

# Package ‘WorldMapR’

January 28, 2025

**Type** Package

**Title** Worldwide or Coordinates-Based Heat Maps

**Version** 1.1.0

**Description** Easily plot heat maps of the world, based on continuous or categorical data. Country labels can also be added to the map.

**License** GPL-3

**URL** <https://github.com/Luigi-Annic/WorldMapR/>

**BugReports** <https://github.com/Luigi-Annic/WorldMapR/issues>

**Encoding** UTF-8

**Depends** R (>= 4.3.0)

**Imports** ggplot2 (>= 3.4.4), dplyr (>= 1.1.4), rnaturalearth (>= 1.0.1), sf (>= 1.0-14), countrycode (>= 1.5.0), utils (>= 4.3.0), ggfx (>= 1.0.1)

**LazyData** true

**RoxygenNote** 7.2.3

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0), rnaturalearthdata (>= 1.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Luigi Annicchiarico [cre, aut]

**Maintainer** Luigi Annicchiarico <luigi.annic@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-01-28 13:10:01 UTC

## Contents

countrycoord_data . . . . .	2
geometries_data . . . . .	3
testdata1 . . . . .	3
testdata1b . . . . .	4
testdata1c . . . . .	4
worldplot . . . . .	5
worldplotCat . . . . .	6

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

---

countrycoord_data	<i>countrycoord_data</i>
-------------------	--------------------------

---

## Description

This function generates a data frame with information about the coordinates of the central point for each country of interest. You can choose whether to keep all the countries or only a subset.

## Usage

```
countrycoord_data(
  countries.list = NULL,
  crs = 4326,
  UK_as_GB = TRUE,
  exclude.iso.na = TRUE
)
```

## Arguments

<code>countries.list</code>	List of the ISO 3166-1 alpha-2 codes of countries that are to be included. By default it is set to NULL and all countries are included.
<code>crs</code>	Coordinate reference system (EPSG). By default the value is 4326, which corresponds to EPSG::4326 (WGS84)
<code>UK_as_GB</code>	Do you want to translate the GB isoa2 code to UK? If FALSE, GB is returned in the output data.frame. If TRUE, UK is returned. Note that you will need to provide GB as the input for United Kingdom, even if you want the UK label to be returned in output.
<code>exclude.iso.na</code>	if TRUE (default), countries that do not have a ISO 3166 code are excluded from the table.

## Value

an object of class `data.frame`

**Examples**

```
countrycoord_data(countries.list = c("IT", "FR", "SE"), crs = 3035)
countrycoord_data(countries.list = c("IT", "FR", "SE"), crs = 3035)
countrycoord_data(countries.list = c("IT", "FR", "SE", "GB"), crs = 3035, UK_as_GB = TRUE)
countrycoord_data(countries.list = c("IT", "FR", "SE", "GB"), crs = 3035, UK_as_GB = FALSE)
```

---

geometries_data	<i>geometries_data</i>
-----------------	------------------------

---

**Description**

This function generates a data frame with information about geometries and centroid coordinates of countries. You can choose whether to keep all the countries or only a subset.

**Usage**

```
geometries_data(exclude.iso.na = TRUE, countries.list = NULL)
```

**Arguments**

`exclude.iso.na` if TRUE (default), countries that do not have a ISO 3166 code are excluded from the table.

`countries.list` List of the ISO 3166-1 alpha-2 codes of countries that are to be included. By default it is set to NULL and all countries are included.

**Value**

an object of class `data.frame` and `sf`.

**Examples**

```
geometries_data(countries.list = c("IT", "FR", "US"))
```

---

testdata1	<i>Simulated data set 1</i>
-----------	-----------------------------

---

**Description**

Data from a random simulation with continuous data.

**Usage**

```
data(testdata1)
```

**Format**

An object of class `data.frame`

**Examples**

```
data(testdata1)
head(testdata1)
```

---

testdata1b

*Simulated data set 1b*

---

**Description**

Data from a random simulation with continuous and categorical data.

**Usage**

```
data(testdata1b)
```

**Format**

An object of class `data.frame`

**Examples**

```
data(testdata1b)
head(testdata1b)
```

---

testdata1c

*Simulated data set 1c*

---

**Description**

Data from a random simulation with continuous and categorical data. This data set contains information about 237 countries (countries without unique ISO 3166 code are excluded).

**Usage**

```
data(testdata1c)
```

**Format**

An object of class `data.frame`

**Examples**

```
data(testdata1c)
head(testdata1c)
```

---

worldplot	<i>worldplot</i>
-----------	------------------

---

## Description

Plot a world heat map based on a continuous variable.

