Package ‘powerPLS’

May 9, 2026

Type Package

Title Power Analysis for PLS Classification

Version 0.2.1

Description It estimates power and sample size for Partial Least Squares-based methods described in Andreella, et al., (2024), <doi:10.48550/arXiv.2403.10289>.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Imports compositions, FKSUM, nipals, MASS, foreach, parallel, simukde, ks, mvtnorm, pROC, caret

Language en-US

BugReports `https://github.com/angeella/powerPLS/issues`

URL `https://github.com/angeella/powerPLS`

Depends R (>= 2.10)

NeedsCompilation no

Author Angela Andreella [aut, cre] (Main author, ORCID: <`https://orcid.org/0000-0002-1141-3041`>)

Maintainer Angela Andreella <`angela.andreella@unitn.it`>

Repository CRAN

Date/Publication 2025-03-06 00:00:02 UTC

Contents

`aqueous_humour` *Aqueous Humour data*

Description

59 post-mortem aqueous humor samples collected from closed and opened sheep eyes

Usage

```
aqueous_humour
```

Format

A data frame with 59 rows and 45 variables:

ID ID observation

group class membership (C, O)

R1 metabolic values

R2 metabolic values

R3 metabolic values

R4 metabolic values

R5 metabolic values

R6 metabolic values

R7 metabolic values

R8 metabolic values

R9 metabolic values

R10 metabolic values

R11 metabolic values

R12 metabolic values

R13	metabolic values
R14	metabolic values
R15	metabolic values
R16	metabolic values
R17	metabolic values
R18	metabolic values
R19	metabolic values
R20	metabolic values
R21	metabolic values
R22	metabolic values
R23	metabolic values
R24	metabolic values
R25	metabolic values
R26	metabolic values
R27	metabolic values
R28	metabolic values
R29	metabolic values
R30	metabolic values
R31	metabolic values
R32	metabolic values
R33	metabolic values
R34	metabolic values
R35	metabolic values
R36	metabolic values
R37	metabolic values
R38	metabolic values
R39	metabolic values
R40	metabolic values
R41	metabolic values
R42	metabolic values
R43	metabolic values

Author(s)

Angela Andreella `<angela.andreella@unive.it>`

References

https://link.springer.com/article/10.1007/s11306-019-1533-2

`AUCTest` *AUC test*

Description

Performs permutation-based test based on AUC

Usage

```
AUCTest(X, Y, nperm = 100, A, randomization = FALSE,
Y.prob = FALSE, eps = 0.01, scaling = 'auto-scaling',
post.transformation = TRUE, cross.validation = FALSE,...)
```

Arguments

`X`	data matrix where columns represent the p variables and rows the n observations.
`Y`	data matrix where columns represent the two classes and rows the n observations.
`nperm`	number of permutations. Default to 200.
`A`	number of score components
`randomization`	Boolean value. Default to `FALSE`. If `TRUE` the permutation p-value is computed
`Y.prob`	Boolean value. Default `FALSE`. IF `TRUE` `Y` is a probability vector
`eps`	Default 0.01. `eps` is used when `Y.prob` = `FALSE` to transform `Y` in a probability vector
`scaling`	Type of scaling, one of `c('auto-scaling', 'pareto-scaling', 'mean-centering')`. Default 'auto-scaling'.
`post.transformation`	Boolean value. `TRUE` if you want to apply post transformation. Default `TRUE`
`cross.validation`	Boolean value. Default `FALSE`. `TRUE` if you want to compute the observed test statistic by Nested cross-validation
`...`	additional arguments related to `cross.validation`. See `repeatedCV_test`

Value

List with the following objects:

pv raw p-value. It equals `NA` if `randomization` = `FALSE`

pv_adj adjusted p-value. It equals `NA` if `randomization` = `FALSE`

test estimated test statistic

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint `https://arxiv.org/abs/2403.10289`.

See Also

Other test statistics implemented: `mccTest`, `scoreTest`, `dQ2Test`, `sensitivityTest`,`F1Test`, `R2Test`, `specificityTest`, `FMTest`.

Examples

```
datas <- simulatePilotData(nvar = 30, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 2)
out <- AUCTest(X = datas$X, Y = datas$Y, A = 1)
out
```

`computePower` *Power estimation*

Description

Estimates power for a given sample size, type I error level and number of score components.

