Package ‘Rcan’

March 25, 2020

Version 1.3.81
Title Cancer Registry Data Analysis and Visualisation
Date 2020-03-03
Author Mathieu Laversanne [aut, cre],
    Jerome Vignat [aut],
    Cancer Surveillance Unit [cph]
Maintainer Mathieu Laversanne <laversannem@iarc.fr>
Depends R (>= 3.5)
Description Tools for basic and advance cancer statistics and graphics.
    Groups individual data, merges registry data and population data, calculates age-
    specific rate, age-standardized rate, cumulative risk, estimated annual percentage rate with stan-
    dards error. Creates graphics across variable and
    time, such as age-specific trends, bar chart and period-cohort trends.
License GPL (>= 2)
URL https://github.com/timat35/Rcan
LazyLoad yes
Imports data.table, stats, ggplot2, grid, scales, grDevices, graphics,
    utils
Suggests testthat
RoxygenNote 6.1.1
NeedsCompilation no
Repository CRAN
Date/Publication 2020-03-25 14:10:13 UTC

R topics documented:

   Rcan-package .......................................................... 2
csu_ageSpecific .......................................................... 4
csu_ageSpecific_top ..................................................... 7
csu_asr ................................................................. 10
Rcan-package

Cancer Registry Data Analysis and Visualisation

Description

Tools for basic and advance cancer statistics and graphics. Groups individual data, merges registry data and population data, calculates age-specific rate, age-standardized rate, cumulative risk, estimated annual percentage rate with standards error. Creates graphics across variable and time, such as age-specific trends, bar chart and period-cohort trends.

Author(s)

Mathieu Laversanne [aut, cre], Jerome Vignat [aut], Cancer Surveillance Unit [cph]

Maintainer: Mathieu Laversanne <laversannem@iarc.fr>

References

https://publications.iarc.fr/_publications/media/download/3753/609d0d7711047dd76d7f3dbaa25d7f041fcdpdf

