Package: afrihealthsites (via r-universe)

September 12, 2024

Title Geographic locations of African health facilities from different sources

Version 0.1.0.9008

Description African healthsite locations from healthsites.io and WHO/KEMRI-WELLCOME.

License GPL (>= 3) | file LICENSE

URL https://github.com/afrimapr/afrihealthsites

BugReports https://github.com/afrimapr/afrihealthsites/issues

Depends R (>= 3.5.0), sf

Imports mapview, countrycode, RColorBrewer, ggplot2, afriadmin

Suggests rhdx, rhealthsites, remotes, knitr, shiny, leaflet, nngeo

Remotes afrimapr/afriadmin

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

Repository https://afrimapr.r-universe.dev

RemoteUrl https://github.com/afrimapr/afrihealthsites

RemoteRef HEAD

RemoteSha aa93ad61f72a2fcfbbdfe9214f801f873bae6423

Contents

`african_country_names` *african country names and iso 3 letter country codes*

Description

african country names and iso 3 letter country codes

Usage

```
afcountries
```

Format

`dataframe`

An object of class `data.frame` with 53 rows and 2 columns.

Slots

`data` dataframe

africa_healthsites *africa healthsite points from healthsites.io*

Description

africa healthsite points from healthsites.io

Usage

```
sf_healthsites_af
```

Format

sf

An object of class sf (inherits from tbl_df, tbl, data.frame) with 56854 rows and 35 columns.

Slots

data healthsite points

Source

https://www.healthsites.io

africa_healthsites_WHO *africa healthsite points from WHO*

Description

africa healthsite points from WHO

Usage

```
df_who_sites
```

Format

dataframe

An object of class tbl_df (inherits from tbl, data.frame) with 98745 rows and 12 columns.

Slots

data healthsite points

Source

https://www.who.int/malaria/areas/surveillance/who-cds-gmp-2019-01-eng.xlsx

`afrihealthsites` *afrihealthsites : healthsite locations for Africa*

Description

locations from WHO and healthsites.io. Part of afrimapr project.

returns healthsite locations for specified countries and optionally plots map

Usage

```
afrihealthsites(
  country,
  datasource = "healthsites",
  plot = "mapview",
  hs_amenity = "all",
  who_type = "all",
  type_filter = "all",
  returnclass = "sf",
  type_column = "Facility Type",
  label_column = "Facility Name",
  lonlat_columns = c("Longitude", "Latitude"),
  admin_level = NULL,
  admin_names = NULL
)
```

Arguments

`country`	a character vector of country names or iso3c character codes.
`datasource`	data source, 'healthsites' predownloaded, 'who', 'healthsites_live' needs API, 'hdx' not working yet
`plot`	option to display map 'mapview' for interactive, 'sf' for static
`hs_amenity`	filter healthsites data by amenity. 'all', 'clinic', 'dentist', 'doctors', 'pharmacy', 'hospital' to exclude dentist hs_amenity=c('clinic', 'doctors', 'pharmacy', 'hospital')
`who_type`	filter by Facility type
`type_filter`	filter by facility type (from the type_column, or internally defined for healthsites & who)
`returnclass`	'sf' or 'dataframe', currently 'dataframe' only offered for WHO so that can have points with no coords
`type_column`	for user provided files which column has information on type of site, default : 'Facility Type'
`label_column`	for user provided files which column has information on name of site, default : 'Facility Name'

lonlat_columns	for user provided files which columns contain longitude, latitude. option of NULL if no coords
admin_level	what admin level to filter regions from FALSE or NULL if no filtering
admin_names	names of admin regions to filter NULL if no filter

Value

sf

See Also

afrihealthsites compare_hs_sources

Examples

```
sfnga <- afrihealthsites("nigeria", datasource='who', plot='sf')

afrihealthsites('chad', datasource='who', plot='sf')
afrihealthsites('chad', datasource='healthsites', plot='sf')

sfnga <- afrihealthsites("nigeria", plot='mapview')

#to return raw dataframe for WHO data including any rows with no coordinates
dfzaf <- afrihealthsites("south africa", datasource='who', plot=FALSE, returnclass='dataframe')
#note that ISO 3 letter codes and country names with upper case letters can also be used
#afrihealthsites("ZAF")
#afrihealthsites("South Africa")


#filter healthsites data by amenity type
afrihealthsites('chad',datasource = 'healthsites', hs_amenity=c('clinic','hospital'))
#filter who data by Facility type
afrihealthsites('chad',datasource = 'who',who_type=c('Regional hospital','Health Centre'))

#filter by admin regions
afrihealthsites('togo', admin_level=1, admin_names=c('Maritime Region', 'Centrale Region'))
```

---

check_rhealthsites *Check whether to install rhealthsites and install if necessary*

Description

If the rhealthsites package is not installed, install it from GitLab using remotes.

