Package 'dppca'

November 10, 2026

Title Differentially Private Principal Component Analysis and Visualization

Version 1.0.0

Description Provides tools for differentially private principal component analysis (PCA) and visualization. Includes private principal component directions, scree and proportion of variance explained (PVE) estimates, and two-dimensional private score histograms with synthetic sampled points. Supports additive and sparse histogram mechanisms, group-wise score visualizations, and optional non-private comparisons. An integrated workflow produces loading plots, scree plots, score plots, variable plots, and biplots from shared private PCA estimates. An interactive 'shiny' application is also provided. Private direction estimation is based on Kim and Jung (2025) <doi:10.1002/sam.70053>. Private scree estimation draws on Dwork and Roth (2014) <doi:10.1561/0400000042>, Ramsay and Spicker (2025) <doi:10.48550/arXiv.2501.14095>, and Yu, Ren and Zhou (2024) <doi:10.3150/23-BEJ1706>. Private score frames use unbounded quantile mechanisms from Durfee (2023) <doi:10.52202/075280-3394>. Private score histograms use methods related to Wasserman and Zhou (2010) <doi:10.1198/jasa.2009.tm08651> and Karwa and Vadhan (2018) <doi:10.4230/LIPIcs.ITCS.2018.44>.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Depends R (>= 4.3.0)

Imports ggplot2, ggrepel, graphics, grDevices, grid, patchwork, Rcpp, Rdpack, rlang, stats, VGAM

LinkingTo Rcpp

Suggests knitr, rmarkdown, shiny

VignetteBuilder knitr

RdMacros Rdpack

LazyData true

LazyDataCompression xz

URL https://github.com/yejinjo0220/dppca,
https://yejinjo0220.github.io/dppca/

BugReports https://github.com/yejinjo0220/dppca/issues

Config/testthat/edition 3

NeedsCompilation yes

Author Yejin Jo [aut, cre],
Minwoo Kim [aut],
Sungkyu Jung [aut]

Maintainer Yejin Jo <yejinjo0220@gmail.com>

Repository CRAN

Date/Publication 2026-11-10 22:00:23 UTC

Contents

clipped_control . . . 3
dppca_app . . . 4
dp_loading . . . 5
dp_pca . . . 7
dp_score . . . 10
dp_scree . . . 13
gau . . . 17
gau_g . . . 17
huber_control . . . 18
loading_plot_control . . . 19
plot.dp_loading . . . 20
plot.dp_pca . . . 22
plot.dp_score . . . 24
plot.dp_score_group . . . 25
plot.dp_scree . . . 26
pmwm_control . . . 27
privacy_control . . . 28
sampling_control . . . 30
score_plot_control . . . 32
scree_plot_control . . . 34
vowel . . . 35

Index **37**

`clipped_control` *Control options for clipped scree estimation*

Description

Creates a control list for `method = "clipped"` in `dp_scree()` or the scree configuration of `dp_pca()`.

Usage

```
clipped_control(C_clip)
```

Arguments

`C_clip`	Required positive public clipping threshold for squared pair differences. Larger values reduce clipping bias but increase Gaussian noise; data-dependent selection requires separate privacy accounting.

Details

The method clips the paired quantities at `C_clip` and adds Gaussian noise to their vector of column means. For $m = \lfloor n/2 \rfloor$ pairs and k components, the calibration uses vector sensitivity $\sqrt{k}\, C_{clip}/m$ under the conditions in `dp_scree()`. Larger thresholds reduce clipping bias but increase the noise scale.

Value

A control list for `method = "clipped"`.

References

Dwork C, Roth A (2014). "The Algorithmic Foundations of Differential Privacy." *Found. Trends Theor. Comput. Sci.*, **9**(3–4), 211–407. ISSN 1551-305X, doi:10.1561/0400000042.

See Also

`dp_scree()` for using clipped scree estimation. `plot.dp_scree()` for plotting private scree estimates.

Examples

```
clipped_control(C_clip = 3)
```

dppca_app *Launch the dppca Shiny app*

Description

Launches an interactive 'shiny' application for exploring differentially private PCA visualizations. The PCA tab uses `dp_pca()` for integrated estimation, summaries, and scree, score, loading, variable, and biplot displays, including grouped score plots when grouping is enabled. The Loading, Scree, and Score tabs provide standalone analyses, including method comparisons and grouped score plots.

Usage

```
dppca_app(X = NULL, group = NULL)
```

Arguments

`X`	A numeric matrix or data frame with observations in rows and variables in columns. Non-numeric data-frame columns are excluded with a warning; numeric values must be finite. Input is not automatically centered or standardized. If provided, the app uses this dataset initially. At least two observations and two numeric features are required.
`group`	Optional group column name in `X` or vector of group labels (`NULL` by default). A selected group column is excluded from PCA features, even when numeric; group labels and membership are assumed public. Only used with supplied `X`; activate grouping in the PCA or Score tab to use these labels.

Details

The app accepts built-in data, supplied `X`, or uploaded CSV files. Grouping is available for PCA and Score analyses; a selected group column is excluded from numeric PCA features. Loading and Scree analyses use all numeric feature columns.

Running an analysis stores its estimates for subsequent plotting. Display settings can redraw an existing fit without consuming another privacy budget. Non-private reference output is optional in the app and is not a private release. Budget tables display allocations but do not track cumulative privacy cost across runs.

Input is not automatically centered or standardized. Any data-dependent preprocessing requires separate privacy accounting; see `dp_pca()` and `dp_score()` for applicable conditions.

Value

Invisibly returns the value produced by `shiny::runApp()` after the application exits.

Examples

```
if (interactive()) {
# Launch the app with built-in example datasets.
dppca_app()

# Launch the app with a user-supplied numeric dataset.
data(gau, package = "dppca")
dppca_app(gau)

# Launch the app with group labels stored in a column.
data(gau_g, package = "dppca")
dppca_app(gau_g, group = "group")
}
```

`dp_loading` *Estimate differentially private PC directions*

Description

Estimate principal component (PC) directions using a noisy spherical Kendall matrix, with optional ordinary PCA directions for comparison.

Display a concise overview of a `dp_loading()` fit, including its data dimensions, privacy parameters, and whether a non-private reference is stored.

Summarize a fitted `dp_loading()` object without recomputing its estimates or consuming additional privacy budget.

Usage

```
dp_loading(X, privacy, cpp.option = TRUE, non_private = FALSE)

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

## S3 method for class 'dp_loading'
summary(object, ...)

## S3 method for class 'summary.dp_loading'
print(x, digits = 3, ...)
```

Arguments

`X`	A numeric matrix or data frame with observations in rows and variables in columns. Non-numeric data-frame columns are excluded with a warning; numeric values must be finite. At least two observations and one numeric feature are required. Observations must be sufficiently separated to compute normalized pairwise differences.

privacy	A privacy_control() object allocating privacy budget to loading alone. Either specify total eps and delta or an absolute loading budget.
cpp.option	Whether to use the compiled spherical-Kendall implementation (TRUE by default); FALSE uses the R implementation.
non_private	Whether to compute and store ordinary PCA reference results (FALSE by default).
x	A summary.dp_loading object.
...	Additional arguments.
object	A dp_loading object.
digits	Number of significant digits used for printing.

Details

The private PC directions are eigenvectors of a spherical Kendall matrix perturbed with symmetric Gaussian noise. Directions are ordered by the noisy eigenvalues; they need not be eigenvectors of the sample covariance matrix. The spherical Kendall construction uses normalized pairwise differences and is invariant to common translations.

