

# Package ‘rnaturalearth’

December 15, 2023

**Title** World Map Data from Natural Earth

**Version** 1.0.1

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

**License** MIT + file LICENSE

**LazyData** true

**LazyDataCompression** xz

**Encoding** UTF-8

**URL** <https://docs.ropensci.org/rnaturalearth/>,  
<https://github.com/ropensci/rnaturalearth>,  
<https://docs.ropensci.org/rnaturalearthhires/>

**BugReports** <https://github.com/ropensci/rnaturalearth/issues>

**Additional\_repositories** <http://packages.ropensci.org>

**Depends** R (>= 3.1.1)

**Imports** httr (>= 1.1.0), jsonlite, sf (>= 0.3-4), terra, utils (>= 3.2.3)

**Suggests** devtools (>= 1.10.0), dplyr, ggplot2, ggrepel, knitr (>= 1.12.3), lifecycle (>= 1.0.3), pbapply, purrr, rlang, rmarkdown, rnaturalearthdata (>= 0.1.0), rnaturalearthhires (>= 0.1.0), testthat (>= 0.9.1), tmap

**Config/Needs/website** ropensci/rnaturalearthhires

**VignetteBuilder** knitr

**RoxygenNote** 7.2.3

**NeedsCompilation** no

**Author** Philippe Massicotte [aut, cre]  
(<https://orcid.org/0000-0002-5919-4116>),  
Andy South [aut],  
Koen Hufkens [ctb] (<https://orcid.org/0000-0002-5070-8109>)

**Maintainer** Philippe Massicotte <[pmassicotte@hotmail.com](mailto:pmassicotte@hotmail.com)>

Repository CRAN

Date/Publication 2023-12-15 11:00:02 UTC

## R topics documented:

check_data_exist . . . . .	2
check_rnaturalearthdata . . . . .	3
check_rnaturalearthhires . . . . .	4
check_scale . . . . .	4
convert_spatial_class . . . . .	5
countries . . . . .	6
df_layers_cultural . . . . .	7
df_layers_physical . . . . .	7
get_data . . . . .	8
install_rnaturalearthdata . . . . .	9
install_rnaturalearthhires . . . . .	9
ne_coastline . . . . .	10
ne_countries . . . . .	11
ne_download . . . . .	12
ne_file_name . . . . .	14
ne_find_vector_data . . . . .	15
ne_load . . . . .	16
ne_states . . . . .	17

<b>Index</b>	<b>19</b>
--------------	-----------

---

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	The 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'

**Details**

Note that the filename of the requested object will be returned if 'load = FALSE'.

**Value**

TRUE or FALSE

**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**

```
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\_rnaturalearth hires

*Check whether to install rnaturalearth hires and install if necessary*

---

### **Description**

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

### **Usage**

check\_rnaturalearth hires()

---

check\_scale

*check that this scale is present in Natural Earth*

---

### **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

---

convert\_spatial\_class *Convert from/to sf/sv objects*

---

### 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

```
convert_spatial_class(x, returnclass = c("sf", "sv"))
```

### Arguments

x	Object to be converted
returnclass	A string determining the spatial object to return. Either "sf" for for simple feature (from 'sf', the default) or "sv" for a 'SpatVector' (from 'terra').

### Details

Note that the filename of the requested object will be returned if 'load = FALSE'.

### Value

Object of class "sf" or "sv"

### 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")

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(terra)
terra::plot(rst)
# end dontrun

## End(Not run)
```

---

countries

*world country polygons from Natural Earth*

---

### Description

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

### Usage

```
countries110
```

### Format

A `sf` object.

An object of class `sf` (inherits from `data.frame`) with 177 rows and 169 columns.

### Slots

`data` A data frame with country attributes.

### Source

[https://www.naturalearthdata.com/http://www.naturalearthdata.com/download/10m/cultural/ne\\_10m\\_admin\\_0\\_countries.zip](https://www.naturalearthdata.com/http://www.naturalearthdata.com/download/10m/cultural/ne_10m_admin_0_countries.zip)

---

*df\_layers\_cultural*      *list of cultural layers available from Natural Earth*

---

**Description**

list of cultural layers available from Natural Earth

**Usage**

`df_layers_cultural`

**Format**

A DataFrame

An object of class `data.frame` with 43 rows and 4 columns.

---

*df\_layers\_physical*      *list of physical layers available from Natural Earth*

---

**Description**

list of physical layers available from Natural Earth

**Usage**

`df_layers_physical`

**Format**

A DataFrame

An object of class `data.frame` with 29 rows and 4 columns.

---

`get_data`*Get data from within the package*

---

**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**

<code>scale</code>	The scale of map to return, one of '110', '50', '10' or 'small', 'medium', 'large'.
<code>type</code>	country type, one of 'countries', 'map_units', 'sovereignty', 'tiny_countries'

**Details**

Note that the filename of the requested object will be returned if `'load = FALSE'`.

**Value**

A sf object.