See Also

csu_group_cases csu_merge_cases_pop csu_asr csu_cumrisk csu_eapc csu_ageSpecific csu_ageSpecific_top csu_bar_top csu_time_trend csu_trendCohortPeriod
Examples

```r
# group individual data by
# 5 year age group
# ICD grouping from dataframe ICD_group_GLOBOCAN
data(ICD_group_GLOBOCAN)
data(data_individual_file)

df_data_year <- csu_group_cases(data_individual_file, 
  var_age="age",
  group_by=c("sex", "regcode", "reglabel"),
  df_ICD = ICD_group_GLOBOCAN,
  var_ICD = "site",
  var_year = "doi")

head(df_data_year)
# individual cases grouped by ICD and 5 years age group and year.
readline(prompt="Press [enter] to continue to merge cases and population")

# data(data_population_file)

data(data_population_file)

df_data <- csu_merge_cases_pop(
  df_data_year, 
  data_population_file, 
  var_age = "age_group", 
  var_cases = "cases", 
  var_py = "pop", 
  group_by = c("sex"))

head(df_data)
# Merge 5-years age grouped data with population by year (automatic) and sex
readline(prompt="Press [enter] to continue to merge cases and population")

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 <- subset(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 CI5XI 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 CI5XI 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 = 16)

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,
Arguments

- **df_data**: Data (need to be R dataframe format, see example to import csv file).
- **var_age**: Age variable. Several format are accepted
  ```
  1  "0-4"  0
  2  "5-9"  5
  3  "10-14"  10
  ...
  17 "80-84"  80
  18 "85+"  85
  ```
  Missing age value must be precise in the option `missing_age`. Last age group will always be considered without size (i.e., 80+, 85+, etc.).
- **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.
- **logscale**: Logical value: if TRUE Y-axis use logscale.
- **plot_title**: Title of the plot.
- **legend**: Legend option: see `csu_trend_legend`. Use only if `group_by` is not NULL.
- **color_trend**: Vector of color for the trend. The color codes are hexadecimal (e.g., "#FF0000") or predefined R color names (e.g., "red").
- **CI5_comparison**: Add a dotted line representing the CI5XI for a specific cancer. Value can be a cancer_label (example: "Liver"), or a cancer_code (example = 16). See `csu_ci5_mean` to get the list of possible values.
- **var_rate**: Name of the age specific variable if a dataframe is return.

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 CI5XI 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://publications.iarc.fr/_publications/media/download/3753/609d0d7711047dd76d7f3dbaa25d7f041fcd.pdf

See Also

csu_group_cases csu_merge_cases_pop csu_asr csu_cumrisk csu_eapc csu_ageSpecific_top
csu_bar_top csu_time_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 <- subset(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 CI5XI 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 CI5XI data using cancer code.
csu_ageSpecific(
    csu_registry_data_1,
group_by="registry_label",
legend=csu_trend_legend(position="bottom", nrow = 1),
plot_title = "Liver, male",
CI5_comparison = 16)
# 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

csu_ageSpecific(
csu_registry_data_1,
group_by="registry_label",
legend=csu_trend_legend(position="bottom", nrow = 2),
plot_title = "Liver, male",
CI5_comparison = 16)
plot.new()

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")

dev.off()
Description

csu_ageSpecific_top calculate and plot Age-Specific Rate for the top X cancer or top X population.

Usage

csu_ageSpecific_top(df_data,
    var_age="age",  
    var_cases="cases", 
    var_py="py", 
    var_top,
    group_by=NULL, 
    missing_age=NULL, 
    db_rate = 100000, 
    logscale = FALSE, 
    nb_top = 5,
    plot_title=NULL, 
    plot_subtitle=NULL, 
    var_color=NULL)

Arguments

df_data  Data (need to be R data.frame format, see example to import csv file).
var_age  Age variable. Several format are accepted
1 "0-4" 0
2 "5-9" 5
3 "10-14" 10
... ...
17 "80-84" 80
18 "85+" 85

Missing age value must be precise in the option missing_age.
Last age group will always be considere without size (ie: 80+,85+, etc..).

var_cases  Number of event (cases, deaths, ...) variable.
var_py  Population year variable.
var_top  Cancer label variable or country/registry variable for example.
group_by  Variable to compare different age specific rate (sex, registry ...).
Only one variable can be chosen. One graph will be produce per variable. If the
variable is a factor, the labels associate to the variable will be the subtitle of the
plot. See factor.

missing_age  Age value representing the missing age cases.
db_rate  The denominator population. Default is 100000.
logscale  Logical value: if TRUE Y-axis use logscale.
nb_top  Lowest Rank include. Default is 5.
plot_title Title of the plot.
plot_subtitle Subtitle of the plot. (For example, "Top 5 cancer").
var_color Variable with a color associate to each cancer.
The color codes are hexadecimal (e.g. "#FF0000") or predefined R color names (e.g. "red").
This allow to keep the same color coding for each graph.

Details
This function keep only the top X cancer and plot their age specific rate. The group_by option allow to compare different population. 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 plots and a data.frame.

Author(s)
Mathieu Laversanne

References
https://publications.iarc.fr/_publications/media/download/3753/609d0d7711047dd76d7f3dbaa25d7f041fcde40d2781210.pdf

See Also

csu_group_cases csu_merge_cases_pop csu_asr csu_cumrisk csu_eapc csu_ageSpecific
csu_bar_top csu_time_trend csu_trendCohortPeriod

Examples

library(Rcan)
data("csu_CI5XI_data")

# get the registry code asssociate to registry_label
# print(unique(csu_CI5XI_data[,c("registry_label", "registry_code")]),nrows = 1000)

# get the cancer code asssociate to cancer_label
# print(unique(csu_CI5XI_data[,c("cancer_label", "cancer_code")]),nrows = 1000)

# remove all cancers:
df_data <- subset(csu_CI5XI_data ,cancer_code < 62)
df_data$sex <- factor(df_data$sex, levels=c(1,2), labels=c("Male", "Female"))

# select Thailand changmai
df_data_1 <- subset(df_data, registry_code==764001)
# plot for Thailand Changmai
dt_result_1 <-
csu_ageSpecific_top(df_data_1,
  var_age="age",
  var_cases="cases",
  var_py="py",
  var_top="cancer_label",
  group_by="sex",
  plot_title= "Thailand, Chiangmai",
  plot_subtitle = "Top 5 cancer",
  missing_age = 19)

#select USAm NPCR
df_data_2 <- subset(df_data,registry_code== 840000 & ethnic_group == 99)

dt_result_2 <-
csu_ageSpecific_top(
  df_data_2,
  var_age="age",
  var_cases="cases",
  var_py="py",
  var_top="cancer_label",
  group_by="sex",
  plot_title= "USA",
  plot_subtitle = "Top 5 cancer",
  missing_age = 19)

