Package 'cases'

May 8, 2026

Type Package

Title Stratified Evaluation of Subgroup Classification Accuracy

Version 0.2.0

Description Enables simultaneous statistical inference for the accuracy of multiple classifiers in multiple subgroups (strata). For instance, allows to perform multiple comparisons in diagnostic accuracy studies with co-primary endpoints sensitivity and specificity (Westphal M, Zapf A. Statistical inference for diagnostic test accuracy studies with multiple comparisons. Statistical Methods in Medical Research. 2024;0(0). <doi:10.1177/09622802241236933>).

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports bindata, boot, copula, corrplot, dplyr, extraDistr, magrittr, Matrix, multcomp, mvtnorm, ggplot2

Suggests testthat (>= 3.0.0), knitr, rmarkdown, covr, badger, glmnet, splitstackshape

RoxygenNote 7.3.2

VignetteBuilder knitr, rmarkdown

Config/testthat/edition 3

URL https://github.com/maxwestphal/cases, https://maxwestphal.github.io/cases/

BugReports https://github.com/maxwestphal/cases/issues

Depends R (>= 2.10)

NeedsCompilation no

Author Max Westphal [aut, cre] (ORCID: <https://orcid.org/0000-0002-8488-758X>)

Maintainer Max Westphal <dev@maxwestphal.io>

Repository CRAN

Date/Publication 2025-01-09 17:50:02 UTC

Contents

`cases` `cases` *package*

Description

Enables simultaneous statistical inference for the accuracy of multiple classifiers in multiple subgroups (strata). For instance, allows to perform multiple comparisons in diagnostic accuracy studies with co-primary endpoints sensitivity (true positive rate, TPR) and specificity (true negative rate, TNR).

Details

The package functionality and syntax is described in the vignettes, see examples.

Author(s)

Maintainer: Max Westphal <dev@maxwestphal.io> (ORCID)

References

Westphal M, Zapf A. Statistical inference for diagnostic test accuracy studies with multiple comparisons. Statistical Methods in Medical Research. 2024;0(0). doi:10.1177/09622802241236933

See Also

Useful links:

- https://github.com/maxwestphal/cases
- https://maxwestphal.github.io/cases/
- Report bugs at https://github.com/maxwestphal/cases/issues

Examples

```
# overview over package functionality:
vignette("package_overview")

# a real-world data example:
vignette("example_wdbc")
```

`categorize` *Categorize continuous values*

Description

This function allows to split continuous values, e.g. (risk) scores or (bio)markers, into two or more categories by specifying one or more cutoff values.

Usage

```
categorize(
  values,
  cutoffs = rep(0, ncol(values)),
  map = 1:ncol(values),
  labels = NULL
)
```

Arguments

`values`	(matrix) numeric matrix of continuous values to be categorized. Assume an (n x r) matrix with n observations (subjects) of r continuous values.
`cutoffs`	(numeric | matrix) numeric matrix of dimension m x k. Each row of cutoffs defines a split into k+1 distinct categories. Each row must contain distinct values. In the simplest case (k=1), cutoffs is a single column matrix whereby each row defines a binary split (<=t vs. >t). In this case (k=1), cutoffs can also be a numeric vector.
`map`	(numeric) integer vector of length k with values in 1:r, whereby r = ncol(values). map_l gives the value which column of values should be categorized by ...
`labels`	(character) character of length m (= number of prediction r)

Value

(matrix)
numeric (n x k) matrix with categorical outcomes after categorizing.

Examples

```
set.seed(123)
M <- as.data.frame(mvtnorm::rmvnorm(20, mean = rep(0, 3), sigma = 2 * diag(3)))
M
categorize(M)
C <- matrix(rep(c(-1, 0, 1, -2, 0, 2), 3), ncol = 3, byrow = TRUE)
C
w <- c(1, 1, 2, 2, 3, 3)
categorize(M, C, w)
```

---

compare *Compare predictions and labels*

Description

Compare predictions and labels

Usage

```
compare(
  predictions,
  labels,
  partition = TRUE,
  names = c(specificity = 0, sensitivity = 1)
)
```

Arguments

predictions	(numeric | character) vector of class predictions, class and unique values+ need to match those of labels.
labels	(numeric | character) vector of true class labels (reference standard)
partition	(logical) should result be split into one matrix per class (TRUE; default) or not (FALSE)
names	(character | NULL) named character. Values specify data values, names specify class names. If names=NULL, the values and names are defined as unique(labels)

Value

(list | matrix)
list of matrices (one for each unique value of `labels`) with values 1 (correct prediction) and 0 (false prediction). If `partition=TRUE`, the matrices are combined in a single matrix with `rbind`.

