

Package ‘naijR’

August 3, 2020

Type Package

Title Operations to Ease Data Analyses Specific to Nigeria

Version 0.1.3

Date 2020-08-03

Depends R (>= 3.6)

Imports RColorBrewer (>= 1.1.2), magrittr (>= 1.5), mapdata (>= 2.3.0), maps (>= 3.3.0), rgdal (>= 1.4.4), rlang (>= 0.4.0), tools (>= 3.6.3), lifecycle (>= 0.2.0)

Suggests covr, knitr, readxl, rmarkdown, testthat

Description A set of convenience functions as well as geographical/political data about Nigeria, aimed at simplifying work with data and information that are specific to the country.

License GPL-3

LazyData TRUE

Encoding UTF-8

RoxygenNote 7.1.1

VignetteBuilder knitr

RdMacros lifecycle

NeedsCompilation no

Author Victor Ordu [aut, cre] (<<https://orcid.org/0000-0003-3716-0668>>)

Maintainer Victor Ordu <victorordu@outlook.com>

Repository CRAN

Date/Publication 2020-08-03 15:30:15 UTC

R topics documented:

fix_mobile	2
is_state	2
lgas_ng	3
lgas_nigeria	3

map_ng	4
naijR	6
states	6

Index	8
--------------	----------

fix_mobile	<i>Fix mobile numbers</i>
------------	---------------------------

Description

Fixes up local mobile phone numbers to a uniform text format.

Usage

```
fix_mobile(x)
```

Arguments

x A character vector of numerical strings.

Details

This format is specific to that used in a given location - for now the function is useful only for Nigeria mobile numbers, which come in the format expressed by the regex pattern " $^0[7-9][0-1][0-9]{8}$ ".

Value

The updated vector, usually the column of a data frame.

is_state	<i>Test an Object for States</i>
----------	----------------------------------

Description

Test an Object for States

Usage

```
is_state(x, test = c("all", "selected"), allow.na = TRUE)
```

Arguments

x A character vector to be tested.

test The type of test to be carried out - on the vector as a whole i.e. all (the default argument) or on the individual elements i.e. selected.

allow.na logical. If TRUE, all NAs are ignored in the result.

Value

A logical vector.

lgas_ng	<i>List Local Government Areas</i>
---------	------------------------------------

Description

List Local Government Areas

Usage

```
lgas_ng(ng.state = NA_character_)
```

Arguments

ng.state Character; State(s) in the Federation of Nigeria. Default is NA_character_.

Value

If length of ng.state == 1L, a character vector containing the names of Local Government Areas; otherwise a named list whose elements are character vectors of the LGAs in each state.

Examples

```
how_many_lgas <- function(state) {
  require(naijR)
  stopifnot(state %in% states())
  cat(sprintf("No. of LGAs in %s State:", state),
        length(lgas_ng(state)),
        fill = TRUE)
}
how_many_lgas("Sokoto")
how_many_lgas("Ekiti")
```

lgas_nigeria	<i>Local Government Areas of Nigeria</i>
--------------	--

Description

A dataset containing the 774 Local Government Areas of Nigeria

Usage

```
lgas_nigeria
```

Format

A dataframe with 774 rows and 2 columns

lga Local Government Area

state State of the Federation

map_ng

Map of Nigeria

Description

Maps of the Federal Republic of Nigeria that are based on the basic plotting idiom utilised by [maps:map](#) and its variants.

Usage

```
map_ng(
  region = character(),
  data = NULL,
  x = NULL,
  y = NULL,
  breaks = NULL,
  categories = NULL,
  col = NULL,
  fill = FALSE,
  title = NULL,
  caption = NULL,
  show.neighbours = FALSE,
  show.text = FALSE,
  leg.x = 13L,
  leg.y = 7L,
  leg.title,
  leg.orient = c("vertical", "horizontal"),
  ...
)
```

Arguments

region	A character vector of a list of Nigerian States to be displayed.
data	An object containing data, principally the variables required to plotted in a map.
x	Numeric object or factor (or coercible to one). See <i>Details</i> .
y	Numeric. See <i>Details</i>
breaks	Numeric. A vector of length ≥ 1 . If a single value i.e. scalar, it denote the expected number of breaks. Internally, the function will attempt to compute appropriate category sizes or fail if out-of bounds. Where length is $\geq 3L$, it is expected to be an arithmetic sequence that represents category bounds as for cut (applicable only to choropleth maps).

