Package 'EpiReport'

May 2, 2025

Type Package

Title Epidemiological Report

Version 1.0.4

Date 2025-04-18

Description Drafting an epidemiological report in 'Microsoft Word' format for a given disease, similar to the Annual Epidemiological Reports published by the European Centre for Disease Prevention and Control. Through standalone functions, it is specifically designed to generate each disease specific output presented in these reports and includes:
- Table with the distribution of cases by Member State over the last five years;
- Seasonality plot with the distribution of cases at the European Union / European Economic Area level,
by month, over the past five years;
- Trend plot with the trend and number of cases at the European Union / European Economic Area level,
by month, over the past five years;
- Age and gender bar graph with the distribution of cases at the European Union / European Economic Area level.
Two types of datasets can be used:
- The default dataset of dengue 2015-2019 data;
- Any dataset specified as described in the vignette.

Depends R (>= 3.5.0)

License EUPL

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports officer, flextable, zoo, png, dplyr, tidyr, tidyselect, ggplot2

Suggests knitr (>= 1.20), rmarkdown

VignetteBuilder knitr

URL https://www.ecdc.europa.eu/en/publications-data/monitoring/all-annual-epidemiological-reports
https://epiconcept-paris.github.io/EpiReport/

NeedsCompilation no

Author Lore Merdrignac [aut, ctr, cre] (Author of the package and original code),
Tommi Karki [aut, fnd],
Esther Kissling [aut, ctr],
Joana Gomes Dias [aut, fnd] (Project manager)

Maintainer Lore Merdrignac <l.merdrignac@epiconcept.fr>

Repository CRAN

Date/Publication 2025-05-02 09:30:12 UTC

Contents

AERparams *Dataset describing the parameters for the epidemiological report production*

Description

A dataset describing the parameters to be used for each output of each disease report for all 53 health topics included in TESSy

Usage

```
AERparams
```

Format

A data frame with 53 rows (corresponding to the 53 health topics) and 24 variables:

HealthTopic Disease code that should match with the health topic code from the disease-specific dataset e.g. ANTH, SALM, etc.

DG (optional) Disease group e.g. FWD

DP (optional) Disease programme e.g. FWD

Label Disease label to be used in the report e.g. salmonellosis, anthrax

FrequencyCategory (optional) Frequency of the disease e.g. VERY RARE, NON-RARE, etc.

MeasurePopulation Type of population presented for this disease i.e. ALL or CONFIRMED cases

DatePublicAtlas Date of latest availability in the public access of the Atlas

TableUse Type of table to present in the report i.e. NO table, ASR table presenting age-standardised rates, RATE table presenting rates or COUNT table presenting the number of cases only.

TableRatesLabel Label to use in the table for rates e.g. RATE PER 100000 POPULATION

TableRatesNoDecimals Number of decimals to use when presenting rates

TableASRNoDecimals Number of descimals to use when presenting ASR

AgeGenderUse Type of age and gender bar graph to present i.e. NO graph, AG-COUNT Bar graph presenting the number of cases by age and gender, AG-RATE Bar graph presenting the rates of cases by age and gender, AG-PROP Bar graph presenting the proportion of cases by age and gender, A-RATE Bar graph presenting the rates of cases by age.

AgeGenderBarGraphLabel Label to use in the age and gender bar graph

AgeGenderGraphNoDecimals Number of decimals to use when presenting rates in the age and gender bar graph

TSTrendGraphUse Logical Y/N specifying whether to include a line graph describing the trend of the disease over the time

TSSeasonalityGraphUse Logical Y/N specifying whether to include a line graph describing the seasonality of the disease

TSSpecific Logical Y/N for specific line graph inclusion

MapNumbersUse Logical Y/N specifying whether to include the map presenting the number of cases by Member State

MapRatesUse Logical Y/N specifying whether to include the map presenting the rates of cases by Member State

MapRatesNoDecimals (optional) Number of decimals to use for presenting maps

MapASRUse Logical Y/N specifying whether to include the map presenting the age-standardised rates of cases by Member State

MapASRNoDecimals (optional) Number of decimals to use for presenting maps

Transmission Not implemented yet

TransmissionNoDecimals Not implemented yet

`body_replace_gg_at_bkm`

Replace a plot at a bookmark location

Description

Replace a plot at a bookmark location saving it as a PNG file in a temporary folder.
A bookmark will be considered as valid if enclosing words within a paragraph; i.e., a bookmark along two or more paragraphs is invalid, a bookmark set on a whole paragraph is also invalid, but bookmarking few words inside a paragraph is valid.

Usage

```
body_replace_gg_at_bkm(doc, gg, bookmark, width = 6, height = 3)
```

Arguments

`doc`	a docx device
`gg`	a ggplot object or any object that can be printed in grDevices::png()
`bookmark`	bookmark id
`width`	the width of the device in inches
`height`	the height of the device.

Value

doc

Examples

```
doc <- officer::read_docx(path = file.path(system.file(package = "EpiReport"),
                                           "template/AER_template.docx" ))
p <- EpiReport::getTrend()
doc <- EpiReport::body_replace_gg_at_bkm(doc = doc,
                                         gg = p,
                                         bookmark = "TS_TREND",
                                         width = 6,
                                         height = 3)
```

`cleanECDCTable` *Cleaning the final table*

Description

Cleaning the final table: identifying missing reports with `'-'`, replacing the Member State codes with Member State names (see correspondence table `MSCode`), identifying not reporting Member States with `'.'`

Usage

```
cleanECDCTable(
  x,
  Country = EpiReport::MSCode$Country,
  GeoCode = EpiReport::MSCode$GeoCode
)
```

Arguments

`x`	dataframe, dataset to clean
`Country`	character vector, full names of the countries / Member States (e.g. Austria, Belgium, etc.) that will replace the GeoCodes included the x dataframe (Default `MSCode$Country`)
`GeoCode`	character vector, corresponding GeoCode of each Member State (e.g. AT, BE, etc.) to replace with the country full names (Default `MSCode$GeoCode`)

Value

cleaned ECDC dataframe

See Also

Global function: `getTableByMS`
Default dataset `MSCode`

`cleanMeasureCode` *Clean the MeasureCode variable*

Description

Clean the MeasureCode variable and replace the specific codes with the generic ones (e.g. `ACCUTE.AGE_GENDER.RATE` will be replaced by `CONFIRMED.AGE_GENDER.RATE`)