Usage

```
computePower(X, Y, A, n, seed = 123,
Nsim = 100, nperm = 200, alpha = 0.05,
scaling = 'auto-scaling', test = 'R2',
Y.prob = FALSE, eps = 0.01, post.transformation = TRUE,
fast = FALSE, transformation = 'clr', ncores = NULL)
```

Arguments

`X`	Data matrix where columns represent the p variables and rows the n observations.
`Y`	Data matrix where columns represent the two classes and rows the n observations.
`A`	Number of score components
`n`	Sample size
`seed`	Seed value
`Nsim`	Number of simulations
`nperm`	Number of permutations
`alpha`	Type I error level
`scaling`	Type of scaling, one of `c('auto-scaling', 'pareto-scaling', 'mean-centering')`. Default to 'auto-scaling'

`test`	Type of test statistic, one of `c('score', 'mcc', 'R2')`. Default to 'R2'.
`Y.prob`	Boolean value. Default `FALSE`. IF `TRUE` `Y` is a probability vector
`eps`	Default 0.01. `eps` is used when `Y.prob` = `FALSE` to transform `Y` in a probability vector.
`post.transformation`	Boolean value. `TRUE` if you want to apply post transformation. Default to `TRUE`
`fast`	Use the function `fk_density` from the `FKSUM` `R` package for kernel density estimation. Default to `FALSE`.
`transformation`	Transformation used to map `Y` in probability data vector. The options are 'ilr' and 'clr'.
`ncores`	Number of cores, default NULL.

Value

Returns a matrix of estimated power for each number of components and tests selected.

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint `https://arxiv.org/abs/2403.10289`.

Examples

```
## Not run:
datas <- simulatePilotData(nvar = 10, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 2)
out <- computePower(X = datas$X, Y = datas$Y, A = 3, n = 20, test = 'R2')

## End(Not run)
```

`computeSampleSize` *Sample size estimation*

Description

Compute optimal sample size

Usage

```
computeSampleSize(n, X, Y, A, alpha, beta,
nperm, Nsim, seed, test = 'R2',...)
```

Arguments

`n`	Vector of sample sizes to consider
`X`	Data matrix where columns represent the p variables and rows the n observations.
`Y`	Data matrix where columns represent the two classes and rows the n observations.
`A`	Number of score components
`alpha`	Type I error level. Default to 0.05
`beta`	Type II error level. Default to 0.2.
`nperm`	Number of permutations. Default to 100.
`Nsim`	Number of simulations. Default to 100.
`seed`	Seed value
`test`	Type of test, one of `c('score', 'mcc', 'R2')`. Default to ’R2’.
`...`	Further parameters.

Value

Returns a data frame that contains the estimated power for each sample size and number of components considered

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint `https://arxiv.org/abs/2403.10289`.

See Also

`computePower`

Examples

```
## Not run:
datas <- simulatePilotData(nvar = 10, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 2)
out <- computeSampleSize(X = datas$X, Y = datas$Y, A = 2, A = 3, n = 20, test = 'R2')

## End(Not run)
```

`dQ2Test` *dQ2 test*

Description

Performs permutation-based test based on dQ2

Usage

```
dQ2Test(X, Y, nperm = 200, A, randomization = FALSE,
Y.prob = FALSE, eps = 0.01, scaling = 'auto-scaling',
post.transformation = TRUE, class = 1, cross.validation = FALSE, ...)
```

Arguments

`X`	data matrix where columns represent the p variables and rows the n observations.
`Y`	data matrix where columns represent the two classes and rows the n observations.
`nperm`	number of permutations. Default to 200.
`A`	number of score components
`randomization`	Boolean value. Default to `FALSE`. If `TRUE` the permutation p-value is computed
`Y.prob`	Boolean value. Default `FALSE`. IF `TRUE` `Y` is a probability vector
`eps`	Default 0.01. `eps` is used when `Y.prob = FALSE` to transform `Y` in a probability vector
`scaling`	Type of scaling, one of `c('auto-scaling', 'pareto-scaling', 'mean-centering')`. Default 'auto-scaling'.
`post.transformation`	Boolean value. `TRUE` if you want to apply post transformation. Default `TRUE`
`class`	Numeric value. Specifiy the reference class. Default `1`
`cross.validation`	Boolean value. Default `FALSE`. `TRUE` if you want to compute the observed test statistic by Nested cross-validation
`...`	additional arguments related to `cross.validation`. See `repeatedCV_test`

Value

List with the following objects:

pv raw p-value. It equals `NA` if `randomization = FALSE`

pv_adj adjusted p-value. It equals `NA` if `randomization = FALSE`

test estimated test statistic

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint `https://arxiv.org/abs/2403.10289`.