With non_private = TRUE, ordinary covariance-based directions are stored after sign alignment with the private directions. This comparison is not a private release.

Under fixed-size replacement adjacency, the matrix has Frobenius sensitivity at most 4 / n. The diagonal Gaussian noise standard deviation is 4 * sqrt(2 * log(1.25 / delta)) / (n * eps); off-diagonal entries use this scale divided by sqrt(2). This classical calibration has a standard sufficient guarantee for 0 < eps < 1; larger eps requires an additional calibration check, which is not performed by the function.

Duplicate or numerically indistinguishable observations are rejected, but these checks do not privatize input validation. Data schema and sample size are assumed public. Data-dependent preprocessing requires separate privacy analysis and accounting.

Value

A dp_loading object with the following components:

directions A p by p matrix of private PC directions.

nonprivate A p by p matrix of ordinary PC directions, or NULL when non_private = FALSE.

metadata Sample size, variable count and privacy parameters.

Rows represent variables, and columns represent PC1, ..., PCp. The columns are orthonormal eigenvectors, without eigenvalue or correlation scaling.

Invisibly returns x.

A summary.dp_loading object containing the stored private and nonprivate direction matrices, with sample size and privacy budget metadata. The print method displays up to six variables and four PCs; the full matrices remain stored. The non-private matrix remains NULL if it was not stored.

Invisibly returns x.

Examples

```
set.seed(123)
X <- matrix(rnorm(150), nrow = 50, ncol = 3)
fit <- dp_loading(
  X, privacy = privacy_control(eps = 0.5, delta = 1e-3),
  cpp.option = FALSE
)
print(fit)
summary(fit)
fit$directions
fit_with_reference <- dp_loading(
  X, privacy = privacy_control(loading = c(eps = 0.5, delta = 1e-3)),
  cpp.option = FALSE, non_private = TRUE
)
fit_with_reference$nonprivate
## Not run:
loading_plot <- plot(fit, type = c("vector", "point", "heatmap"),
                     heatmap_components = 1:3, variables = "all")
loading_plot                 # Display the saved plot
loading_plot$loading$directions # Access the stored directions
loading_plot$plot$all       # Access the combined plot

## End(Not run)
```

---

dp_pca *Estimate shared PCA quantities once*

Description

Summarizes an existing fit without new estimation or random sampling.

Usage

```
dp_pca(
  X,
  privacy,
  scree,
  score = list(),
  axes = c(1, 2),
  cpp.option = TRUE,
  non_private = FALSE,
  V_dp = NULL,
  group = NULL
)

## S3 method for class 'dp_pca'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

```
## S3 method for class 'dp_pca'
summary(object, ...)

## S3 method for class 'summary.dp_pca'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

Arguments

X	A numeric matrix or data frame with observations in rows and variables in columns. Non-numeric data-frame columns are excluded with a warning; numeric values must be finite. At least two observations and two numeric features are required. If directions must be estimated, the pairwise-separation condition of dp_loading() also applies.
privacy	A privacy_control() object allocating budget to loading, scree, and score (equal thirds by default).
scree	Required named list with k = NULL (all components), method = "clipped", control (required), and mono = TRUE. k, when supplied, must be between 2 and the number of numeric features. Use clipped_control(), pmwm_control(), or huber_control() for tuning.
score	Named list specifying method = "sparse" or "add" and bins = c(20, 20) by default. Exactly one histogram method is used.
axes	Two distinct PC indices between 1 and the fitted k (c(1, 2) by default).
cpp.option	Whether to use the compiled spherical-Kendall implementation (TRUE by default) when estimating private directions; ignored if V_dp is supplied.
non_private	Whether to compute and store ordinary PCA reference results (FALSE by default).
V_dp	Optional p by p private PC direction matrix (NULL by default), where p is the number of numeric PCA features. Columns must be orthonormal and rows must follow the feature order in X; matching row names are required when both sets of names exist.
group	Optional group column name in X or vector of group labels (NULL by default). A selected group column is excluded from PCA features, even when numeric; group labels and membership are assumed public.
x	A summary.dp_pca object.
digits	Significant digits used when printing tables.
...	Additional arguments. The summary calculation takes no extra options; print methods pass formatting arguments to the printed tables.
object	A dp_pca() result.

Details

A single private PC direction matrix is estimated or supplied and reused for scree and score estimation. Scree values estimate variance along those directions using disjoint paired differences; an odd final observation is unused for scree, while the score histogram uses all rows. PVE is normalized over the retained k components, not necessarily all original variables.

The selected scree method uses the entire scree allocation. Of the score epsilon budget, 35% goes to the private frame and 65% to the histogram; all score delta goes to the histogram. With supplied `V_dp`, the loading budget records its previous release rather than being reassigned.

Grouped data use one direction estimate and private frame. Their disjoint histograms use parallel composition when group labels and membership are public and fixed. The pooled histogram is post-processed from the grouped histograms.

Input is not automatically centered or standardized. Data-dependent preprocessing needs separate privacy accounting, and sample size, schema, selected components, and tuning settings are assumed public. The loading conditions in `dp_loading()` also apply. The optional non-private reference uses sample-mean-centered ordinary PCA scores and is not private.

PVE is normalized over the `k` retained components; a total of zero produces zero PVE. When `k` is smaller than the number of variables, a cumulative PVE of one does not describe all original variation.

The budget table reports allocated budgets, not cumulative privacy spending. With supplied `V_dp`, the loading allocation records its prior cost. The optional non-private comparison is excluded from the importance table.

Value

A `dp_pca` object containing private `directions`, `scree`, `pve`, `hist`, `frame`, and optional `score_groups`. `nonprivate` contains the optional ordinary PCA reference, or `NULL` when `non_private = FALSE`. `metadata` contains data dimensions, privacy allocations, methods, controls, grouping information and axes. The print method returns the fitted object invisibly.

A `summary.dp_pca` object. `importance` contains DP eigenvalues, PVE and cumulative PVE, with a column for each estimated component. `budget` contains the allocated eps and delta for loading, scree, score frame and score histogram, plus the total. Other fields describe data dimensions, selected axes, methods and whether a non-private reference is stored. Raw scores and loading matrices are not copied into the summary. The print method returns its input invisibly.

Examples

```
data(gau, package = "dppca")
set.seed(123)
fit <- dp_pca(
  gau[1:200, ],
  privacy = privacy_control(eps = 5, delta = 1e-4),
  scree = list(k = 2, control = clipped_control(C_clip = 3)),
  score = list(method = "sparse", bins = c(10, 10)),
  non_private = FALSE
)
fit
summary(fit)
fit$directions
fit$nonprivate # NULL by default
fit$metadata$privacy

set.seed(123)
plots <- plot(fit) # DP panels only
plots$private$plots$biplot
```

dp_score *Differentially private score histograms*

Description

This function projects observations onto supplied private directions or directions estimated by dp_loading() and computes two-dimensional score histograms. Observed coordinates and empirical reference histograms are returned only when non_private = TRUE.