---

**csu_asr**

**csu_asr**

describe 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",
    var_top = "cancer_label",
    group_by = "sex",
    plot_title = "Country, Region",
    plot_subtitle = "Top 5 cancer",
    missing_age = 19)
\begin{verbatim}
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",
crude_rate = FALSE,
var_st_err = NULL,
correction_info = FALSE,
var_asr = "asr",
age_dropped = FALSE)
\end{verbatim}

**Arguments**

- **df_data**  
  Data (need to be R data.frame format, see example to import csv file).

- **var_age**  
  Age variable. Several format are accepted:

<table>
<thead>
<tr>
<th>Age Group</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-4</td>
<td>0</td>
</tr>
<tr>
<td>5-9</td>
<td>5</td>
</tr>
<tr>
<td>10-14</td>
<td>10</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>80-84</td>
<td>80</td>
</tr>
<tr>
<td>85+</td>
<td>85</td>
</tr>
</tbody>
</table>

  Missing age value must be precise in the option `missing_age`.  
  Last age group will always be considere without size (ie: 80+, 85+, etc..).

- **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 (1960).
  "EURO"      EURO is the European standard population (1976).
  "EURO2"     EURO2 is the European standard population (2013).
  "WHO"       WHO is the WHO standard population (2001).

crude_rate   Logical value. if TRUE calculate the crude rate instead of the ASR.
Variables var_age_group, missing_age, pop_base, var_st_err, correction_info
are not used.
Default name of the new variable is "crude_rate".

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, Assume 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://publications.iarc.fr/_publications/media/download/3753/609d0d7711047dd76d7f3dbaa25d7f041fcd0a1.pdf

See Also

csu_group_cases csu_merge_cases_pop csu_cumrisk csu_eapc csu_ageSpecific csu_ageSpecific_top
    csu_bar_top csu_time_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")

csu_bar_top

Description

csu_bar_top plots top X single-sided or double-sided bar chart.

Usage

csu_bar_top(df_data,
    var_value,
    var_bar,
    group_by=NULL,
    nb_top = 10,
    plot_title=NULL,
    plot_subtitle=NULL,
    xtitle=NULL,
    label_by=NULL,
    color=NULL,
    digits = 1)

Arguments

df_data Data (need to be R data.frame format, see examples to import csv file).
var_value Value variable.
    There must be only 1 value for each bar.
var_bar Bar label variable.
group_by • Single-sided bar chart. NULL (default)
    • Double-sided bar chart. Variable name with exactly 2 values. (For example, "sex").
    Must be filled if label_by argument is defined.
nb_top Lowest Rank included. Default is 10.
plot_title Title of the plot. (For example, "Top 10 cancer sites").
plot_sub_title Subtitle of the plot. (For example, "Males").
xtitle x-axe title. (For example, "Number of cases").
label_by 2 values vector. Will overwrite the legend label for double-sided bar chart. (See group_by).
For example: c("Male", "Female").
color The color codes are hexadecimal (e.g. "]FF0000") or predefined R color names (e.g. "red").

• Single-sided bar chart. 1 hexadecimal color code (same color for each bar)
or variable name with a color associated to each bar label variable.
• Double-sided bar chart. 2 values vector. For example: c("#2c7bb6","#b62ca1").
digits Number of decimal digits. Default: 1

Details
This function plots a top X (default is top 10) bar chart, single-sided or double sided.

Value
Return plots and a data.frame.

Author(s)
Mathieu Laversanne

See Also
  csu_group_cases  csu_merge_cases_pop  csu_asr  csu_cumrisk  csu_eapc  csu_ageSpecific
  csu_ageSpecific_top  csu_time_trend  csu_trendCohortPeriod

Examples

data(data_individual_file)
data(data_population_file)
data(ICD_group_GLOBOCAN)

#Group individual data by:
#5 year age group
#ICD grouping from dataframe ICD_group_GLOBOCAN
#year extract from date of incidence

df_data_year <- csu_group_cases(data_individual_file,
  var_age="age",
  group_by=c("sex", "regcode", "reglabel"),
  df_ICD = ICD_group_GLOBOCAN,
  var_ICD ="site",
  var_year = "doi")

#Merge 5-years age grouped data with population by year (automatic) and sex

df_data <- csu_merge_cases_pop(
df_data_year,  
data_population_file,  
var_age = "age_group",  
var_cases = "cases",  
var_py = "pop",  
group_by = c("sex"))

#prepare for calculate ASR  
df_data$age_group_label <- NULL # to avoid warning  
df_data <- subset(df_data , year == 2012) # to keep only 2012 data  
df_data$year <- NULL # to avoid warning

# calculate asr  
df_asr <- csu_asr(df_data,  
"age_group",  
"cases",  
"pop",  
group_by=c("sex", "ICD_group", "LABEL", "relabel", "regcode"),  
missing_age =19)

#remove Other cancer  
df_asr <- subset(df_asr , LABEL != "Other")  
df_asr <- subset(df_asr , LABEL != "Other skin")

#keep male  
df_asr_M <- subset(df_asr , sex==1)

#Single sided bar plot  
data1 <- csu_bar_top(  
df_asr_M,  
var_value="cases",  
var_bar="LABEL",  
nb_top = 10,  
plot_title = "Top 10 cancer sites",  
xtitle= "Number of cases",  
color= c("#2c7bb6"),  
digits=0)

#Double sided bar plot example 1  
data2 <- csu_bar_top(  
df_asr,  
var_value="cases",  
var_bar="LABEL",  
group_by="sex",  
nb_top = 15,  
plot_title = "Top 15 cancer sites",  
xtitle= "Number of cases",  
label_by=c("Male", "Female"),  
color = c("#2c7bb6","#b62ca1"),  
digits=0)
# Double sided bar plot example 2

data3 <- csu_bar_top(
  df_asr,
  var_value="asr",
  var_bar="LABEL",
  group_by="sex",
  nb_top = 10,
  plot_title = "Top 10 cancer sites",
  xtitle = "Age-standardized rate per 100,000",
  label_by=c("Male", "Female"),
  color = c("#2c7bb6","#b62ca1"),
  digits=1)

---

csu_CI5XI_data

*Cancer in five continent volume XI.*

**Description**

CI5 XI summary database.

**Usage**

data("csu_CI5XI_data")

**Format**

A data frame with 1113210 observations on the following 10 variables.

- registry_code  Registry code (integer)
- registry_label  Name of the registry (character)
- sex  sex (integer)
- cancer_code  cancer code (integer)
- cancer_label  cancer label (character)
- 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. (integer)
- cases  Number of cases (incidence) (integer)
- py  Population-year: Reference population of the registry (integer)
- CI5_continent  Continent code (integer)
- ethnic_group  ethnic code (integer)

**Details**

All information are available here: http://ci5.iarc.fr/CI5-XI/Default.aspx
Source


See Also

csu_asr csu_cumrisk csu_eapc csu_ageSpecific

Examples

data("csu_CI5XI_data")

# get the registry code associate to registry_label
# print(unique(csu_CI5XI_data[,c("registry_code","registry_label")]),nrows = 1000)

# get the cancer code associate to cancer_label
# print(unique(csu_CI5XI_data[,c("cancer_label","cancer_code")]),nrows = 1000)

# remove all cancers:
df_data <- subset(csu_CI5XI_data,cancer_code < 62)