Usage

```
check_rhealthsites()
```

compare_hs_sources *compare healthsite points from different sources for a country*

Description

main aim to plot map comparing different sources

Usage

```
compare_hs_sources(
  country,
  datasources = c("healthsites", "who"),
  plot = "mapview",
  plotshow = TRUE,
  plotcex = c(6, 4),
 col.regions = list(RColorBrewer::brewer.pal(9, "YlGn"), RColorBrewer::brewer.pal(9,
    "BuPu")),
  alpha = c(0.1, 0.1),
  alpha.regions = c(0.7, 0.7),
  layer.names = NULL,
  plotlegend = TRUE,
  hs_amenity = "all",
  who_type = "all",
  canvas = FALSE,
  plotlabels1 = FALSE,
  plotlabels2 = FALSE,
  map.types = c("CartoDB.Positron", "OpenStreetMap.HOT"),
  type_column = "Facility Type",
  label_column = "Facility Name",
  lonlat_columns = c("Longitude", "Latitude"),
  admin_level = NULL,
  admin_names = NULL
)
```

Arguments

country	a character vector of country names or iso3c character codes.
datasources	vector of 2 datasources from 'healthsites' predownloaded, 'who', 'healthsites_live' needs API, 'hdx' not working yet
plot	option to display map 'mapview' for interactive, 'sf' for static
plotshow	whether to show the plot, otherwiser just return plot object
plotcex	sizes of symbols for each source default=c(6,3), helps view symbol overlap
col.regions	list of two colour palettes to pass to mapview
alpha	list of two alphas to pass to mapview - low keeps borders light
alpha.regions	list of two alpha.regions to pass to mapview

layer.names	allow mapview layer.names to be set c('a','b')
plotlegend	whether to add legend to mapview plot
hs_amenity	filter healthsites data by amenity. 'all', 'clinic', 'dentist', 'doctors', 'pharmacy', 'hospital'
who_type	filter by Facility type
canvas	mapview plotting option, TRUE by default for better performance with larger data
plotlabels1	whether to add static labels for source1
plotlabels2	whether to add static labels for source2
map.types	optional specification of background map tiles for mapview, default c('CartoDB.Positron','OpenStreetMa
type_column	just for user provided files which column has information on type of site, default : 'Facility Type'
label_column	just for user provided files which column has information on name of site, default : 'Facility Name'
lonlat_columns	for user provided files which columns contain longitude, latitude. option of NULL if no coords
admin_level	what admin level to filter regions from FALSE or NULL if no filtering
admin_names	names of admin regions to filter NULL if no filter

Value

sf

Examples

```
#compare_hs_sources("nigeria", datasources=c('who', 'healthsites'), plot='mapview')
#compare_hs_sources(c('malawi','zambia'))

#filter by admin regions
compare_hs_sources('togo', admin_level=1, admin_names=c('Maritime Region', 'Centrale Region'))
```

---

country2iso *conversion from country names to iso3c code*

Description

#todo vectorise

Usage

```
country2iso(country)
```

Arguments

country	a character vector of country names

Value

character vector of iso3c codes

Examples

```
iso3c <- country2iso("nigeria")
```

---

facility_types *get & plot freq of facility types for a country*

Description

get & plot freq of facility types for a country

Usage

```
facility_types(
  country,
  datasource = "healthsites",
  datasource_title = NULL,
  type_filter = "all",
  type_column = "Facility Type",
  label_column = "Facility Name",
  brewer_palette = "BuPu",
  lonlat_columns = c("Longitude", "Latitude"),
  admin_level = NULL,
  admin_names = NULL,
  plot_title = "default",
  plot = TRUE
)
```