Usage

```
dp_score(
  X,
  privacy,
  bins,
  method = c("add", "sparse"),
  cpp.option = TRUE,
  axes = c(1, 2),
  V_dp = NULL,
  non_private = FALSE,
  group = NULL
)

## S3 method for class 'dp_score'
summary(object, ...)

## S3 method for class 'dp_score_group'
summary(object, ...)

## S3 method for class 'summary.dp_score'
print(x, digits = 4, ...)

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

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

Arguments

X	A numeric matrix or data frame with observations in rows and variables in columns. Non-numeric data-frame columns are excluded with a warning; numeric values must be finite. At least two observations and two numeric features are required. If directions must be estimated, the pairwise-separation condition of dp_loading() also applies.

privacy	A privacy_control() object allocating budget to loading and score (equal shares by default). Each requested histogram method receives the full histogram allocation.
bins	Positive integer vector of length two giving the number of histogram bins along the selected score axes.
method	One or both histogram methods: "add" (Gaussian additive noise) and "sparse" (thresholded sparse histogram). Default: c("add", "sparse").
cpp.option	Whether to use the compiled spherical-Kendall implementation (TRUE by default) when estimating private directions; ignored if V_dp is supplied.
axes	Two distinct PC indices for the horizontal and vertical score axes, respectively (c(1, 2) by default). Each must be between 1 and the number of numeric PCA features.
V_dp	Optional p by p private PC direction matrix (NULL by default), where p is the number of numeric PCA features. Columns must be orthonormal and rows must follow the feature order in X; matching row names are required when both sets of names exist.
non_private	Whether to compute and store ordinary PCA reference results (FALSE by default). For score analysis, these include mean-centered ordinary PCA scores and empirical reference histograms.
group	Optional group column name in X or vector of group labels (NULL by default). A selected group column is excluded from PCA features, even when numeric; group labels and membership are assumed public. All groups share the same private directions, frame, and histogram grid.
object	A result from dp_score(), with or without group.
...	Reserved for S3 generic compatibility.
x	A result from dp_score(), with or without group.
digits	Number of significant digits shown in tables.

Details

Scores are projected onto the selected private PC directions. Rather than releasing individual projected coordinates, the function estimates a private two-dimensional plotting frame followed by private histogram probabilities. The frame uses two private coordinate medians and an inflated private 0.99 quantile of distances from their center (inflation factor 1.2), using unbounded quantile mechanisms.

For a fixed frame and directions under replacement adjacency, the histogram count vector has l1 sensitivity at most 2 and l2 sensitivity at most sqrt(2). Out-of-frame scores are assigned to boundary bins. Adaptive private directions and frame selection require privacy composition. Input is not automatically centered or standardized; data-dependent mean centering or standardization is not covered by the count sensitivity argument.

Two histogram mechanisms are available:

- "add": Add Gaussian noise to all counts, truncate negatives to zero, and normalize. The classical Gaussian calibration has a standard sufficient guarantee for 0 < eps_hist < 1; larger values require a separate calibration check. All-zero truncated counts cause an error.

- "sparse": Add Laplace noise to nonzero bin proportions, retain bins above a privacy-dependent threshold, and normalize positive mass. If none survive, return an all-zero histogram.

The score epsilon allocation is split 35% for the private frame and 65% for the histogram; all score delta goes to the histogram. The frame epsilon is divided equally among two private medians and one radial quantile. Directions and frame are shared across methods. Each requested histogram method receives the full histogram allocation, so jointly releasing "add" and "sparse" requires composition across methods.

For grouped data, the partition and labels are assumed public and fixed; groups share directions, frame and grid. Parallel composition applies to disjoint group histograms under this assumption, while different methods still require composition. non_private = FALSE does not privatize group membership. Input dimensions, feature schema, bins, and tuning choices are assumed public. Repeated releases also require composition.

The optional ordinary PCA reference uses sample-mean-centered scores and a zero-centered comparison frame. It is not differentially private. See dp_loading() for direction-estimation and Gaussian calibration conditions.

Value

A dp_score object containing the private frame and requested histogram methods (add and/or sparse), or groups for grouped fits. Each histogram has xmin, xmax, ymin, ymax and prob columns. nonprivate is NULL by default; when requested it contains scores, frame and the empirical histogram (or group histograms under groups). metadata contains data dimensions, axes, bins, methods and privacy settings. Use summary() and plot() to inspect or visualize stored results.

A summary containing the public sample size, axes, grid, private frame (including its center), allocated privacy budget and a table of positive-bin counts and probability sums. No data-dependent private mechanisms are run.

Invisibly returns x.

Invisibly returns x.

References

Dwork C, Roth A (2014). "The Algorithmic Foundations of Differential Privacy." *Found. Trends Theor. Comput. Sci.*, **9**(3–4), 211–407. ISSN 1551-305X, doi:10.1561/0400000042.

Durfee D (2023). "Unbounded Differentially Private Quantile and Maximum Estimation." In *Advances in Neural Information Processing Systems*, volume 36, 77691–77712. doi:10.52202/0752803394, https://proceedings.neurips.cc/paper_files/paper/2023/hash/f4b6ef2a78684dca2fb3f1c09372e
html.

Wasserman L, Zhou S (2010). "A Statistical Framework for Differential Privacy." *Journal of the American Statistical Association*, **105**(489), 375–389. doi:10.1198/jasa.2009.tm08651.

Karwa V, Vadhan S (2018). "Finite Sample Differentially Private Confidence Intervals." In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference*, volume 94 of *Leibniz International Proceedings in Informatics*, 44:1–44:9. doi:10.4230/LIPIcs.ITCS.2018.44.

Kim M, Jung S (2025). "Robust and Differentially Private Principal Component Analysis." *Statistical Analysis and Data Mining: An ASA Data Science Journal*, **18**(6), e70053. doi:10.1002/sam.70053.

See Also

plot.dp_score() for plotting stored results without new private estimates. dp_pca() and plot.dp_pca() for estimation followed by plots of a stored fit. plot.dp_score_group() for displaying a grouped result. dp_loading() for private principal component direction estimation.

Examples

```
data(gau, package = "dppca")

# Use a small subset to keep the example fast.
X <- gau[1:300, ]

# Release a private histogram with fixed preprocessing and no raw scores.
set.seed(123)
score_gau <- dp_score(
  X,
  privacy = privacy_control(eps = 1.8, delta = 1e-3),
  method = "add",
  bins = c(10, 10),
  non_private = FALSE
)

score_gau$frame
head(score_gau$add)
summary(score_gau)
# plot(score_gau, type = "histogram")
# To use a group column: dp_score(X, privacy, bins, group = "region")
```

---

dp_scree *Differentially private scree values*

Description

Estimates scree values along private principal component directions using means of squared differences from disjoint score pairs. An ordinary PCA reference is computed only when non_private = TRUE.

Usage

```
dp_scree(
  X,
  k,
  privacy,
  method = c("clipped", "pmwm", "huber"),
  control = NULL,
  cpp.option = TRUE,
  mono = TRUE,
```

```
  V_dp = NULL,
  non_private = FALSE
)