**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")  
  
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(terra)
terra::plot(rst)
# end dontrun

## End(Not run)
```

---

install\_rnaturalearthdata

*Install the naturalearthdata package after checking with the user*

---

### **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_coastline	<i>Get natural earth world coastline</i>
--------------	--

---

### Description

returns world coastline at specified scale

### Usage

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

### Arguments

scale	The scale of map to return, one of '110', '50', '10' or 'small', 'medium', 'large'.
returnclass	A string determining the spatial object to return. Either "sf" for for simple feature (from 'sf', the default) or "sv" for a 'SpatVector' (from 'terra').

### Details

Note that the filename of the requested object will be returned if 'load = FALSE'.

### Value

An object of class 'sf' for simple feature (from 'sf', the default) or 'SpatVector' (from 'terra').

### 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

```
if (requireNamespace("rnaturalearthdata")) {  
  coast <- ne_coastline()  
  plot(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("sf", "sv")
)
```

## Arguments

scale	The 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	A string determining the spatial object to return. Either "sf" for for simple feature (from 'sf', the default) or "sv" for a 'SpatVector' (from 'terra').

## Details

Note that the filename of the requested object will be returned if 'load = FALSE'.

## Value

An object of class 'sf' for simple feature (from 'sf', the default) or 'SpatVector' (from 'terra').

## 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**

```

world <- ne_countries()
africa <- ne_countries(continent = "africa")
france <- ne_countries(country = "france")

plot(world$geometry)
plot(africa$geometry)
plot(france$geometry)

# get as SpatVector
world <- ne_countries(returnclass = "sv")
terra::plot(world)

tiny_countries <- ne_countries(type = "tiny_countries", scale = 50)
plot(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("sf", "sv")
)

```

**Arguments**

scale	The 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 <code>tempdir()</code> , <code>getwd()</code> is also possible.

load	'TRUE' load the spatial object into R, 'FALSE' return the filename of the downloaded object.
returnclass	A string determining the spatial object to return. Either "sf" for for simple feature (from 'sf', the default) or "sv" for a 'SpatVector' (from 'terra').

### Details

Note that the filename of the requested object will be returned if 'load = FALSE'.

### Value

An object of class 'sf' for simple feature (from 'sf', the default) or 'SpatVector' (from 'terra').

### 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")

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(terra)
terra::plot(rst)
# end dontrun

## End(Not run)
```

---

ne_file_name	<i>return a natural earth filename based on arguments</i>
--------------	---

---

### 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	The 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

### Details

Note that the filename of the requested object will be returned if 'load = FALSE'.

### Value

string

### 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

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

---

ne\_find\_vector\_data     *Return a dataframe of available vector layers on Natural Earth*

---

### Description

Checks the Natural Earth Github repository for current vector layers and provides the file name required in the type argument of ne\_download.

### Usage

```
ne_find_vector_data(  
  scale = 110,  
  category = c("cultural", "physical"),  
  getmeta = FALSE  
)
```

### Arguments

scale	The scale of map to return, one of '110', '50', '10' or 'small', 'medium', 'large'.
category	one of natural earth categories : 'cultural', 'physical'
getmeta	whether to get url of the metadata for each layer

### Details

Note that the filename of the requested object will be returned if 'load = FALSE'.

### Value

dataframe with two variables: layer and metadata

### 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:  
ne_find_vector_data(scale = 10, category = "physical")  
  
## End(Not run)
```

---

ne_load	<i>load a Natural Earth vector that has already been downloaded to R using <a href="#">ne_download</a></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("sf", "sv")
)
```

### Arguments

scale	The 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	A string determining the spatial object to return. Either "sf" for for simple feature (from 'sf', the default) or "sv" for a 'SpatVector' (from 'terra').

### Details

Note that the filename of the requested object will be returned if 'load = FALSE'.

### Value

An object of class 'sf' for simple feature (from 'sf', the default) or 'SpatVector' (from 'terra').

### 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(terra)
plot(rst)
# end dontrun

## End(Not run)
```

---

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,
  spat_object = NULL,
  returnclass = c("sf", "sv")
)
```

**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
spat_object	an optional alternative states map
returnclass	A string determining the spatial object to return. Either "sf" for for simple feature (from 'sf', the default) or "sv" for a 'SpatVector' (from 'terra').

**Details**

Note that the filename of the requested object will be returned if 'load = FALSE'.

**Value**

An object of class 'sf' for simple feature (from 'sf', the default) or 'SpatVector' (from 'terra').

**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**

```
# comparing using country and geounit to filter
if (requireNamespace("rnaturalearthhires")) {
  spdf_france_country <- ne_states(country = "france")
  spdf_france_geounit <- ne_states(geounit = "france")

  plot(spdf_france_country)
  plot(spdf_france_geounit)

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

# Index

## \* datasets

- countries, [6](#)
- df\_layers\_cultural, [7](#)
- df\_layers\_physical, [7](#)

check\_data\_exist, [2](#)

check\_rnaturalearthdata, [3](#)

check\_rnaturalearthhires, [4](#)

check\_scale, [4](#)

convert\_spatial\_class, [5](#)

countries, [6](#)

countries110 (countries), [6](#)

df\_layers\_cultural, [7](#)

df\_layers\_physical, [7](#)

get\_data, [8](#)

getData, [3](#), [5](#), [8](#), [10](#), [11](#), [13–15](#), [18](#)

install\_rnaturalearthdata, [9](#)

install\_rnaturalearthhires, [9](#)

ne\_admin0 (ne\_countries), [11](#)

ne\_admin1 (ne\_states), [17](#)

ne\_coastline, [10](#)

ne\_countries, [3](#), [5](#), [8](#), [10](#), [11](#), [11](#), [13–15](#), [18](#)

ne\_download, [12](#), [16](#)

ne\_file\_name, [14](#)

ne\_find\_vector\_data, [15](#)

ne\_load, [3](#), [5](#), [8](#), [10–15](#), [16](#), [18](#)

ne\_states, [3](#), [5](#), [8](#), [10](#), [11](#), [13–15](#), [17](#), [18](#)