

Package ‘Rcan’

May 22, 2017

Version 1.3.0

Title R package to help managing and analysing cancer data

Date 2016-09-15

Author Mathieu Laversanne

Maintainer Mathieu Laversanne <laversannem@iarc.fr>

Depends R (>= 2.10)

Description These functions are design by the Cancer Surveillance Unit at IARC. The goal is to provide tools for the unit, but also scientist and other fellows using cancer epidemiological data.

License GPL (>=2)

URL <http://timat.org/matR>

LazyLoad yes

Imports data.table, stats, ggplot2, grid, scales, grDevices, graphics, utils

NeedsCompilation no

R topics documented:

Rcan-package	2
csu_ageSpecific	3
csu_asr	6
csu_ci5x_mean	8
csu_eapc	10
csu_registry_data_1	12
csu_registry_data_2	13
csu_trend	14
csu_trendCohortPeriod	16
csu_trend_legend	20
Index	22

Description

Theses functions are design by the Cancer Surveillance Unit at IARC. The goal is to provide tools for the unit, but also scientist and other fellows using cancer epidemiological data.

Author(s)

Mathieu Laversanne

Maintainer: Mathieu Laversanne <laversannem@iarc.fr>

References

<https://www.iarc.fr/en/publications/pdfs-online/epi/sp95/sp95-chap11.pdf>

See Also

[csu_ageSpecific](#) [csu_asr](#) [csu_eapc](#) [csu_trend](#) [csu_trendCohortPeriod](#)

Examples

```
data(csu_registry_data_2)

# ASR with standard error with missing age.
df_asr <- csu_asr(csu_registry_data_2,
                 "age", "cases", "py",
                 group_by = c("registry", "registry_label", "sex", "year", "ethnic" ),
                 var_age_group = c("registry_label"),
                 var_st_err = "st_err")

df_asr[1:4,]

# ASR with standard error with missing age.
readline(prompt="Press [enter] to continue to EAPC")

# EAPC with standard error

df_eapc <- csu_eapc(df_asr,
                  "asr", "year",
                  group_by=c("registry", "registry_label", "sex", "ethnic" ))

df_eapc[1:4,]

# EAPC with standard error
readline(prompt="Press [enter] to continue to age specific graph")

data(csu_registry_data_1)

# plot age specific rate for 1 population.
df_colombia <- csu_registry_data_1[csu_registry_data_1$registry_label=="Colombia, Cali",]

csu_ageSpecific(df_colombia,
```

```

plot_title = "Colombia, Liver, male")

# plot age specific rate for 1 population, and comparison with CI5X data.
csu_ageSpecific(df_colombia,
plot_title = "Colombia, Liver, male",
CI5_comparison = "Liver")

# plot age specific rate for 4 population, legend at the bottom and comparison with CI5X data.
csu_ageSpecific(csu_registry_data_1,
group_by="registry_label",
legend=csu_trend_legend(position="bottom", nrow = 2),
plot_title = "Liver, male",
CI5_comparison = 7)

```

csu_ageSpecific	<i>csu_ageSpecific</i>
-----------------	------------------------

Description

csu_ageSpecific calculate and plot Age-Specific Rate.

Usage

```

csu_ageSpecific(df_data,
var_age="age",
var_cases="cases",
var_py="py",
group_by = NULL,
missing_age = NULL,
db_rate = 100000,
legend = csu_trend_legend(),
CI5_comparison = NULL,
var_rate = "rate",
plot_title = "csu_title",
format_export = NULL,
graph_dev =FALSE)

```

Arguments

df_data	Data (need to be R data.frame format, see example to import csv file).
var_age	Age group variable: (numerical) 18 5-years age-group, 0-4, 5-9, .. 85+ and an optional missing age. The value can be 1,2,..19, as well as 0,5,..85 or other value as long are they are numerical.
var_cases	Number of event (cases, deaths, ...) variable.
var_py	Population year variable.
group_by	Variable to compare different age specific rate (sex, country, cancer ...). Only one variable can be chosen.
missing_age	Age value representing the missing age cases.