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

## S3 method for class 'dp_scree'
summary(object, ...)

## S3 method for class 'summary.dp_scree'
print(x, digits = 4, ...)
```

Arguments

X	A numeric matrix or data frame with observations in rows and variables in columns. Non-numeric data-frame columns are excluded with a warning; numeric values must be finite. At least two observations and one numeric feature are required. If directions must be estimated, the pairwise-separation condition of dp_loading() also applies.
k	Number of leading components to estimate; an integer from 1 to the number of retained numeric features.
privacy	A privacy_control() object allocating budget to loading and scree (equal shares by default). Each requested scree method receives the full scree allocation.
method	One or more scree estimators: "clipped", "pmwm", or "huber" ("clipped" by default).
control	Tuning parameters from clipped_control(), pmwm_control(), or huber_control(). For multiple methods, supply a named list of method-specific controls; required tuning values must be supplied.
cpp.option	Whether to use the compiled spherical-Kendall implementation (TRUE by default) when estimating private directions; ignored if V_dp is supplied.
mono	Whether to apply nonincreasing isotonic regression (TRUE by default). Negative scree estimates are truncated to zero in either case.
V_dp	Optional p by p private PC direction matrix (NULL by default), where p is the number of numeric PCA features. Columns must be orthonormal and rows must follow the feature order in X; matching row names are required when both sets of names exist.
non_private	Whether to compute and store ordinary PCA reference results (FALSE by default). The reference contains ordinary PCA scree and PVE.
x	A summary.dp_scree object.
...	Further arguments, reserved for compatibility with the generic.
object	A dp_scree object.
digits	Number of significant digits used for printing.

Details

Let `Y = X V` be scores along the private directions. For each method, `m = floor(n / 2)` disjoint pairs of rows are formed and the component-wise values are

$$W_{jl} = (Y_{b_j,l} - Y_{a_j,l})^2/2.$$

An odd final observation is unused. The methods privately estimate the column means of `W`, corresponding to variance along private directions rather than necessarily to sample covariance eigenvalues.

The estimators differ as follows:

- `"clipped"`: Clip the paired values at `C_clip`, then perturb their mean vector with Gaussian noise.
- `"pmwm"`: Estimate private lower and upper quantiles, winsorize the paired values, and perturb their mean vector with Gaussian noise. Quantile calls share half of the method's epsilon; the mean receives the remaining half and all delta.
- `"huber"`: Estimate private component scales and apply noisy Huber gradient updates. The scale step uses `m2_frac` of epsilon; the updates use the remainder and all delta. At least eight observations are required.

Directions are estimated once unless `V_dp` is supplied. The input is not automatically centered or standardized; sensitivity arguments use fixed-size replacement adjacency and condition on the directions. Data-dependent scaling requires separate privacy analysis. Sample size, schema, component selection, and tuning parameters are assumed public.

Each requested scree method uses the full scree allocation. Releasing several methods together, or releasing results from repeated calls, requires composition beyond the stored budget for a single method. The optional ordinary PCA reference is not private.

Scree values are truncated at zero, optionally made nonincreasing, and normalized over the returned `k` components for PVE. A zero total yields zero PVE. These steps are post-processing and do not reorder the directions.

Value

A `dp_scree` list containing one entry per requested method (`clipped`, `pmwm`, and/or `huber`). Each entry has `scree` and `pve` vectors named `PC1`, ..., `PCk`. The `nonprivate` entry holds the optional ordinary PCA reference (or `NULL` when omitted), and `metadata` stores input dimensions, methods and the recorded privacy budget. PVE is normalized over the `k` returned components in every entry. Use `summary()` for scree, PVE and cumulative PVE tables, and `plot()` to draw stored estimates without a new randomized computation.

Invisibly returns `x`.

A `summary.dp_scree` object containing one table per stored method. Each table gives scree values, PVE and cumulative PVE. PVE is normalized over the fitted components, as in `dp_scree()`. Printing the summary shows component importance by method and the privacy allocations.

References

Dwork C, Roth A (2014). "The Algorithmic Foundations of Differential Privacy." *Found. Trends Theor. Comput. Sci.*, **9**(3–4), 211–407. ISSN 1551-305X, doi:10.1561/0400000042.

Ramsay K, Spicker D (2025). "Improved subsample-and-aggregate via the private modified winsorized mean." Code available at https://github.com/12ramsake/PMWM, 2501.14095, https://arxiv.org/abs/2501.14095.

Durfee D (2023). "Unbounded Differentially Private Quantile and Maximum Estimation." In *Advances in Neural Information Processing Systems*, volume 36, 77691–77712. doi:10.52202/0752803394, https://proceedings.neurips.cc/paper_files/paper/2023/hash/f4b6ef2a78684dca2fb3f1c09372e
html.

Yu M, Ren Z, Zhou W (2024). "Gaussian differentially private robust mean estimation and inference." *Bernoulli*, **30**(4), 3059–3088.

Kim M, Jung S (2025). "Robust and Differentially Private Principal Component Analysis." *Statistical Analysis and Data Mining: An ASA Data Science Journal*, **18**(6), e70053. doi:10.1002/sam.70053.

See Also

dp_loading() for private principal component direction estimation. clipped_control(), pmwm_control(), and huber_control() for method-specific tuning parameters.

Examples

```
data(gau, package = "dppca")

# Use a small subset to keep the example fast.
X <- gau[1:100, ]
privacy <- privacy_control(eps = 3, delta = 1e-4)

# Estimate clipped private scree values and an ordinary PCA comparison.
set.seed(123)
out <- dp_scree(
  X,
  k = 2,
  privacy = privacy,
  method = "clipped",
  control = clipped_control(C_clip = 3)
)

out$nonprivate
out$clipped

# Multiple methods can be requested together by using a named control list.
# Loading is estimated once; each method receives the full scree budget.
# Change the split in privacy_control() to choose a different allocation.
```

gau — *Five Gaussian Cluster Dataset*

Description

A synthetic 20-dimensional Gaussian cluster dataset generated from multivariate normal distributions. It is used as an example for principal component analysis and differentially private PCA visualization.

Usage

```
gau
```

Format

A data frame with 1,000 observations and 20 numerical variables named `V1`, `V2`, ..., `V20`.

Details

The dataset `gau` contains 1,000 observations in 20 dimensions, with 200 observations in each of five groups, generated with seed 123. For each group $g = 1, \ldots, 5$, observations are sampled independently from a 20-dimensional multivariate normal distribution $N_{20}(\mu_g, \Sigma_g)$. The first two groups have covariance matrix I_{20}, the third and fourth groups have covariance matrix $5I_{20}$, and the fifth group has covariance matrix $8I_{20}$. The group mean vectors differ across selected coordinate blocks, producing both separated and partially overlapping cluster structures.

Source

Generated by the authors of the package.

gau_g — *Five Gaussian Cluster Dataset with Group Labels*

Description

A grouped version of `gau`, with an additional group label column.

Usage

```
gau_g
```

Format

A data frame with 1,000 observations and 21 variables. The first 20 columns, named `V1`, `V2`, ..., `V20`, are numerical variables identical to those in `gau`. The last column, `group`, is a character variable with values `group1`, `group2`, `group3`, `group4`, and `group5`.

Details

The dataset gau_g contains the same 1,000 observations and 20 numerical variables as gau, with 200 observations per group. An additional column, group, identifies the five Gaussian clusters. The group labels correspond to the multivariate normal distributions described in gau.

Source

Generated by the authors of the package.

---

huber_control *Control options for Huber scree estimation*

Description

Creates a control list for method = "huber" in dp_scree() or the scree configuration of dp_pca().

Usage

