Package 'flagr'

October 13, 2022

Type Package

Title Implementation of Flag Aggregation

Version 0.3.2

Date 2019-04-02

Description
Three methods are implemented in R to facilitate the aggregations of flags in official statistics. From the underlying flags the highest in the hierarchy, the most frequent, or with the highest total weight
is propagated to the flag(s) for EU or other aggregates. Below there are some reference documents for the topic:
<https://sdmx.org/wp-content/uploads/CL_OBS_STATUS_v2_1.docx>,
<https://sdmx.org/wp-content/uploads/CL_CONF_STATUS_1_2_2018.docx>,
<http://ec.europa.eu/eurostat/data/database/information>,
<http://www.oecd.org/sdd/33869551.pdf>,
<https://sdmx.org/wp-content/uploads/CL_OBS_STATUS_implementation_20-10-2014.pdf>.

License EUPL-1.1

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Suggests tidyr, eurostat, knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation no

Author Mátyás Mészáros [aut, cre],
Matteo Salvati [aut]

Maintainer Mátyás Mészáros <matyas.meszaros@ec.europa.eu>

Repository CRAN

Date/Publication 2019-04-04 16:00:02 UTC

R topics documented:

`flag_divide` *Assignment of the weights for the multiple flags*

Description

This function is used when a single value has multiple flags. The same weight is repeated for each single character.

Usage

```
flag_divide(x)
```

Arguments

`x`	A vector with two items. The first item is a string of flags with several characters, the second is a single numerical value of the weight.

Value

`flag_divide` returns a character matrix with the flags as single characters as the first column and the weight is repeated as the second column. The length of the list is equal to the length of the string of flags.

See Also

`flag_weighted`

Examples

```
flags <- tidyr::spread(test_data[, c(1:3)], key = time, value = flags)
weights <- tidyr::spread(test_data[, c(1, 3:4)], key = time, value = values)
input <- as.data.frame(cbind(flags[,5],weights[,5]),stringsAsFactors = FALSE)[!is.na(flags[,5]),]

do.call(rbind, apply(input,1,flag_divide))
```

flag_frequency *Flag aggregation by the frequency count method*

Description

Flag aggregation by the frequency count method

Usage

```
flag_frequency(f)
```

Arguments

f	A vector of flags containing the flags of a series for a given period.

Value

flag_frequency returns a character with a single character flag in case the highest frequency count is unique, or multiple character in case there are several flags with the highest frequency count.

Examples

```
flag_frequency(c("pe","b","p","p","u","e","d"))
flag_frequency(c("pe","b","p","p","eu","e","d"))


flags <- tidyr::spread(test_data[, c(1:3)], key = time, value = flags)
flag_frequency(flags[,5])
apply(flags[, c(2:ncol(flags))],2, flag_frequency)
```

flag_hierarchy *Flag aggregation by the hierarchical inheritance method*

Description

Flag aggregation by the hierarchical inheritance method

Usage

```
flag_hierarchy(f, flag_list)
```

Arguments

f	A vector of flags containing the flags of a series for a given set of flags.
flag_list	The predefined hierarchy of allowed flags as a vector of single characters.

Value

`flag_hierarchy` returns the flag as single character that is the highest place in the predifined hierarchy order for the given set of flags.

Examples

```
flag_hierarchy(c("p","b","s","b","u","e","b"), flag_list = c("e","s","t"))
flag_hierarchy(c("p","b","s","c","u","d"), flag_list = c("e","s","t"))

flags <- tidyr::spread(test_data[, c(1:3)], key = time, value = flags)
flag_hierarchy(flags[,4],flag_list = c("p","b","s","c","u","e","d"))
apply(flags[, c(2:ncol(flags))],2, flag_hierarchy, flag_list = c("p","b","s","c","u","e","d"))
```

`flag_weighted` *Flag aggregation by the weighted frequency method*

Description

This method can be used when you want to derive the flag of an aggregate that is a weighted average, index, quantile, etc.

Usage

```
flag_weighted(i, f, w)
```

Arguments

`i`	An integer column identifier of data.frame or a matrix containing the flags and weights used to derived the flag for the aggregates.
`f`	A data.frame or a matrix containing the flags of the series (one column per period)
`w`	A data.frame or a matrix with same size and dimesion as `f` containing the corresponding weights for each flags.