# change group_by option to factor
df_data$sex <- factor(df_data$sex, levels=c(1,2), labels=c("Male","Female"))

# select Thailand changmai
df_data_1 <- subset(df_data, registry_code==764001)
dt_result_1 <- csu_ageSpecific_top(df_data_1,
   var_age="age",
   var_cases="cases",
   var_py="py",
   var_top="cancer_label",
   group_by="sex",
   plot_title= "Thailand, Chiangmai",
   plot_subtitle = "Top 5 cancer",
   missing_age = 19)

# select USA
df_data_2 <- subset(df_data, registry_code==840000 & ethnic_group == 99)
dt_result_2 <- csu_ageSpecific_top(df_data_2,
   var_age="age",
   var_cases="cases",
   var_py="py",
   var_top="cancer_label",
   group_by="sex",
plot_title = "USA",
plot_subtitle = "Top 5 cancer",
missing_age = 19)

---

<table>
<thead>
<tr>
<th>csu_ci5_mean</th>
<th>cancer registry data</th>
</tr>
</thead>
</table>

**Description**

CI5 XI data, all population grouped.

**Usage**

data("csu_ci5_mean")

**Format**

A data frame with 1026 observations on the following 5 variables.

- **ci5_cancer_code**: cancer code
- **ci5_cancer_label**: cancer label
- **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

**Details**

This dataset have been created in order to compare age specific rate with the CI5XI mean directly on the age specific rate graph. Here is the table of cancer_label and cancer_code available:

<table>
<thead>
<tr>
<th>ci5_cancer_code</th>
<th>ci5_cancer_label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Lip</td>
</tr>
<tr>
<td>2</td>
<td>Tongue</td>
</tr>
<tr>
<td>3</td>
<td>Mouth</td>
</tr>
<tr>
<td>4</td>
<td>Salivary glands</td>
</tr>
<tr>
<td>5</td>
<td>Tonsil</td>
</tr>
<tr>
<td>6</td>
<td>Other oropharynx</td>
</tr>
<tr>
<td>7</td>
<td>Nasopharynx</td>
</tr>
<tr>
<td>8</td>
<td>Hypopharynx</td>
</tr>
<tr>
<td>9</td>
<td>Pharynx unspecified</td>
</tr>
<tr>
<td>10</td>
<td>Oesophagus</td>
</tr>
<tr>
<td>11</td>
<td>Stomach</td>
</tr>
<tr>
<td>12</td>
<td>Small intestine</td>
</tr>
</tbody>
</table>
13 Colon
14 Rectum
15 Anus
16 Liver
17 Gallbladder
18 Pancreas
19 Nose, sinuses
20 Larynx
21 Lung
23 Bone
24 Melanoma of skin
25 Other skin
26 Mesothelioma
27 Kaposi sarcoma
28 Connective and soft tissue
29 Breast
30 Vulva
31 Vagina
32 Cervix uteri
33 Corpus uteri
35 Ovary
37 Placenta
38 Penis
39 Prostate
40 Testis
42 Kidney
43 Renal pelvis
44 Ureter
45 Bladder
47 Eye
48 Brain, nervous system
49 Thyroid
50 Adrenal gland
51 Other endocrine
52 Hodgkin disease
53 Non-Hodgkin lymphoma
54 Immunoproliferative diseases
55 Multiple myeloma
56 Lymphoid leukaemia
57 Myeloid leukaemia
58 Leukaemia unspecified
59 Myeloproliferative disorders
60 Myelodysplastic syndromes
62 All sites
63 All sites but skin

See: csu_ageSpecific
Description

csu_cumrisk Calculate cumulative risk across different population (Registry, year, sex...)

Usage

csu_cumrisk(df_data,
var_age = "age",
var_cases = "cases",
var_py = "py",
group_by=NULL,
missing_age = NULL,
last_age = 15,
var_st_err=NULL,
correction_info=FALSE,
var_cumrisk="cumrisk")

Arguments

df_data Data (need to be R data.frame format, see example to import csv file).
var_age Age variable. Several format are accepted

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&quot;0-4&quot;</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>&quot;5-9&quot;</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>&quot;10-14&quot;</td>
<td>10</td>
</tr>
</tbody>
</table>

See Also
csu_ageSpecific

Examples

data(csu_registry_data_1)
test <- subset(csu_registry_data_1 , registry_label == "Colombia, Cali")
csu_ageSpecific(test,

plot_title = "Colombia, Liver, male",
CIS_comparison = "Liver")

#See more examples here:
help(csu_ageSpecific)
Missing age value must be precise in the option `missing_age`. Last age group will always be considered without size (i.e., 80+, 85+, etc.).

- **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 ...).
- **missing_age**: Age value representing the missing age cases. If not precise, missing age value will not affect the calculation.
- **last_age**: Last age group included, must be between 2 and 17. 2 represents 5-9, ..., 5 represents 20-24, ..., 17 represents 80-84, 18 represents 85+, etc. Default is 15 (calculates cumulative risk 0-74).
- **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_cumrisk**: Name of the new variable for the cumulative risk.

**Details**

Calculate cumulative risk across different population
the age group include in cumulative risk will always exclude the last age group since we do not know the size of the last age group (i.e., 85+, 80+ etc.)

**Value**

Return a `data.frame`.

**Author(s)**

Mathieu Laversanne

**References**

https://publications.iarc.fr/_publications/media/download/3753/609d0d7711047dd76d7f3dbaa25d7f041fcd5ac2.pdf

**See Also**

csu_group_cases csu_merge_cases_pop csu_asr csu_cumrisk csu_eapc csu_ageSpecific csu_ageSpecific_top csu_bar_top csu_time_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=";")

# Cumulative risk, 0-74 years, with no missing age cases.
result <- csu_cumrisk(csu_registry_data_1,  
"age", "cases", "py",  
group_by = c("registry", "registry_label" )
)

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

# Cumulative risk, 0-74 years, with the percentage of correction due to missing age cases.
result <- csu_cumrisk(csu_registry_data_1,  
"age", "cases", "py",  
group_by = c("registry", "registry_label" ),  
missing_age = 19,  
correction_info = TRUE)

# Cumulative risk 0-74 years and standard error with missing age.
result <- csu_cumrisk(csu_registry_data_2,  
"age", "cases", "py",  
group_by = c("registry", "registry_label", "sex", "year", "ethnic" ),  
var_st_err = "st_err",  
missing_age = 99)

# Cumulative risk, 0-69 years.
result <- csu_cumrisk(csu_registry_data_2,  
"age", "cases", "py",  
group_by = c("registry", "registry_label", "sex", "year", "ethnic" ),  
var_st_err = "st_err",  
last_age = 14,  
missing_age = 99)

csu_eapc

csu_eapc

Description

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

Usage

csu_eapc(df_data,  
var_rate="asr",  
var_st_err="st_err")
Arguments

data 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.
CI_level Confidence interval level. Default is 0.95.

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

See Also

csu_group_cases csu_merge_cases_pop csu_asr csu_cumrisk csu_ageSpecific csu_ageSpecific_top
csu_bar_top csu_time_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_group_cases**

**csu_group_cases**

---

**Description**

`csu_group_cases` groups individual data into 5 years age-group data and other user defined variable (sex, registry, etc...).

Optionally: Group cancer based on a standard ICD10 coding; Extract year from custom year format.

**Usage**