```
huber_control(
  k_min_m2,
  k_max_m2,
  m2_frac,
  mu0 = 0,
  eta0 = 1,
  T = NULL,
  M = NULL
)
```

Arguments

k_min_m2, k_max_m2	Required public integer bounds for the dyadic scale grid, with k_min_m2 < k_max_m2.
m2_frac	Required fraction in (0, 1) of epsilon used for the private scale step.
mu0	Public initial mean estimate (0 by default).
eta0	Positive public gradient step size (1 by default).
T	Number of gradient iterations; NULL (default) uses ceiling(log(m)), where m = floor(n / 2).
M	Number of scale-estimation blocks; NULL (default) uses floor(sqrt(m / 2)), capped by available input pairs.

Details

The Huber method privately estimates a robust scale for each component using a noisy dyadic histogram, then performs projected noisy gradient updates to estimate the mean of squared pair differences. The scale proxy sets the Huber threshold; it is not a raw second moment. At least eight observations are required.

The private scale step uses `m2_frac * epsilon / k` for each component and no delta. Joint gradient updates use the remaining epsilon and all delta, calibrated across components and iterations. Each update is projected to nonnegative values; optional monotone adjustment follows estimation. Tuning parameters must be public or have separate privacy accounting.

Value

A control list for `method = "huber"`.

References

Yu M, Ren Z, Zhou W (2024). "Gaussian differentially private robust mean estimation and inference." *Bernoulli*, **30**(4), 3059–3088.

See Also

`dp_scree()` for computing differentially private scree estimates using these control options. `plot.dp_scree()` for plotting scree estimates.

Examples

```
huber_control(k_min_m2 = -10, k_max_m2 = 10, m2_frac = 1 / 4)
```

`loading_plot_control` *Control loading plot appearance*

Description

Specify colors, text sizes, labels, and heatmap settings for loading plots. Pass the resulting list through the `plot_control` argument of `plot.dp_loading()`.

Usage

```
loading_plot_control(
  nonprivate_color = "black",
  private_color = "#1F4E79",
  arrow_size = 0.7,
  point_size = 2.8,
  label_size = 4,
  heatmap_values = TRUE,
  heatmap_digits = 2L,
```

```
  heatmap_text_size = 3.5,
  base_size = 12,
  title_size = 12,
  nonprivate_title = "Non-private",
  private_title = "DP",
  xlab = NULL,
  ylab = NULL,
  heatmap_colors = c(low = "#2C5D8A", high = "#B44A4A")
)
```

Arguments

nonprivate_color, private_color	Colors of the non-private and private vectors or points, respectively.
arrow_size	Width of arrows in vector plots.
point_size	Size of points in point plots.
label_size	Size of variable labels in vector and point plots.
heatmap_values	Whether to display loading coefficients in heatmap cells. Default: TRUE.
heatmap_digits	Number of decimal places for heatmap labels.
heatmap_text_size	Size of coefficient labels in heatmaps.
base_size	Base font size for all loading plots.
title_size	Font size for panel titles.
nonprivate_title, private_title	Text used for the corresponding panel titles. The display name is appended in parentheses; NULL hides a title.
xlab, ylab	Axis labels for vector and point plots. NULL uses the selected PC names.
heatmap_colors	Named character vector with low and high colors for negative and positive loading coefficients. White represents zero. Private and non-private heatmaps use separate, symmetric color ranges based on their own displayed coefficients.

Value

A named list of validated loading plot settings.

`plot.dp_loading` *Plot stored private PC directions*

Description

Create loading plots from an existing dp_loading() fit. Plotting uses the stored direction matrices and does not re-estimate PCs or spend additional privacy budget.

Draw the combined plot stored in a dp_loading_plot object. The original directions remain available in x$loading.

Summarize the fitted PC directions stored in a dp_loading_plot object.

Display a previously constructed plot or redraw it using new plotting options without re-estimating the PC directions.

Usage

```
## S3 method for class 'dp_loading'
plot(
  x,
  axes = c(1, 2),
  type = c("vector", "point", "heatmap"),
  heatmap_components = NULL,
  variables = 10,
  plot_control = NULL,
  ...
)

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

## S3 method for class 'dp_loading_plot'
summary(object, ...)

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

Arguments

x	A dp_loading_plot object.
axes	Two distinct PC indices for vector and point plots, displayed on fixed axes from -1.2 to 1.2. Default: c(1, 2).
type	One or more of "vector", "point", and "heatmap". The default shows all three types. With only one stored component, the default shows a heatmap because vector and point plots need two axes.
heatmap_components	PC indices to include in a heatmap; NULL (default) uses axes.
variables	Variables to display. A single positive integer selects the variables with the largest Euclidean norms across the displayed PC coefficients (top 10 by default). Use "all" or NULL for every variable, or provide variable names or a vector of column indices. Selection is performed separately for each version and display.

plot_control	Optional appearance settings from loading_plot_control() or a named list of its options.
...	Options passed to plot.dp_loading(). With no additional options, redraw the stored plot using its original settings.
object	A dp_loading_plot object.

Details

Vector and point plots use fixed axes from -1.2 to 1.2 and show variable labels. Each heatmap scales colors over its own displayed coefficients, so comparable coefficients can have different colors between versions. An ordinary PCA reference is shown automatically only when it was stored at fitting; it is not a private release.

Value

Draws the combined plot and invisibly returns a dp_loading_plot object with loading (the original fit) and plot (individual plots and the combined plot$all). Printing the saved object displays the plot.

Invisibly returns x.

A summary.dp_loading object.

Invisibly returns a dp_loading_plot object.

---

plot.dp_pca *Plot an integrated PCA fit*

Description

Uses the stored estimates without estimating loading, scree or score again. Use set.seed() before plotting to reproduce synthetic samples.

Usage

```
## S3 method for class 'dp_pca'
plot(
  x,
  display = "all",
  scree = list(),
  score = list(),
  loading = list(),
  variable = list(),
  biplot = list(),
  sampling = NULL,
  center = TRUE,
  variables = 10,
  base_size = 12,
  ...
)
```

Arguments

x	A fitted dp_pca() object.
display	"all" (default) or a vector selecting "scree", "score", "loading", "variable", and "biplot".
scree	Plot options: type = "pve" or "scree", and plot_control = NULL (scree_plot_control()).
score	Plot options: type = "histogram" or "sample", and plot_control = NULL (score_plot_control()). For grouped plots, the color map and legend settings also apply to the biplot.
loading	Plot options: type = "vector", "point", or "heatmap", heatmap_components = NULL, and plot_control = NULL (loading_plot_control()). Multiple types require display = "loading".
variable	Plot options: plot_control = NULL. Shows labeled arrows and the correlation-style unit circle.
biplot	Plot options: alpha = 0 (between 0 and 1), labels = TRUE, and plot_control = NULL.
sampling	A sampling_control() object with one method, shared by the score sample plot and biplot. By default, DP sample size is the input sample size, and non-private plots use observed scores. An explicit sample_size samples both versions from their histograms. For groups, sizes are allocated across groups; zero-mass groups yield no points.
center	Whether to shift displayed private scores by the midpoint of the private frame (TRUE by default). Shared by score and biplot; the biplot arrows start at the origin when TRUE. Ordinary reference scores remain centered at zero. This is display-only post-processing.
variables	Variables to display: positive top-N count (10 by default), "all", NULL, names, or indices. Top-N selection uses norms of private loading coefficients on the two selected axes. Selection is shared by loading, variable and biplot panels, including non-private comparisons.
base_size	Base font size (12 by default); local plot controls override it.
...	Unused arguments are rejected.

Details

Plotting reuses the stored estimates; non-private panels appear automatically if the fit contains them. Synthetic sampling is post-processing, and set.seed() can reproduce sampled displays.

Variable-plot coordinates use the covariance reconstructed from the k private scree estimates and directions; they need not equal observed variable–score correlations. Variables with zero implied variance are omitted.