Usage

```
cleanMeasureCode(var)
```

Arguments

`var`	character string vector variable, variable to clean

Details

- `ALL.COUNT` will replace the following codes:
  - `ALL.DOMESTIC.COUNT`
  - `AGELT1.COUNT`
- `ALL.RATE` will replace the following codes:
  - `ALL.DOMESTIC.AGE.RATE`
- `ALL.AGE.RATE` will replace the following codes:
  - `ALL.DOMESTIC.AGE.RATE`
- `ALL.AGESTANDARDISED.RATE` will replace the following codes:
  - `ALL.DOMESTIC.AGESTANDARDISED.RATE`
- `CONFIRMED.COUNT` will replace the following codes:
  - `ALL.LABCONFIRMED.COUNT`
  - `CONFIRMED.LABCONFIRMED.COUNT`
  - `CONFIRMED.AGELT1.COUNT`
  - `TYPHOID.COUNT`
- `CONFIRMED.RATE` will replace the following codes:
  - `CONFIRMED.LABCONFIRMED.RATE`
  - `CONFIRMED.AGELT1.RATE`
  - `TYPHOID.RATE`
- `CONFIRMED.AGESTANDARDISED.RATE` will replace the following codes:
  - `CONFIRMED.LABCONFIRMED.AGESTANDARDISED.RATE`
- `CONFIRMED.AGE_GENDER.RATE` will replace the following codes:
  - `CONFIRMED.LABCONFIRMED.AGE_GENDER.RATE`
  - `TYPHOID.AGE_GENDER.RATE`
  - `ACCUTE.AGE_GENDER.RATE`

Value

cleaned vector variable

See Also

SALM2016

Examples

```
x <- EpiReport::SALM2016
x$MeasureCode <- cleanMeasureCode(x$MeasureCode)
```

DENGUE2019 *Dataset including Dengue data for 2015-2019*

Description

A dataset containing the data and indicators required to build the epidemiological report for Dengue 2019 TESSy data (default dataset used throughout EpiReport)

Usage

```
DENGUE2019
```

Format

A data frame with 44,332 rows and 11 variables:

HealthTopicCode Disease code e.g. ANTH, SALM, etc.

MeasureCode Code of the measure indicator

TimeUnit Unit of the time variable i.e. Y for yearly data or M for monthly data

TimeCode Time variable including dates in any formats available (according to the unit defined in TimeUnit) yearly data (e.g. 2001) or monthly data (e.g. 2001-01)

GeoCode Geographical level in coded format including country names (e.g. AT for Austria, BE for Belgium, BG for Bulgaria, see also the EpiReport::MSCode table, correspondence table for Member State labels and codes)

XValue XValue

XLabel The label associated with the x-axis in the epidemiological report (see getAgeGender() and plotAgeGender() bar graph for the age variable)

YValue The value associated with the y-axis in the epidemiological report (see plotAge() bar graph for the variable age, or getTableByMS() for the number of cases, rate or age-standardised rate in the table by Member States by year)

YLabel The label associated with the y-axis in the epidemiological report (see getAgeGender() and plotAgeGender() bar graph for the grouping variable gender)

ZValue The value associated with the stratification of XLabel and YLabel in the age and gender bar graph (see getAgeGender() and plotAgeGender())

N Number of cases (see getTrend() and getSeason() line graph)

See Also

The correspondence table for Member State labels and codes MSCode
and the functions mentioned above: getAgeGender, plotAgeGender, plotAge, getTableByMS, getTrend and getSeason.

EcdcColors *Colour palettes following the March 2018 ECDC guidelines for presentation of surveillance data*

Description

Full document: European Centre for Disease Prevention and Control. Guidelines for presentation of surveillance data. Stockholm: ECDC; 2018. Available from: Guidelines for presentation of surveillance data

Usage

```
EcdcColors(col_scale = "green", n = NULL, grey_shade = NULL, hot_cols = NULL)
```

Arguments

col_scale	Selected colour scale, defaults to 'green'. Select from 'green', 'blue', 'red', 'grey', 'qual(itative)' or 'hot(cold)'
n	Number of colours from each colour scale, apart from grey, in order indicated in the guidelines. Defaults to one colour, apart from two colours for the hotcold scale, max 7-8 colours for each scale. To select grey shades, use the argument grey_shade; to select number of hot (warm) colours in the hotcold scale, use the argument hot_cols.
grey_shade	(Optional: use only for 'grey') Selected shade(s) of grey in selected order; c('light', 'mediumlight','medium','mediumdark','dark'). Overrides given number of colours (n). Defaults to 'medium'.
hot_cols	(Optional: use only for 'hotcold') Selected number of hot (warm) colours in the hotcold colour scale. Must be smaller than the total number of colours (n). Defaults to floored half of total hotcold colours.

Author(s)

Tommi Karki

Examples

```
# Select three first green colours
EcdcColors("green", n=3)

# Select two first qualitative colours
EcdcColors("qual", n=2)
```

```
# Select seven red colours
EcdcColors("red", n=7)

EcdcColors("grey", grey_shade = c("mediumlight", "dark"))

# Use in a graph
# Dummy data
mydat <- data.frame(ID = c(seq(1,10,1)),
                    Gender = c(rep(c("F", "M"),5)))
barplot(table(mydat$Gender),
        col = EcdcColors(col_scale = "qual", n=2))

# Hot-cold colour scale
barplot(c(1:4),
        col = EcdcColors(col_scale = "hotcold", n = 4, hot_cols = 2))
```

`filterDisease` *Filter disease parameters*

Description

Filter the table of parameters for the report on the given disease

Usage

```
filterDisease(dis, reportParameters)
```

Arguments

`dis`	character string, disease code
`reportParameters`	dataset of parameters for the report (default `AERparams`)

Value

dataframe with one row (from the `AERparams` dataframe) corresponding to the parameters of the selected disease

See Also

`AERparams`

Examples

```
disease <- "SALM"
reportParameters <- EpiReport::AERparams
reportParameters <- filterDisease(disease, reportParameters)
```

getAER *Get full disease-specific epidemiological report*

Description

Function to generate the 'Microsoft Word' epidemiological report (similar to the ECDC Annual Epidemiological Report (AER)) including all disease-specific outputs at each output-specific bookmarks exact location.
(for further information on the outputs and the corresponding bookmarks, please see the package vignette "The Epidemiological Report Package" with `browseVignettes("EpiReport")`)
(see ECDC AER https://www.ecdc.europa.eu/en/publications-data/monitoring/all-annual-epidemiological

Usage

```
getAER(
 template = file.path(system.file(package = "EpiReport"), "template/AER_template.docx"),
  outputPath = getwd(),
  x = EpiReport::DENGUE2019,
  disease = "DENGUE",
  year = 2019,
  reportParameters = EpiReport::AERparams,
  MSCode = EpiReport::MSCode,
  pathPNG = system.file("maps", package = "EpiReport")
)
```