db_rate	The denominator population. Default is 100000.
legend	Legend option: see csu_trend_legend . Use only if group_by is not NULL.
CI5_comparison	Add a dotted line representing the CI5X for a specific cancer. Value can be a cancer_label (example: "Liver"), or a cancer_code (example = 7). See csu_ci5x_mean to get the list of possible values.
var_rate	Name of the age specific variable if a dataframe is return.
plot_title	Title of the plot.
format_export	export the graph in different format:
NULL	Plot in R studio windows.
"pdf"	Export in PDF format.
"tiff"	Export in TIFF 300dpi format.
"png"	Export in PNG 200dpi format.
"svg"	Export in SVG format. Can be edit with https://inkscape.org/fr/ .
	The filename is the plot_title option.
graph_dev	If the plot is embedded in a graphics Device function (such as pdf()), the graph_dev option should be set to TRUE for the first graph to avoid a blank page.

Details

This function calculate and plot the age specific rate. The group_by option allow to compare different population or cancer. The CI5_comparison option allow to compare with the CI5X and therefore test the quality of the data. If the population data stops before 85+ (75+ for instance), the population data **must** be 0 when the population data is unknown so, the program can detect automatically the last age group (70+,75+,80+ or 85+) for population.

Value

Return a plot and a data.frame.

Author(s)

Mathieu Laversanne

References

<https://www.iarc.fr/en/publications/pdfs-online/epi/sp95/sp95-chap11.pdf>

See Also

[csu_asr](#) [csu_eapc](#) [csu_trend](#) [csu_trendCohortPeriod](#)

Examples

```
data(csu_registry_data_1)
data(csu_registry_data_2)
```

```
# you can import your data from csv file using read.csv:
# mydata <- read.csv("mydata.csv", sep=",")

# to select only 1 population.
test <- csu_registry_data_1[csu_registry_data_1$registry_label=="Colombia, Cali",]

# plot age specific rate for 1 population.
csu_ageSpecific(test,
  plot_title = "Colombia, Liver, male")

# plot age specific rate for 1 population, and comparison with CI5X data.
csu_ageSpecific(test,
  plot_title = "Colombia, Liver, male",
  CI5_comparison = "Liver")

# plot age specific rate for 4 population, legend at the bottom and comparison with CI5X data.
csu_ageSpecific(csu_registry_data_1,
  group_by="registry_label",
  legend=csu_trend_legend(position="bottom", nrow = 2),
  plot_title = "Liver, male",
  CI5_comparison = 7)

# plot age specific rate for 4 population, legend at the right.
csu_ageSpecific(csu_registry_data_1,
  group_by="registry_label",
  legend=csu_trend_legend(position="right", right_space_margin = 6.5),
  plot_title = "Liver, male")

# Plot embedded in a graphic device
pdf("test.pdf")
csu_ageSpecific(csu_registry_data_1,
  group_by="registry_label",
  legend=csu_trend_legend(position="right", right_space_margin = 6.5),
  plot_title = "Liver, male",
  graph_dev=TRUE)

csu_ageSpecific(csu_registry_data_1,
  group_by="registry_label",
  legend=csu_trend_legend(position="bottom", nrow = 2),
  plot_title = "Liver, male",
  CI5_comparison = 7,
  graph_dev=FALSE)

dev.off()
```

 csu_asr

csu_asr

Description

csu_asr calculate Age-Standardized Rate (ASR) and Truncated Age-Standardized Rate (TASR) across different population (Registry, year, sex...)

Usage

```
csu_asr(df_data,
  var_age = "age",
  var_cases = "cases",
  var_py = "py",
  group_by=NULL,
  var_age_group=NULL,
  missing_age = NULL,
  db_rate = 100000,
  first_age = 1,
  last_age = 18,
  pop_base = "SEGI",
  var_st_err = NULL,
  correction_info = FALSE,
  var_asr = "asr",
  age_dropped = FALSE)
```