Biplot coordinates divide sampled scores by d^alpha and multiply private direction coefficients by d^alpha, where d = sqrt((n - 1) * scree) on the selected axes. A zero scree value with positive alpha suppresses that axis. Arrows use one display multiplier computed from the private sample and shared with the ordinary comparison; empty or zero-radius samples omit arrows. The comparison uses the same square biplot frame.

Centering translates only displayed coordinates. Combined grouped pages show one group legend, while individual ggplot objects retain their own. The returned object stores both the combined pages and individual plots.

Value

Invisibly returns `private` and, when requested, `nonprivate` lists. Each contains `plots`, `page`, `sampled_points` (unshifted), `biplot`, `frames` and `direction_sign`. Biplot metadata includes the alpha-transformed and displayed coordinates, `scale`, `arrow_scale`, `score_center`, `score_radius`, `estimated_center`, `centered`, `arrow_start`, `arrow_endpoints` and display frame. The plots are always drawn; `out <- plot(fit)` also keeps these results for reuse. Multiple loading displays with both versions additionally return a `comparison` patchwork.

`plot.dp_score` *Plot a stored private score histogram*

Description

Reuses the stored frame and histograms. Synthetic sampling is post-processing; no direction, frame or histogram mechanism is run again.

Usage

```
## S3 method for class 'dp_score'
plot(
  x,
  type = c("histogram", "sample"),
  sampling_control = NULL,
  plot_control = NULL,
  center = TRUE,
  ...
)

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

## S3 method for class 'dp_score_plot'
summary(object, ...)

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

Arguments

`x`	A plotting result returned by `plot.dp_score()` or `plot.dp_score_group()`.
`type`	`"histogram"`, `"sample"`, or both (`c("histogram", "sample")` by default). Samples are synthetic for DP histograms; ordinary-reference sample plots show centered observed scores unless an explicit `sample_size` is requested.

`sampling_control`
: Optional `sampling_control()` object used for `"sample"`; `NULL` uses defaults.

`plot_control`
: Optional `score_plot_control()` appearance settings; `NULL` uses defaults.

`center`
: Whether to shift displayed private coordinates by the midpoint of the private frame (`TRUE` by default). Ordinary reference coordinates remain centered at zero. This changes only the display.

`...`
: Passed to the summary print method.

`object`
: A plotting result returned by `plot.dp_score()` or `plot.dp_score_group()`.

Details

Non-private reference panels appear automatically when retained at fitting. Combined grouped displays show one legend per histogram method; individual plots retain their legends. Groups with no sampled points are omitted from sampled-point legends.

Value

Invisibly returns a `dp_score_plot` object with `score`, optional `sample`, and `plot` components, while displaying `plot$all`.

Invisibly returns the original or rebuilt comparison result.

A score-histogram summary with stored synthetic point counts.

Invisibly returns `x`.

Examples

```
# After estimating once:
# summary(score_fit)
# plots <- plot(score_fit, type = "histogram")
# plot(score_fit, type = "sample",
#      sampling_control = sampling_control(sample_size = 500))
```

---

plot.dp_score_group *Plot stored group-wise private score histograms*

Description

Uses the same display options as `plot.dp_score()`. Group sample sizes follow observed proportions if non-private counts were retained; otherwise they are allocated equally across public group labels.

Usage

```
## S3 method for class 'dp_score_group'
plot(
  x,
  type = c("histogram", "sample"),
  sampling_control = NULL,
  plot_control = NULL,
  center = TRUE,
  ...
)

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

## S3 method for class 'dp_score_plot_group'
summary(object, ...)

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

Arguments

x	A grouped result from dp_score() with group supplied.
type	"histogram", "sample", or both (c("histogram", "sample")) by default). Samples are synthetic for DP histograms; ordinary-reference sample plots show centered observed scores unless an explicit sample_size is requested.
sampling_control	Optional sampling_control() object used for "sample"; NULL uses defaults.
plot_control	Optional score_plot_control() appearance settings; NULL uses defaults.
center	Whether to shift displayed private coordinates by the midpoint of the private frame (TRUE by default). Ordinary reference coordinates remain centered at zero. This changes only the display.
...	Passed to the summary print method.
object	A plotting result returned by plot.dp_score() or plot.dp_score_group().

Value

Invisibly returns a dp_score_plot_group object while displaying plot$all.

---

plot.dp_scree *Plot stored private scree estimates*

Description

Plot stored private scree estimates

Usage

```
## S3 method for class 'dp_scree'
plot(x, type = c("pve", "scree"), plot_control = NULL, ...)
```

Arguments

x	A dp_scree object.
type	One or both quantities to plot: "pve" for proportions of variance explained and "scree" for scree values. The default draws both side by side. Supply one value to draw only that quantity.
plot_control	Optional list from scree_plot_control(). If NULL, default plotting settings are used.
...	Further arguments, reserved for compatibility with the generic.

Details

Uses stored estimates only, so repeated plotting consumes no new privacy budget. The ordinary reference is non-private comparison output and is displayed automatically when present in the fitted result.

Value

Invisibly returns x after drawing the requested plot(s).

---

pmwm_control *Control options for private modified winsorized scree estimation*

Description

Creates a control list for method = "pmwm" in dp_scree() or the scree configuration of dp_pca().

Usage

```
pmwm_control(a, b, trim_const, eta, beta = 1.001, split_mode = FALSE)
```

Arguments

a, b	Required finite public bounds for private quantile cutoffs, with a < b, on the scale of squared pair differences.
trim_const	Required positive public trimming constant.
eta	Required public tuning value in [0, 0.5).
beta	Geometric quantile-search base greater than 1 (1.001 by default).
split_mode	Whether to split score pairs between quantile and mean estimation (FALSE by default).

Details

PMWM estimates lower and upper private quantiles of squared pair differences, truncates the cutoffs to `[a, b]`, winsorizes the data, and perturbs the mean vector. The cutoffs, not the original data, are bounded. If the estimated upper cutoff is smaller than the lower, the cutoffs are reset to `a` and `b`.

With `k` components, each of `2k` private quantiles receives `epsilon / (4k)`; Gaussian mean estimation receives the remaining half of epsilon and all delta. If `n_q` is the number of paired rows used for quantile estimation, the trimming proportion is

$$p = \min\{\max(\text{trim_const}/n_q, \eta), 0.49\}.$$

With `split_mode = TRUE`, the pair rows are split between quantile and mean estimation (at least four observations are required); otherwise all rows are reused under the same privacy allocation. Tunings must be public or selected through an accounted private procedure.

Value

A control list for `method = "pmwm"`.

References

Ramsay K, Spicker D (2025). "Improved subsample-and-aggregate via the private modified winsorized mean." Code available at https://github.com/12ramsake/PMWM, 2501.14095, https://arxiv.org/abs/2501.14095.

Durfee D (2023). "Unbounded Differentially Private Quantile and Maximum Estimation." In *Advances in Neural Information Processing Systems*, volume 36, 77691–77712. doi:10.52202/0752803394, https://proceedings.neurips.cc/paper_files/paper/2023/hash/f4b6ef2a78684dca2fb3f1c09372e
html.

See Also

`dp_scree()` for computing differentially private scree estimates using these control options. `plot.dp_scree()` for plotting scree estimates.