Arguments

country	a character vector of country names or iso3c character codes.
datasource	data source, 'healthsites' predownloaded, 'who', 'healthsites_live' needs API, 'hdx' not working yet
datasource_title	optional title for datasource to be used in plots - particularly if a filname has been passed for datasource
type_filter	filter by facility type - will depend on the data source

type_column	just for user provided files which column has information on type of site, default : 'Facility Type'
label_column	just for user provided files which column has information on name of site, default : 'Facility Name'
brewer_palette	ColorBrewer palette default 'BuPu',
lonlat_columns	just for user provided files which columns contain longitude, latitude
admin_level	what admin level to filter regions from FALSE or NULL if no filtering
admin_names	names of admin regions to filter NULL if no filter
plot_title	title for plot, 'default', string or NULL for no title
plot	whether to display plot

Value

ggplot2 object

Examples

```
ggnga <- facility_types("nigeria", datasource='who')

facility_types('chad', datasource='who')
facility_types('chad', datasource='healthsites')


#filter healthsites data by amenity type
facility_types('chad',datasource = 'healthsites', type_filter=c('clinic','hospital'))
#filter who data by Facility type
facility_types('chad',datasource = 'who', type_filter=c('Regional hospital','Health Centre'))

# from an sf object
data(sfssd)
ggssd <- facility_types("south sudan",
                        datasource=sfssd,
                        type_column = "type")

# using consistent 9 class facility types for WHO data, specify type_column='facility_type_9'
facility_types('all', datasource='who', type_column='facility_type_9')
```

---

install_rhealthsites *Install the rhealthsites package after checking with the user*

Description

Install the rhealthsites package after checking with the user

Usage

```
install_rhealthsites()
```

iso2country *conversion from iso3c code to country names*

Description

#todo vectorise

Usage

```
iso2country(iso3c)
```

Arguments

iso3c	a character vector of country codes

Value

character vector of country names

Examples

```
name <- iso2country("nga")
```

merge_points *merge healthsite points from different sources for a country*

Description

IN DEVELOPMENT aim is to identify and remove duplicates first version just tests distance, between points in layer1 and 2 BUT problem maybe that there are points in each layer that are duplicates too may want to move this to separate package later

Usage

```
merge_points(
  country,
  datasources = c("healthsites", "who"),
  hs_amenity = c("clinic", "doctors", "pharmacy", "hospital"),
  dist_same_m = 50,
  toreturn = "summary"
)
```

Arguments

country	a character vector of country names or iso3c character codes.
datasources	vector of 2 datasources from 'healthsites' predownloaded, 'who', 'healthsites_live' needs API, 'hdx' not working yet
hs_amenity	filter healthsites data by amenity. 'all', 'clinic', 'dentist', 'doctors', 'pharmacy', 'hospital' to exclude dentist hs_amenity=c('clinic', 'doctors', 'pharmacy', 'hospital')
dist_same_m	distance below which a site from source 1 and 2 is considered same
toreturn	whether to return 'summary' or the merged 'sf' object

Value

sf

Examples

```
#merge_points("nigeria", datasources=c('who', 'healthsites'), plot='mapview')
#merge_points(c('malawi','zambia'))
```

---

nameof_labcol *name of label column according to datasource*

Description

name of label column according to datasource

Usage

```
nameof_labcol(datasource, label_column)
```

Arguments

datasource	vector of 2 datasources from 'healthsites' predownloaded, 'who', 'healthsites_live' needs API, 'hdx' not working yet
label_column	just for user provided files which column has information on name of site, default : 'Facility Name'

Value

character column name

Examples

```
nameof_labcol('who')
```

nameof_zcol *name of z column according to datasource*

Description

name of z column according to datasource

Usage

```
nameof_zcol(datasource, type_column = NULL)
```

Arguments

datasource	'healthsites' predownloaded, 'who', 'healthsites_live' needs API, 'hdx' not working yet
type_column	just for user provided files which column has information on type of site, default : 'Facility Type'

Value

character column name

Examples

```
nameof_zcol('who')
```

national_list_avail *availability of a national master facility list by country*

Description

returns data availability dataframe for 'all' countries or 1 row for a single country TODO add calculation of stats on numbers of countries satisfying criteria

Usage

```
national_list_avail(country = "all")
```

Arguments

country	'all' (default) or a single country name or iso3c character code.