Arguments

`template`	doc (see `'officer'` package), the empty 'Word' document template in which to include the table and plots disease-specific outputs. Default value is the empty template included in the package. See `getTemplate()`.
`outputPath`	character string, the full path where to generate the epidemiological report 'Word' output. Default value is the current working directory `getwd()`.
`x`	dataframe, raw disease-specific dataset (see specification of the dataset in the package vignette with `browseVignettes("EpiReport")`) (default `DENGUE2019`)
`disease`	character string, disease code (default `"DENGUE"`). Please make sure the disease code is included in the disease-specific dataset x in the `HealthTopicCode` variable.
`year`	numeric, year to produce the report for (default `2019`). Please make sure the year is included in the disease-specific dataset x in the `TimeCode` variable.
`reportParameters`	dataframe, dataset including the required parameters for the report production (default `AERparams`) (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`)
`MSCode`	dataframe, correspondence table of GeoCode names and codes (default `MSCode`) (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`)

pathPNG — character string, the full path to the folder containing the maps (in PNG) to include in the final report

Value

A 'Word' document

See Also

Default template: getTemplate
Default datasets: MSCode AERparams SALM2016 DENGUE2019
Disease-specific outputs: getTableByMS getSeason getTrend getMap getAgeGender

Examples

```
## Not run:
# --- Generating the AER report using the default Dengue dataset
getAER()

## End(Not run)

## Not run:
# --- Or using external data (example below)
ZIKV2016 <- read.table("data/ZIKV2016.csv", sep = ",", header = TRUE, stringsAsFactors = FALSE)
output <- "C:/EpiReport/doc/"
pathMap <- "C:/EpiReport/maps/"
getAER(disease = "ZIKV", year = 2016, x = ZIKV2016, outputPath = output, pathPNG = pathMap)

## End(Not run)
```

---

getAgeGender *Get disease-specific age and gender bar graph*

Description

Function returning the age and gender bar graph that will be included in the epidemiological report at the bookmark location 'BARGPH_AGEGENDER' of the template report.
The bar graph presents the distribution of cases at EU/EEA level using either:

- AG-COUNT: The number of cases by age and gender
- AG-RATE: The rate per 100 000 cases by age and gender
- AG-PROP: The proportion of cases by age and gender
- A-RATE: The rate per 100 000 cases by age only

The choice of the type of bar graph is set in the report parameters table AERparams.
(see ECDC reports https://www.ecdc.europa.eu/en/publications-data/monitoring/all-annual-epidemiologica

Usage

```
getAgeGender(
  x = EpiReport::DENGUE2019,
  disease = "DENGUE",
  year = 2019,
  reportParameters = EpiReport::AERparams,
  geoCode = "EU_EEA31",
  index = 1,
  doc
)
```

Arguments

x	dataframe, raw disease-specific dataset (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`) (default `DENGUE2019`)
disease	character string, disease code (default `"DENGUE"`). Please make sure the disease code is included in the disease-specific dataset x in the `HealthTopicCode` variable.
year	numeric, year to produce the graph for (default `2019`). Please make sure the year is included in the disease-specific dataset x in the `TimeCode` variable.
reportParameters	dataframe, dataset including the required parameters for the graph and report production (default `AERparams`) (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`)
geoCode	character string, GeoCode to run the analysis on (default `"EU_EEA31"`)
index	integer, figure number
doc	'Word' document (see `'officer'` package) in which to add the graph at the bookmark location. If doc is missing, `getAgeGender` returns the `ggplot2` object.

Value

'Word' doc or a ggplot2 object

See Also

Global function for the full epidemilogical report: `getAER`
Required Packages: `ggplot` `body_replace_text_at_bkm`
Internal functions: `plotBarGrouped` (use of `plotAgeGender` discouraged) `plotBar` (use of `plotAge` discouraged) `EcdcColors`
Default datasets: `AERparams`

Examples

```
# --- Plot using the default dataset
getAgeGender()
```

```
# --- Plot using external dataset
# --- Please see examples in the vignette
browseVignettes(package = "EpiReport")
```

---

getMap *Get disease-specific map: distribution of cases by Member State*

Description

Function returning the disease-specific PNG map previously created and stored in a specific folder (see pathPNG argument) and that will be included in the epidemiological report at the bookmark location of the template report, depending of the type of map. Three type of maps can be included in the report:

- Bookmark 'MAP_NB': Distribution of cases by country. An additional caption will be included at the location of the bookmark 'MAP_NB_CAPTION'.
- Bookmark 'MAP_RATE': Distribution of cases per 100 000 population by country. An additional caption will be included at the location of the bookmark 'MAP_RATE_CAPTION'.
- Bookmark 'MAP_ASR': Distribution of cases using age-strandardised rates per 100 000 population by country. An additional caption will be included at the location of the bookmark 'MAP_ASR_CAPTION'.

(see ECDC reports https://www.ecdc.europa.eu/en/publications-data/monitoring/all-annual-epidemiologica

Usage

```
getMap(
  disease = "DENGUE",
  year = 2019,
  reportParameters = EpiReport::AERparams,
  index = 1,
  pathPNG = system.file("maps", package = "EpiReport"),
  doc
)
```

Arguments

disease	character string, disease code (default "DENGUE").
year	numeric, year to produce the map for (default 2019).
reportParameters	dataframe, dataset including the required parameters for the map and report production (default AERparams) (see specification of the dataset in the package vignette with browseVignettes(package = "EpiReport"))
index	integer, figure number

pathPNG	character string, full path to the folder containing the maps in PNG (default 'maps' folder included in the package `system.file("maps", package = "EpiReport")`)
doc	'Word' document (see `'officer'` package) in which to add the maps at the bookmark location. If doc is missing, `getMap` returns a preview of the PNG image.

Value

'Word' doc an image preview

See Also

Global function for the full epidemilogical report: `getAER`
Required Packages: `body_replace_img_at_bkm`
Internal functions: `includeMap` `previewMap`
Default datasets: `AERparams`

Examples

```
# --- Preview of the PNG map using the default Dengue dataset
getMap()

# --- Plot using external PNG image
# --- Please see examples in the vignette
browseVignettes(package = "EpiReport")
```

`getSeason` *Get disease-specific seasonality graph: distribution of cases by month*

Description

Function returning the plot describing the seasonality of the disease that will be included in the epidemiological report at the bookmark location `'TS_SEASON'` of the template report.

The graph includes the distribution of cases at EU/EEA level, by month, over the past five years, with:

- The number of cases by month in the reference year (green solid line)
- The mean number of cases by month in the four previous years (grey dashed line)
- The minimum number of cases by month in the four previous years (grey area)
- The maximum number of cases by month in the four previous years (grey area)

(see ECDC reports https://www.ecdc.europa.eu/en/publications-data/monitoring/all-annual-epidemiologica

Usage

```
getSeason(
  x = EpiReport::DENGUE2019,
  disease = "DENGUE",
  year = 2019,
  reportParameters = EpiReport::AERparams,
  MSCode = EpiReport::MSCode,
  index = 1,
  doc
)
```

Arguments

x	dataframe, raw disease-specific dataset (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`) (default `DENGUE2019`)
disease	character string, disease code (default `"DENGUE"`). Please make sure the disease code is included in the disease-specific dataset x in the `HealthTopicCode` variable.
year	numeric, year to produce the graph for (default `2019`). Please make sure the year is included in the disease-specific dataset x in the `TimeCode` variable.
reportParameters	dataframe, dataset including the required parameters for the graph and report production (default `AERparams`) (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`)
MSCode	dataframe, correspondence table of GeoCode names and codes (default `MSCode`) (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`)
index	integer, figure number
doc	'Word' document (see `'officer'` package) in which to add the graph at the bookmark location. If doc is missing, `getSeason` returns the `ggplot2` object.

