

Package ‘rnaturalearth’

March 21, 2017

Title World Map Data from Natural Earth

Version 0.1.0

Description Facilitates mapping by making natural earth map data from <http://www.naturalearthdata.com/> more easily available to R users.

License MIT + file LICENSE

LazyData true

LazyDataCompression xz

URL <https://github.com/ropenscilabs/rnaturalearth>

BugReports <https://github.com/ropenscilabs/rnaturalearth/issues>

Additional_repositories <http://packages.ropensci.org>

Depends R (>= 3.1.1)

Imports sp (>= 1.0.15), utils (>= 3.2.3), sf (>= 0.3-4)

Suggests rgdal, knitr (>= 1.12.3), testthat (>= 0.9.1), httr (>= 1.1.0), devtools (>= 1.10.0), rnaturalearthdata (>= 0.1.0), rnaturalearthhires (>= 0.1.0), raster (>= 2.5-2), rgeos

VignetteBuilder knitr

RoxygenNote 6.0.1

NeedsCompilation no

Author Andy South [aut, cre]

Maintainer Andy South <southandy@gmail.com>

Repository CRAN

Date/Publication 2017-03-21 21:58:11 UTC

R topics documented:

check_data_exist	2
check_rnaturalearthdata	3
check_rnaturalearthhires	3
check_scale	4

countries	4
get_data	5
install_rnaturalearthdata	5
install_rnaturalearth hires	6
ne_as_sf	6
ne_coastline	7
ne_countries	7
ne_download	8
ne_file_name	10
ne_load	11
ne_states	12
rnaturalearth	13

Index	14
--------------	-----------

check_data_exist	<i>check whether the requested data exist on Natural Earth</i>
------------------	--

Description

checks from a list dependent on type, category and scale. If it returns FALSE the data may still exist on the website. Doesn't yet do checking on raster names because I found the naming convention too tricky.

Usage

```
check_data_exist(scale = 110, type, category = c("cultural", "physical",
"raster"))
```

Arguments

scale	scale of map to return, one of 110, 50, 10 or 'small', 'medium', 'large'
type	type of natural earth file to download one of 'countries', 'map_units', 'map_subunits', 'sovereignty', 'states' OR the portion of any natural earth vector url after the scale and before the . e.g. for 'ne_50m_urban_areas.zip' this would be 'urban_areas' OR the raster filename e.g. for 'MSR_50M.zip' this would be 'MSR_50M'
category	one of natural earth categories : 'cultural', 'physical', 'raster'

Value

TRUE or FALSE

Examples

```
check_data_exist( scale = 110, category = 'cultural', type = 'countries' )
# type not in list for this category
check_data_exist( scale = 110, category = 'physical', type = 'airports' )
# type in list but scale shows FALSE
check_data_exist( scale = 110, category = 'cultural', type = 'airports' )
```

check_rnaturalearthdata

Check whether to install rnaturalearthdata and install if necessary

Description

If the rnaturalearthdata package is not installed, install it from GitHub using devtools. If it is not up to date, reinstall it.

Usage

```
check_rnaturalearthdata()
```

check_rnaturalearththires

Check whether to install rnaturalearththires and install if necessary

Description

If the rnaturalearththires package is not installed, install it from GitHub using devtools. If it is not up to date, reinstall it.

Usage

```
check_rnaturalearththires()
```

check_scale	<i>check that this scale is present in Natural Earth</i>
-------------	--

Description

check name or numeric scale representations, return numeric one

Usage

```
check_scale(x)
```

Arguments

x scale of map to return, one of 110, 50, 10 or 'small', 'medium', 'large'

Value

integer scale of map

Examples

```
# commented out because not exported
# check_scale(110)
# check_scale("small")
```

countries	<i>world country polygons from Natural Earth</i>
-----------	--

Description

at 1:110m scale (small). Other data and resolutions are in the packages `rnaturalearthdata` and `rnaturalearth hires`.