```r
csu_group_cases(df_data,
  var_age ,
  group_by=NULL,
  var_cases = NULL,
  df_ICD = NULL,
  var_ICD=NULL,
  var_year = NULL,
  all_cancer=FALSE)
```

**Arguments**

- **df_data**: Individual data (need to be R data.frame format, see examples to import csv file).
- **var_age**: Age variable. (Numeric). Value > 150 will be considered as missing age.
- **group_by**: (Optional) A vector of variables to create the different population (sex, country, etc...).
csu_group_cases

var_cases  (Optional) cases variable: If there is already a variable for the number of cases.
df_ICD    (Optional) ICD file for ICD grouping information. Must have 2 fields: "ICD", "LABEL"
          . 2 formats are possible:
          Each ICD code separated by ICD group

          ICD    LABEL
          C82    NHL
          C83    NHL
          C84    NHL
          C85    NHL
          C96    NHL

          ICD code already grouped.

          ICD_group    LABEL
          C82-85,C96   NHL

          2 ICD codes separated by ";" includes all the ICD code between.
          2 ICD codes separated by "," includes only these 2 ICD code.
          For instance, C82-85, C96 (or C82-C85, C96) includes:
          C82, C83, C84, C85 and C96
          Must be filled if var_ICD argument is defined

          example: ICD_group_GLOBOCAN

var_ICD    (Optional) ICD variable: ICD variable in the individual data.
           Must be filled if df_ICD argument is defined

var_year   (Optional) Year variable: Extract year from custom format , as long as the year
           is expressed with 4 digits (i.e. "yyyymmdd","ddmmyyyy", "yyyy/mm","dd-mm-yyyy", etc..) and group data by year.

all_cancer (Optional) If TRUE, will calculate the number of cases for all cancers (C00-97)
            and all cancers but non-melanoma of skin (C00-97 but C44)
            Need var_ICD and df_ICD arguments to be defined

Details

For most analysis, individual cases database need to be grouped by category.
This function groups data by 5 years age-group and other user defined variable.
Next step will be to add 5 years population data. (see csu_merge_cases_pop).

Value

Return a dataframe.
Author(s)

Mathieu Laversanne

See Also

csu_merge_cases_pop csu_asr csu_cumrisk csu_eapc csu_ageSpecific csu_ageSpecific_top
csu_bar_top csu_time_trend csu_trendCohortPeriod

Examples

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

data(ICD_group_GLOBOCAN)
data(data_individual_file)

#group individual data by
# 5 year age group
df_data_age <- csu_group_cases(data_individual_file,
   var_age="age",
   group_by=c("sex", "regcode", "reglabel", "site"))

#group individual data by
# 5 year age group
# ICD grouping from dataframe ICD_group_GLOBOCAN

df_data_icd <- csu_group_cases(data_individual_file,
   var_age="age",
   group_by=c("sex", "regcode", "reglabel"),
   df_ICD = ICD_group_GLOBOCAN,
   var_ICD = "site")

#group individual data by
# 5 year age group
# ICD grouping from dataframe ICD_group_GLOBOCAN
# year (extract from date of incidence)

df_data_year <- csu_group_cases(data_individual_file,
   var_age="age",
   group_by=c("sex", "regcode", "reglabel"),
   df_ICD = ICD_group_GLOBOCAN,
   var_ICD = "site",
   var_year = "doi"
# you can export your result as csv file using write.csv:
# write.csv(result, file="result.csv")

### Description

csu_merge_cases_pop merges registry data and population data, group by year and other user defined variable (sex, registry, etc...).

### Usage

csu_merge_cases_pop(df_cases, 
                     df_pop, 
                     var_age, 
                     var_cases="cases", 
                     var_py=NULL, 
                     group_by=NULL)

### Arguments

df_cases Registry data group by 5 years-age group (need to be R data.frame format, see examples to import csv file).
df_pop Population data group by 5-years age group (need to be R data.frame format, see examples to import csv file).
var_age Age variable. Several format are accepted

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>5-9</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>10-14</td>
<td>10</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>17</td>
<td>80-84</td>
<td>80</td>
</tr>
<tr>
<td>18</td>
<td>85+</td>
<td>85</td>
</tr>
</tbody>
</table>

This variable must be a variable with the same column name in both dataset (df_cases and df_pop).
Age >= 85 in the df_pop dataset will be aggregated as 85+.

var_cases Cases variable in the df_cases dataset.

var_py (Optional) If population is "long format", name of the population variable in the df_pop dataset.
If population data is wide format (see details), var_py must be NULL.