Value

'Word' doc or a ggplot2 object

See Also

Global function for the full epidemilogical report: `getAER`
Required Packages: `ggplot` `body_replace_text_at_bkm`
Internal functions: `plotSeasonality`
Default datasets: `AERparams` `MSCode`

Examples

```
# --- Plot using the default dataset
getSeason()
```

```
# --- Plot using external dataset
# --- Please see examples in the vignette
browseVignettes(package = "EpiReport")
```

`getTableByMS` *Get disease-specific table: distribution of cases by Member State (GeoCode)*

Description

Function returning the table (`'flextable'`) that will be included in the epidemiological report at the bookmark location `'TABLE1'` of the template report. An additional caption will be included at the location of the bookmark `'TABLE1_CAPTION'`.
(see Table 1 of the ECDC annual reports https://www.ecdc.europa.eu/en/publications-data/monitoring/all-annual-epidemiological-reports)

Usage

```
getTableByMS(
  x = EpiReport::DENGUE2019,
  disease = "DENGUE",
  year = 2019,
  reportParameters = EpiReport::AERparams,
  MSCode = EpiReport::MSCode,
  index = 1,
  doc
)
```

Arguments

`x`	dataframe, raw disease-specific dataset (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`) (default `DENGUE2019`)
`disease`	character string, disease code (default `"DENGUE"`). Please make sure the disease code is included in the disease-specific dataset x in the `HealthTopicCode` variable.
`year`	numeric, year to produce the table for (default `2019`). Please make sure the year is included in the disease-specific dataset x in the `TimeCode` variable.
`reportParameters`	dataframe, dataset including the required parameters for the report production (default `AERparams`) (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`)
`MSCode`	dataframe, correspondence table of GeoCode names and codes (default `MSCode`) (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`)

index	integer, figure number
doc	'Word' document (see officer package) in which to add the table at the bookmark location. If doc is missing, getTable returns the flextable table object.

Details

The current version of the 'EpiReport' package includes three types of table (see detailed specification of the tables in the package vignette with browseVignettes(package = "EpiReport")):

- COUNT - Table presenting the number of cases by Member State (GeoCode) over a 5-year period;
- RATE - Table presenting the number of cases and rates by Member State (GeoCode) over a 5-year period;
- ASR - Table presenting the number of cases and rates by Member State (GeoCode) over a 5-year period, including age-standardised rates for the most recent year.

Value

'Word' doc or flextable object (see 'flextable' package)

See Also

Global function for the full epidemiological report: getAER
Required Packages: flextable cursor_bookmark
Internal functions: shapeECDCFlexTable cleanECDCTable
Default datasets: AERparams MSCode

Examples

```
# --- Draft the table using the default Dengue dataset
getTableByMS()
```

---

getTemplate *Get epidemiological report (empty) template*

Description

Function to export the generic 'Microsoft Word' empty template (included in the 'EpiReport' package) used to produce the epidemiological report similar to the ECDC Annual Epidemiological Report (AER). The modified version of the template can then be used to produce the final epidemiological report using getAER(template = 'NewTemplate.docx', ...)
(see the package vignette "The Epidemiological Report Package" with browseVignettes("EpiReport"))
(see ECDC annual epidemiological reports https://www.ecdc.europa.eu/en/publications-data/monitoring/all-annual-epidemiological-reports)

Usage

```
getTemplate(output_path)
```

Arguments

output_path	character string, the full path where to create the 'Word' output. Defaut location will be the current working directory (default `getwd()`)

Value

A 'Word' document

See Also

getAER

Examples

```
## Not run:
# --- Export the template in the default folder: working directory
getTemplate()

# --- Or specify the full path
getTemplate(output_path = getwd())

## End(Not run)
```

---

getTrend *Get disease-specific trend plot: trend and number of cases by month*

Description

Function returning the plot describing the trend of the disease over time that will be included in the epidemiological report at the bookmark location `'TS_TREND'` on the template report.

The graph includes the number of cases at EU/EEA level, by month, over the past five years, with:

- The number of cases by month over the 5-year period (grey solid line)
- The 12-month moving average of the number of cases by month (green solid line)

(see ECDC reports https://www.ecdc.europa.eu/en/publications-data/monitoring/all-annual-epidemiologica

Usage

```
getTrend(
  x = EpiReport::DENGUE2019,
  disease = "DENGUE",
  year = 2019,
  reportParameters = EpiReport::AERparams,
  MSCode = EpiReport::MSCode,
  index = 1,
  doc
)
```

Arguments

x	dataframe, raw disease-specific dataset (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`) (default `DENGUE2019`)
disease	character string, disease code (default `"DENGUE"`). Please make sure the disease code is included in the disease-specific dataset x in the `HealthTopicCode` variable.
year	numeric, year to produce the graph for (default `2019`). Please make sure the year is included in the disease-specific dataset x in the `TimeCode` variable.
reportParameters	dataframe, dataset including the required parameters for the graph and report production (default `AERparams`) (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`)
MSCode	dataframe, correspondence table of GeoCode names and codes (default `MSCode`) (see specification of the dataset in the package vignette with `browseVignettes(package = "EpiReport")`)
index	integer, figure number
doc	'Word' document (see `officer` package) in which to add the graph at the bookmark location. If doc is missing, `getTrend` returns the `ggplot2` object.

Value

'Word' doc or a ggplot2 preview

See Also

Global function for the full epidemilogical report: `getAER`
Required Packages: `ggplot` `body_replace_text_at_bkm`
Internal functions: `plotTS12MAvg`
Default datasets: `AERparams` `MSCode`

Examples

```
# --- Plot using the default dataset
getTrend()
```

```
# --- Plot using external dataset
# --- Please see examples in the vignette
browseVignettes(package = "EpiReport")
```

---

includeMap *Including PNG map in the 'Microsoft Word' template*

Description

Function including the disease-specific PNG map in the 'Word' document at the specific bookmark location.

Usage

```
includeMap(
  disease,
  year,
  reportParameters,
  index,
  pathPNG,
  doc,
  pop,
  namePNGsuffix,
  unit,
  mapBookmark,
  captionBookmark
)
```

Arguments

disease	character string, disease code (default "DENGUE").
year	numeric, year to produce the graph for (default 2019).
reportParameters	dataframe, dataset including the required parameters for the graph and report production (default AERparams) (see specification of the dataset in the package vignette with browseVignettes(package = "EpiReport"))
index	integer, figure number
pathPNG	character string, full path to the folder containing the maps in PNG (default 'maps' folder included in the package system.file("maps", package = "EpiReport"))
doc	'Word' document (see 'officer' package) in which to add the maps at the bookmark location
pop	character string, label of the type of population to use in the caption (e.g. confirmed)
namePNGsuffix	character string, suffix of the PNG file name of the map (i.e. "COUNT", "RATE" or "AGESTANDARDISED".)