Usage

```
countries110
```

Format

A `SpatialPolygonsDataFrame`

Slots

data A data frame with country attributes

Source

http://www.naturalearthdata.com/download/110m/cultural/ne_110m_admin_0_countries.zip

get_data	<i>Get data from within the package</i>
----------	---

Description

returns world country polygons at a specified scale, used by ne_countries()

Usage

```
get_data(scale = 110, type = c("countries", "map_units", "sovereignty",  
  "tiny_countries"))
```

Arguments

scale	scale of map to return, one of 110, 50, 10, 'small', 'medium', 'large'
type	country type, one of 'countries', 'map_units', 'sovereignty', 'tiny_countries'

Value

A SpatialPolygonsDataFrame object.

install_rnaturalearthdata	<i>Install the naturalearthdata package after checking with the user</i>
---------------------------	--

Description

Install the naturalearthdata package after checking with the user

Usage

```
install_rnaturalearthdata()
```

```
install_rnaturalearthhires
```

Install the naturalearthhires package after checking with the user

Description

Install the naturalearthhires package after checking with the user

Usage

```
install_rnaturalearthhires()
```

```
ne_as_sf
```

coerce return object to sf if option set

Description

coerce return object to sf if option set

Usage

```
ne_as_sf(x, returnclass = c("sp", "sf"))
```

Arguments

x	scale of map to return, one of 110, 50, 10 or 'small', 'medium', 'large'
returnclass	'sp' default or 'sf' for Simple Features #not exported

Value

an sf or sp object

ne_coastline	<i>Get natural earth world coastline</i>
--------------	--

Description

returns world coastline at specified scale

Usage

```
ne_coastline(scale = 110, returnclass = c("sp", "sf"))
```

Arguments

scale scale of map to return, one of 110, 50, 10 or 'small', 'medium', 'large'
returnclass 'sp' default or 'sf' for Simple Features

Value

SpatialLinesDataFrame or sf

Examples

```
if (requireNamespace("rnaturalearthdata")) {  
  sldf_coast <- ne_coastline()  
  
  if (require(sp)) {  
    plot(sldf_coast)  
  }  
}
```

ne_countries	<i>Get natural earth world country polygons</i>
--------------	---

Description

returns world country polygons at a specified scale, or points of tiny_countries

Usage

```
ne_countries(scale = 110, type = "countries", continent = NULL,  
  country = NULL, geounit = NULL, sovereignty = NULL,  
  returnclass = c("sp", "sf"))
```

Arguments

scale	scale of map to return, one of 110, 50, 10 or 'small', 'medium', 'large'
type	country type, one of 'countries', 'map_units', 'sovereignty', 'tiny_countries'
continent	a character vector of continent names to get countries from.
country	a character vector of country names.
geounit	a character vector of geounit names.
sovereignty	a character vector of sovereignty names.
returnclass	'sp' default or 'sf' for Simple Features

Value

SpatialPolygonsDataFrame, SpatialPointsDataFrame or sf

Examples

```

spdf_world <- ne_countries()
spdf_africa <- ne_countries(continent = 'africa')
spdf_france <- ne_countries(country = 'france')

if (require(sp)) {
  plot(spdf_world)
  plot(spdf_africa)
  plot(spdf_france)
}

# get as sf
if (require(sf)) {
  sf_world <- ne_countries(returnclass='sf')
  plot(sf_world)
}

if (require(rnaturalearthdata) & require(sp)) {
  spdf_tiny_countries <- ne_countries(type = 'tiny_countries', scale = 50)
  plot(spdf_tiny_countries)
}

```

ne_download

download data from Natural Earth and (optionally) read into R

Description

returns downloaded data as a spatial object or the filename if load=FALSE. if destdir is specified the data can be reloaded in a later R session using [ne_load](#) with the same arguments.