Examples

```
pmwm_control(a = 0, b = 20, trim_const = 10, eta = 0.01)
```

`privacy_control` *Allocate privacy budgets for shared PCA estimation*

Description

Allocate privacy budgets for shared PCA estimation

Usage

```
privacy_control(
  eps = NULL,
  delta = NULL,
  split = c(loading = 1/3, scree = 1/3, score = 1/3),
  loading = NULL,
  scree = NULL,
  score = NULL
)
```

Arguments

eps, delta	Total privacy parameters. eps must be positive and finite, and delta must lie in (0, 1). Leave both NULL when supplying absolute component budgets instead.
split	Named positive proportions that sum to one. The same proportions apply to eps and delta. Include loading alone, or loading with scree, score, or both. When split is omitted, the fitting function automatically divides the total budget equally among its required components: loading for dp_loading(), loading and scree for dp_scree(), loading and score for dp_score(), or all three for dp_pca(). An explicitly supplied split is used exactly as specified, without automatic redistribution.
loading, scree, score	Optional absolute component budgets, each a named c(eps = ..., delta = ...) vector. Supply loading alone or with scree, score, or both, without total budgets or a split argument. Each component must have positive finite eps and delta in (0, 1); the component deltas must sum to less than one. Absolute budgets may allocate different proportions of eps and delta.

Details

The estimation function resolves the required component budgets: loading for dp_loading(), loading plus scree for dp_scree(), loading plus score for dp_score(), and all three for dp_pca(). With total eps and delta and no explicit split, the constructor initially shows equal thirds, but the fitting function resolves the actual equal shares. An explicit split or absolute budget must match the required components.

With supplied V_dp, the loading allocation records the prior direction release; it is not spent again or redistributed. Score epsilon is divided 35% to the private frame and 65% to the histogram; all score delta goes to the histogram. When multiple scree or score estimators are requested, each receives its respective full allocation. Releasing them together requires privacy composition; the recorded totals are allocations, not cumulative privacy expenditure. Preprocessing and supplied directions require their own privacy justification.

Value

A dppca_privacy_control list with total, a named c(eps, delta) vector, and budgets, a list of the supplied component budget vectors in loading, scree, score order.

Examples

```
# With no split, the fitting function resolves equal shares:
# loading only (100%), scree/score (50%-50%), or PCA (one-third each).
privacy_control(eps = 6, delta = 1e-4)

# Alternatively, specify the loading-only absolute budget.
privacy_control(loading = c(eps = 6, delta = 1e-4))

# Explicit splits customize allocation for scree or score.
privacy_control(
  eps = 6,
  delta = 1e-4,
  split = c(loading = 0.5, scree = 0.5)
)

# Score estimation: a different allocation for its two components.
privacy_control(
  eps = 6,
  delta = 1e-4,
  split = c(loading = 0.4, score = 0.6)
)

# Absolute budgets, without total budgets or split.
privacy_control(
  loading = c(eps = 2, delta = 2e-5),
  scree = c(eps = 4, delta = 8e-5)
)
```

---

sampling_control *Control options for synthetic score sampling*

Description

Creates a control list for generating synthetic score points from a score histogram. Sampling from a released private histogram is post-processing under the conditions below; it does not alter the histogram mechanism. `plot.dp_pca()` can also use these controls to sample an empirical reference histogram, which does not make that reference private.

Usage

```
sampling_control(
  method = "uniform",
  sample_size = NULL,
  bandwidth_scale = NULL
)
```

Arguments

method	Character vector specifying the sampling construction. "center" selects a histogram bin according to its histogram probability and samples around that bin center using Gaussian jitter. "uniform" selects a histogram bin according to its histogram probability, samples uniformly inside the selected bin, and then adds Gaussian jitter. The default is "uniform". Both methods can be requested with c("center", "uniform") in plot.dp_score(); plot.dp_pca() requires exactly one sampling method.
sample_size	Optional positive integer giving the synthetic sample size for each requested sampling construction. For plot.dp_score(), this is the number of synthetic score points, or the total across all groups for a grouped fit. For plot.dp_pca(), it is the size shared by score and biplot samples within each version; an explicit value also samples non-private reference histograms when comparison panels are requested. With NULL, private samples use the number of input observations and integrated non-private panels use observed scores. The default private sample size assumes that the input sample size is public. See dp_score() and dp_pca() for grouped sampling weights and the applicable privacy conditions.
bandwidth_scale	Gaussian smoothing scale. If NULL, method-specific defaults are used: 0.5 for "center" and 0.25 for "uniform". A single nonnegative number applies the same scale to all requested sampling methods. A named numeric vector can specify method-specific values, for example c(center = 0.5, uniform = 0.2); it must include every requested method. A scale of zero disables Gaussian jitter.

Details

For released histogram probabilities p_k and cells B_k, first draw

$$J \sim \text{Categorical}(p_1, \ldots, p_m).$$

With method = "center", sample S^* = c_J + Z, where c_J is the selected cell center. With method = "uniform", sample U uniformly in B_J and use S^* = U + Z. In both cases Gaussian jitter has covariance

$$\text{Cov}(Z) = c^2 \, \text{diag}(\Delta_x^2, \Delta_y^2),$$

where c is bandwidth_scale and the deltas are bin widths.

Sampling uses only released histogram probabilities and bin geometry. With publicly chosen settings it is post-processing and requires no new privacy budget. Jitter can place points outside the frame; an all-zero histogram yields no sampled points and a warning.

Value

A dppca_sampling_control list with components method, sample_size, and bandwidth_scale. The bandwidth values are resolved to a named numeric vector, while sample_size = NULL is resolved later by the plotting function.

Examples

```
sampling_control()

sampling_control(
  method = c("center", "uniform"),
  sample_size = 1000,
  bandwidth_scale = c(center = 0.5, uniform = 0.2)
)
```

`score_plot_control` *Control options for score plots*

Description

Creates a control list for the appearance of plots produced by `plot.dp_score()`, including grouped score plots. Score plots are constructed with `ggplot2` and combined with `patchwork`.

Usage

```
score_plot_control(
  color = "#6A5ACD",
  scatter_alpha = 0.6,
  scatter_size = 1.8,
  hist_alpha_range = c(0, 1),
  base_size = 12,
  title_size = 14,
  scatter_title = "Non-Private Scatter",
  nonprivate_title = "Non-Private Histogram",
  add_title = "Add DP Histogram",
  sparse_title = "Sparse DP Histogram",
  xlab = NULL,
  ylab = NULL,
  group_colors = NULL,
  group_ncol = 3L,
  show_group_legend = TRUE
)
```

Arguments

`color`	Single color used for non-group score plots. The default is `"#6A5ACD"`.
`scatter_alpha`	Number in `[0, 1]` controlling point transparency in non-private and sampled scatter panels. The default is `0.6`.
`scatter_size`	Positive number controlling point size in non-private and sampled scatter panels. The default is `1.8`.

`hist_alpha_range`
: Numeric vector of length 2 giving the minimum and maximum alpha values used for histogram cells. Values must satisfy `0 <= hist_alpha_range[1] <= hist_alpha_range[2] <= 1`. The default is `c(0, 1)`.

`base_size`
: Positive base font size for score plots. The default is `12`.

`title_size`
: Positive plot-title font size. The default is `14`.

`scatter_title`
: Optional title for the non-private scatter panel. The default is `"Non-Private Scatter"`.

`nonprivate_title`
: Optional title for the non-private histogram panel. The default is `"Non-Private Histogram"`.

`add_title`
: Optional title for the additive private histogram panel. The default is `"Add DP Histogram"`.

`sparse_title`
: Optional title for the sparse private histogram panel. The default is `"Sparse DP Histogram"`.

`xlab`
: Optional x-axis label. If `NULL`, the selected PC name is used.

`ylab`
: Optional y-axis label. If `NULL`, the selected PC name is used.

