Package 'graphPAF'

October 20, 2022

Title Estimating and Displaying Population Attributable Fractions

Version 1.0.1

Description Estimation and display of various types of population attributable fraction and impact fractions. As well as the usual calculations of attributable fractions and impact fractions, functions are provided for attributable fraction nomograms and fan plots, continuous exposures, for pathway specific population attributable fractions, and for joint, average and sequential population attributable fractions.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.2.1

URL https://github.com/johnfergusonNUIG/graphPAF

BugReports https://github.com/johnfergusonNUIG/graphPAF/issues

Imports boot, ggplot2, ggrepel, gridExtra, gtools, MASS, reshape2, survival, splines, dplyr

Depends R (>= 3.5.0)

Suggests R.rsp

VignetteBuilder R.rsp

NeedsCompilation no

Author John Ferguson [aut, cre]

Maintainer John Ferguson <john.ferguson@universityofgalway.ie>

Repository CRAN

Date/Publication 2022-10-20 11:45:07 UTC

R topics documented:

`automatic_fit` *Automatic fitting of probability models in a pre-specified Bayesian network.*

Description

Main effects models are fit by default. For continuous variables, lm is used, for binary (numeric 0/1 variables), glm is used and for factor valued variables polr is used. For factors, ensure that the factor levels are ordered by increasing levels of risk. If interactions are required for certain models, it is advisable to populate the elements of model_list separately.

Usage

```
automatic_fit(
  data,
  parent_list,
  node_vec,
  prev = 0.09,
  common = "",
  spline_nodes = c(),
  df_spline_nodes = 3
)
```

Arguments

data	Data frame. A data frame containing variables used for fitting the models. Must contain all variables used in fitting
parent_list	A list. The ith element is the vector of variable names that are direct causes of ith variable in node_vec
node_vec	A vector corresponding to the nodes in the Bayesian network. This must be specified from root to leaves - that is ancestors in the causal graph for a particular node are positioned before their descendants. If this condition is false the function will return an error.
prev	Prevalence of disease. Set to NULL for cohort or cross sectional studies
common	character text for part of the model formula that doesn't involve any variable in node_vec. Useful for specifying confounders involved in all models automatically
spline_nodes	Vector of continuous variable names that are fit as splines (when involved as parents). Natural splines are used.
df_spline_nodes	How many degrees of freedom for each spline (Default 3). At the moment, this can not be specified separately for differing variables.

Value

A list of fitted models corresponding to node_vec and parent_vec.

Examples

```
# More complicated example (slower to run)
library(splines)
parent_exercise <- c("education")
parent_diet <- c("education")
parent_smoking <- c("education")
parent_alcohol <- c("education")
parent_stress <- c("education")
parent_high_blood_pressure <- c("education","exercise","diet",
"smoking","alcohol","stress")
parent_lipids <- c("education","exercise","diet","smoking",
"alcohol","stress")
parent_waist_hip_ratio <- c("education","exercise","diet","smoking",
"alcohol","stress")
parent_early_stage_heart_disease <- c("education","exercise","diet",
"smoking","alcohol","stress","lipids","waist_hip_ratio","high_blood_pressure")
parent_diabetes <- c("education","exercise","diet","smoking","alcohol",
"stress","lipids","waist_hip_ratio","high_blood_pressure")
parent_case <- c("education","exercise","diet","smoking","alcohol","stress",
"lipids","waist_hip_ratio","high_blood_pressure","early_stage_heart_disease","diabetes")
parent_list <- list(parent_exercise,parent_diet,parent_smoking,
parent_alcohol,parent_stress,parent_high_blood_pressure,
parent_lipids,parent_waist_hip_ratio,parent_early_stage_heart_disease,
parent_diabetes,parent_case)
node_vec=c("exercise","diet","smoking","alcohol","stress","high_blood_pressure",
```

```
"lipids","waist_hip_ratio","early_stage_heart_disease",
"diabetes","case")

model_list=automatic_fit(data=stroke_reduced, parent_list=parent_list,
node_vec=node_vec, prev=.0035,common="region*ns(age,df=5)+
sex*ns(age,df=5)", spline_nodes = c("waist_hip_ratio","lipids","diet"))
```

`average_paf` *Calculation of average and sequential paf taking into account risk factor sequencing*

Description

Calculation of average and sequential paf taking into account risk factor sequencing

Usage

```
average_paf(
  data,
  model_list,
  parent_list,
  node_vec,
  prev = 0.09,
  exact = TRUE,
  nperm = NULL,
  correct_order = 2,
  vars = NULL,
  ci = FALSE,
  boot_rep = 100,
  ci_type = c("norm"),
  ci_level = 0.95,
  ci_level_ME = 0.95,
  weight_vec = NULL
)
```

Arguments

`data`	Data frame. A dataframe containing variables used for fitting the models. Must contain all variables used in fitting
`model_list`	List. A list of fitted models corresponding for the outcome variables in node_vec, with parents as described in parent_vec. This list must be in the same order as node_vec and parent_list. Models can be linear (lm), logistic (glm) or ordinal logistic (polr). Non-linear effects of variables (if necessary) should be specified via ns(x, df=y), where ns is the natural spline function from the splines library.
`parent_list`	A list. The ith element is the vector of variable names that are direct causes of ith variable in node_vec (Note that the variable names should be columns in data)

node_vec	A character vector corresponding to the nodes in the Bayesian network (The variable names should be column names in data). This must be specified from root to leaves - that is ancestors in the causal graph for a particular node are positioned before their descendants. If this condition is false the function will return an error.
prev	numeric. Prevalence of disease. Only relevant to set for case control datasets.
exact	logical. Default TRUE. If TRUE, an efficient calculation is used to calculate average PAF, which enables the average PAF from N! permutations, over all N risk factors to be calculated with only 2^N-1 operations. If FALSE, permutations are sampled
nperm	Default NULL Number of random permutations used to calculate average and sequential PAF. If correct_order is set to an integer value, nperm is reset to an integer multiple of factorial(N)/factorial(N-correct_order) depending on the size of nperm. If nperm is NULL or less than factorial(N)/factorial(N-correct_order), factorial(N)/factorial(N-correct_order) permutations will be sampled. If nperm is larger than factorial(N)/factorial(N-correct_order), nperm will be reset to the smallest integer multiple of factorial(N)/factorial(N-correct_order) less than the input value of nperm
correct_order	Default 3. This enforces stratified sampling of permutations where the first correct_order positions of the sampled permutations are evenly distributed over the integers 1 ... N, N being the number of risk factors of interest, over the sampled permutations. The other positions are randomly sampled. This automatically sets the number of simulations when nperm=NULL. For interest, if N=10 and correct_order=3, nperm is set to factorial(10)/factorial(10-3) = 720. This special resampling reduces Monte Carlo variation in estimated average and sequential PAFs.
vars	A subset of risk factors for which we want to calculate average, sequential and joint PAF
ci	Logical. If TRUE, a bootstrap confidence interval is computed along with a point estimate (default FALSE). If ci=FALSE, only a point estimate is produced. A simulation procedure (sampling permutations and also simulating the effects of eliminating risk factors over the descendant nodes in a Bayesian network) is required to produce the point estimates. The point estimate will change on repeated runs of the function. The margin of error of the point estimate is given when ci=FALSE
boot_rep	Integer. Number of bootstrap replications (Only necessary to specify if ci=TRUE). Note that at least 50 replicates are recommended to achieve stable estimates of standard error. In the examples below, values of boot_rep less than 50 are sometimes used to limit run time.
ci_type	Character. Default norm. A vector specifying the types of confidence interval desired. "norm", "basic", "perc" and "bca" are the available methods
ci_level	Numeric. Default 0.95. A number between 0 and 1 specifying the level of the confidence interval (when ci=TRUE)
ci_level_ME	Numeric. Default 0.95. A number between 0 and 1 specifying the level of the margin of error for the point estimate (only revelant when ci=FALSE and exact=FALSE)

weight_vec — An optional vector of inverse sampling weights (note with survey data, the variance may not be calculated correctly if sampling isn't independent). Note that this vector will be ignored if prev is specified, and the weights will be calibrated so that the weighted sample prevalence of disease equals prev. This argument can be ignored if data has a column weights with correctly calibrated weights

Value

A SAF_summary object with average joint and sequential PAF for all risk factors in node_vec (or alternatively a subset of those risk factors if specified in vars).

References

Ferguson, J., O'Connell, M. and O'Donnell, M., 2020. Revisiting sequential attributable fractions. Archives of Public Health, 78(1), pp.1-9.