`unit` character string, label of the unit used in the caption (e.g. `"per 100 000 population"`)

`mapBookmark` character string, label of the bookmark where to add the map in the 'Word' document

`captionBookmark` character string, label of the bookmark where to add the caption in the 'Word' document

Value

'Word' doc

See Also

Global function: `getMap`

`MSCode` *Dataset correspondence table between country names and country code*

Description

Dataframe providing the correspondence table of the geographical code `GeoCode` used in the disease dataset, and the geographical label `Country` to use throughout the report. Additional information on the EU/EEA affiliation is also available in column `EUEEA`.

Usage

```
MSCode
```

Format

A data frame with 32 rows and 3 variables:

Country Full name of the country / Member State e.g. Austria, Belgium, etc.

TheCountry Full name of the country / Member State including 'the' article for NL and UK e.g. Austria, Belgium, the Netherlands, the United Kingdom etc.

GeoCode Associated code (see `GeoCode` variable on the `SALM2016` internal dataset) e.g. AT, BE, BG, etc.

EUEEA For each Member State, variable specifying in the country is part of the EU or EEA.

See Also

`SALM2016`

`orderQuasinum` *Order 'quasinumerical' categorical vectors (increasing order)*

Description

A function to order 'quasinumerical' (i.e. categorical with values such as "15-30" or "<18") integer vectors into increasing order. Currently handles away the following non-numerical characters "-", ">", "<", ">=", "<=", "+".

Usage

```
orderQuasinum(x)
```

Arguments

x	character vector with 'quasinumerical' values

Author(s)

Tommi Karki

See Also

Used in `getAgeGender` and `plotAgeGender` / `plotAge`

Examples

```
age1 <- c("<1", "1-15", "16-25", ">65", "26-65")
age2 <- c("0-4", "5-10", ">65", "25-64", "11-25")
age3 <- c("5-10", ">65", "25-64", "11-25", "<=4")
age4 <- c(">=65", "<18", "18-64")
age5 <- c("5-10", "+65", "25-64", "11-25", "0-4")

age1
orderQuasinum(age1)
age2
orderQuasinum(age2)
age3
orderQuasinum(age3)
age4
orderQuasinum(age4)
age5
orderQuasinum(age5)
```

plotAge *Age bar graph*

Description

(Discouraged function. Please use `plotBarGrouped()` instead.)

Usage

```
plotAge(
  .data,
  xvar = "XLabel",
  yvar = "YValue",
  fill_color1 = "#65B32E",
  ytitle = "Rate"
)
```

Arguments

`.data`	dataframe containing the variables to plot
`xvar`	character string, name of the variable to plot on the x-axis in quotes (default `"XLabel"`)
`yvar`	character string, name of the variable to plot on the y-axis in quotes (default `"YValue"`)
`fill_color1`	character string, hexadecimal colour to use in the graph; (default to ECDC green `"#65B32E"`, see EcdcColors(col_scale = "qual", n = 1))
`ytitle`	character string, y-axis title; (default `"Rate"`).

Details

This function draws a bar graph by age group (or possibly other grouping).
The bar graph presents the distribution of cases at EU/EEA level using the rate per 100 000 cases by age.
Expects aggregated data.

See Also

Global function: `getAgeGender`
Internal function: `EcdcColors`
Required Packages: `ggplot`

Examples

```
# --- Create dummy data
mydat <- data.frame(AgeGroup = c("0-25", "26-65", "65+"),
                    NumberOfCases = c(54,32,41))
```

```
# --- Plot the dummy data
plotAge(mydat,
        xvar = "AgeGroup",
        yvar = "NumberOfCases",
        ytitle = "Number of cases")
```

`plotAgeGender` *Age and Gender bar graph*

Description

(Discouraged function. Please use `plotBarGrouped()` instead.)

Usage

```
plotAgeGender(
  .data,
  xvar = "XLabel",
  yvar = "ZValue",
  group = "YLabel",
  fill_color1 = "#65B32E",
  fill_color2 = "#7CBDC4",
  ytitle = "Rate"
)
```

Arguments

`.data`	dataframe containing the variables to plot
`xvar`	character string, name of the variable to plot on the x-axis in quotes (default `"XLabel"`)
`yvar`	character string, name of the variable to plot on the y-axis in quotes (default `"ZValue"`)
`group`	character string, name of the grouping variable in quotes, e.g. gender. (default `"YLabel"`)
`fill_color1`	character string, hexadecimal colour to use in the graph for bar 1; (default to ECDC green `"#65B32E"`, see EcdcColors(col_scale = "qual", n = 2))
`fill_color2`	character string, hexadecimal colour to use in the graph for bar 2; (default to ECDC blue `"#7CBDC4"`, see EcdcColors(col_scale = "qual", n = 2))
`ytitle`	character string, y-axis title; (default `"Rate"`).

Details

This function draws a bar graph of the distribution of cases by age group and gender (or possibly other grouping).
The bar graph presents the distribution of cases at EU/EEA level using either:

- AG-COUNT: The number of cases by age and gender
- AG-RATE: The rate per 100 000 cases by age and gender
- AG-PROP: The proportion of cases by age and gender

Expects aggregated data.

See Also

Global function: getAgeGender
Internal function: EcdcColors
Required Packages: ggplot

Examples

```
# --- Create dummy data
mydat <- data.frame(Gender=c("F", "F", "M", "M"),
                    AgeGroup = c("0-65", "65+", "0-65", "65+"),
                    NumberOfCases = c(54,43,32,41))

# --- Plot the dummy data
plotAgeGender(mydat,
              xvar = "AgeGroup",
              yvar = "NumberOfCases",
              group = "Gender",
              ytitle = "Number of cases")
```

---

plotBar *Bar graph*

Description

This function draws a bar graph of the values of variable 'Yvar' with the categorical variable 'Xvar' on the x-axis.
Expects aggregated data.