Value

`flag_weighted` Returns a character vector with the flag that has the highest weighted frequency or multiple flags in alphabetical order (in case there are more than one flag with the same highest weight) as the first value, and the sum of weights for the given flag(s) as the second value for the given columns of `f`,`w` defined by the parameter `i`.

See Also

`flag_divide`

Examples

```
flag_weighted(1,
              data.frame(f=c("pe","b","p","p","u","e","d"), stringsAsFactors = FALSE),
              data.frame(w=c(10,3,7,12,31,9,54)))
flag_weighted(1,
              data.frame(f=c("pe","b","p","p","up","e","d"), stringsAsFactors = FALSE),
              data.frame(w=c(10,3,7,12,31,9,54)))
flag_weighted(1,
              data.frame(f=c("pe",NA,"pe",NA,NA,"d"), stringsAsFactors = FALSE),
              data.frame(w=c(10,3,7,12,31,9)))


flags <- tidyr::spread(test_data[, c(1:3)], key = time, value = flags)
weights <- tidyr::spread(test_data[, c(1, 3:4)], key = time, value = values)
flag_weighted(7,flags[, c(2:ncol(flags))],weights[, c(2:ncol(weights))])

weights<-apply(weights[, c(2:ncol(weights))],2,function(x) x/sum(x,na.rm=TRUE))
weights[is.na(weights)] <- 0
flags<-flags[, c(2:ncol(flags))]
sapply(1:ncol(flags),flag_weighted,f=flags,w=weights)
```

`propagate_flag` *Derive flags for an aggregates using diffrent methods*

Description

The wrapper function to use the different method and provide a structured return value independently from the method used.

Usage

```
propagate_flag(flags, method = "", codelist = NULL, flag_weights = 0,
  threshold = 0.5)
```

Arguments

`flags`	A data.frame or a matrix containing the flags of the series (one column per period) without row identifiers (e.g. country code).
`method`	A string contains the method to to derive the flag for the aggregate. It can take the value, "hierarchy", "frequency" or "weighted".
`codelist`	A string or character vector defining the list of acceptable flags in case the method "hierarchy" is chosen. In case of the string equals to "estat" or "sdmx" then the predefined standard Eurostat and SDMX codelist is used, otherwise the characters in the sring will define the hierarchical order.
`flag_weights`	A data.frame or a matrix containing the corresponding weights of the series (one column per period) without row identifiers (e.g. country code). It has the same size and dimesion as the `flags` parameter.

threshold The threshold which above the should be the waights in order the aggregate to receive a flag. Defalut value is 0.5, but can be changed to any value.

Value

propagate_flag returns a list with the same size as the number of periods (columns) in the flags parameter. In case of the methods is "hierarchy" or "frequency", then only the derived flag(s) is returned. In case of weighted it returns the flag(s) and the sum of weights if it is above the threshold, otherwise the list contains NA where the sum of weights are below the threshold.

See Also

flag_hierarchy, flag_frequency, flag_weighted

Examples

```
flags <- tidyr::spread(test_data[, c(1:3)], key = time, value = flags)
weights <- tidyr::spread(test_data[, c(1, 3:4)], key = time, value = values)

propagate_flag(flags[, c(2:ncol(flags))],"hierarchy","puebscd")
propagate_flag(flags[, c(2:ncol(flags))],"hierarchy","estat")
propagate_flag(flags[, c(2:ncol(flags))],"frequency")

flags<-flags[, c(2:ncol(flags))]
weights<-weights[, c(2:ncol(weights))]
propagate_flag(flags,"weighted",flag_weights=weights)
propagate_flag(flags,"weighted",flag_weights=weights,threshold=0.1)
```

---

test_data

This data set is a fictive data set with fictive values and flags for testing purposes.

Description

This data set is a fictive data set with fictive values and flags for testing purposes.

Usage

```
test_data
```

Format

A data frame with 195 rows and 4 variables:

geo 2 digit country code

flags flag of the value

time date of observation

values value of the element

Source

The source is in *.csv* format also available in the package.

Index