Ferguson, J., Alvarez-Iglesias, A., Newell, J., Hinde, J. and O'Donnell, M., 2018. Estimating average attributable fractions with confidence intervals for cohort and case–control studies. Statistical methods in medical research, 27(4), pp.1141-1152

Examples

```
library(splines)
library(survival)
library(parallel)
options(boot.parallel="snow")
options(boot.ncpus=2)
# The above could be set to the number of available cores on the machine
#  Simulated data on occupational and environmental exposure to chronic cough from Eide, 1995
# First specify the causal graph, in terms of the parents of each node.  Then put into a list
parent_urban.rural <- c()
parent_smoking.category <- c("urban.rural")
parent_occupational.exposure <- c("urban.rural")
parent_y <- c("urban.rural","smoking.category","occupational.exposure")
parent_list <- list(parent_urban.rural, parent_smoking.category,
 parent_occupational.exposure, parent_y)
# also specify nodes of graph, in order from root to leaves
node_vec <- c("urban.rural","smoking.category","occupational.exposure", "y")
# specify a model list according to parent_list
# here we use the auxillary function 'automatic fit'
model_list=automatic_fit(data=Hordaland_data, parent_list=parent_list,
 node_vec=node_vec, prev=.09)
# By default the function works by stratified simulation of permutations and
# subsequent simulation of the incremental interventions on the distribution of risk
# factors.  The permutations are stratified so each factor appears equally often in
# the first correct_order positions.  correct_order has a default of 2.

# model_list$data objects have fitting weights included
# Including weight column in data
# necessary if Bootstrapping CIs

out <- average_paf(data=model_list[[length(model_list)]]$data,
```

```
 model_list=model_list, parent_list=parent_list,
 node_vec=node_vec, prev=.09, nperm=10,vars = c("urban.rural",
 "occupational.exposure"),ci=FALSE)
 print(out)


# More complicated example (slower to run)
parent_exercise <- c("education")
parent_diet <- c("education")
parent_smoking <- c("education")
parent_alcohol <- c("education")
parent_stress <- c("education")
parent_high_blood_pressure <- c("education","exercise","diet","smoking",
"alcohol","stress")
parent_lipids <- c("education","exercise","diet","smoking","alcohol",
"stress")
parent_waist_hip_ratio <- c("education","exercise","diet","smoking",
"alcohol","stress")
parent_early_stage_heart_disease <- c("education","exercise","diet",
"smoking","alcohol","stress","lipids","waist_hip_ratio","high_blood_pressure")
parent_diabetes <- c("education","exercise","diet","smoking","alcohol",
"stress","lipids","waist_hip_ratio","high_blood_pressure")
parent_case <- c("education","exercise","diet","smoking","alcohol","stress",
"lipids","waist_hip_ratio","high_blood_pressure",
"early_stage_heart_disease","diabetes")
parent_list <- list(parent_exercise,parent_diet,parent_smoking,
parent_alcohol,parent_stress,parent_high_blood_pressure,
parent_lipids,parent_waist_hip_ratio,parent_early_stage_heart_disease,
parent_diabetes,parent_case)
node_vec=c("exercise","diet","smoking","alcohol","stress",
"high_blood_pressure","lipids","waist_hip_ratio","early_stage_heart_disease",
"diabetes","case")
model_list=automatic_fit(data=stroke_reduced, parent_list=parent_list,
 node_vec=node_vec, prev=.0035,common="region*ns(age,df=5)+sex*ns(age,df=5)",
  spline_nodes = c("waist_hip_ratio","lipids","diet"))
out <- average_paf(data=stroke_reduced, model_list=model_list,
parent_list=parent_list, node_vec=node_vec, prev=.0035,
vars = c("high_blood_pressure","smoking","stress","exercise","alcohol",
"diabetes","early_stage_heart_disease"),ci=TRUE,boot_rep=10)
print(out)
plot(out,max_PAF=0.5,min_PAF=-0.1,number_rows=3)
# plot sequential and average PAFs by risk factor
# similar calculation, but now sampling permutations (stratified, so
# that each risk factor will appear equally often in the first correct_order positions)
out <- average_paf(data=stroke_reduced, model_list=model_list,
parent_list=parent_list, node_vec=node_vec, prev=.0035, exact=FALSE,
 correct_order=2, vars = c("high_blood_pressure","smoking","stress",
 "exercise","alcohol","diabetes","early_stage_heart_disease"),ci=TRUE,
 boot_rep=10)
 print(out)
 plot(out,max_PAF=0.5,min_PAF=-0.1,number_rows=3)
```

`data_clean` *Clean a dataset to make model fitting more efficient*

Description

Strip out unneeded variables from original data (based on fitted model, or alternatively based on specifying a list of variables), and remove rows with NA values. The function works for logistic, survival and conditional logistic regressions. The function also creates a column of weights, which will be just a vector of 1s if prevalence is unspecified.

Usage

```
data_clean(data, model = NULL, vars = NULL, response = "case", prev = NULL)
```

Arguments

`data`	A data frame that was used to fit the model
`model`	A glm (with logistic or log link, with binomial family), clogit or coxph model.
`vars`	Default NULL. Variables required in output data set. If set to NULL and model is specified, the variables kept are the response and covariates assumed in model
`response`	Default "case". response variable in dataset. Used when recalculating weights (if the argument prev is set) If set to NULL, the response is inferred from the model
`prev`	Default NULL. Prevalence of disease (or yearly incidence of disease in healthy controls). Only relevant to set in case control studies and if path specific PAF or sequential joint PAF calculations are required. The purpose of this is to create a vector of weights that reweights the cases and controls to reflect the general population

Value

A cleaned data frame

Examples

```
# example of using dataclean to strip out NAs, redundant columns and recalculate weights
library(survival)
library(splines)
stroke_reduced_2 <- stroke_reduced
stroke_reduced_2$case[sample(1:length(stroke_reduced_2$case),50)] <- NA
stroke_reduced_2$random <- rnorm(length(stroke_reduced_2$case))
stroke_reduced_3 <- data_clean(stroke_reduced_2,vars=colnames(stroke_reduced),prev=0.01)
dim(stroke_reduced_2)
dim(stroke_reduced_3)
mymod <- clogit(case ~ high_blood_pressure + strata(strata),data=stroke_reduced_2)
stroke_reduced_3 <- data_clean(stroke_reduced_2,model=mymod,prev=0.01)
dim(stroke_reduced_2)
dim(stroke_reduced_3)
```

do_sim — *Internal: Simulate a column from the post intervention distribution corresponding to eliminating a risk factor*

Description

Internal: Simulate a column from the post intervention distribution corresponding to eliminating a risk factor

Usage

```
do_sim(colnum, current_mat, model, SN = TRUE)
```

Arguments

colnum	The column indicator for the variable being simulated
current_mat	The current value of the data frame
model	A fitted model for simulating values of the variable, given the parent values
SN	Logical. If TRUE (default) simulations are achieved via adding the original model residuals, to the new fitted values based on the updated values of parents in current_mat.

Value

An updated data frame (a new version of current_mat) with a new column simulated

graphPAF — *Estimating and Displaying Population Attributable Fractions*

Description

Estimation and display of various types of population attributable fraction and impact fractions. As well as the usual calculations of attributable fractions and impact fractions, functions are provided for attributable fraction nomograms and fan plots, continuous exposures, for pathway specific population attributable fractions, and for joint, average and sequential population attributable fractions.

References

Ferguson, J., O'Leary, N., Maturo, F., Yusuf, S. and O'Donnell, M., 2019. Graphical comparisons of relative disease burden across multiple risk factors. BMC medical research methodology, 19(1), pp.1-9

Ferguson, J., Maturo, F., Yusuf, S. and O'Donnell, M., 2020. Population attributable fractions for continuously distributed exposures. Epidemiologic Methods, 9(1).

Pathway specific Population attributable fractions. O'Connell, M.M. and Ferguson, J.P., 2022. IEA. International Journal of Epidemiology, 1, p.13.

Ferguson, J., O'Connell, M. and O'Donnell, M., 2020. Revisiting sequential attributable fractions. Archives of Public Health, 78(1), pp.1-9.

Ferguson, J., Alvarez-Iglesias, A., Newell, J., Hinde, J. and O'Donnell, M., 2018. Estimating average attributable fractions with confidence intervals for cohort and case–control studies. Statistical methods in medical research, 27(4), pp.1141-1152

`Hordaland_data` *Simulated case control dataset for 5000 cases (individuals with chronic cough) and 5000 controls*

Description

Simulated case control dataset for 5000 cases (individuals with chronic cough) and 5000 controls

Usage

```
Hordaland_data
```

Format

A data frame with 10000 rows and 4 variables:

y Chronic Cough, 1: Yes, 0: No

urban.rural 1: resident in urban setting, 0: resident in rural setting

smoking.category Smoking level: 1 No smoker, 2: ex smoker, 3: 1-9 cigarettes per day, 4: 10-19 cigarettes per day, 4:>= 20 cigarettes per day

occupational.exposure Exposed to dust/gas at work. 1: Yes, 0: no

Source

Data simulated based on "Sequential and average attributable fractions as aids in the selection of preventive strategies." Journal of clinical epidemiology 48, no. 5 (1995): 645-655.

if_bruzzi *Internal: Calculation of an impact fraction using the Bruzzi approach*

Description

Internal: Calculation of an impact fraction using the Bruzzi approach

Usage

```
if_bruzzi(data, ind, model, model_type, new_data, response, weight_vec)
```

Arguments

data	A dataframe containing variables used for fitting the model
ind	An indicator of which rows will be used from the dataset
model	Either a clogit or glm fitted model object. Non-linear effects should be specified via ns(x, df=y), where ns is the natural spline function from the splines library.
model_type	Either a "clogit", "glm" or "coxph" model object
new_data	A dataframe (of the same variables and size as data) representing an alternative distribution of risk factors
response	A string representing the name of the outcome variable in data
weight_vec	An optional vector of inverse sampling weights

Value

A numeric estimated impact fraction.