Usage

```
ne_download(scale = 110, type = "countries", category = c("cultural",
  "physical", "raster"), destdir = tempdir(), load = TRUE,
  returnclass = c("sp", "sf"))
```

Arguments

scale scale of map to return, one of 110, 50, 10 or 'small', 'medium', 'large'

type type of natural earth file to download one of 'countries', 'map_units', 'map_subunits', 'sovereignty', 'states' OR the portion of any natural earth vector url after the scale and before the . e.g. for 'ne_50m_urban_areas.zip' this would be 'urban_areas'. See Details. OR the raster filename e.g. for 'MSR_50M.zip' this would be 'MSR_50M'

category one of natural earth categories : 'cultural', 'physical', 'raster'

destdir where to save files, defaults to tempdir(), getwd() is also possible.

load TRUE/FALSE whether to load file into R and return

returnclass 'sp' default or 'sf' for Simple Features

Details

A non-exhaustive list of datasets available according to scale specified by the type param

	scale = 'small'	scale = 'medium'	scale = 'large'
category = 'physical', type = '[below]'			
coastline	y	y	y
land	y	y	y
ocean	y	y	y
rivers_lake_centerlines	y	y	y
lakes	y	y	y
glaciated_areas	y	y	y
antarctic_ice_shelves_polys	-	y	y
geographic_lines	y	y	y
graticules_1	y	y	y
graticules_30	y	y	y
wgs84_bounding_box	y	y	y
playas	-	y	y
minor_islands	-	-	y
reefs	-	-	y
category = 'cultural', type = '[below]'			
populated_places	y	y	y
boundary_lines_land	y	y	y
breakaway_disputed_areas	-	y	y
airports	-	y	y
ports	-	y	y
urban_areas	-	y	y
roads	-	-	y
railroads	-	-	y

Value

A Spatial object depending on the data (points, lines, polygons or raster), unless load=FALSE in which case it returns the name of the downloaded shapefile (without extension).

See Also

[ne_load](#), pre-downloaded data are available using [ne_countries](#), [ne_states](#). Other geographic data are available in the raster package : [getData](#).

Examples

```
## Not run:
spdf_world <- ne_download( scale = 110, type = 'countries' )

if (require(sp)) {
  plot(spdf_world)
  plot(ne_download(type = 'populated_places'))
}

# reloading from the saved file in the same session with same arguments
spdf_world2 <- ne_load( scale = 110, type = 'countries' )

# download followed by load from specified directory will work across sessions
spdf_world <- ne_download( scale = 110, type = 'countries', destdir = getwd() )
spdf_world2 <- ne_load( scale = 110, type = 'countries', destdir = getwd() )

# for raster, here an example with Manual Shaded Relief (MSR)
# download & load
rst <- ne_download(scale = 50, type = 'MSR_50M', category = 'raster', destdir = getwd())

# load after having downloaded
rst <- ne_load(scale = 50, type = 'MSR_50M', category = 'raster', destdir = getwd())

# plot
library(raster)
raster::plot(rst)

## End(Not run) # end dontrun
```

ne_file_name

return a natural earth filename based on arguments

Description

returns a string that can then be used to download the file.

Usage

```
ne_file_name(scale = 110, type = "countries", category = c("cultural",
  "physical", "raster"), full_url = FALSE)
```

Arguments

scale	scale of map to return, one of 110, 50, 10 or 'small', 'medium', 'large'
type	type of natural earth file to download one of 'countries', 'map_units', 'map_subunits', 'sovereignty', 'states' OR the portion of any natural earth vector url after the scale and before the . e.g. for 'ne_50m_urban_areas.zip' this would be 'urban_areas' OR the raster filename e.g. for 'MSR_50M.zip' this would be 'MSR_50M'
category	one of natural earth categories : 'cultural', 'physical', 'raster'
full_url	whether to return just the filename [default] or the full URL needed for download

Value

string

Examples

```
ne_name <- ne_file_name( scale = 110, type = 'countries' )
ne_url  <- ne_file_name( scale = 110, type = 'countries', full_url = TRUE )
```

ne_load	<i>load a Natural Earth vector that has already been downloaded to R using ne_download</i>
---------	--

Description

returns loaded data as a spatial object.