Arguments

df_data	Data (need to be R data.frame format, see example to import csv file).
var_age	Age group variable: (numerical) 18 5-years age-group, 0-4, 5-9, .. 85+ and an optional missing age. The value can be 1,2..19, as well as 0,5,..85 or other value as long are they are numerical.
var_cases	Number of event (cases, deaths, ...) variable.
var_py	Population year variable.
group_by	A vector of variables to compare different ASR (sex, country, cancer ...). Any variable included in the next option var_age_group must be also include here.
var_age_group	Variables over which the number of population age-group might change. For example the last age group (70+, 75+, 80+ or 85+) can vary across registries, or year, but not across sex.
missing_age	Age value representing the missing age cases. If not precise, missing age value will not affect the calculation.
db_rate	The denominator population. Default is 100000.
first_age	First age group included (for Truncated ASR), must be between 1 and 17. 1 represents 0-4, 2 represents 5-9, ... 5 represents 20-24 etc.

last_age	Last age group included (for Truncated ASR), must be between 2 and 18. 2 represents 5-9, ... 5 represents 20-24, ... 18 represents 85+ etc. To calculate ASR for 15-69 years, use: first_age(4), last_age(14).
pop_base	Select the standard population: "SEGI" SEGI world standard population. "EURO" EURO is the european standard population.
var_st_err	Calculate the Standard error and name of the new variable.
correction_info	Logical value. if TRUE compute the variable "correction" : percentage of missing age cases.
var_asr	Name of the new variable for the ASR.
age_dropped	Only for truncated ASR. Logical value: if TRUE, considere the age group not used are already dropped from the data. First_age and last_age option must still be filled.

Details

This function take automatically account the number of age group (last age group can be 70+,75+,80+,85+):
If the population data stops before 85+ (75+ for instance), the population data **must** be 0 when the population data is unknown so, the program can detect automatically the last age group (70+,75+,80+ or 85+) for population.

Value

Give the list of population with less than 18 age group (last age group: 70+, 75+, 80+).
Return a data.frame.

Author(s)

Mathieu Laversanne

References

<https://www.iarc.fr/en/publications/pdfs-online/epi/sp95/sp95-chap11.pdf>

See Also

[csu_ageSpecific](#) [csu_eapc](#) [csu_trend](#) [csu_trendCohortPeriod](#)

Examples

```
data(csu_registry_data_1)
data(csu_registry_data_2)

# you can import your data from csv file using read.csv:
# mydata <- read.csv("mydata.csv", sep=",")

# Age standardized rate (ASR) with no missing age cases.
```

```

result <- csu_asr(csu_registry_data_1,
  "age", "cases", "py",
  group_by = c("registry", "registry_label" ),
  var_age_group = c("registry_label"))

# you can export your result as csv file using write.csv:
# write.csv(result, file="result.csv")

# ASR, with the percentage of correction due to missing age cases.
result <- csu_asr(csu_registry_data_1,
  "age", "cases", "py",
  group_by = c("registry", "registry_label" ),
  var_age_group = c("registry_label"),
  missing_age = 19,
  correction_info = TRUE)

# ASR and standard error with missing age.
result <- csu_asr(csu_registry_data_2,
  "age", "cases", "py",
  group_by = c("registry", "registry_label", "sex", "year", "ethnic" ),
  var_age_group = c("registry_label"),
  var_st_err = "st_err",
  missing_age = 99)

# Truncated ASR, 25-69 years.
result <- csu_asr(csu_registry_data_2,
  "age", "cases", "py",
  group_by = c("registry", "registry_label", "sex", "year", "ethnic" ),
  var_age_group = c("registry_label"),
  var_st_err = "st_err",
  first_age = 6,
  last_age = 14,
  missing_age = 99)

# Truncated ASR, 0-15 with denominator population = 1000000.
result <- csu_asr(csu_registry_data_2,
  "age", "cases", "py",
  group_by = c("registry", "registry_label", "sex", "year", "ethnic" ),
  var_age_group = c("registry_label"),
  var_st_err = "st_err",
  first_age = 1,
  last_age = 3,
  missing_age = 99,
  db_rate = 1000000)

# ASR with EURO population as reference (instead of SEGI)
result <- csu_asr(csu_registry_data_1,
  "age", "cases", "py",
  group_by = c("registry", "registry_label" ),
  var_age_group = c("registry_label"),
  missing_age = 19,
  pop_base = "EURO")

```


Description

CI5 X data, all population grouped.

Usage

```
data("csu_ci5x_mean")
```

Format

A data frame with 522 observations on the following 5 variables.

CSU_age_factor age variable from 1 to 18. 1 is 0-4 years, 2 is 5-9, etc..., 17 is 80-84, 18 is 85+.