References

Bruzzi, P., Green, S.B., Byar, D.P., Brinton, L.A. and Schairer, C., 1985. Estimating the population attributable risk for multiple risk factors using case-control data. American journal of epidemiology, 122(5), pp.904-914

if_direct *Internal: Calculation of an impact fraction using the direct approach*

Description

Internal: Calculation of an impact fraction using the direct approach

Usage

```
if_direct(
  data,
  ind,
  model,
  model_type,
  new_data,
  prev,
  t_vector,
  response,
  weight_vec
)
```

Arguments

data	A dataframe containing variables used for fitting the model
ind	An indicator of which rows will be used from the dataset
model	Either a clogit, glm or coxph fitted model object. Non-linear effects should be specified via ns(x, df=y), where ns is the natural spline function from the splines library.
model_type	Either a "clogit", "glm" or "coxph" model object
new_data	A dataframe (of the same variables and size as data) representing an alternative distribution of risk factors
prev	Population prevalence of disease (default NULL)
t_vector	A vector of times at which PAF estimates are desired (for a coxph model)
response	A string representing the name of the outcome variable in data
weight_vec	An optional vector of inverse sampling weights

Value

A numeric estimated impact fraction.

---

impact_fraction *General calculations of impact fractions*

Description

General calculations of impact fractions

Usage

```
impact_fraction(
  model,
  data,
  new_data,
  calculation_method = "B",
  prev = NULL,
  ci = FALSE,
  boot_rep = 100,
  t_vector = NULL,
  ci_level = 0.95,
  ci_type = c("norm"),
  weight_vec = NULL
)
```

Arguments

model	Either a clogit, glm or coxph fitted model object. Non-linear effects should be specified via ns(x, df=y), where ns is the natural spline function from the splines library.
data	A dataframe containing variables used for fitting the model
new_data	A dataframe (of the same variables and size as data) representing an alternative distribution of risk factors
calculation_method	A character either 'B' (Bruzzi) or 'D' (Direct method). For case control data, the method described in Bruzzi 1985 is recommended. Bruzzi's method estimates PAF from relative risks and prevalence of exposure to the risk factor. The Direct method estimates PAF by summing estimated probabilities of disease in the absence of exposure on the individual level
prev	estimated prevalence of disease. This only needs to be specified if the data source is from a case control study, and the direct method is used
ci	Logical. If TRUE, a bootstrap confidence interval is computed along with point estimate (default FALSE)
boot_rep	Integer. Number of bootstrap replications (Only necessary to specify if ci=TRUE)
t_vector	Numeric. A vector of times at which to calculate PAF (only specified if model is coxph)
ci_level	Numeric. Default 0.95. A number between 0 and 1 specifying the confidence level
ci_type	Character. Defalt norm. A vector specifying the types of confidence interval desired. "norm", "basic", "perc" and "bca" are the available methods
weight_vec	An optional vector of inverse sampling weights for survey data (note that variance will not be calculated correctly if sampling isn't independent). Note that this vector will be ignored if prev is specified, and the weights will be calibrated so that the weighted sample prevalence of disease equals prev.

Value

A numeric estimated impact fraction if ci=FALSE, or for survival data a vector of estimated impact corresponding to event times in the data. If ci=TRUE, a vector with elements corresponding to the raw estimated impact fraction, estimated bias, bias corrected estimate and lower and upper elements of any confidence procedures requested. If ci=TRUE, and a coxph model is fit, a matrix will be returned, with rows corresponding to the times at which the impact fraction is calculated.

References

Bruzzi, P., Green, S.B., Byar, D.P., Brinton, L.A. and Schairer, C., 1985. Estimating the population attributable risk for multiple risk factors using case-control data. American journal of epidemiology, 122(5), pp.904-914

Examples

```
library(splines)
library(survival)
new_data <- stroke_reduced
N <- nrow(new_data)
inactive_patients <- (1:N)[stroke_reduced$exercise==1]
N_inactive <- sum(stroke_reduced$exercise)
newly_active_patients <- inactive_patients[sample(1:N_inactive,0.2*N_inactive)]
new_data$exercise[newly_active_patients] <- 0
model_exercise <- clogit(formula = case ~ age + education +exercise +
ns(diet, df = 3) + smoking + alcohol + stress + ns(lipids,df = 3) +
ns(waist_hip_ratio, df = 3) + high_blood_pressure +strata(strata),
data=stroke_reduced)
impact_fraction(model=model_exercise,stroke_reduced,new_data,
calculation_method = "B")
```

---

joint_paf *Calculation of joint attributable fractions over several risk factors taking into account risk factor sequencing*

Description

Calculation of joint attributable fractions over several risk factors taking into account risk factor sequencing

Usage

```
joint_paf(
  data,
  model_list,
  parent_list,
  node_vec,
  prev = NULL,
  vars = NULL,
```

```
  ci = FALSE,
  boot_rep = 100,
  ci_type = c("norm"),
  ci_level = 0.95,
  nsim = 1,
  weight_vec = NULL
)
```

Arguments

data	Data frame. A dataframe containing variables used for fitting the models. Must contain all variables used in fitting
model_list	List. A list of fitted models corresponding for the outcome variables in node_vec, with parents as described in parent_vec. This list must be in the same order as node_vec and parent_list. Non-linear effects should be specified via ns(x, df=y), where ns is the natural spline function from the splines library. Linear (lm), logistic (glm) and ordinal logistic (polr) models are permitted
parent_list	A list. The ith element is the vector of variable names that are direct causes of ith variable in node_vec
node_vec	A vector corresponding to the nodes in the Bayesian network. This must be specified from root to leaves - that is ancestors in the causal graph for a particular node are positioned before their descendants. If this condition is false the function will return an error.
prev	prevalence of the disease (default is NULL)
vars	A subset of risk factors for which we want to calculate joint PAF
ci	Logical. If TRUE, a bootstrap confidence interval is computed along with a point estimate (default FALSE). If ci=FALSE, only a point estimate is produced. A simulation procedure (sampling permutations and also simulating the effects of eliminating risk factors over the descendant nodes in a Bayesian network) is required to produce the point estimates. The point estimate will change on repeated runs of the function. The margin of error of the point estimate is given when ci=FALSE
boot_rep	Integer. Number of bootstrap replications (Only necessary to specify if ci=TRUE). Note that at least 50 replicates are recommended to achieve stable estimates of standard error. In the examples below, values of boot_rep less than 50 are sometimes used to limit run time.
ci_type	Character. Default norm. A vector specifying the types of confidence interval desired. "norm", "basic", "perc" and "bca" are the available method
ci_level	Numeric. Confidence level. Default 0.95
nsim	Numeric. Number of independent simulations of the dataset. Default of 1
weight_vec	An optional vector of inverse sampling weights (note with survey data, the variance may not be calculated correctly if sampling isn't independent). Note that this vector will be ignored if prev is specified, and the weights will be calibrated so that the weighted sample prevalence of disease equals prev. This argument can be ignored if data has a column weights with correctly calibrated weights

Value

A numeric estimate of the joint PAF for all risk factors (if ci=FALSE), or a data frame giving joint PAF and confidence intervals (if ci=TRUE)

References

Ferguson, J., O'Connell, M. and O'Donnell, M., 2020. Revisiting sequential attributable fractions. Archives of Public Health, 78(1), pp.1-9.