```
group_by
```

(Optional) A vector of variables to create the different population (sex, country, etc...).

Each variable must be a variable with the same column name in both dataset
(df_cases and df_pop).
Do not include the "year" variable since it is automatically detected (see details).

### Details

This function merges registry data and population for further analysis.
Both datasets must be group by 5-years age group.

**If present, the year information in format "yyyy" will be detected automatically.**

2 formats are accepted for population data:
- Long format: (year and population are 2 variables)

```
<table>
<thead>
<tr>
<th>sex</th>
<th>age</th>
<th>pop</th>
<th>year</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>116128</td>
<td>2005</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>130995</td>
<td>2005</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>137556</td>
<td>2005</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>27171</td>
<td>2007</td>
</tr>
<tr>
<td>2</td>
<td>17</td>
<td>13585</td>
<td>2007</td>
</tr>
<tr>
<td>2</td>
<td>18</td>
<td>13585</td>
<td>2007</td>
</tr>
</tbody>
</table>
```

- Wide format: (One column per year and no population variable, "yyyy" year format must be included in columns name)

```
<table>
<thead>
<tr>
<th>sex</th>
<th>age</th>
<th>Y2013</th>
<th>Y2014</th>
<th>Y2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0-4</td>
<td>215607</td>
<td>237346</td>
<td>247166</td>
</tr>
<tr>
<td>1</td>
<td>5-9</td>
<td>160498</td>
<td>152190</td>
<td>152113</td>
</tr>
<tr>
<td>1</td>
<td>10-14</td>
<td>175676</td>
<td>171794</td>
<td>165406</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>75-79</td>
<td>20625</td>
<td>20868</td>
<td>23434</td>
</tr>
<tr>
<td>2</td>
<td>80-84</td>
<td>7187</td>
<td>7276</td>
<td>7620</td>
</tr>
<tr>
<td>2</td>
<td>85+</td>
<td>2551</td>
<td>2597</td>
<td>2617</td>
</tr>
</tbody>
</table>
```

### Value

Return a dataframe.

### Author(s)

Mathieu Laversanne
See Also

csu_group_cases csu_asr csu_cumrisk csu_eapc csu_ageSpecific csu_ageSpecific_top
csu_bar_top csu_time_trend csu_trendCohortPeriod

Examples

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

data(ICD_group_GLOBOCAN)
data(data_individual_file)
data(data_population_file)

#group individual data by
# 5 year age group
# ICD grouping from dataframe ICD_group_GLOBOCAN
# year (extract from date of incidence)

df_data_year <- csu_group_cases(data_individual_file,
    var_age="age",
    group_by=c("sex", "regcode", "reglabel"),
    df_ICD = ICD_group_GLOBOCAN,
    var_ICD ="site",
    var_year = "doi")

#Merge 5-years age grouped data with population by year (automatic) and sex

df_data <- csu_merge_cases_pop(
    df_data_year,
    data_population_file,
    var_age = "age_group",
    var_cases = "cases",
    var_py = "pop",
    group_by = c("sex"))

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

csu_registry_data_1 cancer registry data

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


See Also

csu_asr csu_cumrisk csu_eapc csu_ageSpecific csu_ageSpecific_top csu_bar_top

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

```r
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


### See Also

- `csu_asr`
- `csu_cumrisk`
- `csu_eapc`
- `csu_ageSpecific`
- `csu_ageSpecific_top`
- `csu_bar_top`
- `csu_time_trend`
- `csu_trendCohortPeriod`

### Examples

```r
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")
```
### csu_time_trend

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

---

#### Description

`csu_time_trend` plot stats over year.