CSU_C Number of cases (incidence)

CSU_P Population-year: Reference population

ci5_cancer_label cancer label

ci5_cancer_code cancer code

Details

This dataset have been created in order to compare age specific rate with the CI5X mean directly on the age specific rate graph. Here is the table of cancer_label and cancer_code available:

Code	Label
1	Lip, oral cavity
2	Nasopharynx
3	Other pharynx
4	Oesophagus
5	Stomach
6	Colorectum
7	Liver
8	Gallbladder
9	Pancreas
10	Larynx
11	Lung
12	Melanoma of skin
13	Mesothelioma
14	Kaposi sarcoma
15	Breast
16	Cervix uteri
17	Corpus uteri
18	Ovary
19	Prostate
20	Testis
21	Kidney
22	Bladder
23	Brain, nervous system
24	Thyroid
25	Hodgkin lymphoma
26	Non-Hodgkin lymphoma
27	Multiple myeloma
28	Leukaemia
29	All sites but skin

See: [csu_ageSpecific](#)

Source

<http://ci5.iarc.fr/CI5-X/Default.aspx>

See Also

[csu_ageSpecific](#)

Examples

```
data(csu_registry_data_1)
test <- csu_registry_data_1[csu_registry_data_1$registry_label=="Colombia, Cali",]
csu_ageSpecific(test,
  plot_title = "Colombia, Liver, male",
  CI5_comparison = "Liver")

#See more examples here:
help(csu_ageSpecific)
```

csu_eapc

csu_eapc

Description

csu_eapc calculate the Estimated Annual Percentage Change (EAPC) of rates during a time period with the Confidence Intervalle (CI) across different population (Registry, year, sex...)

Usage

```
csu_eapc(df_data,
  var_rate="asr",
  var_year="year",
  group_by=NULL,
  var_eapc = "eapc")
```

Arguments

df_data	Data (need to be R data.frame format, see example to import csv file).
var_rate	Rate variable. (Standardized or not, incidence, mortality, etc..)
var_year	Period variable. (Year, month, etc...)
group_by	A vector of variables to compare different EAPC (sex, country, cancer ...).
var_eapc	Name of the new variable for the EAPC. The variable for the CI will be name based on var_eapc + "_" + up/Low.

Details

This function use Generalized Linear Model (GLM):

```
glm(rate ~ year, family=poisson(link="log")).
```

We use the poisson family instead of Gaussian, so we can compute EAPC even if the is a rate of 0.

Value

Return a dataframe.

Author(s)

Mathieu Laversanne

References

http://rht.iconcologia.net/stats/sart/eapc/eapc_method.pdf

See Also

[csu_ageSpecific](#) [csu_asr](#) [csu_trend](#) [csu_trendCohortPeriod](#)

Examples

```
data(csu_registry_data_2)

# you import your data from csv file using read.csv:
# mydata <- read.csv("mydata.csv", sep=",")

# Estimated Annual Percentage Change (EAPC) base on ASR.

df_asr <- csu_asr(csu_registry_data_2,
                  "age", "cases", "py",
                  group_by = c("registry", "registry_label", "sex", "year", "ethnic" ),
                  var_age_group = c("registry_label"),
                  missing_age = 99)

result <- csu_eapc(df_asr,
                  "asr", "year",
                  group_by=c("registry", "registry_label", "sex", "ethnic" ))

# you can export your result as csv file using write.csv:
# write.csv(result, file="result.csv")
```

csu_registry_data_1	<i>cancer registry data</i>
---------------------	-----------------------------

Description

Cancer registry data for liver cancer, males, 2007, 4 registries.

Usage

```
data("csu_registry_data_1")
```

Format

A data frame with 76 observations on the following 5 variables.

age age variable from 1 to 19. 1 is 0-4 years, 2 is 5-9, etc..., 17 is 80-84, 18 is 85+, 19 represents missing age.

cases Number of cases (incidence)

py Population-year: Reference population of the registry

registry_label Name of the registry

registry Registry code

Details

this 4 registries have been selected for this example as they different number of age group (75+, 80+, 85+) and some have missing age cases.