Examples

```
library(splines)
library(survival)
library(parallel)
options(boot.parallel="snow")
options(boot.ncpus=2)
# The above could be set to the number of available cores on the machine
#  Simulated data on occupational and environmental exposure to
# chronic cough from Eide, 1995
# First specify the causal graph, in terms of the parents of each node.
# Then put into a list.
parent_urban.rural <- c()
parent_smoking.category <- c("urban.rural")
parent_occupational.exposure <- c("urban.rural")
parent_y <- c("urban.rural","smoking.category","occupational.exposure")
parent_list <- list(parent_urban.rural, parent_smoking.category,
 parent_occupational.exposure, parent_y)
# also specify nodes of graph, in order from root to leaves
node_vec <- c("urban.rural","smoking.category","occupational.exposure", "y")
# specify a model list according to parent_list
# here we use the auxillary function 'automatic fit'
model_list=automatic_fit(data=Hordaland_data, parent_list=parent_list,
node_vec=node_vec, prev=.09)
# model_list$data objects have fitting weights included
# Including weight column in data
# necessary if Bootstrapping CIs
joint_paf(data=model_list[[length(model_list)]]$data,
 model_list=model_list, parent_list=parent_list,
 node_vec=node_vec, prev=.09, vars = c("urban.rural",
 "occupational.exposure"),ci=FALSE)

# More complicated example (slower to run)
parent_exercise <- c("education")
parent_diet <- c("education")
parent_smoking <- c("education")
parent_alcohol <- c("education")
parent_stress <- c("education")
parent_high_blood_pressure <- c("education","exercise","diet","smoking","alcohol","stress")
parent_lipids <- c("education","exercise","diet","smoking","alcohol","stress")
parent_waist_hip_ratio <- c("education","exercise","diet","smoking",
"alcohol","stress")
parent_early_stage_heart_disease <- c("education","exercise","diet",
```

```
"smoking","alcohol","stress","lipids","waist_hip_ratio","high_blood_pressure")
parent_diabetes <- c("education","exercise","diet","smoking","alcohol",
"stress","lipids","waist_hip_ratio","high_blood_pressure")
parent_case <- c("education","exercise","diet","smoking","alcohol",
"stress","lipids","waist_hip_ratio","high_blood_pressure","early_stage_heart_disease","diabetes")
parent_list <- list(parent_exercise,parent_diet,parent_smoking,parent_alcohol,
parent_stress,parent_high_blood_pressure,parent_lipids,parent_waist_hip_ratio,
parent_early_stage_heart_disease,parent_diabetes,parent_case)
node_vec=c("exercise","diet","smoking","alcohol","stress","high_blood_pressure",
"lipids","waist_hip_ratio","early_stage_heart_disease","diabetes","case")
model_list=automatic_fit(data=stroke_reduced, parent_list=parent_list,
node_vec=node_vec, prev=.0035,common="region*ns(age,df=5)+sex*ns(age,df=5)",
 spline_nodes = c("waist_hip_ratio","lipids","diet"))
jointpaf <- joint_paf(data=stroke_reduced, model_list=model_list,
parent_list=parent_list, node_vec=node_vec, prev=.0035,
vars = c("high_blood_pressure","smoking","stress","exercise","alcohol",
"diabetes","early_stage_heart_disease"),ci=TRUE,boot_rep=10)
```

PAF_calc_continuous *Calculation of attributable fractions with a continuous exposure*

Description

Calculation of attributable fractions with a continuous exposure

Usage

```
PAF_calc_continuous(
  model,
  riskfactor_vec,
  q_vec = c(0.01),
  data,
  calculation_method = "B",
  prev = NULL,
  ci = FALSE,
  boot_rep = 10,
  t_vector = NULL,
  ci_level = 0.95,
  ci_type = c("norm"),
  S = 1,
  weight_vec = NULL
)
```

Arguments

model	Either a clogit, glm or coxph R model object. Non-linear effects should be specified via ns(x, df=y), where ns is the natural spline function from the splines library.

`riskfactor_vec` A character vector of names for continuous exposures/riskfactors that are predictors the model.

`q_vec` A vector of 'risk quantiles' for the continuous exposure. q_vec=c(0.01) (the default) calculates an estimate of the PAF that is in some way analogous to eliminating a categorical risk factor. Other values in q_vec correspond to interventions on the continuous risk factor that results in a risk level for all individuals thresholded above by the corresponding quantile of pre-intervention population risk. For survival regressions only single element values of q_vec are allowed

`data` A dataframe containing variables used for fitting the model

`calculation_method`
A character either 'B' (Bruzzi) or 'D' (Direct method). For case control data, the method described in Bruzzi 1985 is recommended. Bruzzi's method estimates PAF from relative risks and prevalence of exposure to the risk factor. The Direct method estimates PAF via summing estimated probabilities of disease in the absence of exposure over differing individuals.

`prev` The estimated prevalence of disease (A number between 0 and 1). This only needs to be specified if the data source is from a case control study, and the direct calculation method is used

`ci` Logical. If TRUE, a bootstrap confidence interval is computed along with point estimate (default FALSE)

`boot_rep` Integer. Number of bootstrap replications (Only necessary to specify if ci=TRUE). Note that at least 50 replicates are recommended to achieve stable estimates of standard error. In the examples below, values of boot_rep less than 50 are sometimes used to limit run time.

`t_vector` Numeric. A vector of times at which to calculate PAF (only specified if model is coxph)

`ci_level` Numeric. A number between 0 and 1 specifying the confidence level

`ci_type` Character. A vector specifying the types of confidence interval desired, as available in the 'Boot' package. The default is c('norm'), which calculates a symmetric confidence interval: (Est-Bias +- 1.96*SE), with the standard error calculated via Bootstrap. Other choices are 'basic', 'perc' and 'bca'. Increasing the number of Bootstrap repetitions is recommended for the 'basic', 'perc' and 'bca' methods.

`S` Integer (default 1). Only relevant to change if there is an interaction between the continuous exposure and other variables in the model. In this case, marginal comparisons of disease risk at differing levels of the exposure need to be averaged over many individuals. S is the number of individuals used in this averaging. May be slow for large S

`weight_vec` An optional vector of inverse sampling weights for survey data (note that variance will not be calculated correctly if sampling isn't independent). Note that this vector will be ignored if prev is specified, and the weights will be calibrated so that the weighted sample prevalence of disease equals prev.

Value

A PAF_q object. When ci=FALSE, this will essentially be a vector of estimated PAF corresponding to the quantiles specified in q_vec. If ci=TRUE, a data frame with columns corresponding to the raw

estimate, estimated bias, bias corrected estimate and lower and upper elements of any confidence procedures requested, and rows corresponding to the quantiles in q_vec.

References

Ferguson, J., Maturo, F., Yusuf, S. and O'Donnell, M., 2020. Population attributable fractions for continuously distributed exposures. Epidemiologic Methods, 9(1).

Examples

```
library(splines)
library(survival)
library(parallel)
options(boot.parallel="snow")
options(boot.ncpus=2)
# The above could be set to the number of available cores on the machine
# Example with logistic regression.  PAF_q (as in Ferguson, 2020)
# estimated at q=0.01, 0.1, 0.3, 0.5, 0.7. 0.9.  PAF_0.01 is roughly
# analogous to 'eliminating' a discrete risk factor, but its estimation
# may be unstable for some exposures, and the corresponding intervention
# may be impractical.  Comparing PAF_q for q >= 0.1 over different risk factors
# may lead to more sensible comparisons of disease burden.
# Either method (direct, D, or Bruzzi )
# reduce dataset to improve run time (not recommended on real data!)
stroke_small <- stroke_reduced[sample(1:nrow(stroke_reduced),1000),]
model_continuous <- glm(formula = case ~ region * ns(age, df = 5) +
sex * ns(age, df = 5) + education +exercise + ns(diet, df = 3) +
alcohol + stress + ns(lipids,df = 3) + ns(waist_hip_ratio, df = 3) +
 high_blood_pressure, family = "binomial", data = stroke_small)
out <- PAF_calc_continuous(model_continuous,riskfactor_vec=
c("diet","lipids","waist_hip_ratio"),q_vec=c(0.1,0.5,0.9),
ci=FALSE,calculation_method="D",data=stroke_small, prev=0.0035)
print(out)
plot(out)

 # with confidence intervals (via bootstrap) on full dataset.  Slower.
model_continuous_clogit <- clogit(formula = case ~ region * ns(age, df = 5) +
sex * ns(age, df = 5) + education +exercise + ns(diet, df = 3)  +
alcohol + stress + ns(lipids,df = 3) + ns(waist_hip_ratio, df = 3) +
 high_blood_pressure + strata(strata), data = stroke_reduced)
out <- PAF_calc_continuous(model_continuous_clogit,riskfactor_vec=c("diet",
"lipids","waist_hip_ratio"),q_vec=c(0.01, 0.1,0.3,0.5,0.7,0.9),
ci=TRUE,calculation_method="B",data=stroke_reduced, prev=0.01)
print(out)
plot(out)
```

PAF_calc_discrete *Calculation of attributable fractions using a categorized risk factor*

Description

Calculation of attributable fractions using a categorized risk factor

Usage

```
PAF_calc_discrete(
  model,
  riskfactor,
  refval,
  data,
  calculation_method = "B",
  prev = NULL,
  ci = FALSE,
  boot_rep = 100,
  t_vector = NULL,
  ci_level = 0.95,
  ci_type = c("norm"),
  weight_vec = NULL
)
```