Usage

```
plotBar(
  .data,
  xvar = "XLabel",
  xlabel = "",
  yvar = "YValue",
```

```
  ylabel = "",
  fill_color = EcdcColors(col_scale = "qual", n = 1)
)
```

Arguments

.data	dataframe containing the variables to plot
xvar	character string, name of the variable to plot on the x-axis in quotes (default "XLabel")
xlabel	character string, label of the x axis
yvar	character string, name of the variable to plot on the y-axis in quotes (default "YValue")
ylabel	character string, label of the y axis
fill_color	character string, hexadecimal colour to use in the graph; (default to ECDC green "#65B32E", see EcdcColors(col_scale = "qual", n = 1))

See Also

Global function: getAgeGender
Internal function: EcdcColors
Required Packages: ggplot

Examples

```
# --- Create dummy data
mydat <- data.frame(AgeGroup = c("0-25", "26-65", "65+"),
                    NumberOfCases = c(54,32,41))

# --- Plot the dummy data
plotBar(mydat,
        xvar = "AgeGroup",
        xlabel = "Age",
        yvar = "NumberOfCases",
        ylabel = "Number of cases")
```

---

plotBarGrouped *Grouped bar graph*

Description

This function draws a vertical grouped bar graph of the values of variable 'Yvar' with the categorical variable 'Xvar' on the x-axis and grouped by 'Group' categorical variable.
Expects aggregated data.

Usage

```
plotBarGrouped(
  .data,
  xvar = "XLabel",
  xlabel = "",
  yvar = "ZValue",
  ylabel = "",
  group = "YLabel",
 fill_color = EcdcColors(col_scale = "qual", n = length(unique(.data[[group]]))),
  position = "dodge"
)
```

Arguments

.data	dataframe containing the variables to plot
xvar	character string, name of the variable to plot on the x-axis in quotes (default "XLabel")
xlabel	character string, label of the x axis
yvar	character string, name of the variable to plot on the y-axis in quotes (default "ZValue")
ylabel	character string, label of the y axis
group	character string, name of the grouping variable in quotes, e.g. gender. (default "YLabel").
fill_color	vector of character strings, hexadecimal colour to use in the graph for bars; the vector should contain the number categories in "group" variable. (default to ECDC blue "#7CBDC4" and ECDC green "#65B32E", see EcdcColors(col_scale = "qual", n = 2))
position	character string, position of the bars, either "dodge" or "stack" (default "dodge", see geom_bar(position = ...)).

See Also

Global function: getAgeGender
Internal function: EcdcColors
Required Packages: ggplot

Examples

```
# --- Create dummy data
mydat <- data.frame(Gender=c("F", "F", "M", "M"),
                    AgeGroup = c("0-65", "65+", "0-65", "65+"),
                    NumberOfCases = c(30, 35, 70, 65))

# --- Plot the dummy data
plotBarGrouped(mydat,
               xvar = "AgeGroup",
               xlabel = "Age",
```

```
              yvar = "NumberOfCases",
              ylabel = "Number of cases",
              group = "Gender")

# -- Create dummy data
mydat <- data.frame(VaccStatus = rep(c("Unvaccinated", "1 dose", "2 doses", "3 doses"), 3),
                    AgeGroup = rep(c("<1", "1-4", "5-9") , each = 4),
                    Proportion = c(90, 10, 0, 0,
                                   30, 50, 20, 0,
                                   10, 25, 35, 30))
mydat$VaccStatus <- factor(mydat$VaccStatus,
                           levels = c("Unvaccinated", "1 dose", "2 doses", "3 doses"))
plotBarGrouped(mydat,
               xvar = "AgeGroup",
               xlabel = "Age (years)",
               yvar = "Proportion",
               ylabel = "Proportion of cases %",
               group = "VaccStatus",
               position = "stack")
```

---

plotBarGroupedH *Horizontal grouped bar graph*

Description

This function draws an horizontal bar graph of the values of variable 'Yvar' with the categorical variable 'Xvar' on the x-axis.
Expects aggregated data.

Usage

```
plotBarGroupedH(
  .data,
  xvar = "",
  xlabel = "",
  yvar = "",
  ylabel = "",
  group = "",
 fill_color = EcdcColors(col_scale = "qual", n = length(unique(.data[[group]]))),
  log10_scale = FALSE
)
```

Arguments

.data	dataframe containing the variables to plot
xvar	character string, name of the categorical variable to plot on the x-axis in quotes

xlabel	character string, label of the x axis
yvar	character string, name of the numerical variable to plot on the y-axis in quotes
ylabel	character string, label of the y axis
group	character string, name of the grouping variable in quotes, e.g. gender.
fill_color	character string, hexadecimal colour to use in the graph; (default to ECDC green "#65B32E", see EcdcColors(col_scale = "qual", n = 1))
log10_scale	boolean, TRUE if y-axis should be log scale (default FALSE ,see ggplot2::scale_y_log10)

See Also

Internal function: EcdcColors
Required Packages: ggplot

Examples

```
# --- Create dummy data
mydat <- data.frame(Gender=c("F", "F", "M", "M"),
                    AgeGroup = c("0-65", "65+", "0-65", "65+"),
                    NumberOfCases = c(54,43,32,41))

# --- Plot the dummy data
plotBarGroupedH(mydat,
               xvar = "AgeGroup",
               xlabel = "Age",
               yvar = "NumberOfCases",
               ylabel = "Number of cases",
               group = "Gender")
```

---

plotBarH *Horizontal bar graph*

Description

This function draws an horizontal bar graph of the values of variable 'Yvar' with the categorical variable 'Xvar' on the x-axis.
Expects aggregated data.

Usage

```
plotBarH(
  .data,
  xvar = "",
  xlabel = "",
  yvar = "",
  ylabel = "",
  fill_color = EcdcColors(col_scale = "qual", n = 1),
```

```
  log10_scale = FALSE,
  xlabel_black = ""
)
```

Arguments

.data	dataframe containing the variables to plot
xvar	character string, name of the categorical variable to plot on the x-axis in quotes
xlabel	character string, label of the x axis
yvar	character string, name of the numerical variable to plot on the y-axis in quotes
ylabel	character string, label of the y axis
fill_color	character string, hexadecimal colour to use in the graph; (default to ECDC green "#65B32E", see EcdcColors(col_scale = "qual", n = 1))
log10_scale	boolean, TRUE if y-axis should be log scale (default FALSE ,see ggplot2::scale_y_log10)
xlabel_black	(optional) character string, value of the categorical variable for which the bar should be black

See Also

Internal function: EcdcColors
Required Packages: ggplot

Examples

```
# --- Create dummy data
mfratio <- data.frame( Country = sample(EpiReport::MSCode$Country, 28),
                       Ratio = runif(28, min = 0, max = 28))

# --- Plot the dummy data
plotBarH(mfratio,
         xvar = "Country",
         xlabel = "",
         yvar = "Ratio",
         ylabel = "Male-to-Female ratio",
         log10_scale = FALSE)
plotBarH(mfratio,
         xvar = "Country",
         xlabel = "",
         yvar = "Ratio",
         ylabel = "Male-to-Female ratio",
         log10_scale = TRUE,
         xlabel_black = "EU-EEA")
```

plotPie *Pie chart*

Description

This function draws a pie chart of the values of variable 'Xvar' with the labels from the categorical variable 'Labels'.
Expects aggregated data.