See Also

Other test statistics implemented: `mccTest`, `scoreTest`, `sensitivityTest`, `specificityTest`,`AUCTest`, `R2Test`, `FMTest`, `F1Test`.

Examples

```
datas <- simulatePilotData(nvar = 30, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 1)
out <- dQ2Test(X = datas$X, Y = datas$Y, A = 1)
out
```

`F1Test` *F1 test*

Description

Performs permutation-based test based on F1

Usage

```
F1Test(X, Y, nperm = 200, A, randomization = FALSE,
Y.prob = FALSE, eps = 0.01, scaling = 'auto-scaling',
post.transformation = TRUE,cross.validation = FALSE,...)
```

Arguments

`X`	data matrix where columns represent the p variables and rows the n observations.
`Y`	data matrix where columns represent the two classes and rows the n observations.
`nperm`	number of permutations. Default to 200.
`A`	number of score components
`randomization`	Boolean value. Default to `FALSE`. If `TRUE` the permutation p-value is computed
`Y.prob`	Boolean value. Default `FALSE`. IF `TRUE` `Y` is a probability vector
`eps`	Default 0.01. `eps` is used when `Y.prob = FALSE` to transform `Y` in a probability vector

`scaling`	Type of scaling, one of `c('auto-scaling', 'pareto-scaling', 'mean-centering')`. Default 'auto-scaling'.
`post.transformation`	Boolean value. `TRUE` if you want to apply post transformation. Default `TRUE`
`cross.validation`	Boolean value. Default `FALSE`. `TRUE` if you want to compute the observed test statistic by Nested cross-validation
`...`	additional arguments related to `cross.validation`. See `repeatedCV_test`

Value

List with the following objects:

pv raw p-value. It equals `NA` if `randomization = FALSE`

pv_adj adjusted p-value. It equals `NA` if `randomization = FALSE`

test estimated test statistic

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint `https://arxiv.org/abs/2403.10289`.

See Also

Other test statistics implemented: `mccTest`, `scoreTest`, `dQ2Test`, `sensitivityTest`,`AUCTest`, `R2Test`, `specificityTest`, `FMTest`.

Examples

```
datas <- simulatePilotData(nvar = 30, clus.size = c(15,15),m = 6,nvar_rel = 5,A = 1)
out <- F1Test(X = datas$X, Y = datas$Y, A = 1)
out
```

`FMTest` *FM test*

Description

Performs permutation-based test based on FM

Usage

```
FMTest(X, Y, nperm = 200, A, randomization = FALSE,
Y.prob = FALSE, eps = 0.01, scaling = 'auto-scaling',
post.transformation = TRUE,cross.validation = FALSE,...)
```

Arguments

`X`	data matrix where columns represent the p variables and rows the n observations.
`Y`	data matrix where columns represent the two classes and rows the n observations.
`nperm`	number of permutations. Default to 200.
`A`	number of score components
`randomization`	Boolean value. Default to `FALSE`. If `TRUE` the permutation p-value is computed
`Y.prob`	Boolean value. Default `FALSE`. IF `TRUE` `Y` is a probability vector
`eps`	Default 0.01. `eps` is used when `Y.prob` = `FALSE` to transform `Y` in a probability vector
`scaling`	Type of scaling, one of `c('auto-scaling', 'pareto-scaling', 'mean-centering')`. Default 'auto-scaling'.
`post.transformation`	Boolean value. `TRUE` if you want to apply post transformation. Default `TRUE`
`cross.validation`	Boolean value. Default `FALSE`. `TRUE` if you want to compute the observed test statistic by Nested cross-validation
`...`	additional arguments related to `cross.validation`. See `repeatedCV_test`

Value

List with the following objects:

pv raw p-value. It equals `NA` if `randomization` = `FALSE`

pv_adj adjusted p-value. It equals `NA` if `randomization` = `FALSE`

test estimated test statistic

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint `https://arxiv.org/abs/2403.10289`.