Arguments

model	Either a clogit, glm or coxph fitted regression object. Non-linear effects can be specified in these models if necessary via ns(x, df=y), where ns is the natural spline function from the splines library.
riskfactor	The name of the risk factor of interest in the dataset. The risk factor values can be 0/1 numeric, categorical or factor valued.
refval	The reference value for the risk factor. If a risk factor is 0/1 numeric, 0 is assumed as the default value, otherwise refval must be specified.
data	A dataframe containing variables used for fitting the model
calculation_method	A character either 'B' (Bruzzi) or 'D' (Direct method). For case control data, the method described in Bruzzi 1985 is recommended. Bruzzi's method estimates PAF from relative risks and prevalence of exposure to the risk factor. The Direct method estimates PAF by summing estimated probabilities of disease in the absence of exposure on the individual level
prev	The estimated prevalence of disease (A number between 0 and 1). This only needs to be specified if the data source is from a case control study, and the direct method is used
ci	Logical. If TRUE, a bootstrap confidence interval is computed along with point estimate (default FALSE)
boot_rep	Integer. Number of bootstrap replications (Only necessary to specify if ci=TRUE). Note that at least 50 replicates are recommended to achieve stable estimates of standard error. In the examples below, values of boot_rep less than 50 are sometimes used to limit run time.
t_vector	Numeric. A vector of times at which to calculate PAF (only specified if model is coxph)

ci_level	Numeric. A number between 0 and 1 specifying the confidence level
ci_type	Character. A vector specifying the types of confidence interval desired, as available in the 'Boot' package. The default is c('norm'), which calculates a symmetric confidence interval: (Est-Bias +- 1.96*SE), with the standard error calculated via Bootstrap. Other choices are 'basic', 'perc' and 'bca'. Increasing the number of Bootstrap repetitions is recommended for the 'basic', 'perc' and 'bca' methods.
weight_vec	An optional vector of inverse sampling weights for survey data (note that variance will not be calculated correctly if sampling isn't independent). Note that this will be ignored if prev is specified and calculation_method="D", in which case the weights will be constructed so the empirical re-weighted prevalence of disease is equal to prev.

Value

An estimated PAF if ci=FALSE, or for survival data a vector of estimated PAF corresponding to event times in the data. If ci=TRUE, a vector with elements corresponding to the raw estimate, estimated bias, bias corrected estimate and lower and upper elements of any confidence procedures requested. If ci=TRUE, and a coxph model is fit, a matrix will be returned, with rows corresponding to differing times at which the PAF might be calculated.

References

Bruzzi, P., Green, S.B., Byar, D.P., Brinton, L.A. and Schairer, C., 1985. Estimating the population attributable risk for multiple risk factors using case-control data. American journal of epidemiology, 122(5), pp.904-914

Examples

```
library(splines)
library(survival)
library(parallel)
options(boot.parallel="snow")
options(boot.ncpus=2)
# The above could be set to the number of available cores on the machine
data(stroke_reduced)
model_exercise <- glm(formula = case ~ region * ns(age, df = 5) +
 sex * ns(age, df = 5) + education + exercise + ns(diet, df = 3) +
 smoking + alcohol + stress, family = "binomial", data = stroke_reduced)
# calculate discrete PAF using Bruzzi method
PAF_calc_discrete(model_exercise, "exercise", refval=0,
data=stroke_reduced, calculation_method="B",ci=FALSE)

# calculate discrete PAF using Direct method
# Use bootstrap resampling to calculate a confidence interval
PAF_calc_discrete(model_exercise, "exercise", refval=0,
data=stroke_reduced, calculation_method="D", prev=0.005, ci=TRUE, boot_rep=10)
### use the Bruzzi method derived by Bruzzi, 1985, instead
PAF_calc_discrete(model_exercise, "exercise", refval=0, data=stroke_reduced,
 calculation_method="B", ci=TRUE, boot_rep=10)
```

```
# examples of clogit and coxph regressions

model_high_blood_pressure_clogit <- clogit(formula = case ~ age +
education +exercise + ns(diet, df = 3) + smoking + alcohol + stress +
 ns(lipids,df = 3) + ns(waist_hip_ratio, df = 3) + high_blood_pressure +
 strata(strata),data=stroke_reduced)
PAF_calc_discrete(model_high_blood_pressure_clogit, "high_blood_pressure",
refval=0, data=stroke_reduced, calculation_method="B",ci=TRUE, boot_rep=100,
 ci_type=c('norm'))

model_high_blood_pressure_coxph <- coxph(formula = Surv(time,event) ~
 ns(age,df=5) + education +exercise + ns(diet, df = 3) + smoking + alcohol +
  stress + ns(lipids,df = 3) + ns(waist_hip_ratio, df = 3) +
  high_blood_pressure, data = stroke_reduced)
PAF_calc_discrete(model_high_blood_pressure_coxph, "high_blood_pressure",
refval=0, data=stroke_reduced, calculation_method="D", ci=TRUE,
boot_rep=10, ci_type=c('norm'),t_vector=c(1,2,3,4,5,6,7,8,9))
```

`plot.PAF_q` *Plot impact fractions corresponding to risk-quantiles over several risk factors*

Description

Plot impact fractions corresponding to risk-quantiles over several risk factors

Usage

```
## S3 method for class 'PAF_q'
plot(x, ...)
```

Arguments

`x`	A PAF_q object. This is a dataframe that is created by running the function PAF_calc_continuous.
`...`	Other global arguments inherited by that might be passed to the ggplot routine

Value

A ggplot2 ploting object for PAF_q over the differing risk factors in x

Examples

```
library(splines)
library(survival)
library(parallel)
options(boot.parallel="snow")
options(boot.ncpus=2)
```

```
# The above could be set to the number of available cores on the machine
model_continuous <- glm(formula = case ~ region * ns(age, df = 5) +
sex * ns(age, df = 5) + education +exercise + ns(diet, df = 3) +
 alcohol + stress + ns(lipids,df = 3) + ns(waist_hip_ratio, df = 3) +
  high_blood_pressure, family = "binomial", data = stroke_reduced)
out <- PAF_calc_continuous(model_continuous,riskfactor_vec=
c("diet","lipids","waist_hip_ratio"),q_vec=c(0.1,0.9),
ci=FALSE,calculation_method="B",data=stroke_reduced)
plot(out)

# example with more quantile points and including confidence intervals
# (more useful - but a bit slower to run)
out <- PAF_calc_continuous(model_continuous,riskfactor_vec=
c("diet","lipids","waist_hip_ratio"),q_vec=c(0.01, 0.1,0.3,0.5,0.7,0.9),
ci=TRUE,calculation_method="B",data=stroke_reduced)
plot(out)
```

`plot.rf.data.frame` *Create a fan_plot of a rf.data.frame object*

Description

Create a fan plot displaying approximate PAF, risk factor prevalence and risk ratios

Usage

```
## S3 method for class 'rf.data.frame'
plot(
  x,
  type = "f",
  rf_prevmarks = c(0.02, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.7, 0.9),
  ormarks = c(1.05, 1.1, 1.4, 1.7, 2, 3),
  ...
)
```

Arguments

`x`	A rf.data.frame object
`type`	A character representing the type of plot. "f" for a fan_plot, "n" for a PAF nomogram and "rn" for a reverse PAF nomogram. See Ferguson et al.. "Graphical comparisons of relative disease burden across multiple risk factors." BMC medical research methodology 19, no. 1 (2019): 1-9 for more details
`rf_prevmarks`	Axis marks for risk factor prevalence (only used for type="n" and type = "rn") Default c(0.02, 0.05,0.1,0.2,0.3,0.4,0.5,0.7,0.9)
`ormarks`	Axis marks for odds ratios (only used for type="n" and type = "rn") Default c(1.05,1.1,1.4,1.7,2.0,3.0)
`...`	Other arguments that can be passed to the plotting routine

Value

fanplot or PAF nomogram (each is a ggplotting object)

References

Ferguson, J., O'Leary, N., Maturo, F., Yusuf, S. and O'Donnell, M., 2019. Graphical comparisons of relative disease burden across multiple risk factors. BMC medical research methodology, 19(1), pp.1-9.

Examples

```
rfs <- rf_summary(rf_names=c('Hypertension','Inactivity','ApoB/ApoA',
'Diet','WHR','Smoking','Cardiac causes','Alcohol','Global Stress','Diabetes'),
rf_prev=c(.474,.837,.669,.67,.67,.224,.049,.277,.144,.129),
risk=c(1.093,0.501,0.428,0.378,0.294,0.513,1.156,0.186,0.301,0.148),log=TRUE)
# fanplot
plot(rfs,type="f")
# nomogram
plot(rfs,type="n")
# reverse nomogram
plot(rfs,type="rn")
```

---

plot.SAF_summary *Produce plots of sequential and average PAF*

Description

Produce plots of sequential and average PAF

Usage

```
## S3 method for class 'SAF_summary'
plot(x, number_rows = 3, max_PAF = 0.4, min_PAF = 0, ...)
```

Arguments

x	An SAF_summary R object produced by running the average_paf function.
number_rows	integer How many rows of plots will be included on the associated figure.
max_PAF	upper limit of y axis on PAF plots (default = 0.4)
min_PAF	lower limit of y axis on PAF plots (default = 0)
...	Other global arguments inherited by that might be passed to the ggplot routine

Value

A ggplot2 plotting object illustrating average sequential PAF by position and average PAF by risk factor.