Value

dataframe or single value

Examples

```
national_list_avail('Togo')
```

`national_list_url` *data url for national master facility lists by country*

Description

returns urls for 'all' available countries or url for a single country

Usage

```
national_list_url(country = "all")
```

Arguments

`country`	'all' (default) or a single country name or iso3c character code.

Value

dataframe or single value

Examples

```
national_list_url('South Sudan')

#example of reading in data direct from a url and mapping
#will only work for countries where "machine_readable" is TRUE
#dfgha <- read.csv(national_list_url("Ghana"))
#sfgha <- sf::st_as_sf(dfgha, coords=c("Longitude","Latitude"), crs=4326, na.fail=FALSE)
#mapview::mapview(sfgha, zcol='Type')
#afrihealthsites('gha',datasource=sfgha, type_column='Type')
#compare_hs_sources("ghana", datasources=list(sfgha,"who"), type_column='Type')
#3756 facilities from mfl
#facility_types('gha',datasource=sfgha, type_column='Type')
#1878 facilities in who-kemri data
#facility_types('gha',datasource='who', type_column='Type')


#to return a dataframe with all countries that have urls
national_list_url()
```

near_points *find near points for a country initially in one datasource*

Description

IN DEVELOPMENT aim is to help identify and remove duplicates

Usage

```
near_points(
  country,
  datasources = c("healthsites", "who"),
  hs_amenity = c("clinic", "doctors", "pharmacy", "hospital"),
  dist_same_m = 50,
  toreturn = "summary"
)
```

Arguments

country	a character vector of country names or iso3c character codes.
datasources	vector of 2 datasources from 'healthsites' predownloaded, 'who', 'healthsites_live' needs API, 'hdx' not working yet
hs_amenity	filter healthsites data by amenity. 'all', 'clinic', 'dentist', 'doctors', 'pharmacy', 'hospital' to exclude dentist hs_amenity=c('clinic', 'doctors', 'pharmacy', 'hospital')
dist_same_m	distance below which a site from source 1 and 2 is considered same
toreturn	whether to return 'summary' or the merged 'sf' object

Value

sf

Examples

```
#near_points("burundi", datasources=c('who', 'healthsites'), plot='mapview')
```

runviewer *to run the shiny web application.*

Description

to run the shiny web application.

Usage

```
runviewer()
```

south_sudan_moh *south sudan health facility points from moh*

Description

south sudan health facility points from moh

Usage

```
sfssd
```

Format

`sf`

An object of class `sf` (inherits from `data.frame`) with 2889 rows and 18 columns.

Slots

`data` health facility points

Source

https://www.southsudanhealth.info/facility/fac.php?list&s=0&p=0&ps=2889

tiers4 *takes a vector of facility types and classes them into 4 tiers following Falchetta type definition*

Description

to reclass facility types into a 4 tier classification, any types not passed will be classed as 0

Usage

```
tiers4(
  types,
  tier1 = NULL,
  tier2 = NULL,
  tier3 = NULL,
  tier4 = NULL,
  nametier0 = "Tier0 unknown",
  nametier1 = "Tier1 health post",
  nametier2 = "Tier2 health centre",
  nametier3 = "Tier3 provincial hospital",
  nametier4 = "Tier4 central hospital",
  to_return = "Tier"
)
```

Arguments

types	facility types, one per health facility as a vector
tier1	types to be classed as tier1
tier2	types to be classed as tier2
tier3	types to be classed as tier3
tier4	types to be classed as tier4
nametier0	name for tier0 i.e. when it isn't classed into other tiers
nametier1	name for tier1
nametier2	name for tier2
nametier3	name for tier3
nametier4	name for tier4
to_return	"Tier" or "Tier_name" potentially others later

Value

vector of numeric values or strings representing tiers

Examples

```
tiers4(c("a","b","c"), tier1=c("a","b"), tier2="c", to_return="Tier_name")
```

---

who_type_lookup *lookup table to convert 173 WHO facility types to 9 broad categories*

Description

lookup table to convert 173 WHO facility types to 9 broad categories

Usage

```
who_type_lookup
```

Format

dataframe

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 318 rows and 3 columns.

Slots

`data` who type lookup

Source

https://link.springer.com/article/10.1186/s12916-019-1459-6

Index