Source

<http://ci5.iarc.fr/CI5plus/Default.aspx>

See Also

[csu_asr](#) [csu_eapc](#) [csu_ageSpecific](#)

Examples

```
data(csu_registry_data_1)

# Age standardized rate (ASR) with no missing age cases.
result <- csu_asr(csu_registry_data_1,
  "age", "cases", "py",
  group_by = c("registry", "registry_label" ),
  var_age_group = c("registry_label"))

#See more examples here:
help(csu_asr)
```

csu_registry_data_2 *cancer registry data*

Description

Cancer registry data for liver cancer

Usage

```
data("csu_registry_data_2")
```

Format

A data frame with 125856 observations on the following 8 variables.

sex sex variable: 1 male, 2 female

year year variable, from 1953 to 2007

age age variable from 0 to 85 with missing age.

0 is 0-4 years, 5 is 5-9, ..., 80 is 80-84, 85 is 85+, 99 represents missing age.

cases Number of cases (incidence)

py Population-year: Reference population of the registry

registry_label Name of the registry (118 populations)

registry Registry code (102 registries)

ethnic ethnic code: white (10), black (30), ..., all(99).

Source

<http://ci5.iarc.fr/CI5plus/Default.aspx>

See Also

[csu_asr](#) [csu_eapc](#) [csu_ageSpecific](#)

Examples

```
data(csu_registry_data_2)

# ASR and standard error with missing age.
result <- csu_asr(csu_registry_data_2,
  "age", "cases", "py",
  group_by = c("registry", "registry_label", "sex", "year", "ethnic" ),
  var_age_group = c("registry_label"),
  missing_age = 99,
  var_st_err = "st_err")

#See more examples here:
help(csu_asr)
```

csu_trend

*csu_trend***Description**

csu_trend plot stats over year.

Usage

```
csu_trend(df_data,
var_trend = "asr",
var_year = "year",
group_by = NULL,
logscale = TRUE,
smoothing = NULL,
legend = csu_trend_legend(),
yaxes_title = "Age standardized rate per 100000",
plot_title = "csu_title",
format_export = NULL,
graph_dev = FALSE)
```

Arguments

df_data	Data (need to be R data.frame format, see example to import csv file).
var_trend	Statistics variable to be plot on Y axis. Usually for the ASR, but it could be the cumulative risk, or even the number of cases.
var_year	Time variable.
group_by	Variable to compare different age specific rate (sex, country, cancer ...). Only one variable can be chosen.
logscale	Logical value: if TRUE Y-axis use logscale.
smoothing	Apply a smoothing using the R loess function. . The numerical parameter controls the degree of smoothing. See option span in loess .
legend	legend option: see csu_trend_legend . Use only if group_by is not NULL.
yaxes_title	Y-axis title. To choose in function for the var_trend option.
plot_title	Title of the plot.
format_export	export the graph in different format:
NULL	Plot in R studio windows.
"pdf"	Export in PDF format.
"tiff"	Export in TIFF 300dpi format.
"png"	Export in PNG 200dpi format.
"svg"	Export in SVG format. Can be edit with https://inkscape.org/fr/ .

The filename is the `plot_title` option.

`graph_dev` If the plot is embedded in a graphics Device function (such as `pdf()`), the `graph_dev` option should be set to `TRUE` for the first graph to avoid a blank page.

Details

This function is design the plot a statistics over time. It has been design for the ASR by year, but can be used for other statistics over time period. The `group_by` option allow to compare different population or cancer.

Value

Return a plot.

Author(s)

Mathieu Laversanne

See Also

[csu_ageSpecific](#) [csu_asr](#) [csu_eapc](#) [csu_trendCohortPeriod](#)