References

Ferguson, J., O'Connell, M. and O'Donnell, M., 2020. Revisiting sequential attributable fractions. Archives of Public Health, 78(1), pp.1-9. Ferguson, J., Alvarez-Iglesias, A., Newell, J., Hinde, J. and O'Donnell, M., 2018. Estimating average attributable fractions with confidence intervals for cohort and case–control studies. Statistical methods in medical research, 27(4), pp.1141-1152

Examples

```
library(splines)
library(survival)
library(parallel)
options(boot.parallel="snow")
options(boot.ncpus=2)
#  Simulated data on occupational and environmental exposure to
# chronic cough from Eide, 1995
# First specify the causal graph, in terms of the parents of each node.  Then put into a list
parent_urban.rural <- c()
parent_smoking.category <- c("urban.rural")
parent_occupational.exposure <- c("urban.rural")
parent_y <- c("urban.rural","smoking.category","occupational.exposure")
parent_list <- list(parent_urban.rural, parent_smoking.category,
parent_occupational.exposure, parent_y)
# also specify nodes of graph, in order from root to leaves
node_vec <- c("urban.rural","smoking.category","occupational.exposure", "y")
model_list=automatic_fit(Hordaland_data,
parent_list=parent_list, node_vec=node_vec, prev=.09)
out <- average_paf(data=model_list[[length(model_list)]]$data,
 model_list=model_list,
parent_list=parent_list, node_vec=node_vec, prev=.09, nperm=10,
vars = c("urban.rural","occupational.exposure"),ci=FALSE)
plot(out)

# plot with confidence intervals for average and sequential PAF
# (This is probably more useful for more than 2 risk factors).
# Separate axes for each risk factor so confidence intervals can be clearly displayed
out <- average_paf(data=model_list[[length(model_list)]]$data,
 model_list=model_list,
parent_list=parent_list, node_vec=node_vec, prev=.09, nperm=10,
vars = c("urban.rural","occupational.exposure"),ci=TRUE,boot_rep=8)
plot(out)
# Here we plot, with margin of error of point estimate when 50 permutations are used
out <- average_paf(data=model_list[[length(model_list)]]$data,
 model_list=model_list,
parent_list=parent_list, node_vec=node_vec, prev=.09, nperm=50,
vars = c("urban.rural","occupational.exposure"),ci=FALSE,exact=FALSE)
plot(out)
```

plot_continuous *Plot hazard ratios, odds ratios or risk ratios comparing differing values of a continuous exposure to a reference level*

Description

Plot hazard ratios, odds ratios or risk ratios comparing differing values of a continuous exposure to a reference level

Usage

```
plot_continuous(
  model,
  riskfactor,
  data,
  S = 10,
  ref_val = NA,
  ci_level = 0.95,
  min_risk_q = 0.1,
  plot_region = TRUE,
  plot_density = TRUE,
  n_x = 1000,
  theylab = "OR",
  qlist = seq(from = 0.001, to = 0.999, by = 0.001),
  interact = FALSE
)
```

Arguments

model	A fitted model (either glm, clogit or coxph)
riskfactor	The string name of a continuous exposure or risk factor represented in the data and model
data	Data frame used to fit the model
S	Default 10. The integer number of random samples used to calculate average differences in linear predictors. Only relevant to set when interact=TRUE
ref_val	The reference value used in plotting. If left at NA, the median value of the risk factor is used
ci_level	Numeric. A number between 0 and 1 specifying the confidence level
min_risk_q	Default .1. A number between 0 and 1 representing the desired risk quantile for the continuous risk factor
plot_region	Default TRUE. Logical specifying whether the targeted region corresponding to an intervention setting the continuous risk factor at a quantile min_risk_q or lower is to be plotted
plot_density	Default TRUE. Logical specifying whether density of distribution of risk factor is to be added to the plot

n_x	Default 1000. How many values of riskfactor will be used to plot spline (when interact=FALSE)
theylab	Default "OR". Y-axis label of the plot
qlist	Vector of quantile values for q, corresponding to the plotted values of PAF_q for each risk factor/exposure
interact	Default "FALSE". Set to TRUE spline models enter as interactions.

Value

A ggplot2 plotting object

References

Ferguson, J., Maturo, F., Yusuf, S. and O'Donnell, M., 2020. Population attributable fractions for continuously distributed exposures. Epidemiologic Methods, 9(1)

Examples

```
library(survival)
library(splines)
model_continuous <- glm(formula = case ~ region * ns(age, df = 5) +
 sex * ns(age, df = 5) + education +exercise + ns(diet, df = 3) +
 alcohol + stress + ns(lipids,df = 3) + ns(waist_hip_ratio, df = 3) +
  high_blood_pressure, family = "binomial", data = stroke_reduced)
plot_continuous(model_continuous,riskfactor="diet",data=stroke_reduced)
```

---

predict_df_continuous *Internal: Create a data frame for predictions (when risk factor is continuous).*

Description

Internal: Create a data frame for predictions (when risk factor is continuous).

Usage

```
predict_df_continuous(riskfactor, q_val, risk_q, data)
```

Arguments

riskfactor	The name of the risk factor of interest in the dataset
q_val	The risk quantile to match to
risk_q	Estimated risk quantiles
data	A dataframe containing variables used to fit the model

Value

A data frame where the distribution continuous risk factor so at an individual level, risk is at the q_val-quantile or below

predict_df_discrete *Internal: Create a data frame for predictions (when risk factor is discrete).*

Description

Internal: Create a data frame for predictions (when risk factor is discrete).

Usage

```
predict_df_discrete(riskfactor, refval, data)
```

Arguments

riskfactor	The name of the risk factor of interest in the dataset
refval	The reference value for the risk factor
data	A dataframe containing variables used to fit the model

Value

A data frame where the categorical variable is set to its reference level

print.PAF_q *Print out PAF_q for differing risk factors*

Description

Print out PAF_q for differing risk factors

Usage

```
## S3 method for class 'PAF_q'
print(x, ...)
```

Arguments

x	A PAF_q object. This is a dataframe that is created by running the function PAF_calc_continuous. The final 3 columns of the data frame are assumed to be (in order), PAF and lower and upper confidence bounds.
...	Other arguments to be passed to print

Value

No return value, prints the PAF_q object to the console.

Examples

```
library(splines)
library(survival)
library(parallel)
options(boot.parallel="snow")
options(boot.ncpus=2)
# The above could be set to the number of available cores on the machine
model_continuous <- glm(formula = case ~ region * ns(age, df = 5) +
sex * ns(age, df = 5) + education +exercise + ns(diet, df = 3) +
 alcohol + stress + ns(lipids,df = 3) + ns(waist_hip_ratio, df = 3) +
high_blood_pressure, family = "binomial", data = stroke_reduced)
out <- PAF_calc_continuous(model_continuous,
riskfactor_vec=c("diet","lipids","waist_hip_ratio"),
q_vec=c(0.01, 0.1,0.3,0.5,0.7,0.9),ci=FALSE,calculation_method="B",
data=stroke_reduced)
print(out)
```

---

print.SAF_summary *Print out a SAF_summary object*

Description

Print out a SAF_summary object

Usage

```
## S3 method for class 'SAF_summary'
print(x, ...)
```

Arguments

x	A SAF_summary object. This is a special dataframe that is created by running the function average_PAF.
...	Other arguments to be passed to print

Value

No return value. Prints the SAF_summary object to the console.

Examples

```
library(splines)
library(survival)
library(parallel)
options(boot.parallel="snow")
options(boot.ncpus=2)
# The above could be set to the number of available cores on the machine
#  Simulated data on occupational and environmental exposure to chronic cough from Eide, 1995
# First specify the causal graph, in terms of the parents of each node.  Then put into a list
```

```
parent_urban.rural <- c()
parent_smoking.category <- c("urban.rural")
parent_occupational.exposure <- c("urban.rural")
parent_y <- c("urban.rural","smoking.category","occupational.exposure")
parent_list <- list(parent_urban.rural, parent_smoking.category,
parent_occupational.exposure, parent_y)
node_vec <- c("urban.rural","smoking.category","occupational.exposure", "y")
model_list=automatic_fit(data=Hordaland_data, parent_list=parent_list,
 node_vec=node_vec, prev=.09)
# model_list$data objects have fitting weights
# included in data frame
# Including weight column in data
# necessary if Bootstrapping CIs
out <- average_paf(data=model_list[[length(model_list)]]$data,
model_list=model_list,
parent_list=parent_list, node_vec=node_vec, prev=.09, nperm=10,
vars = c("urban.rural","occupational.exposure"),ci=FALSE)
print(out)
```

`pspaf_discrete` *Internal, pathway specific PAF when the mediator is discrete*

Description

Internal, pathway specific PAF when the mediator is discrete

Usage

```
pspaf_discrete(
  data,
  refval,
  riskfactor_col,
  mediator_col,
  mediator_model,
  response_model,
  weight_vec
)
```

Arguments

`data`	dataframe. A dataframe (with no missing values) containing the data used to fit the mediator and response models. You can run data_clean to the input dataset if the data has missing values as a pre-processing step
`refval`	For factor valued risk factors, the reference level of the risk factor. If the risk factor is numeric, the reference level is assumed to be 0
`riskfactor_col`	Integer indicator for the risk factor column in data
`mediator_col`	Integer indicator for the discrete mediator column in data

`mediator_model` A glm or polr model for the mediator, depending on the same confounders and risk factor as specified in the response model.