Examples

```
pred <- matrix(c(1, 1, 0), 5, 3)
labels <- c(1, 1, 0, 0, 1)
compare(pred, labels, FALSE)
compare(pred, labels, TRUE)
```

---

cormat_ar1 *Create an AR(1) correlation matrix*

Description

Create an AR(1) correlation matrix

Usage

```
cormat_ar1(m, rho, d = TRUE)
```

Arguments

`m`	(numeric) dimensions of the (square) matrix
`rho`	(numeric) correlation parameter in (0,1)
`d`	(logical | numeric) binary vector of length m, whereby TRUE/FALSE (alternatively 1/0) indicate active/inactive components of underlying random vector.

Value

(matrix)
AR(1) correlation matrix R with entries $R_{ij} = \rho^{|i-j|}$

cormat_equi *Create an equicorrelation matrix*

Description

Create an equicorrelation matrix

Usage

```
cormat_equi(m, rho, d = TRUE)
```

Arguments

m	(numeric) dimensions of the (square) matrix
rho	(numeric) correlation parameter in (0,1)
d	(logical | numeric) binary vector of length m, whereby TRUE/FALSE (alternatively 1/0) indicate active/inactive components of underlying random vector.

Value

(matrix)
AR(1) correlation matrix R with entries $R_{ij} = \rho, i \neq j$

data_wdbc *Breast Cancer Wisconsin (Diagnostic) Data Set*

Description

Dataset documentation can be found at the source website and references below.

Usage

```
data_wdbc
```

Format

`data_wdbc`**:**
A data frame with 569 rows (patients) and 31 columns (1 target, 30 features).

Details

The ID variable was removed. Diagnosis (1= malignant, 0 = benign). Feature variables have been renamed.

Source

https://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(diagnostic)

References

- W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993.
- O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), pages 570-577, July-August 1995.

`define_contrast` *Define a contrast (matrix) to specify exact hypothesis system*

Description

Define a contrast (matrix) to specify exact hypothesis system

Usage

```
define_contrast(type = c("raw", "one", "all"), comparator = NA)
```

Arguments

`type`	(character) either "raw", "one" or "all", see details.
`comparator`	(numeric | character) either integer (index of comparator) or character (name of comparator)

Details

- "raw" contrast: compare all candidates against specified benchmark values
- "one" ('all vs. one' or 'Dunnett') contrast: compare all candidates to a single comparator.
- "all" ('all vs. all' or 'Tukey') contrast: compare all candidates against each other.

Value

(`cases_contrast`)
object to be passed to `evaluate`

Examples

```
define_contrast("one", 1)
```

`draw_data` *Generate binary data*

Description

Generate binary data

Usage

```
draw_data(
  n = 200,
  prev = c(0.5, 0.5),
  random = FALSE,
  m = 10,
  method = c("roc", "lfc", "pr"),
  pars = list(),
  ...
)
```

Arguments

`n`	(numeric) integer, overall sample size
`prev`	(numeric) vector of class prevalences (adding up to 1)
`random`	(logical) random sampling (TRUE) or fixed group sample sizes (FALSE)
`m`	(numeric) integer, number of models
`method`	(character) either "roc", "lfc" (multiple subgroups) or "prob" (no subgroups)
`pars`	(list) containing further named parameters passed to `draw_data_roc`, `draw_data_lfc`
`...`	(any) further named parameters passed

Value

(matrix)
generated binary data (possibly stratified for subgroups)

Examples

```
draw_data()
```

`draw_data_lfc` *Generate binary data (LFC model)*

Description

Generate binary data (LFC model)