Usage

```
plotPie(
  .data,
  xvar = "",
  labels = "",
  fill_colors = EcdcColors(col_scale = "qual", n = nrow(.data))
)
```

Arguments

.data	dataframe containing the variables to plot
xvar	character string, name of the numerical variable to plot in quotes
labels	character string, name of the character variable including the corresponding labels
fill_colors	vector of character strings, hexadecimal colours to use for each labels in the piechart; the vector should contain the exact number of categories defined in "labels" variable. (default to ECDC colors, see EcdcColors(col_scale = "qual", n = nrow(.data)))

See Also

Internal function: EcdcColors
Required Packages: ggplot

Examples

```
# --- Create dummy data
piechart <- data.frame(Labels = c("Heterosexual females",
                                  "Heterosexual males",
                                  "MSM",
                                  "Unknow"),
                       Values = c(1633,2937,15342,2854))

# --- Plot the dummy data
plotPie(piechart,
         xvar = "Values",
         labels = "Labels")
```

`plotSeasonality` *Seasonality line graph*

Description

This function draws a line graph describing the seasonality of the selected disease over the past 5 years.
The graph includes the distribution of cases, by month, over the past five years, with:

- `yvar`: The number of cases by month in the reference year (green solid line)
- `mean4years`: The mean number of cases by month in the four previous years (grey dashed line)
- `min4years`: The minimum number of cases by month in the four previous years (grey area)
- `max4years`: The maximum number of cases by month in the four previous years (grey area)

Expects aggregated data and pre-calculated min, max and mean figures.

Usage

```
plotSeasonality(
  .data,
  xvar = "TimeCode",
  yvar = "N",
  min4years = "Min4Years",
  max4years = "Max4Years",
  mean4years = "Mean4Years",
  year = 2016
)
```

Arguments

`.data`	dataframe containing the variables to plot
`xvar`	character string, name of the time variable on the x-axis in quotes (default `"TimeCode"`)
`yvar`	character string, name of the variable to plot on the y-axis in quotes (default `"N"`), number of cases by month in the reference year (green solid line)
`min4years`	character string, name of the variable to plot in quotes including the minimum number of cases by month over the past 4 years (default `"Min4Years"`)
`max4years`	character string, name of the variable to plot in quotes including the maximum number of cases by month over the past 4 years (default `"Max4Years"`)
`mean4years`	character string, name of the variable to plot in quotes including the mean of the number of cases by month over the past 4 years (default `"Mean4Years"`)
`year`	numeric, year to produce the graph for (default `2016`).

See Also

Global function: getSeason
Required Packages: ggplot

Examples

```
# --- Plot using external dataset

# Create a dummy dataset
test <- data.frame(Time = as.Date(paste0("2019-",c(1:12), "-01")),
                   N = sample(c(5000:7000), 12),
                   mean = sample(c(4000:5000), 12),
                   low = sample(c(3000:4000), 12),
                   high = sample(c(5000:6000), 12))

# Plot the dummy data
plotSeasonality(test,
                xvar = "Time",
                yvar = "N",
                min4years = "low",
                max4years = "high",
                mean4years = "mean",
                year = 2019)

# --- Please see examples in the vignette
browseVignettes(package = "EpiReport")

# --- Plot using the default dataset
getSeason()
```

---

plotTS *Time series plot*

Description

This function draws a time series of the values of variable 'Yvar' with the time variable 'Xvar' on the x-axis.
Expects aggregated data.

Usage

```
plotTS(
  .data,
  xvar = "",
  xlabel = "",
  yvar = "",
  ylabel = "",
```

```
  fill_color = EcdcColors(col_scale = "qual", n = 1),
  log10_scale = FALSE,
  xvar_format = "%Y",
  xvar_breaks = "1 year"
)
```

Arguments

.data	dataframe containing the variables to plot
xvar	character string, name of the time variable (expects date format) to plot on the x-axis in quotes
xlabel	character string, label of the x axis
yvar	character string, name of the numerical variable to plot on the y-axis in quotes
ylabel	character string, label of the y axis
fill_color	character string, hexadecimal colour to use in the graph; (default to ECDC green "#65B32E", see EcdcColors(col_scale = "qual", n = 1)
log10_scale	boolean, TRUE if y-axis should be log scale (default FALSE ,see ggplot2::scale_y_log10)
xvar_format	character string, time format to use to plot the x-axis ("%Y" for yearly labels or "%b %Y" for monthly labels)
xvar_breaks	character string, time unit to use to plot the x-axis between breaks ("1 year" or "1 month", see ggplot2::scale_x_date(date_breaks = ...))

See Also

Internal function: EcdcColors
Required Packages: ggplot

Examples

```
# --- Create dummy data
mydat <- data.frame(TimeCode = seq(as.Date("2008/1/1"), as.Date("2017/1/1"), "years"),
                    YValue = sample(1:500/10, 10))

# --- Plot the dummy data
plotTS(mydat,
       xvar = "TimeCode",
       xlabel = "Year",
       yvar = "YValue",
       ylabel = "Rate per 100 000 population",
       log10_scale = FALSE,
       xvar_format = "%Y",
       xvar_breaks = "1 year")
```

plotTS12MAvg *Time series with 12-month moving average*

Description

This function draws a line graph describing the trend of the selected disease over the past 5 years. The graph includes the trend and number of cases at EU/EEA level, by month, over the past five years, with:

- `yvar`: The number of cases by month over the 5-year period (grey solid line)
- `movAverage`: The 12-month moving average of the number of cases by month (green solid line)

Expects aggregated data and pre-calculated 12-month moving average.

Usage

```
plotTS12MAvg(.data, xvar = "TimeCode", yvar = "N", movAverage = "MAV")
```

Arguments

`.data`	dataframe containing the variables to plot
`xvar`	character string, name of the time variable to plot on the x-axis in quotes (default `"TimeCode"`)
`yvar`	character string, name of the variable to plot on the y-axis in quotes (default `"N"`), number of cases by month over the 5-year period (grey solid line)
`movAverage`	character string, name of the variable to plot in quotes including the moving average per each time unit (default `"MAV"`)

See Also

Global function: `getTrend`
Required Packages: `ggplot`

Examples

```
# --- Plot using external dataset

# Create a dummy dataset
test <- data.frame(Time = as.Date(paste0("2019-",c(1:12), "-01")),
                   N = sample(c(4000:6000), 12),
                   mean = sample(c(4000:5000), 12))

# Plot the dummy data
plotTS12MAvg(test,
             xvar = "Time",
             yvar = "N",
             movAverage = "mean")
```

plotTSGrouped *Multiple time series plot*

Description

This function draws a time series of the values of variable 'Yvar' with the time variable 'Xvar' on the x-axis. The categorical variable that specify the group of the observations for which there will be one time series each.
Expects aggregated data.