`response_model` A R model object for a binary outcome that involves a risk factor, confounders and mediators of the risk factor outcome relationship. Note that a weighted model should be used for case control data. Non-linear effects should be specified via ns(x, df=y), where ns is the natural spline function from the splines library.

`weight_vec` A numeric column of weights

Value

A numeric vector (if ci=FALSE), or data frame (if CI=TRUE) containing estimated PS-PAF for each mediator referred to in mediator_models, together with estimated direct PS-PAF and possibly confidence intervals.

`ps_paf` *Estimate pathway specific population attributable fractions*

Description

Estimate pathway specific population attributable fractions

Usage

```
ps_paf(
  response_model,
  mediator_models,
  riskfactor,
  refval,
  data,
  prev = NULL,
  ci = FALSE,
  boot_rep = 100,
  ci_level = 0.95,
  ci_type = c("norm"),
  weight_vec = NULL
)
```

Arguments

`response_model` A R model object for a binary outcome that involves a risk factor, confounders and mediators of the risk factor outcome relationship. Note that a weighted model should be used for case control data. Non-linear effects should be specified via ns(x, df=y), where ns is the natural spline function from the splines library.

`mediator_models`
: A list of fitted models describing the risk factor/mediator relationship (the predictors in the model will be the risk factor and any confounders) Note a weighted model should be fit when data arise from a case control study. Models can be specified for linear responses (lm), binary responses (glm) and ordinal factors (through polr). Non-linear effects should be specified via ns(x, df=y), where ns is the natural spline function from the splines library.

`riskfactor`
: character. Represents the name of the risk factor

`refval`
: For factor valued risk factors, the reference level of the risk factor. If the risk factor is numeric, the reference level is assumed to be 0.

`data`
: dataframe. A dataframe (with no missing values) containing the data used to fit the mediator and response models. You can run data_clean to the input dataset if the data has missing values as a pre-processing step

`prev`
: numeric. A value between 0 and 1 specifying the prevalence of disease: only relevant to set if data arises from a case control study.

`ci`
: logical. If TRUE a confidence interval is calculated using Bootstrap

`boot_rep`
: Integer. Number of bootstrap replications (Only necessary to specify if ci=TRUE). Note that at least 50 replicates are recommended to achieve stable estimates of standard error. In the examples below, values of boot_rep less than 50 are sometimes used to limit run time.

`ci_level`
: Numeric. Default 0.95. A number between 0 and 1 specifying the confidence level (only necessary to specify when ci=TRUE)

`ci_type`
: Character. Defalt norm. A vector specifying the types of confidence interval desired. "norm", "basic", "perc" and "bca" are the available methods

`weight_vec`
: An optional vector of inverse sampling weights for survey data (note that variance will not be calculated correctly if sampling isn't independent). Note that this will be ignored if prev is specified and calculation_method="D", in which case the weights will be constructed so the empirical re-weighted prevalence of disease is equal to prev

Value

A numeric vector (if ci=FALSE), or data frame (if CI=TRUE) containing estimated PS-PAF for each mediator referred to in mediator_models, together with estimated direct PS-PAF and possibly confidence intervals.

References

Pathway specific Population attributable fractions. O'Connell, M.M. and Ferguson, J.P., 2022. IEA. International Journal of Epidemiology, 1, p.13. Accessible at: https://academic.oup.com/ije/advance-article/doi/10.1093/ije/dyac079/6583255?login=true

Examples

```
library(splines)
library(survival)
library(parallel)
```

```
options(boot.parallel="snow")
# User could set the next option to number of cores on machine:
options(boot.ncpus=2)
# Direct and pathway specific attributable fractions estimated
# on simulated case control stroke data:
# Note that the models here are weighted regressions (based on a column in the
# dataframe named 'weights') to rebalance the case control structure to make it
# representative over the population, according to the prev argument.
# Unweighted regression is fine to use if the data arises from cohort or
# cross sectional studies, in which case prev should be set to NULL
response_model <- glm(case ~ region * ns(age, df = 5) + sex * ns(age, df = 5) +
 education + exercise + ns(diet, df = 3) + smoking + alcohol + stress +
  ns(lipids, df = 3) + ns(waist_hip_ratio, df = 3) + high_blood_pressure,
  data=stroke_reduced,family='binomial', weights=weights)
mediator_models <- list(glm(high_blood_pressure ~ region * ns(age, df = 5) +
 sex * ns(age, df = 5) + education   +exercise + ns(diet, df = 3) + smoking +
 alcohol + stress,data=stroke_reduced,family='binomial',weights=weights),
 lm(lipids ~ region * ns(age, df = 5) + sex * ns(age, df = 5) +education +
  exercise + ns(diet, df = 3) + smoking + alcohol + stress, weights=weights,
   data=stroke_reduced),lm(waist_hip_ratio ~ region * ns(age, df = 5) +
   sex * ns(age, df = 5) + education + exercise + ns(diet, df = 3) +
    smoking + alcohol + stress, weights=weights, data=stroke_reduced))
# point estimate
ps_paf(response_model=response_model, mediator_models=mediator_models ,
riskfactor="exercise",refval=0,data=stroke_reduced,prev=0.0035, ci=FALSE)
# confidence intervals

ps_paf(response_model=response_model, mediator_models=mediator_models ,
riskfactor="exercise",refval=0,data=stroke_reduced,prev=0.0035, ci=TRUE,
boot_rep=100,ci_type="norm")
```

`rf_summary` *Create a rf.data.frame object*

Description

Create a rf.data.frame object for risk factors, prevalence and risk ratios. This will be used in fan plots and nomograms (by simply sending the rf.dat.frame object to plot)

Usage

```
rf_summary(rf_names, rf_prev, risk, log = FALSE)
```

Arguments

`rf_names`	A character vector of risk factor names
`rf_prev`	A numeric vector specifying prevalence of risk factor in disease controls (estimates of population prevalence can also be used if the disease is rare)

risk	A numeric vector of relative risks or Odds ratios for disease corresponding to each risk factor (if log=FALSE). Log-relative risks or log-odds ratios can be alternatively specified (if log=TRUE)
log	default TRUE. Set to TRUE if relative risks/odds ratios are specified on log-scale

Value

A rf.data.frame object

References

Ferguson, J., O'Leary, N., Maturo, F., Yusuf, S. and O'Donnell, M., 2019. Graphical comparisons of relative disease burden across multiple risk factors. BMC medical research methodology, 19(1), pp.1-9.

Examples

```
rfs <- rf_summary(rf_names=c('Hypertension','Inactivity','ApoB/ApoA','Diet',
'WHR','Smoking','Cardiac causes','Alcohol','Global Stress','Diabetes'),
rf_prev=c(.474,.837,.669,.67,.67,.224,.049,.277,.144,.129),
risk=c(1.093,0.501,0.428,0.378,0.294,0.513,1.156,0.186,0.301,0.148),log=TRUE)
# fanplot
plot(rfs,type="f")
# nomogram
plot(rfs,type="n")
# reverse nomogram
# plot(rfs,type="rn")
```

---

risk_quantiles *Return the vector of risk quantiles for a continuous risk factor.*

Description

Return the vector of risk quantiles for a continuous risk factor.