## Usage

```
worldplot(
  data,
  ColName,
  CountryName,
  CountryNameType = "isoa2",
  rangeVal,
  longitude = c(-180, 180),
  latitude = c(-90, 90),
  crs = 4326,
  title = "",
  legendTitle = as.character(ColName),
  legend.position = "right",
  annotate = FALSE,
  div = 1,
  palette_option = "D",
  label.color = "white",
  label.size = 2,
  na_colour = "grey80",
  transform_limits = TRUE
)
```

## Arguments

<code>data</code>	Data set containing the list of nations and the variable that we want to plot.
<code>ColName</code>	Character variable with the name of the variable of interest.
<code>CountryName</code>	Character variable with the name of the country names column.
<code>CountryNameType</code>	Character variable with the coding for <code>CountryName</code> . One of <code>isoa2</code> (default, standing for ISO 3166-1 alpha-2 code), <code>isoa3</code> , or <code>name</code> .
<code>rangeVal</code>	Limit values (minimum and maximum) that are to be defined for the map. If not specified, the minimum and maximum are taken, and a message is displayed.
<code>longitude</code>	Longitude limits. Default is <code>c(-180, 180)</code> (whole world with crs as EPSG::4326).
<code>latitude</code>	Latitude limits. Default is <code>c(-90, 90)</code> (whole world with crs as EPSG::4326).
<code>crs</code>	Coordinate reference system (EPSG). By default the value is 4326, which corresponds to EPSG::4326 (WGS84)
<code>title</code>	Title of the plot. Default is no title.

legendTitle	Title of the legend. Default is the name of the filling variable.
legend.position	Position of the legend. If set to "none", no legend is displayed
annotate	Do you want to plot country labels (ISO 3166-1 alpha-2 code) on the map? Default is set to FALSE.
div	Parameter for modifying the elements dimensions in the map. Usually, it does not need to be modified. Default value is 1.
palette_option	Character string indicating the palette to be used. Available options range between "A" and "H".
label.color	Color of the labels if annotate = TRUE. Default is white
label.size	Size of the labels if annotate = TRUE
na_colour	The colour to be used for countries with missing information. Default is grey80
transform_limits	Only if crs is specified and different from 4326. If TRUE (the default) the program expects to receive values of longitude and latitude as in EPSG 4326, (i.e., within -180, +180 for longitude and within -90, +90 for latitude) and automatically updates to the new crs. Set to FALSE if you want to define longitude and latitude limits based on the new crs

## Value

a map

## Examples

```
data(testdata1b)
worldplot(data = testdata1b,
          div = 1,
          ColName = "VNum",
          CountryName = "Cshort",
          CountryNameType = "isoa2",
          rangeVal = c(0,50),
          annotate = FALSE)
```

---

worldplotCat

*worldplotCat*

---

## Description

Plot a world heat map based on a categorical variable.

**Usage**

```
worldplotCat(
  data,
  ColName,
  CountryName,
  CountryNameType = "isoa2",
  longitude = c(-180, 180),
  latitude = c(-90, 90),
  crs = 4326,
  title = "",
  legendTitle = as.character(ColName),
  legend.position = "right",
  Categories = levels(factor(map_df$MapFiller)),
  na.as.category = TRUE,
  label.color = "white",
  label.size = 2,
  annotate = FALSE,
  div = 1,
  palette_option = "D",
  na_colour = "grey80",
  transform_limits = TRUE
)
```

**Arguments**

<code>data</code>	Data set containing the list of nations and the variable that we want to plot.
<code>ColName</code>	Character variable with the name of the variable of interest.
<code>CountryName</code>	Character variable with the name of the country names column.
<code>CountryNameType</code>	Character variable with the coding for CountryName. One of <code>isoa2</code> (default, standing for ISO 3166-1 alpha-2 code), <code>isoa3</code> , or <code>name</code> .
<code>longitude</code>	Longitude limits. Default is <code>c(-180, 180)</code> (whole world with crs as EPSG::4326).
<code>latitude</code>	Latitude limits. Default is <code>c(-90, 90)</code> (whole world with crs as EPSG::4326).
<code>crs</code>	Coordinate reference system (EPSG). By default the value is 4326, which corresponds to EPSG::4326 (WGS84)
<code>title</code>	Title of the plot. Default is no title.
<code>legendTitle</code>	Title of the legend. Default is the name of the filling variable.
<code>legend.position</code>	Position of the legend. If set to "none", no legend is displayed
<code>Categories</code>	categories labels to be plotted in the legend.
<code>na.as.category</code>	Treat NA as a separate category? If 'TRUE, NA will also appear in the legend as one of the categories.
<code>label.color</code>	Color of the labels if <code>annotate = TRUE</code> . Default is white
<code>label.size</code>	Size of the labels if <code>annotate = TRUE</code>

<code>annotate</code>	Do you want to plot country labels (ISO 3166-1 alpha-2 code) on the map? Default is set to FALSE.
<code>div</code>	Parameter for modifying the elements dimensions in the map. Usually, it does not need to be modified. Default value is 1.
<code>palette_option</code>	Character string indicating the palette to be used. Available options range between "A" and "H". You can also enter a string with a colour for each category
<code>na_colour</code>	The colour to be used for countries with missing information. Default is grey80
<code>transform_limits</code>	Only if crs is specified and different from 4326. If TRUE (the default) the program expects to receive values of longitude and latitude as in EPSG 4326, (i.e., within -180, +180 for longitude and within -90, +90 for latitude) and automatically updates to the new crs. Set to FALSE if you want to define longitude and latitude limits based on the new crs

### Value

a map

### Examples

```
data(testdata1b)
worldplotCat(data = testdata1b,
             div = 1,
             ColName = "VCat",
             CountryName = "Cshort",
             CountryNameType = "isoa2",
             annotate = FALSE)
```



# Index

## \* datasets

testdata1, [3](#)

testdata1b, [4](#)

testdata1c, [4](#)

countrycoord\_data, [2](#)

geometries\_data, [3](#)

testdata1, [3](#)

testdata1b, [4](#)

testdata1c, [4](#)

worldplot, [5](#)

worldplotCat, [6](#)