See Also

Other test statistics implemented: `mccTest`, `scoreTest`, `dQ2Test`, `sensitivityTest`,`AUCTest`, `R2Test`, `specificityTest`, `F1Test`.

Examples

```
datas <- simulatePilotData(nvar = 30, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 1)
out <- FMTest(X = datas$X, Y = datas$Y, A = 1)
out
```

`mccTest` *MCC test*

Description

Performs permutation-based test based on Matthews Correlation Coefficient

Usage

```
mccTest(X, Y, nperm = 200, A, randomization = FALSE,
Y.prob = FALSE, eps = 0.01, scaling = 'auto-scaling',
post.transformation = TRUE, cross.validation = FALSE, seed = 123, ...)
```

Arguments

`X`	data matrix where columns represent the p variables and rows the n observations.
`Y`	data matrix where columns represent the two classes and rows the n observations.
`nperm`	number of permutations. Default to 200.
`A`	number of score components
`randomization`	Boolean value. Default to `FALSE`. If `TRUE` the permutation p-value is computed
`Y.prob`	Boolean value. Default `FALSE`. IF `TRUE` `Y` is a probability vector
`eps`	Default 0.01. `eps` is used when `Y.prob` = `FALSE` to transform `Y` in a probability vector
`scaling`	Type of scaling, one of `c('auto-scaling', 'pareto-scaling', 'mean-centering')`. Default 'auto-scaling'.
`post.transformation`	Boolean value. `TRUE` if you want to apply post transformation. Default `TRUE`

`cross.validation`	Boolean value. Default `FALSE`. `TRUE` if you want to compute the observed test statistic by nested cross-validation
`seed`	Seed value
`...`	additional arguments related to `cross.validation`. See `repeatedCV_test`

Value

List with the following objects:

pv raw p-value. It equals `NA` if `randomization` = `FALSE`

pv_adj adjusted p-value. It equals `NA` if `randomization` = `FALSE`

test estimated test statistic

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint `https://arxiv.org/abs/2403.10289`.

See Also

Other test statistics implemented: `AUCTest`, `scoreTest`, `dQ2Test`, `sensitivityTest`,`AUCTest`, `R2Test`, `specificityTest`, `FMTest`.

Examples

```
datas <- simulatePilotData(nvar = 30, clus.size = c(15,15),m = 6,nvar_rel = 5,A = 1)
out <- mccTest(X = datas$X, Y = datas$Y, A = 1)
out
```

`PLSc` *PLS classification*

Description

Performs Partial Least Squares classification

Usage

```
PLSc(X, Y, A, scaling = 'auto-scaling', post.transformation = TRUE,
eps = 0.01, Y.prob = FALSE, transformation = 'ilr')
```

Arguments

`X`	Data matrix where columns represent the p variables and rows the n observations.
`Y`	Data matrix where columns represent the two classes and rows the n observations.
`A`	Number of score components
`scaling`	Type of scaling, one of `c('auto-scaling', 'pareto-scaling', 'mean-centering')`. Default to 'auto-scaling'
`post.transformation`	Boolean value. `TRUE` if you want to apply post transformation. Default `TRUE`
`eps`	Default 0.01. `eps` is used when `Y.prob = FALSE` to transform `Y` in a probability vector
`Y.prob`	Boolean value. Default `FALSE`. IF `TRUE` `Y` is a probability vector
`transformation`	Transformation used to map `Y` in probability data vector. The options are 'ilr' and 'clr'. Default @ilr.

Value

List with the following objects:

W Matrix of weights

X_loading Matrix of `X` loading

Y_loading Matrix of `Y` loading

X Matrix of `X` data (predictor variables)

Y Matrix of `Y` data (dependent variable)

T_score Matrix of scores

Y_fitted Fitted `Y` matrix

B Matrix regression coefficients

M Number of orthogonal components if `post.transformation=TRUE` is applied.

Author(s)

Angela Andreella

References

Stocchero, M., De Nardi, M., & Scarpa, B. (2021). PLS for classification. Chemometrics and Intelligent Laboratory Systems, 216, 104374.

