identifier of the country,
- iso3: two letters identifier of the country,
- type: either "main" (the main part of a sovereign country, the whole country for most of them), "part", "dependency" or "indeterminate"

- country: the name of the entity,
- sovereign: the sovereign country the entity belongs to,
- capital: the name of the capital of the country (NA for parts and dependencies)
- status: United Nations' status
- en, fr, de, es, it: the name of the country in different languages
- region: the name of the region (United Nations' definition)
- subregion: the name of the subregion (United Nations' definition)
- wbreion: the name of the region (World Bank's definition)
- pop: the population of the entity,
- gdp: currently undocumented
- economy: economic group
- income: income groupe
- polygon a geometry column containing the administrative borders
- point a geometry column containing the point coordinate of the capital

 ne_towns

Populated places of natureearth

Description

A sf containing 7342 cities

Format

a sf containing

- id: the id of the country,
- name: the name of the city,
- capital: a boolean, TRUE for a capital
- pop: the population of the city,
- point: a point sfc containing the coordinates of the city

plot.countries *Basic plot function for countries objects*

Description

As the plot method of `sf`, this function is intended to obtain quickly a map for a set of countries. Countries' boundaries are represented and filling can be used, some cities can also be represented and labels can be added. `ggplot` is used and should be used directly when more enhanced maps are required

Usage

```
## S3 method for class 'countries'
plot(
  x,
  ...,
  labels = NULL,
  fill = NULL,
  capital = NULL,
  centroid = NULL,
  towns = NULL,
  bks = NULL,
  n = 6,
  size = c(1, 10),
  shape = c(15, 16),
  style = NULL,
  palette = NULL,
  bw = FALSE,
  oceans = "lightblue",
  background = "lightgrey"
)
```

Arguments

<code>x</code>	a countries object,
<code>...</code>	further arguments (currently unused)
<code>labels</code>	a character vector containing the variables that should be labeled: country, capital and/or towns
<code>fill</code>	a variable use to fill countries' polygons
<code>capital, centroid, towns</code>	a variable associated with the shape or the size of points; for towns, the only available variable is "pop", it town is a boolean fixed points are used with a specific shape for capitals if shape is of length 2
<code>bks</code>	an optional vector of breaks in order to use a continuous variable for fill
<code>n</code>	the number of class (passed to <code>classIntervals</code>)

size	either an integer for a fixed size or a couple of integer to define the range of the sizes of the points
shape	either one integer for a fixed shape or a couple of integer to get a specific shape for capitals
style	the style (passed to <code>classIntervals</code>)
palette	the palette (selected in <code>scale_fill_brewer</code>)
bw	a boolean, if TRUE, a black and white map is produced
oceans, background	a character indicating the color for the oceans and the background

Value

a gg object.

Examples

```
we <- countries("Western Europe")
plot(we)
```

 slave_trade

Slave trade and economic development

Description

a cross-section of 52 countries from 2000

Format

a tibble containing:

- country: the country name
- region: one of 'north', 'east', 'central', 'south', 'west'
- disteq: distance from equator
- longitude: longitude
- area: area in thousands of km squared
- pop: average population during the slave trade period
- coastline: log coastlines divided by the country area
- island: island indicator
- islam: percent islamic
- colony: previous colonizator, one of 'none', 'uk', 'france', 'portugal', 'belgium', 'spain', 'germany', 'italy'
- legor: legal origin, one of 'french' and 'british'
- gdp: log real gdp per capita in 2000

- slaves: number of slaves
- slavesarea: number of slaves divided by the country area
- humidmax: average maximum humidity
- rainmin: lowest month rainfall
- lowtemp: average minimum temperature
- gold: log gold production per inhabitant
- oil: log oil production per inhabitant
- diamond: log diamonds production per inhabitant
- atlantic: distance to the atlantic ocean
- indian: distance to the indian ocean
- redsea: distance to the red sea
- sahara: distance to sahara

Source

Nathan Nunn's website <https://nathannunn.arts.ubc.ca/>

References

Adummy A (2024). "Not available." Failed to insert reference with key = NUNN:08 from package = 'necountries'. Possible cause — missing REFERENCES.bib in package 'necountries' or 'necountries' not installed..

sp_solow

Solow's growth model with spatial correlation

Description

a cross-section of 91 countries from 1995

Format

a tibble containing:

- name: the name of the country
- code: the id of the country
- gdp60: per capita gdp in 1960
- gdp95: per capita gdp in 1995
- saving: saving rate
- labgwth: growth rate of the labor force

Source

JAE data archive

References

Adummy A (2024). “Not available.” Failed to insert reference with key = ERTU:KOCH:07 from package = 'necountries'. Possible cause — missing REFERENCES.bib in package 'necountries' or 'necountries' not installed..

towns

Populated places from naturalearth

Description

Select a set of cities; the set can be defined using the id of the country, the fact that it is a capital and the size

Usage

```
towns(x, size = NULL, capital = FALSE, crs = NULL, shift = FALSE)
```

Arguments

x	a sf (typically computed using the <code>countries</code> function), or a character that is passed to <code>countries</code> ,
size	the minimum size of the cities that have to be retrieved (the default value is NULL and all the cities are retrieved)
capital	if TRUE always retrieve the capitals, even if their size is below the one specified using the <code>size</code> argument
crs	an optional crs which is passed to <code>st_transform</code>
shift	a boolean, if TRUE, <code>st_shift_longitude</code> is used

Value

a sf containing five columns:

- iso2: the id of the country,
- iso3: the id of the country,
- name: the name of the city,
- capital: a boolean, TRUE for a capital
- pop: the population of the city,
- point: a point sfc containing the coordinates of the city

Examples

```
we <- countries("Western Europe")
towns(we, size = 1E06, capital = TRUE)
```

utm	<i>Universal Transverse Mercator projection</i>
-----	---

Description

utm returns the relevant UTM crs (in the 'proj4string' form)

Usage

```
utm(x)
```

Arguments

x either an integer (from 1L to 60L) or a 'sf

Details

There is one utm projections for each of the 60 zones that divide the world. The zone can be indicated as an integer (ie 12L, and not 12) or can be computed from a sf object

Value

a character string
a character (a crs i, the 'proj4string' format)

Examples

```
we <- countries("Western Europe")  
utm(we)  
utm(32L)
```

Index

- * **datasets**
 - ne_countries, 5
 - ne_towns, 6
- * **dataset**
 - slave_trade, 8
 - sp_solow, 9
- check_join (dplyr.methods), 4
- countries, 2
- dplyr.methods, 4
- labels.countries, 5
- left_join.countries (dplyr.methods), 4
- ne_countries, 5
- ne_towns, 6
- plot.countries, 7
- select.countries (dplyr.methods), 4
- slave_trade, 8
- sp_solow, 9
- towns, 10
- utm, 11