

# Package: tidycountries (via r-universe)

October 28, 2024

**Type** Package

**Title** Access and Manipulate Comprehensive Country Level Data in Tidy Format

**Version** 0.1.0

**Description** A comprehensive and user-friendly interface for accessing, manipulating, and analyzing country-level data from around the world. It allows users to retrieve detailed information on countries, including names, regions, continents, populations, currencies, calling codes, and more, all in a tidy data format. The package is designed to work seamlessly within the 'tidyverse' ecosystem, making it easy to filter, arrange, and visualize country-level data in R.

**License** MIT + file LICENSE

**URL** <https://github.com/denironyx/tidycountries>

**BugReports** <https://github.com/denironyx/tidycountries/issues>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**Depends** R (>= 3.5.0)

**Imports** dplyr, stringr, sf

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Config/Needs/website** rmarkdown

**NeedsCompilation** no

**Author** Dennis Irorere [aut, cre, cph]

**Maintainer** Dennis Irorere <denironyx@gmail.com>

**Repository** CRAN

**Date/Publication** 2024-10-27 04:20:02 UTC

## Contents

get_countries_by_currency . . . . .	2
get_countries_by_region . . . . .	3
get_country_by_calling_code . . . . .	4
get_country_info . . . . .	5
restcountries_tidy_data . . . . .	6
world_administrative_boundaries . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

get\_countries\_by\_currency  
*get\_countries\_by\_currency*

---

### Description

This function retrieves a list of countries where a specified currency is used. The function is case-insensitive and matches the currency name or part of the name. The output is ordered alphabetically by country name.

### Usage

```
get_countries_by_currency(currency_input, geometry = FALSE)
```

### Arguments

**currency\_input** A character string representing the currency name or part of the name. The input is case-insensitive.

**geometry** A logical value indicating whether to include geographic boundary data. Defaults to FALSE. If TRUE, the function performs a left join with boundary data and returns a spatial sf data frame.

### Value

A data frame containing the list of countries that use the specified currency, ordered alphabetically by country name. The columns include country codes (CCA2 and CCA3), common name, capital, continents, currency name, currency symbol, latitude, and longitude. If `geometry = TRUE`, an additional column for geographic boundaries is included.

### Note

The function utilizes the pre-loaded `restcountries_data` dataset. Ensure that this dataset is loaded before invoking the function. The function uses a case-insensitive regular expression to match the currency name, allowing partial matches. If `geometry = TRUE`, the function joins with the `world_administrative_boundaries` dataset, which must also be pre-loaded.

## Examples

```
# Example usage: Find all countries that use the Euro
euro_countries <- get_countries_by_currency("Euro")
print(euro_countries)

# Example usage: Find all countries that use a currency with "dollar" in its name
dollar_countries <- get_countries_by_currency("dollar", geometry = TRUE)
print(dollar_countries)

# Example usage: Find all countries that use the Yen
yen_countries <- get_countries_by_currency("Yen")
print(yen_countries)
```

---

```
get_countries_by_region
      get_countries_by_region
```

---

## Description

This function retrieves a list of countries based on a specified region, subregion, or continent. The function is case-insensitive and orders the countries alphabetically by their common names. If the input does not match any region, subregion, or continent, the function provides a list of all available regions, subregions, and continents.

## Usage

```
get_countries_by_region(country_region_value, geometry = FALSE)
```

## Arguments

country_region_value	A character string representing the region, subregion, or continent. The input is case-insensitive.
geometry	Logical. If TRUE, includes spatial geometry data for each country (boundaries). Defaults to FALSE. When TRUE, an additional column for geographic boundaries is included.

## Value

A data frame with information on countries within the specified region, subregion, or continent. If `geometry = TRUE`, the result includes a geometry column with boundary data. If no match is found, a message lists all available regions, subregions, and continents.

## Note

This function returns relevant information on countries in a specified region. When `geometry = TRUE`, it returns an `sf` object, including spatial data.

## Examples

```
# Example usage: Get a list of countries in Africa
africa_countries <- get_countries_by_region("Africa")
print(africa_countries)

# Example usage: Get countries in a specific continent with geometry included
western_europe_countries <- get_countries_by_region("Western Europe", geometry = TRUE)
print(western_europe_countries)

# Example usage: Get a list of countries in the continent of Asia
asia_countries <- get_countries_by_region("Asia")
print(asia_countries)
```

---

```
get_country_by_calling_code
  get_country_by_calling_code
```

---

## Description

This function retrieves information about countries based on a specified calling code or part of it. The input can be a root calling code, suffix, or a full calling code, and the function is case-insensitive.

## Usage

```
get_country_by_calling_code(call_code, geometry = FALSE)
```

## Arguments

<code>call_code</code>	A character string representing the calling code, root calling code, or suffix. The input is case-insensitive.
<code>geometry</code>	A logical value indicating whether to include geographic boundary data. Defaults to FALSE. If TRUE, the function performs a left join with boundary data and returns a spatial sf data frame.

## Value

A data frame containing the information of countries corresponding to the specified calling code, root, or suffixes. If `geometry = TRUE`, the result will include an additional column for geographic boundaries (as spatial features).