Examples

```
datas <- simulatePilotData(nvar = 30, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 2)
out <- PLSc(X = datas$X, Y = datas$Y, A = 3)
```

`R2Test` *R2 test*

Description

Performs permutation-based test based on R2

Usage

```
R2Test(X, Y, nperm = 100, A, randomization = FALSE,
Y.prob = FALSE, eps = 0.01, scaling = 'auto-scaling',
post.transformation = TRUE, cross.validation = FALSE, seed = 123, ...)
```

Arguments

`X`	data matrix where columns represent the p variables and rows the n observations.
`Y`	data matrix where columns represent the two classes and rows the n observations.
`nperm`	number of permutations. Default to 200.
`A`	number of score components
`randomization`	Boolean value. Default to `FALSE`. If `TRUE` the permutation p-value is computed
`Y.prob`	Boolean value. Default `FALSE`. IF `TRUE` `Y` is a probability vector
`eps`	Default 0.01. `eps` is used when `Y.prob = FALSE` to transform `Y` in a probability vector
`scaling`	Type of scaling, one of `c('auto-scaling', 'pareto-scaling', 'mean-centering')`. Default 'auto-scaling'.
`post.transformation`	Boolean value. `TRUE` if you want to apply post transformation. Default `TRUE`
`cross.validation`	Boolean value. Default `FALSE`. `TRUE` if you want to compute the observed test statistic by Nested cross-validation
`seed`	Seed value
`...`	additional arguments related to `cross.validation`. See `repeatedCV_test`

Value

List with the following objects:

pv raw p-value. It equals `NA` if `randomization = FALSE`

pv_adj adjusted p-value. It equals `NA` if `randomization = FALSE`

test estimated test statistic

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint `https://arxiv.org/abs/2403.10289`.

See Also

Other test statistics implemented: `mccTest`, `scoreTest`, `sensitivityTest`, `specificityTest`,`AUCTest`, `dQ2Test`, `FMTest`, `F1Test`.

Examples

```
datas <- simulatePilotData(nvar = 30, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 2)
out <- R2Test(X = datas$X, Y = datas$Y, A = 1)
out
```

`repeatedCV_test` *Repeated k-Fold Cross-Validation with Custom Test Metrics*

Description

This function performs repeated k-fold cross-validation and computes a selected performance metric across all repetitions and folds. It allows for different types of performance tests, such as MCC, sensitivity, specificity, R2, F1, and more.

Usage

```
repeatedCV_test(
  data,
  labels,
  k_folds = 5,
  repeats = 3,
  A = 1,
  test_type = "mccTest",
  seed = 1234
)
```

Arguments

`data`	A data frame or matrix of features (predictor variables).
`labels`	A vector of class labels corresponding to the rows of `data`.
`k_folds`	An integer specifying the number of cross-validation folds (default = 5).

`repeats`	An integer specifying the number of times the cross-validation is repeated (default = 3).
`A`	number of score components
`test_type`	A character string specifying the type of test to use. Options include: • 'mccTest' for Matthews Correlation Coefficient (MCC), • 'sensitivityTest' for Sensitivity, • 'specificityTest' for Specificity, • 'R2Test' for R-squared, • 'scoreTest' for Score, • 'F1Test' for F1 Score, • 'FMTest' for Fowlkes-Mallows Index (FM), • 'AUCTest' for Area Under the Curve (AUC), • 'dQ2Test' for dQ2. Default is 'mccTest'.
`seed`	An integer for setting the random seed to ensure reproducibility (default = 1234).

Value

A numeric value representing the average performance metric across the outer folds.

Examples

```
datas <- simulatePilotData(nvar = 30, clus.size = c(15,15),m = 6,nvar_rel = 5,A = 1)
data <- datas$X
labels <- datas$Y
mean_mcc <- repeatedCV_test(data, labels, A = 1, test_type = 'mccTest')
cat('Mean MCC:', mean_mcc, '\n')

mean_score <- repeatedCV_test(data, labels, A = 1, test_type = 'scoreTest')
cat('Mean Sensitivity:', mean_score, '\n')
```

`scoreTest` *Score test*

Description

Performs permutation-based test based on predictive score vector

Usage

```
scoreTest(X, Y, nperm = 200, A, randomization = FALSE,
Y.prob = FALSE, eps = 0.01, scaling = 'auto-scaling',
post.transformation = TRUE, cross.validation = FALSE, seed = 123, ...)
```