Examples

```
data(csu_registry_data_2)

# you can import your data from csv file using read.csv:
# mydata <- read.csv("mydata.csv", sep=",")

# to select only 1 population
test <- csu_registry_data_2[csu_registry_data_2$registry_label=="Colombia, Cali",]

# to change sex variable to factor with label
test$sex <- factor(test$sex, levels=c(1,2), labels=c("Male", "Female"))

# to calculate the asr
df_asr <- csu_asr(test,missing_age = 99,
                  group_by = c("registry", "registry_label", "year", "sex"),
                  var_age_group = c("registry", "registry_label"))

# plot ASR ove year, by sex.
csu_trend(df_asr, group_by="sex",
          plot_title = "Colombia, Liver")

# plot ASR over year, by sex, with small smoothing.
csu_trend(df_asr, group_by="sex",
          plot_title = "Colombia, Liver",
          smoothing = 0.3)

# plot ASR over year, by sex, with high smoothing.
csu_trend(df_asr, group_by="sex",
          plot_title = "Colombia, Liver",
          smoothing = 0.5)
```

```
# Plot embedded in a graphic device
pdf("example_test.pdf")
csu_trend(df_asr, group_by="sex",
  plot_title = "Colombia, Liver",
  smoothing = 0.3,
  graph_dev=TRUE)

csu_trend(df_asr, group_by="sex",
  plot_title = "Colombia, Liver",
  smoothing = 0.5)

dev.off()
```

csu_trendCohortPeriod *csu_trendCohortPeriod*

Description

csu_trendCohortPeriod plot cohort period age specific graph.

Usage

```
csu_trendCohortPeriod(
  df_data,
  var_age = "age",
  var_cases="cases",
  var_py="py",
  var_year = "year",
  type = "Cohort",
  missing_age = NULL,
  logscale = TRUE,
  db_rate = 100000,
  first_age = 6,
  last_age = 16,
  year_group = 5,
  age_dropped=FALSE,
  plot_title = "csu_title",
  format_export = NULL,
  graph_dev =FALSE)
```

Arguments

df_data	Data (need to be R data.frame format, see example to import csv file).
var_age	Age group variable: (numerical) 18 5-years age-group, 0-4, 5-9, .. 85+ and an optional missing age. The value can be 1,2..19, as well as 0,5,..85 or other value as long are they are numerical.
var_cases	Number of event (cases, deaths, ...) variable.
var_py	Population year variable.
var_year	Time variable.

type Type of the plot:

	"Cohort"	Cohort graph.
	"Period"	Period graph.
	"Both"	Cohort Period graph.
missing_age	Age value representing the missing age cases.	
logscale	Logical value: if TRUE Y-axis use logscale.	
db_rate	The denominator population. Default is 100000.	
first_age	First age group included, must be between 1 and 17. 1 represents 0-4, 2 represents 5-9, ... 5 represents 20-24 etc. Default is 6 (25-29 years).	
last_age	Last age group included, must be between 2 and 18. 2 represents 5-9, ... 5 represents 20-24, ... 18 represents 85+ etc. Default is 16 (75-79 years). To plot the age group from 15-19 years until 65-69 years, use: first_age(4), last_age(14)	
year_group	Usually, data are regrouped in 5 years period. . The numerical parameter controls the size of the group.	
age_dropped	Only if some age grouped are missing in the data. Logical value: if TRUE, considere the age group not used are already dropped from the data. First_age and last_age option must still be filled.	
plot_title	Title of the plot.	
format_export	export the graph in different format:	
	NULL	Plot in R studio windows.
	"pdf"	Export in PDF format.
	"tiff"	Export in TIFF 300dpi format.
	"png"	Export in PNG 200dpi format.
	"svg"	Export in SVG format. Can be edit with https://inkscape.org/fr/ .
	The filename is the plot_title option.	
graph_dev	If the plot is embedded in a graphics Device function (such as pdf()), the graph_dev option should be set to TRUE for the first graph to avoid a blank page.	

Details

This function is design the plot a the age-specific cohort and period plot. The type option allow to choose between the 3 different graphics: "Cohort", "Period", or "Both". Please note than the cohort plot and the period plot can be superimposed if the first_age is too low.

Value

Return a plot.