Usage

```
risk_quantiles(
  riskfactor,
  data,
  model,
  S = 1,
  q = seq(from = 0.01, to = 0.99, by = 0.01)
)
```

Arguments

riskfactor	The name of the risk factor of interest in the dataset. A character vector
data	A dataframe containing variables used to fit the model
model	The fitted model
S	The number of randomly selected individuals for which risk is measured (defaults to 1). Let to perhaps 100 if risk factor involved in interactions in model
q	The desired risk quantiles

Value

A named vector of size S giving the risk factor quantiles

---

seq_paf *Calculation of sequential paf taking into account risk factor sequencing*

Description

Calculation of sequential paf taking into account risk factor sequencing

Usage

```
seq_paf(
  data,
  model_list,
  parent_list,
  node_vec,
  prev = NULL,
  vars = NULL,
  ci = FALSE,
  boot_rep = 100,
  ci_type = c("norm"),
  ci_level = 0.95,
  nsim = 1,
  weight_vec = NULL
)
```

Arguments

data	Data frame. A dataframe containing variables used for fitting the models. Must contain all variables used in fitting
model_list	List. A list of fitted model objects corresponding for the outcome variables in node_vec, with parents as described in parent_vec. Linear (lm), logistic (glm) and ordinal (polr) objects are allowed. This list must be in the same order as node_vec and parent_list. Non-linear effects should be specified via ns(x, df=y), where ns is the natural spline function from the splines library.

parent_list	A list. The ith element is the vector of variable names that are direct causes of ith variable in node_vec
node_vec	A vector corresponding to the nodes in the Bayesian network. This must be specified from root to leaves - that is ancestors in the causal graph for a particular node are positioned before their descendants. If this condition is false the function will return an error.
prev	prevalence of the disease (default is NULL)
vars	A character vector of riskfactors. Sequential PAF is calculated for the risk factor specified in the last position of the vector, conditional on the other risk factors
ci	Logical. If TRUE, a bootstrap confidence interval is computed along with a point estimate (default FALSE). If ci=FALSE, only a point estimate is produced. A simulation procedure (sampling permutations and also simulating the effects of eliminating risk factors over the descendant nodes in a Bayesian network) is required to produce the point estimates. The point estimate will change on repeated runs of the function. The margin of error of the point estimate is given when ci=FALSE
boot_rep	Integer. Number of bootstrap replications (Only necessary to specify if ci=TRUE). Note that at least 50 replicates are recommended to achieve stable estimates of standard error. In the examples below, values of boot_rep less than 50 are sometimes used to limit run time.
ci_type	Character. Default norm. A vector specifying the types of confidence interval desired. "norm", "basic", "perc" and "bca" are the available methods
ci_level	Numeric. Confidence level. Default 0.95
nsim	Numeric. Number of independent simulations of the dataset. Default of 1
weight_vec	An optional vector of inverse sampling weights (note with survey data, the variance may not be calculated correctly if sampling isn't independent). Note that this vector will be ignored if prev is specified, and the weights will be calibrated so that the weighted sample prevalence of disease equals prev. This argument can be ignored if data has a column weights with correctly calibrated weights

Value

A numeric estimate of sequential PAF (if ci=FALSE), or else a data frame giving estimates and confidence limits of sequential PAF (if ci=TRUE)

References

Ferguson, J., O'Connell, M. and O'Donnell, M., 2020. Revisiting sequential attributable fractions. Archives of Public Health, 78(1), pp.1-9.

Examples

```
library(splines)
library(survival)
library(parallel)
options(boot.parallel="snow")
options(boot.ncpus=2)
```

```
# The above could be set to the number of available cores on the machine

# Simulated data on occupational and environmental exposure to
# chronic cough from Eide, 1995
# First specify the causal graph, in terms of the parents of each node.
# Then put into a list.
parent_urban.rural <- c()
parent_smoking.category <- c("urban.rural")
parent_occupational.exposure <- c("urban.rural")
parent_y <- c("urban.rural","smoking.category","occupational.exposure")
parent_list <- list(parent_urban.rural, parent_smoking.category,
 parent_occupational.exposure, parent_y)
# also specify nodes of graph, in order from root to leaves
node_vec <- c("urban.rural","smoking.category","occupational.exposure", "y")
# specify a model list according to parent_list
# here we use the auxillary function 'automatic fit'
model_list=automatic_fit(data=Hordaland_data, parent_list=parent_list,
node_vec=node_vec, prev=.09)
# sequential paf for occupational exposure conditional on elimination of urban.rural
# Including weight column in data
# necessary if Bootstrapping CIs
seq_paf(data=model_list[[length(model_list)]]$data,
model_list=model_list, parent_list=parent_list,
 node_vec=node_vec, prev=.09, vars = c("urban.rural",
 "occupational.exposure"),ci=FALSE)

# More complicated example (slower to run)
parent_exercise <- c("education")
parent_diet <- c("education")
parent_smoking <- c("education")
parent_alcohol <- c("education")
parent_stress <- c("education")
parent_high_blood_pressure <- c("education","exercise","diet","smoking","alcohol",
"stress")
parent_lipids <- c("education","exercise","diet","smoking","alcohol","stress")
parent_waist_hip_ratio <- c("education","exercise","diet","smoking",
"alcohol","stress")
parent_early_stage_heart_disease <- c("education","exercise","diet",
"smoking","alcohol","stress","lipids","waist_hip_ratio","high_blood_pressure")
parent_diabetes <- c("education","exercise","diet","smoking","alcohol",
"stress","lipids","waist_hip_ratio","high_blood_pressure")
parent_case <- c("education","exercise","diet","smoking","alcohol",
"stress","lipids","waist_hip_ratio","high_blood_pressure",
"early_stage_heart_disease","diabetes")
parent_list <- list(parent_exercise,parent_diet,parent_smoking,parent_alcohol,
parent_stress,parent_high_blood_pressure,parent_lipids,parent_waist_hip_ratio,
parent_early_stage_heart_disease,parent_diabetes,parent_case)
node_vec=c("exercise","diet","smoking","alcohol","stress","high_blood_pressure",
"lipids","waist_hip_ratio","early_stage_heart_disease","diabetes","case")
model_list=automatic_fit(data=stroke_reduced, parent_list=parent_list,
node_vec=node_vec, prev=.0035,common="region*ns(age,df=5)+sex*ns(age,df=5)",
 spline_nodes = c("waist_hip_ratio","lipids","diet"))
# calculate sequential PAF for stress, conditional on smoking
```

```
# and blood pressure being eliminated from the population
seqpaf <- seq_paf(data=stroke_reduced, model_list=model_list, parent_list=
parent_list, node_vec=node_vec, prev=.0035, vars = c("high_blood_pressure",
"smoking","stress"),ci=TRUE,boot_rep=10)
```

`sim_outnode` *Internal: Simulate from the post intervention distribution corresponding to eliminating a risk factor*

Description

Internal: Simulate from the post intervention distribution corresponding to eliminating a risk factor

Usage

```
sim_outnode(data, col_num, current_mat, parent_list, col_list, model_list)
```

Arguments

`data`	Data frame. A dataframe containing the original variables used for fitting the models. Must contain all variables used in fitting
`col_num`	The indicator for the risk factor that is being eliminated
`current_mat`	The current value of the data frame
`parent_list`	A list. The ith element is the vector of variable names that are direct causes of ith variable in node_vec (Note that the variable names should be columns in data)
`col_list`	Column indicators for the variables in node_vec (note that node_vec is ordered from root to leaves)
`model_list`	List. A list of fitted models corresponding for the outcome variables in node_vec, with parents as described in parent_vec. This list must be in the same order as node_vec and parent_list. Models can be linear (lm), logistic (glm) or ordinal logistic (polr). Non-linear effects of variables (if necessary) should be specified via ns(x, df=y), where ns is the natural spline function from the splines library

Value

An updated data frame (a new version of current_mat) with new columns simulated for variables that the risk factor causally effects.

`stroke_reduced` *Simulated case control dataset for 6856 stroke cases and 6856 stroke controls*

Description

Dataset containing simulated data on risk factors for 6856 stroke cases and 6856 stroke control, based on risk factors and associations in the INTERSTROKE study

Usage

```
stroke_reduced
```

Format

A data frame with 13712 rows and 19 variables:

region Geographic region, 1: Western Europe, 2: Eastern/central Europe/Middle East 3: Africa, 4: South Asia, 5: China, 6: South East Asia, 7: South America

case case control status, (1 for stroke cases)

sex Gender of individual, 0: male, 1:female

age Age of individual

smoking Smoking status, 0: Never, 1: Current

stress 1: sometimes stressed, 0: never stressed

waist_hip_ratio Waist hip ratio

exercise Physical Activity. 1: mainly inactive, 0: mainly active

alcohol Alcohol history and frequency, 1:never, 2:low/moderate, 3:high intake

diabetes Diabetes, 0: No, 1: Yes

diet Healthy eating score (higher is better)

early_stage_heart_disease presence of risk factors for heart disease. 0: No, 1: yes

lipids Ratio of Apolipoprotein B to Apolipoprotein A

education Years of education. 1: No education, 2: 1-8 years, 3:9-12 years, 3:Technical college, 4: University

high_blood_pressure Diagnosed hypertension: 0 No, 1: yes

weights weights that are proportional to inverse sampling probabilities. We have scaled the weights to be 0.005 for a case and 0.995 for a control to reflect any approximate incidence of 1 serious stroke in every 200 person years in the population

time simulated time variable (for illustrating survival models)

event simulated event indicator (0 if censored, 1 if event happened): for illustrating survival models

strata Strata number based on sex and region. For illustrating conditional regression

Source

Data simulated based on relationships described in https://www.thelancet.com/journals/lancet/article/PIIS0140-6736(16)30506-2/fulltext

Index