Usage

```
draw_data_lfc(
  n = 100,
  prev = c(0.5, 0.5),
  random = FALSE,
  m = 10,
  se = 0.8,
  sp = 0.8,
  B = round(m/2),
  L = 1,
  Rse = diag(rep(1, m)),
  Rsp = diag(rep(1, m)),
  modnames = paste0("model", 1:m),
  ...
)
```

Arguments

`n`	(numeric) integer, total sample size
`prev`	(numeric) disease and healthy prevalence (length 2, adds up to 1)
`random`	(logical) random sampling (TRUE) or fixed prevalence (FALSE)
`m`	(numeric) integer, number of models
`se`	(numeric) sensitivity (length 1)
`sp`	(numeric) specificity (length 1)
`B`	(numeric) integer, between 1 and m, specifies how many sensitivity values are projected to 1
`L`	(numeric) worst alternative is computed under side condition Acc <= L (default value L=1 corresponds to true LFC where values are projected to 1)

`Rse`	(matrix) correlation matrix for empirical sensitivities (m x m)
`Rsp`	(maxtrix) correlation matrix for empirical specificities (m x m)
`modnames`	(modnames) character, model names (length m)
`...`	(any) further arguments (currently unused)

Value

(list)
list of matrices including generated binary datasets (1: correct prediction, 0: incorrect prediction) for each subgroup (specificity, sensitivity)

Examples

```
data <- draw_data_lfc()
head(data)
```

`draw_data_prb` *Sample binary data (single sample)*

Description

This function is wrapper for `rmvbin`.

Usage

```
draw_data_prb(n = 100, pr = c(0.8, 0.8), R = diag(length(pr)))
```

Arguments

`n`	(numeric) integer, sample size
`pr`	(numeric) vector with marginal success probabilities
`R`	(matrix) square correlation matrix

Value

(matrix)
matrix with n rows and length(pr) columns of randomly generated binary (0, 1) data

draw_data_roc *Generate binary data (ROC model)*

Description

Generate binary data (ROC model)

Usage

```
draw_data_roc(
  n = 100,
  prev = c(0.5, 0.5),
  random = FALSE,
  m = 10,
  auc = seq(0.85, 0.95, length.out = 5),
  rho = c(0.25, 0.25),
  dist = c("normal", "exponential"),
  e = 10,
  k = 100,
  delta = 0,
  modnames = paste0("model", 1:m),
  corrplot = FALSE,
  ...
)
```

Arguments

n	(numeric) total sample size
prev	(numeric) disease and healthy prevalence (adds up to 1)
random	(logical) random sampling (TRUE) or fixed prevalence (FALSE)
m	(numeric) integer, number of models
auc	(numeric) vector of AUCs of biomarkers
rho	(numeric) vector (length 2) of correlations between biomarkers
dist	(character) either "normal" or "exponential" specifying the subgroup biomarker distributions
e	(numeric) emulates better (worse) model selection quality with higher (lower) values of e

k	(numeric) technical parameter which adjusts grid size
delta	(numeric) specify importance of sensitivity and specificity (default 0)
modnames	(character) model names (length m)
corrplot	(logical) if TRUE do not return data but instead plot correlation matrices for final binary data (default: FALSE)
...	(any) further arguments (currently unused)

Value

(list)
list of matrices including generated binary datasets (1: correct prediction, 0: incorrect prediction) for each subgroup (specificity, sensitivity)

Examples

```
data <- draw_data_roc()
head(data)
```

---

evaluate *Evaluate the accuracy of multiple (candidate) classifiers in several subgroups*

Description

Assess classification accuracy of multiple classifcation rules stratified by subgroups, e.g. in diseased (sensitivity) and healthy (specificity) individuals.