Arguments

`X`	data matrix where columns represent the p variables and rows the n observations.
`Y`	data matrix where columns represent the two classes and rows the n observations.
`nperm`	number of permutations. Default to 200.
`A`	number of score components
`randomization`	Boolean value. Default to `FALSE`. If `TRUE` the permutation p-value is computed
`Y.prob`	Boolean value. Default `FALSE`. IF `TRUE` `Y` is a probability vector
`eps`	Default 0.01. `eps` is used when `Y.prob = FALSE` to transform `Y` in a probability vector
`scaling`	Type of scaling, one of `c('auto-scaling', 'pareto-scaling', 'mean-centering')`. Default 'auto-scaling'.
`post.transformation`	Boolean value. `TRUE` if you want to apply post transformation. Default `TRUE`
`cross.validation`	Boolean value. Default `FALSE`. `TRUE` if you want to compute the observed test statistic by Nested cross-validation
`seed`	Seed value
`...`	additional arguments related to `cross.validation`. See `repeatedCV_test`

Value

List with the following objects:

pv raw p-value. It equals `NA` if `randomization = FALSE`

pv_adj adjusted p-value. It equals `NA` if `randomization = FALSE`

test estimated test statistic

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint `https://arxiv.org/abs/2403.10289`.

See Also

Other test statistics implemented: `mccTest`, `R2Test`, `sensitivityTest`, `specificityTest`,`AUCTest`, `dQ2Test`, `FMTest`, `F1Test`.

Examples

```
datas <- simulatePilotData(nvar = 30, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 2)
out <- scoreTest(X = datas$X, Y = datas$Y, A = 1)
out
```

`sensitivityTest` *sensitivity test*

Description

Performs permutation-based test based on sensitivity

Usage

```
sensitivityTest(X, Y, nperm = 200, A, randomization = FALSE,
Y.prob = FALSE, eps = 0.01, scaling = 'auto-scaling',
post.transformation = TRUE, cross.validation = FALSE, ...)
```

Arguments

`X`	data matrix where columns represent the p variables and rows the n observations.
`Y`	data matrix where columns represent the two classes and rows the n observations.
`nperm`	number of permutations. Default to 200.
`A`	number of score components
`randomization`	Boolean value. Default to `FALSE`. If `TRUE` the permutation p-value is computed
`Y.prob`	Boolean value. Default `FALSE`. IF `TRUE` `Y` is a probability vector
`eps`	Default 0.01. `eps` is used when `Y.prob` = `FALSE` to transform `Y` in a probability vector
`scaling`	Type of scaling, one of `c('auto-scaling', 'pareto-scaling', 'mean-centering')`. Default 'auto-scaling'.
`post.transformation`	Boolean value. `TRUE` if you want to apply post transformation. Default `TRUE`
`cross.validation`	Boolean value. Default `FALSE`. `TRUE` if you want to compute the observed test statistic by Nested cross-validation
`...`	additional arguments related to `cross.validation`. See `repeatedCV_test`

Value

List with the following objects:

pv raw p-value. It equals `NA` if `randomization` = `FALSE`

pv_adj adjusted p-value. It equals `NA` if `randomization` = `FALSE`

test estimated test statistic

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint `https://arxiv.org/abs/2403.10289`.

See Also

Other test statistics implemented: `mccTest`, `scoreTest`, `dQ2Test`, `specificityTest`,`AUCTest`, `R2Test`, `FMTest`, `F1Test`.

Examples

```
datas <- simulatePilotData(nvar = 30, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 1)
out <- sensitivityTest(X = datas$X, Y = datas$Y, A = 1)
out
```

`simulatePilotData` *Simulate pilot data*

Description

Simulate cluster pilot data

Usage

```
simulatePilotData(seed = 123, nvar, clus.size, nvar_rel,m, A = 2, S1 = NULL, S2 = NULL)
```

Arguments

`seed`	Seed value
`nvar`	Number of variables
`clus.size`	Vector of two elements, specifying the size of classes (only two classes are considered)
`nvar_rel`	Number of variables relevant to predict the dependent variable
`m`	Effect size of separation between classes
`A`	Oracle number of score components
`S1`	Covariance matrix for the first class. Default `NULL`, i.e., the identity is considered.
`S2`	Covariance matrix for the second class. Default`NULL`, i.e., the identity is considered.