#### Usage

```r
csu_time_trend(df_data,
var_trend = "asr",
var_year = "year",
group_by = NULL,
logscale = FALSE,
smoothing = NULL,
legend = csu_trend_legend(),
color_trend = NULL,
ytitle = "Age standardized rate per 100,000",
plot_title = "csu_title")
```

#### 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 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`.
- `color_trend`: Vector of color for the trend. The color codes are hexadecimal (e.g. "#FF0000") or predefined R color names (e.g. "red").
- `ytitle`: Y-axis title. Default is "Age standardized rate per 100,000".
- `plot_title`: Title of the plot.
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_group_cases csu_merge_cases_pop csu_asr csu_cumrisk csu_eapc csu_ageSpecific csu_ageSpecific_top csu_bar_top csu_trendCohortPeriod`

Examples
```r
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 <- subset(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", "ethnic"),
  var_age_group = c("registry", "registry_label")
)

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

# plot ASR over year, by sex, with small smoothing.
csu_time_trend(df_asr, group_by="sex",
               plot_title = "Colombia, Liver",
               smoothing = 0.3)
```
# plot ASR over year, by sex, with high smoothing.
csu_time_trend(df_asr, group_by="sex",
    plot_title = "Colombia, Liver",
    smoothing = 0.5)

# Plot embedded in a graphic device
csu_time_trend(df_asr, group_by="sex",
    plot_title = "Colombia, Liver",
    smoothing = 0.3)
csu_time_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 variable. Several format are accepted
1  "0-4"  0
2  "5-9"  5
3  "10-14"  10
... ...
17  "80-84"  80
18  "85+"  85

Missing age value must be precise in the option missing_age.
Last age group will always be considere without size (ie: 80+, 85+, etc..).

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, assume 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_group_cases csu_merge_cases_pop csu_asr csu_cumrisk csu_eapc csu_ageSpecific csu_ageSpecific_top csu_bar_top csu_time_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 <- subset(csu_registry_data_2,registry == 84020 & 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 30-34 years until 70-74 years.
csu_trendCohortPeriod(df_data=test,
```r
# plot Cohort-Period graph from 30-34 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)
```

```r
# 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)
```

```r
# Plot embedded in a graphic device
dev.off()
```

---

**Description**

csu_trend_legend legend option use in Rcan package for trends.
Usage

```r
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`. If there is already a plot in the device, use `plot.new()` to avoid overprint.
- **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

```r
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")
```
data_individual_file

Description

Data individual example for the function `csu_group_cases`
1 line = 1 cases.

Usage

data("data_individual_file")

Format

A data frame with 19284 observations on the following 10 variables.

- regcode: registry code
- reglabel: registry label
- sex: sex
- age: age
- doi: date of birth (yyyymmdd)
- site: ICD10 code
- histo: histology
- beh: behavior code
- grade: grade
- basis: basis

Details

This dataset provide an example how to group individual cases to 5 years data, grouped by ICD code and year using the function `csu_group_cases` and the database `ICD_group_GLOBOCAN`.

See Also

csu_group_cases ICD_group_GLOBOCAN ICD_group_CI5
Examples

```
data(ICD_group_GLOBOCAN)
data(data_individual_file)

# group individual data by
# 5 year age group

df_data_age <- csu_group_cases(data_individual_file,
    var_age="age",
    group_by=c("sex", "regcode", "reglabel", "site"))

# group individual data by
# 5 year age group
# ICD grouping from dataframe ICD_group_GLOBOCAN

df_data_icd <- csu_group_cases(data_individual_file,
    var_age="age",
    group_by=c("sex", "regcode", "reglabel"),
    df_ICD = ICD_group_GLOBOCAN,
    var_ICD ="site")
```

---

data_population_file  Population data example

Description

Population data example for the function `csu_merge_cases_pop`.

Usage