Usage

```
evaluate(
  data,
  contrast = define_contrast("raw"),
  benchmark = 0.5,
  alpha = 0.05,
  alternative = c("two.sided", "greater", "less"),
  adjustment = c("none", "bonferroni", "maxt", "bootstrap", "mbeta"),
  transformation = c("none", "logit", "arcsin"),
  analysis = c("co-primary", "full"),
  regu = FALSE,
  pars = list(),
  ...
)
```

Arguments

`data`	(list) of n_g x m binary matrix or data.frame (n_g observations of m binary decisions), g is the index of subgroups/classes, usually created via `compare`.
`contrast`	(`cases_contrast`) specified via `define_contrast`
`benchmark`	(numeric) value to compare against (RHS), should have same length as data.
`alpha`	(numeric) significance level (default: 0.05)
`alternative`	(character) specification of alternative hypothesis
`adjustment`	(character) specification of statistical adjustment taken to address multiplicity. The default 'none' does not perform any adjustment for multiplicity.
`transformation`	(character) define transformation to ensure results (e.g. point estimates, confidence limits) lie in unit interval ("none" (default), "logit", or "arcsin" (sqrt))
`analysis`	(character) "co-primary" or "full"
`regu`	(numeric | logical) vector of length 3, specify type of shrinkage. Alternatively, logical of length one (TRUE := c(1, 1/2, 1/4), FALSE := c(0, 0, 0))
`pars`	(list) further parameters given as named list list(type="pairs", nboot=2000)
`...`	(any) additional named parameters, can be used instead of (in in conjunction with) `pars`

Details

Adjustment methods (`adjustment`) and additional parameters (`pars` or `...`):

"none" (default): no adjustment for multiplicity

"bonferroni": Bonferroni adjustment

"maxt": maxT adjustment, based on a multivariate normal approximation of the vector of test statistics

"bootstrap": Bootstrap approach

- nboot: number of bootstrap draws (default: 2000)
- type: type of bootstrap, "pairs" (default) or "wild"

- dist: residual distribution for wild bootstrap, "Normal" (default) or "Rademacher"
- proj_est: should bootstrapped estimates for wild bootstrap be projected into unit interval? (default: TRUE)
- res_tra: type of residual transformation for wild boostrap, 0,1,2 or 3 (default: 0 = no transformation) (for details on res_tra options, see this presentation by James G. MacKinnon (2012) and references therein)

"mbeta": A heuristic Bayesian approach which is based on a multivariate beta-binomial model.

- nrep: number of posterior draws (default: 5000)
- lfc_pr: prior probability of 'least-favorable parameter configuration' (default: 1 if analysis == "co-primary", 0 if analysis == "full").

Value

(cases_results)
list of analysis results including (adjusted) confidence intervals and p-values

Examples

```
#
data <- draw_data_roc()
evaluate(data)
```

`generate_instance_lfc` *Generate data sets under least favorable parameter configurations*

Description

Generates a (simulation) instance, a list of multiple datasets to be processed (analyzed) with process_instance. Ground truth parameters (Sensitvity & Specificity) are least-favorable in the sense that the type-I error rate of the subsequently applied multiple test procedures is maximized.

This function is only needed for simulation via batchtools, not relevant in interactive use!

Usage

```
generate_instance_lfc(
  nrep = 10,
  n = 100,
  prev = 0.5,
  random = FALSE,
  m = 10,
  se = 0.8,
  sp = 0.8,
  L = 1,
  rhose = 0,
  rhosp = 0,
```

```
  cortype = "equi",
  ...,
  data = NULL,
  job = NULL
)
```

Arguments

nrep	(numeric) integer, number of instances
n	(numeric) integer, total sample size
prev	(numeric) disease prevalence
random	(logical) fixed prevalence (FALSE) or simple random sampling (TRUE)
m	(numeric) integer, number of candidates
se	(numeric) sensitivity
sp	(numeric) specificity
L	(numeric) worst alternative is computed under side condition Acc <= L
rhose	(numeric) correlation parameter for sensitivity
rhosp	(numeric) correlation parameter for specificity
cortype	(character) correlation type ("equi" or "ak1")
...	(any) further (named) arguments
data	(NULL) ignored (for batchtools compatibility)
job	(NULL) ignored (for batchtools compatibility)

Details

Utilizes same arguments as draw_data_lfc unless mentioned otherwise above.