Author(s)

Angela Andreella @return List with the following objects:

- **X** matrix of predictor variables with `nvar` columns and the sum of `clus.size` values as number of rows.
- **Y** vector of dependent variable with the sum of `clus.size` values as length

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint https://arxiv.org/abs/2403.10289.

Examples

```
datas <- simulatePilotData(nvar = 10, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 2)
```

`sim_XY` *Simulate pilot data*

Description

Simulate data matrix under the alternative hypothesis with `n` observations by kernel density estimation

Usage

```
sim_XY(out, n, seed = 123, post.transformation = TRUE, A, fast = FALSE)
```

Arguments

`out`	Output from `PLSc`
`n`	Number of observations to simulate
`seed`	Seed value
`post.transformation`	Boolean value. Default to `TRUE`, i.e., post transformation is applied in `PLSc`
`A`	Number of score components used in `PLSc`.
`fast`	Use the function `fk_density` from the `FKSUM` R package for kernel density estimation. Default to `FALSE`.

Value

Returns a list:

- **Y_H1** dependent variable, matrix with 2 columns and `n` rows (observations)
- **X_H1** predictor variables, matrix with `n` rows (observations) and number of columns equal to `out$X` (i.e., original dataset)

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint https://arxiv.org/abs/2403.10289.

See Also

`PLSc`, `ptPLSc`

Examples

```
datas <- simulatePilotData(nvar = 10, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 2)
out <- PLSc(X = datas$X, Y = datas$Y, A = 3)
out_sim <- sim_XY(out = out, n = 10, A = 3)
```

`specificityTest` *specificity test*

Description

Performs permutation-based test based on specificity

Usage

```
specificityTest(X, Y, nperm = 200, A, randomization = FALSE,
Y.prob = FALSE, eps = 0.01, scaling = 'auto-scaling',
post.transformation = TRUE,cross.validation = FALSE,...)
```

Arguments

`X`	data matrix where columns represent the p variables and rows the n observations.
`Y`	data matrix where columns represent the two classes and rows the n observations.
`nperm`	number of permutations. Default to 200.
`A`	number of score components
`randomization`	Boolean value. Default to `FALSE`. If `TRUE` the permutation p-value is computed
`Y.prob`	Boolean value. Default `FALSE`. IF `TRUE` `Y` is a probability vector
`eps`	Default 0.01. `eps` is used when `Y.prob = FALSE` to transform `Y` in a probability vector

`scaling`	Type of scaling, one of `c('auto-scaling', 'pareto-scaling', 'mean-centering')`. Default 'auto-scaling'.
`post.transformation`	Boolean value. `TRUE` if you want to apply post transformation. Default `TRUE`
`cross.validation`	Boolean value. Default `FALSE`. `TRUE` if you want to compute the observed test statistic by Nested cross-validation
`...`	additional arguments related to `cross.validation`. See `repeatedCV_test`

Value

List with the following objects:

pv raw p-value. It equals `NA` if `randomization = FALSE`

pv_adj adjusted p-value. It equals `NA` if `randomization = FALSE`

test estimated test statistic

Author(s)

Angela Andreella

References

For the general framework of power analysis for PLS-based methods see:

Andreella, A., Fino, L., Scarpa, B., & Stocchero, M. (2024). Towards a power analysis for PLS-based methods. arXiv preprint `https://arxiv.org/abs/2403.10289`.

See Also

Other test statistics implemented: `mccTest`, `scoreTest`, `dQ2Test`, `sensitivityTest`,`AUCTest`, `R2Test`, `FMTest`, `F1Test`.

Examples

```
datas <- simulatePilotData(nvar = 30, clus.size = c(5,5),m = 6,nvar_rel = 5,A = 1)
out <- specificityTest(X = datas$X, Y = datas$Y, A = 1)
out
```

`wheezing` *Wheezing data*

Description

32 urine samples from children at risk of early-onset asthma and those with transient wheezing.

Usage

```
wheezing
```

Format

A data frame with 32 rows and 176 variables

Author(s)

Angela Andreella <`angela.andreella@unive.it`>

References

`https://onlinelibrary.wiley.com/doi/10.1111/pai.12879`

Index