Usage

```
ne_load(scale = 110, type = "countries", category = c("cultural",
  "physical", "raster"), destdir = tempdir(), file_name = NULL,
  returnclass = c("sp", "sf"))
```

Arguments

scale	scale of map to return, one of 110, 50, 10 or 'small', 'medium', 'large'
type	type of natural earth file one of 'countries', 'map_units', 'map_subunits', 'sovereignty', 'states' OR the portion of any natural earth vector url after the scale and before the . e.g. for 'ne_50m_urban_areas.zip' this would be 'urban_areas' OR the raster filename e.g. for 'MSR_50M.zip' this would be 'MSR_50M'
category	one of natural earth categories : 'cultural', 'physical', 'raster'
destdir	folder to load files from, default=tempdir()
file_name	OPTIONAL name of file (excluding path) instead of natural earth attributes
returnclass	'sp' default or 'sf' for Simple Features

Value

A Spatial object depending on the data (points, lines, polygons or raster).

See Also

[ne_download](#)

Examples

```
## Not run:
# download followed by load from tempdir() works in same R session
spdf_world <- ne_download( scale = 110, type = 'countries' )
spdf_world2 <- ne_load( scale = 110, type = 'countries' )

# download followed by load from specified directory works between R sessions
spdf_world <- ne_download( scale = 110, type = 'countries', destdir = getwd() )
spdf_world2 <- ne_load( scale = 110, type = 'countries', destdir = getwd() )

# for raster
# download & load
rst <- ne_download(scale = 50, type = '0B_50M', category = 'raster', destdir = getwd())

# load after having downloaded
rst <- ne_load(scale = 50, type = '0B_50M', category = 'raster', destdir = getwd())

# plot
library(raster)
plot(rst)

## End(Not run) # end dontrun
```

ne_states

Get natural earth world state (admin level 1) polygons

Description

returns state polygons (administrative level 1) for specified countries

Usage

```
ne_states(country = NULL, geounit = NULL, iso_a2 = NULL, spdf = NULL,
  returnclass = c("sp", "sf"))
```

Arguments

country	a character vector of country names.
geounit	a character vector of geounit names.
iso_a2	a character vector of iso_a2 country codes
spdf	an optional alternative states map
returnclass	'sp' default or 'sf' for Simple Features

Value

SpatialPolygonsDataFrame or sf

Examples

```
# comparing using country and geounit to filter
if (requireNamespace("rnaturalearth") {
  spdf_france_country <- ne_states(country = 'france')
  spdf_france_geounit <- ne_states(geounit = 'france')
  if (require(sp)) {
    plot(spdf_france_country)
    plot(spdf_france_geounit)

    plot(ne_states(country = 'united kingdom'))
    plot(ne_states(geounit = 'england'))
  }
}
```

rnaturalearth

rnaturalearth : world map data from Natural Earth

Description

Facilitates world mapping by making **Natural Earth** map data more easily available to R users.

See Also

[ne_countries](#) [ne_states](#) [ne_download](#)

Index

*Topic **datasets**

- countries, [4](#)

- check_data_exist, [2](#)
- check_rnaturalearthdata, [3](#)
- check_rnaturalearth hires, [3](#)
- check_scale, [4](#)
- countries, [4](#)
- countries110 (countries), [4](#)

- get_data, [5](#)
- getData, [10](#)

- install_rnaturalearthdata, [5](#)
- install_rnaturalearth hires, [6](#)

- ne_admin0 (ne_countries), [7](#)
- ne_admin1 (ne_states), [12](#)
- ne_as_sf, [6](#)
- ne_coastline, [7](#)
- ne_countries, [7](#), [10](#), [13](#)
- ne_download, [8](#), [11–13](#)
- ne_file_name, [10](#)
- ne_load, [8](#), [10](#), [11](#)
- ne_states, [10](#), [12](#), [13](#)

- rnaturalearth, [13](#)
- rnaturalearth-package (rnaturalearth),
[13](#)