Value

(list)
a single (LFC) simulation instance of length nrep

`generate_instance_roc` *Generate data sets under realistic parameter configurations*

Description

Generates a (simulation) instance, a list of multiple datasets to be processed (analyzed) with process_instance. Ground truth parameters (Sensitvity & Specificity) are initially generated according to a generative model whereby multiple decision rules (with different parameter values) are derived by thresholding multiple biomarkers.

This function is only needed for simulation via batchtools, not relevant in interactive use!

Usage

```
generate_instance_roc(
  nrep = 10,
  n = 100,
  prev = 0.5,
  random = FALSE,
  m = 10,
  auc = "seq(0.85, 0.95, length.out = 5)",
  rhose = 0.5,
  rhosp = 0.5,
  dist = "normal",
  e = 10,
  k = 100,
  delta = 0,
  ...,
  data = NULL,
  job = NULL
)
```

Arguments

`nrep`	(numeric) integer, number of instances
`n`	(numeric) integer, total sample size
`prev`	(numeric) disease prevalence
`random`	(logical) fixed prevalence (FALSE) or simple random sampling (TRUE)
`m`	(numeric) integer, number of candidates
`auc`	(numeric) vector of AUCs of biomarkers

`rhose`	(numeric) correlation parameter for sensitivity
`rhosp`	(numeric) correlation parameter for specificity
`dist`	(character) either "normal" or "exponential" specifying the subgroup biomarker distributions
`e`	(numeric) emulates better (worse) model selection quality with higher (lower) values of e
`k`	(numeric) technical parameter which adjusts grid size
`delta`	(numeric) specify importance between sensitivity and specificity (default 0: equal importance)
`...`	(any) further arguments
`data`	(NULL) ignored (for batchtools compatibility)
`job`	(NULL) ignored (for batchtools compatibility)

Details

Utilizes same arguments as draw_data_roc unless mentioned otherwise above.

Value

(list)
a single (ROC) simulation instance of length `nrep`

`process_instance` *Analyze simulated synthetic datasets.*

Description

Process data instances, a list of multiple datasets generated via generate_instance_lfc or generate_instance_roc. This function applies evaluate to all datasets.

This function is only needed for simulation via batchtools, not relevant in interactive use!

Usage

```
process_instance(
  instance = NULL,
  contrast = "cases::define_contrast('raw', NA)",
  benchmark = 0.5,
  alpha = 0.05,
  alternative = "greater",
  adjustment = "none",
  transformation = "none",
  analysis = "co-primary",
  regu = "c(1,1/2,1/4)",
  pars = "list()",
  ...,
  data = NULL,
  job = list(id = NA)
)
```

Arguments

instance	(list) generated via generate_instance_lfc or generate_instance_roc.
contrast	(cases_contrast) specified via define_contrast
benchmark	(numeric) value to compare against (RHS), should have same length as data or length one if all benchmark values are identical.
alpha	(numeric) significance level (default: 0.05)
alternative	(character) specify alternative hypothesis
adjustment	(character) specify type of statistical adjustment taken to address multiplicity
transformation	(character) define transformation to ensure results (e.g. point estimates, confidence limits) lie in unit interval ("none" (default) or "logit")
analysis	(character) "co-primary" (default; only option currently)
regu	(numeric | logical) vector of length 3, specify type of shrinkage. Alternatively, logical of length one (TRUE := c(2, 1, 1/2), FALSE := c(0, 0, 0))
pars	(list) further parameters given as named list
...	(any) additional named parameters

`data`	(NULL) ignored (for batchtools compatibility)
`job`	(NULL) for batchtools compatibility, do not change

Details

Utilizes same arguments as evaluate unless mentioned otherwise above.

Value

(list)
standardized evaluation results

`visualize` *Visualize evaluation results*

Description

Currently, this implementation is only intended for situations with ...

- two groups (e.g. healthy (<-> specificity) and diseased (<-> sensitivity))
- alternative = "greater"
- contrast = define_contrast("raw)

Usage

```
visualize(x, ...)
```

Arguments

`x`	`cases_results` produced by `evaluate`
`...`	`any` further arguments (currently ignored)

Value

a ggplot

Index