## Note

The function utilizes the pre-loaded `restcountries_data` dataset. Ensure that this dataset is loaded before invoking the function. If `geometry = TRUE`, the function joins with the `world_administrative_boundaries` dataset, which must also be pre-loaded.

## Examples

```
# Example usage: Find country information by root calling code
us_info <- get_country_by_calling_code("+1")
print(us_info)

# Example usage: Find country information by calling code suffix and include geometry
uk_info <- get_country_by_calling_code("44", geometry = TRUE)
print(uk_info)

# Example usage: Find country information by full calling code
india_info <- get_country_by_calling_code("+91")
print(india_info)
```

---

get\_country\_info      *get\_country\_info*

---

## Description

This function retrieves information about a specific country based on its country code (cca2 or cca3) or common name. The function is case-insensitive and provides a comprehensive overview of the selected country. If "all" is passed as the input, it returns data for all countries. If the input does not match any country, the function returns a list of all available country names.

## Usage

```
get_country_info(country_value, geometry = FALSE)
```

## Arguments

country_value	A character string representing the country code (cca2 or cca3) or common name. The input is case-insensitive. If "all" is passed, the function returns data for all countries.
geometry	Logical. If TRUE, includes spatial geometry data for the country (boundaries). Defaults to FALSE. When TRUE, an additional column for geographic boundaries is included.

## Value

A data frame with selected country information. If geometry = TRUE, the result includes a geometry column with boundary data, making the returned object ready to be converted to an sf (simple features) data frame for spatial analysis. If the input is "all", it returns data for all countries. If no match is found, a list of all available country names is printed.

## Note

The returned data frame includes relevant country details. If geometry = TRUE, an additional column for geographic boundaries is included.

## Examples

```
# Examples usage: Get information for Nigeria
nigeria_info <- get_country_info("Nigeria")
print(nigeria_info)

# Example usage: Get information for a country using it's cca2 code
usa_info <- get_country_info("US")
print(usa_info)
```

---

```
restcountries_tidy_data
  restcountries_tidy_data
```

---

## Description

A dataset containing tidied information about countries from the Restcountries API.

## Usage

```
restcountries_tidy_data
```

## Format

A data frame with several rows and the following columns:

**tld** Top-level domain(s) associated with the country.  
**common\_name** Common name of the country.  
**official\_name** Official name of the country.  
**cca2** Country code (2-letter).  
**cca3** Country code (3-letter).  
**fifa** FIFA code of the country.  
**independent** Independence status (TRUE/FALSE).  
**status** Country status (e.g., officially assigned).  
**un\_member** Whether the country is a UN member (TRUE/FALSE).  
**region** Geographic region.  
**subregion** Subregion.  
**population** Population of the country.  
**capital** Capital city of the country.  
**capital\_lat** Latitude of the capital city.  
**capital\_lon** Longitude of the capital city.  
**continents** Continent(s) the country is part of.  
**lat** Latitude of the country.

**lon** Longitude of the country.

**landlocked** Whether the country is landlocked (TRUE/FALSE).

**borders** Countries that share a border.

**area** Total area of the country in square kilometers.

**start\_of\_week** Day the week starts (e.g., Monday).

**timezones** Timezones applicable to the country.

**root** Root of the country calling code.

**suffixes** Suffixes of the country calling code.

**car\_side** Which side of the road cars drive on.

**googlemaps** Google Maps link for the country.

**openstreetmaps** OpenStreetMap link for the country.

**flags\_png** URL to PNG image of the country flag.

**flags\_svg** URL to SVG image of the country flag.

**flags\_alt** Alternative text for the country flag.

**currencies** Currencies used in the country.

**languages** Languages spoken in the country.

**currency\_name** Name of the primary currency used.

**currency\_symbol** Symbol of the primary currency used.

**calling\_code** Calling code(s) associated with the country.

## Details

This dataset includes a variety of country-level data such as country codes, names, capitals, regions, subregions, continents, currencies, population, geographic coordinates, languages, and more.

## Source

Data obtained from the Restcountries Json file and processed for use in this package.

## Examples

```
# Load the dataset and view the first few rows
data(restcountries_tidy_data)
head(restcountries_tidy_data)
```

---

```
world_administrative_boundaries  
  world_administrative_boundaries
```

---

**Description**

A dataset containing information about world administrative boundaries.

**Usage**

```
world_administrative_boundaries
```

**Format**

A data frame with several rows and the following columns:

**cca3** Country code (3-letter).

**geometry** geographic boundaries

**Details**

This dataset includes country code and the geometries

**Source**

Data obtained from the World Food Programme (UN agency) processed for use in this package.

**Examples**

```
# Load the dataset and view the first few rows  
data(world_administrative_boundaries)  
head(world_administrative_boundaries)
```



# Index

## \* datasets

restcountries\_tidy\_data, [6](#)

world\_administrative\_boundaries, [8](#)

[get\\_countries\\_by\\_currency, 2](#)

[get\\_countries\\_by\\_region, 3](#)

[get\\_country\\_by\\_calling\\_code, 4](#)

[get\\_country\\_info, 5](#)

[restcountries\\_tidy\\_data, 6](#)

[world\\_administrative\\_boundaries, 8](#)