Usage

```
plotTSGrouped(
  .data,
  xvar = "",
  xlabel = "",
  yvar = "",
  ylabel = "",
  group = "",
 fill_color = EcdcColors(col_scale = "qual", n = length(unique(.data[[group]]))),
  log10_scale = FALSE,
  xvar_format = "%Y",
  xvar_breaks = "1 year"
)
```

Arguments

.data	dataframe containing the variables to plot
xvar	character string, name of the time variable (expects date format) to plot on the x-axis in quotes
xlabel	character string, label of the x axis
yvar	character string, name of the numerical variable to plot on the y-axis in quotes
ylabel	character string, label of the y axis
group	character string, name of the grouping variable in quotes, e.g. gender.
fill_color	character string, hexadecimal colour to use in the graph; (default to ECDC green "#65B32E", see EcdcColors(col_scale = "qual", n = length(unique(.data[[group]]))))
log10_scale	boolean, TRUE if y-axis should be log scale (default FALSE ,see ggplot2::scale_y_log10)
xvar_format	character string, time format to use to plot the x-axis ("%Y" for yearly labels or "%b %Y" for monthly labels)
xvar_breaks	character string, time unit to use to plot the x-axis between breaks ("1 year" or "1 month", see ggplot2::scale_x_date(date_breaks = ...))

See Also

Internal function: EcdcColors
Required Packages: ggplot

Examples

```
# --- Create dummy data
mydat <- data.frame(TimeCode = seq(as.Date("2008/1/1"), as.Date("2017/1/1"), "years"),
                    YValue = sample(1:79/10, 20),
                    YLabel = rep(c("Acute", "Chronic"), each = 10))

# --- Plot the dummy data
plotTSGrouped(mydat,
              xvar = "TimeCode",
              xlabel = "Year",
              yvar = "YValue",
              ylabel = "Rate per 100 000 population",
              group = "YLabel",
              log10_scale = TRUE,
              xvar_format = "%Y",
              xvar_breaks = "1 year")

# --- Create dummy data
mydat <- data.frame(TimeCode = rep(seq(as.Date("2008/1/1"), as.Date("2017/1/1"), "years"), 5),
                    YValue = c(sample(1:50/10, 10),
                               sample(1:100/10, 10),
                               sample(1:300/10, 10),
                               sample(1:400/10, 10),
                               sample(1:500/10, 10)),
                    YLabel = rep(c("United Kingdom",
                                   "France",
                                   "Spain",
                                   "Netherlands",
                                   "Belgium"), each = 10))

# --- Plot the dummy data
plotTSGrouped(mydat,
              xvar = "TimeCode",
              xlabel = "Year",
              yvar = "YValue",
              ylabel = "Rate per 100 000 population",
              group = "YLabel",
              log10_scale = FALSE,
              xvar_format = "%Y",
              xvar_breaks = "1 year")
```

---

previewMap *Previewing the PNG map*

Description

Function previewing the disease-specific PNG map

Usage

```
previewMap(disease, year, reportParameters, pathPNG, namePNGsuffix)
```

Arguments

disease	character string, disease code (default "DENGUE").
year	numeric, year to produce the graph for (default 2019).
reportParameters	dataframe, dataset including the required parameters for the graph and report production (default AERparams) (see specification of the dataset in the package vignette with browseVignettes(package = "EpiReport"))
pathPNG	character string, full path to the folder containing the maps in PNG (default 'maps' folder included in the package system.file("maps", package = "EpiReport"))
namePNGsuffix	character string, suffix of the PNG file name of the map (i.e. "COUNT", "RATE" or "AGESTANDARDISED".)

Value

Preview

See Also

Global function: getMap

SALM2016 *Dataset including Salmonellosis data for 2012-2016*

Description

A dataset containing the data and indicators required to build the epidemiological report for Salmonellosis 2016 TESSy data (default dataset used throughout EpiReport)

Usage

```
SALM2016
```

Format

A data frame with 60,775 rows and 18 variables:

HealthTopicCode Disease code e.g. ANTH, SALM, etc.

MeasureLabel optional) Label of the measure indicator

MeasurePopulation Population targeted by the measure indicator

MeasureCode Code of the measure indicator

MeasureId (optional) Measure indicator ID

MeasureType (optional) Type of measure indicator

TimeUnit Unit of the time variable i.e. `Y` for yearly data or `M` for monthly data

GeoLevel (optional) Geographical level e.g. 1, 2, etc

TimeCode Time variable including dates in any formats available (according to the unit defined in `TimeUnit`) yearly data (e.g. 2001) or monthly data (e.g. 2001-01)

GeoCode Geographical level in coded format including country names (e.g. `AT` for Austria, `BE` for Belgium, `BG` for Bulgaria, see also the `EpiReport::MSCode` table, correspondence table for Member State labels and codes)

XValue (optional) XValue

XLabel The label associated with the x-axis in the epidemiological report (see `getAgeGender()` and `plotAgeGender()` bar graph for the age variable)

YValue The value associated with the y-axis in the epidemiological report (see `plotAge()` bar graph for the variable age, or `getTableByMS()` for the number of cases, rate or age-standardised rate in the table by Member States by year)

YLabel The label associated with the y-axis in the epidemiological report (see `getAgeGender()` and `plotAgeGender()` bar graph for the grouping variable gender)

ZValue The value associated with the stratification of XLabel and YLabel in the age and gender bar graph (see `getAgeGender()` and `plotAgeGender()`)

N Number of cases (see `getTrend()` and `getSeason()` line graph)

NMissing (optional)

NLowerResolution (optional)

See Also

The correspondence table for Member State labels and codes `MSCode`
and the functions mentioned above: `getAgeGender`, `plotAgeGender`, `plotAge`, `getTableByMS`, `getTrend` and `getSeason`.

`shapeECDCFlexTable` *Shaping the final table (layout, title, color, font)*

Description

Shaping the final table including titles, adding background color, specifying font name and size.

Usage

```
shapeECDCFlexTable(ft, headers, fsize, fname, maincolor, lastbold)
```

Arguments

ft	flextable (see 'flextable' package), table to shape into ECDC table layout
headers	dataframe including the multiple headers to add to the flextable object. Please note that the column col_keys should contain the names of the flextable object (i.e. col_key = names(x)), accordingly to set_header_df.
fsize	numeric, font to use (Default 7)
fname	character, font name (Default "Tahoma")
maincolor	character string, hexadecimal code for the header background color (Default EcdcColors(col_scale = "green", n=1))
lastbold	bolean, last row in bold (Default TRUE), usually used when the last row includes totals (EU/EEA totals)

Value

flextable object (see flextable package)

See Also

Global function: getTableByMS
Required package: flextable

---

toCapTitle *Capitalise first letter*

Description

Capitalise the first letter of a character string in order to use it as title

Usage

```
toCapTitle(str)
```

Arguments

str	character string to capitalise as a title

Value

character string

Examples

```
my_title <- "number of salmonellosis cases by age group"
toCapTitle(my_title)
```

Index