```
data("data_population_file")
```

Format

A data frame with 160 observations on the following 4 variables.

- **year**: year
- **sex**: sex
- **age_group**: 5 year age group
- **pop**: Count of population year

Details

This dataset provide an example to merge 5 years age grouped data with population data using the function `csu_merge_cases_pop`.
ICD_group_CI5

ICD10 group detailed example

Description

ICD10 group example for the function csu_group_cases based on CI5XI ICD grouping

Usage

data("ICD_group_CI5")

Format

A data frame with 97 observations on the following 2 variables.

ICD  ICD10 code
LABEL label for cancer group

Details

This dataset provide an example how to regroup ICD code using the function csu_group_cases
For instance this group

See Also
csu_merge_cases_pop

Examples

data(ICD_group_GLOBOCAN)
data(data_individual_file)
data(data_population_file)

#group individual data by
# 5 year age group
# ICD grouping from dataframe ICD_group_GLOBOCAN
# year (extract from date of incidence)

df_data_year <- csu_group_cases(data_individual_file,
   var_age="age",
   group_by=c("sex", "regcode", "reglabel"),
   df_ICD = ICD_group_GLOBOCAN,
   var_ICD ="site",
   var_year = "doi")

#Merge 5-years age grouped data with population by year (automatic) and sex

df_pop <- csu_merge_cases_pop(df_data_year, data_population_file, var_age = "age_group",
   var_cases = "cases", var_py = "pop", group_by = c("sex"))

---

ICD_group_CI5  ICD10 group detailed example
ICD_group_GLOBOCAN

ICD   LABEL
C18   COLORECTUM
C19   COLORECTUM
C20   COLORECTUM
C21   COLORECTUM

Will become:

ICD_group   LABEL
C18-C21    COLORECTUM

See: csu_group_cases

See Also

csu_group_cases ICD_group_GLOBOCAN

Examples

data(ICD_group_CI5)
data(data_individual_file)

#group individual data by
# 5 year age group
# Icd grouping from dataframe ICD_group_CI5

df_data_icd <- csu_group_cases(data_individual_file, var_age="age",
    group_by=c("sex", "regcode", "relabel"),
    df_ICD = ICD_group_CI5,
    var_ICD ="site")

ICD_group_GLOBOCAN ICD10 group example

Description

ICD10 group example for the function csu_group_cases based on GLOBOCAN

Usage

data("ICD_group_GLOBOCAN")
Format
A data frame with 97 observations on the following 2 variables.

ICD  ICD10 code
LABEL  label for cancer group

Details
This dataset provide an example how to regroup ICD code using the function `csu_group_cases`
For instance this group

<table>
<thead>
<tr>
<th>ICD</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>C18</td>
<td>COLORECTUM</td>
</tr>
<tr>
<td>C19</td>
<td>COLORECTUM</td>
</tr>
<tr>
<td>C20</td>
<td>COLORECTUM</td>
</tr>
<tr>
<td>C21</td>
<td>COLORECTUM</td>
</tr>
</tbody>
</table>

Will become:

<table>
<thead>
<tr>
<th>ICD_group</th>
<th>LABEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>C18-C21</td>
<td>COLORECTUM</td>
</tr>
</tbody>
</table>

See: `csu_group_cases`

See Also

`csu_group_cases` `ICD_group_CI5`

Examples

data(ICD_group_GLOBOCAN)
data(data_individual_file)

#group individual data by
# 5 year age group
# Icd grouping from dataframe ICD_group_GLOBOCAN
df_data_icd <- csu_group_cases(data_individual_file,
   var_age="age",
group_by=c("sex", "regcode", "reglabel"),
df_ICD = ICD_group_GLOBOCAN,
   var_ICD  ="site")
Index

+Topic **datasets**
  - csu_ci5_mean, 19
  - csu_CI5XI_data, 17
  - csu_registry_data_1, 30
  - csu_registry_data_2, 32
  - data_individual_file, 40
  - data_population_file, 41
  - ICD_group_CI5, 42
  - ICD_group_GLOBOCAN, 43

+Topic **package**
  - Rcan-package, 2

  csu_ageSpecific, 2, 4, 9, 12, 15, 18, 20–22, 24, 27, 30–32, 34, 37, 39
  csu_ageSpecific_top, 2, 6, 7, 12, 15, 22, 24, 27, 30–32, 34, 37
  csu_asr, 2, 6, 9, 10, 15, 18, 22, 24, 27, 30–32, 34, 37
  csu_bar_top, 2, 6, 9, 12, 14, 22, 24, 27, 30–32, 34, 37
  csu_ci5_mean, 5, 19
  csu_CI5XI_data, 17
  csu_cumrisk, 2, 6, 9, 12, 15, 18, 21, 22, 24, 27, 30–32, 34, 37
  csu_eapc, 2, 6, 9, 12, 15, 18, 22, 23, 27, 30–32, 34, 37
  csu_group_cases, 2, 6, 9, 12, 15, 22, 24, 25, 30, 34, 37, 40, 42–44
  csu_merge_cases_pop, 2, 6, 9, 12, 15, 22, 24, 27, 26, 27, 28, 34, 37, 41, 42
  csu_registry_data_1, 30
  csu_registry_data_2, 32
  csu_time_trend, 2, 6, 9, 12, 15, 22, 24, 27, 30, 32, 33, 37
  csu_trend_legend, 5, 33, 38
  csu_trendCohortPeriod, 2, 6, 9, 12, 15, 22, 24, 27, 30, 32, 34, 35

data_individual_file, 40
data_population_file, 41

factor, 8
ICD_group_CI5, 40, 42, 44
ICD_group_GLOBOCAN, 26, 40, 43, 43
loess, 33
Rcan (Rcan-package), 2
Rcan-package, 2