Author(s)

Mathieu Laversanne

See Also

[csu_asr](#) [csu_eapc](#) [csu_ageSpecific](#) [csu_trend](#)

Examples

```
data(csu_registry_data_2)

# you can import your data from csv file using read.csv:
# mydata <- read.csv("mydata.csv", sep=",")

# to select only 1 population
test <- csu_registry_data_2[csu_registry_data_2$registry == 84020,]
test <- test[test$sex==1,]

# plot cohort graph from 25-29 years until 75-79 years.
csu_trendCohortPeriod(df_data=test,
                      missing_age =99,
                      plot_title = "USA, Liver, males")

# plot Period graph from 0-5 until 85+.
csu_trendCohortPeriod(df_data=test,
                      missing_age =99,
                      plot_title = "USA, Liver, males",
                      type="Period",
                      first_age=1,
                      last_age=18)

# plot Cohort-Period graph from 30-64 years until 70-74 years.
csu_trendCohortPeriod(df_data=test,
                      missing_age =99,
                      plot_title = "USA, Liver, males",
                      type="Both",
                      first_age=7,
                      last_age=15)

# plot Cohort-Period graph from 30-64 years until 70-74 years with Y axis normal scale.
csu_trendCohortPeriod(df_data=test,
                      missing_age =99,
                      plot_title = "USA, Liver, males",
                      type="Both",
                      first_age=7,
                      last_age=15,
                      logscale=FALSE)

# plot Cohort graph from 25-29 years until 75-79 years, with data grouped in 2 years period.
csu_trendCohortPeriod(df_data=test,
                      missing_age =99,
                      plot_title = "USA, Liver, males",
                      type="Cohort",
                      year_group = 2)

# Plot embedded in a graphic device
pdf("example_test.pdf")
csu_trendCohortPeriod(df_data=test,
```

```

                                missing_age =99,
                                plot_title = "USA, Liver, males",
    type="Both",
    first_age=7,
    last_age=15,
    graph_dev=TRUE)

    csu_trendCohortPeriod(df_data=test,
                                missing_age =99,
                                plot_title = "USA, Liver, males",
    type="Both",
    first_age=7,
    last_age=15,
    logscale=FALSE)

    dev.off()

```

csu_trend_legend

csu_trend_legend

Description

csu_trend_legend legend option use in Rcan package for trends.

Usage

```
csu_trend_legend(title=NULL, position="bottom",nrow=1, right_space_margin=1)
```

Arguments

title Title of the legend. (only if position = "bottom").

position Position of the legend:

"bottom" The legend is place at the bottom of the graph. use with option nrow.

"right" The legend is place at the right of each trend. use with option right_space_margin.

nrow Number of row of the legend (only if position = "bottom").

right_space_margin

If the position = "right", the right margin need to be enough big, to contain the text legend. value should be between 1 and 10 depending the size of the text legend.

Details

This function return a list for the trend graphics of the package Rcan

Value

Return a structured list.

Author(s)

Mathieu Laversanne

See Also

[csu_ageSpecific](#)

Examples

```
data(csu_registry_data_1)

csu_ageSpecific(csu_registry_data_1,
  group_by="registry_label",
  legend=csu_trend_legend(title="registry", position="bottom", nrow = 2),
  plot_title = "Legend: bottom")

csu_ageSpecific(csu_registry_data_1,
  group_by="registry_label",
  legend=csu_trend_legend(position="right", right_space_margin = 2),
  plot_title = "Legend: right, cut")

csu_ageSpecific(csu_registry_data_1,
  group_by="registry_label",
  legend=csu_trend_legend(position="right", right_space_margin = 6.5),
  plot_title = "Legend: right")

#See more examples here:
help(csu_ageSpecific)
```

Index

*Topic **datasets**

- csu_ci5x_mean, [8](#)
- csu_registry_data_1, [12](#)
- csu_registry_data_2, [13](#)

*Topic **package**

- Rcan-package, [2](#)

csu_ageSpecific, [2](#), [3](#), [7](#), [10–13](#), [15](#), [19](#), [21](#)

csu_asr, [2](#), [4](#), [6](#), [11–13](#), [15](#), [19](#)

csu_ci5x_mean, [4](#), [8](#)

csu_eapc, [2](#), [4](#), [7](#), [10](#), [12](#), [13](#), [15](#), [19](#)

csu_registry_data_1, [12](#)

csu_registry_data_2, [13](#)

csu_trend, [2](#), [4](#), [7](#), [11](#), [14](#), [19](#)

csu_trend_legend, [4](#), [14](#), [20](#)

csu_trendCohortPeriod, [2](#), [4](#), [7](#), [11](#), [15](#), [16](#)

loess, [14](#)

Rcan (Rcan-package), [2](#)

Rcan-package, [2](#)