<code>categories</code>	The legend for the choropleth-plotted categories. If not defined, internally created labels are used.
<code>col</code>	Colour to be used for the plot.
<code>fill</code>	Logical. Whether to colour the plotted map region(s). When drawing a choropleth map <code>fill == TRUE</code> is implied.
<code>title</code>	Character vector of length 1.
<code>caption</code>	Character vector of length 1.
<code>show.neighbours</code>	Logical; TRUE to display borders of neighbouring countries.
<code>show.text</code>	Logical. Apply labels to the regions of the map.
<code>leg.x</code>	Numeric. Position of the legend.
<code>leg.y</code>	Numeric. Position of the legend.
<code>leg.title</code>	Character. The legend Title
<code>leg.orient</code>	The orientation of the legend i.e. whether horizontal or vertical.
<code>...</code>	Further arguments for function map

Details

The default value for `region` is to print all States. `NULL` will print an outline map, i.e. without internal boundaries. `data` enables the extraction of data for plotting from an object of class `data.frame`. Columns containing States are identified. The argument also provided context for quasiquotation when providing the `x` and `y` arguments. For `x` and `y`, when both arguments are supplied, they are taken to be point coordinates and are plotted as such. If only `x` is supplied, it is presumed that the intention is to make a choropleth map, and thus, numeric vector arguments are converted into factors i.e. number classes. Otherwise factors or any object that can be coerced to a factor should be used. For plain plots, the `col` argument works the same as with [map](#) and variants. For choropleth maps, the colour provided represents a (sequential) colour palette based on `RColorBrewer::brewer.pal`. The possible colour options can be checked with `getOption("choropleth.colours")` and this can also be modified by the user.

Value

An object of class `map`, invisibly; as a side-effect, results in the drawing of a map of Nigeria.

An object of class `maps` containing the data used to draw the map and which can be used for additional calls to [map](#) or other similar functions (e.g. `graphics::plot.default`).

Note

When adjusting the default colour choiced for choropleth maps, it is advisable to use one of the sequential palettes. For a list of of available palettes, especially for more advanced use, review `RColorBrewer::display.brewer.all`

Examples

```
map_ng() # Draw a map with default settings
map_ng(states("sw"))
map_ng("Kano")
```

 naijR

naijR: Operations to Ease Data Analyses Specific to Nigeria

Description

The naijR package is essentially an R package about Nigeria and for Nigeria.

Details

naijR contains a number of functions that facilitate the management of data sets of interest including data cleaning and wrangling, as well as making available a number of facilities for geo-spatial data visualisation.

 states

Display States of the Federal Republic of Nigeria

Description

Display States of the Federal Republic of Nigeria

Usage

```
states(gpz = NULL, all = TRUE)
```

Arguments

gpz	Geopolitical zone. Default is NULL; optionally "nc", "ne", "nw", "se", "ss" and "sw" (see Details).
all	logical; whether to include FCT in the result

Details

gpz A geo-political zone, in the Nigerian context, is a national subdivision that groups contiguous states. Historically, they arise from subnational administrative divisions known as 'regions' that existed at the time of the country's independence. There are 6 zones - North-Central, North-East, North-West, South-East, South-South and South-West.

Value

The States of Nigeria as a whole or by zones, as a character vector

states

7

Examples

```
states() # lists names of all States  
states("se") # lists States in South-East zone
```

Index

* datasets

lgas_nigeria, 3

cut, 4

fix_mobile, 2

is_state, 2

lgas_ng, 3

lgas_nigeria, 3

map, 5

map_ng, 4

maps:map, 4

naijR, 6

naijR-package (naijR), 6

states, 6