`group_colors`
: Optional nonempty character vector of colors for grouped score plots. Named colors are matched to group labels; missing groups receive generated colors. Unnamed colors are recycled in group order. If `NULL`, valid color labels are used directly, otherwise a palette is generated automatically.

`group_ncol`
: Positive integer retained for backward compatibility. Grouped score plots are currently overlaid by group color rather than arranged as separate group-specific panels, so this option is not used by `plot.dp_score()`. The default is `3`.

`show_group_legend`
: Logical value indicating whether to display the group legend in grouped score plots. The default is `TRUE`. In combined grouped displays, each histogram method uses one legend; standalone panels retain their own legends. Groups with no sampled points are omitted from sampled-point legends.

Value

A list of plotting options for `plot.dp_score()`.

Examples

```
score_plot_control()

score_plot_control(
  scatter_size = 2.2,
  title_size = 13,
  nonprivate_title = "Non-Private score histogram"
)
```

scree_plot_control *Control options for scree and PVE plots*

Description

Creates a control list for the appearance of plots produced by `plot.dp_scree()`. The plotting function uses base R graphics.

Usage

```
scree_plot_control(
  title = NULL,
  xlab = NULL,
  ylab = NULL,
  legend_position = "topright",
  legend_labels = NULL,
  col = NULL,
  lty = NULL,
  pch = NULL,
  lwd = 1,
  point_cex = 1,
  cex_main = 1.5,
  cex_lab = 1.2,
  cex_axis = 1.1,
  cex_legend = 1.1,
  legend_bty = "n"
)
```

Arguments

`title`	Optional plot title. If `NULL`, `plot.dp_scree()` uses `"DP PVE Plot"` when `type = "pve"` and `"DP Scree Plot"` when `type = "scree"`.
`xlab`	Optional x-axis label. If `NULL`, `"Component"` is used.
`ylab`	Optional y-axis label. If `NULL`, `"PVE"` is used for PVE and `"Scree Value"` for scree values.
`legend_position`	Legend position passed to `graphics::legend()`. The default is `"topright"`.
`legend_labels`	Optional named character vector giving legend labels. Recognized names are `"nonprivate"`, `"clipped"`, `"pmwm"`, and `"huber"`.
`col`	Optional named character vector of plotting colors, using the same series names as `legend_labels`.
`lty`	Optional named vector of line types, using the same series names as `legend_labels`.
`pch`	Optional named vector of point symbols, using the same series names as `legend_labels`.
`lwd`	Positive line width. The default is `1`.

point_cex	Positive point-size multiplier. The default is 1.
cex_main	Positive title-size multiplier. The default is 1.5.
cex_lab	Positive axis-label-size multiplier. The default is 1.2.
cex_axis	Positive axis-text-size multiplier. The default is 1.1.
cex_legend	Positive legend-text-size multiplier. The default is 1.1.
legend_bty	Box type passed to graphics::legend(). The default "n" suppresses the legend border.

Value

A list of plotting options for plot.dp_scree().

Examples

```
scree_plot_control()

scree_plot_control(
  title = "PVE comparison",
  legend_position = "topright",
  cex_main = 1.4
)
```

---

vowel *Vowel Recognition Dataset*

Description

Acoustic measurements of eleven British English vowels spoken by fifteen speakers.

Usage

```
vowel
```

Format

A data frame with 990 observations and 14 variables:

Split Training or test set (Train, Test).

Speaker Speaker identifier (15 levels).

Sex Speaker sex (Male, Female).

F0 Acoustic feature 0.

F1 Acoustic feature 1.

F2 Acoustic feature 2.

F3 Acoustic feature 3.

F4 Acoustic feature 4.

F5 Acoustic feature 5.

F6 Acoustic feature 6.

F7 Acoustic feature 7.

F8 Acoustic feature 8.

F9 Acoustic feature 9.

Class Vowel category (11 levels).

Details

The dataset contains 990 observations and 14 variables, including ten numeric acoustic features and four categorical variables.

This dataset was obtained from the UCI file `vowel-context.data`. Categorical codes were converted to factors; numeric acoustic measurements were preserved.

Original dataset license: CC BY 4.0. https://creativecommons.org/licenses/by/4.0/

Source

UCI Machine Learning Repository. doi:10.24432/C58P4S

References

Deterding, D., Niranjan, M., and Robinson, T. (1988). Connectionist Bench (Vowel Recognition - Deterding Data). UCI Machine Learning Repository.

Turney, P. (1993). Robust Classification With Context-Sensitive Features.

Index

∗ **datasets**
gau, 17
gau_g, 17
vowel, 35

clipped_control, 3
clipped_control(), *8*, *14*, *16*

dp_loading, 5
dp_loading(), *5*, *8–10*, *12–14*, *16*, *21*, *29*
dp_pca, 7
dp_pca(), *3, 4*, *8*, *13*, *18*, *23*, *27*, *29*, *31*
dp_score, 10
dp_score(), *4*, *11*, *26*, *29*, *31*
dp_scree, 13
dp_scree(), *3*, *15*, *18, 19*, *27–29*
dppca_app, 4

gau, *17*, 17, *18*
gau_g, 17
graphics::legend(), *34, 35*

huber_control, 18
huber_control(), *8*, *14*, *16*

loading_plot_control, 19
loading_plot_control(), *22, 23*

plot.dp_loading, 20
plot.dp_loading(), *19*, *22*
plot.dp_loading_plot *(*plot.dp_loading*)*, 20
plot.dp_pca, 22
plot.dp_pca(), *13*, *30, 31*
plot.dp_score, 24
plot.dp_score(), *13*, *24–26*, *31–33*
plot.dp_score_group, 25
plot.dp_score_group(), *13*, *24–26*
plot.dp_score_plot *(*plot.dp_score*)*, 24
plot.dp_score_plot_group *(*plot.dp_score_group*)*, 25
plot.dp_scree, 26
plot.dp_scree(), *3*, *19*, *28*, *34, 35*
pmwm_control, 27
pmwm_control(), *8*, *14*, *16*
print.dp_loading *(*dp_loading*)*, 5
print.dp_loading_plot *(*plot.dp_loading*)*, 20
print.dp_pca *(*dp_pca*)*, 7
print.dp_score *(*dp_score*)*, 10
print.dp_score_group *(*dp_score*)*, 10
print.dp_score_plot *(*plot.dp_score*)*, 24
print.dp_score_plot_group *(*plot.dp_score_group*)*, 25
print.dp_scree *(*dp_scree*)*, 13
print.summary.dp_loading *(*dp_loading*)*, 5
print.summary.dp_pca *(*dp_pca*)*, 7
print.summary.dp_score *(*dp_score*)*, 10
print.summary.dp_scree *(*dp_scree*)*, 13
privacy_control, 28
privacy_control(), *6*, *8*, *11*, *14*

sampling_control, 30
sampling_control(), *23*, *25, 26*
score_plot_control, 32
score_plot_control(), *23*, *25, 26*
scree_plot_control, 34
scree_plot_control(), *23*, *27*
summary.dp_loading *(*dp_loading*)*, 5
summary.dp_loading_plot *(*plot.dp_loading*)*, 20
summary.dp_pca *(*dp_pca*)*, 7
summary.dp_score *(*dp_score*)*, 10
summary.dp_score_group *(*dp_score*)*, 10
summary.dp_score_plot *(*plot.dp_score*)*, 24
summary.dp_score_plot_group *(*plot.dp_score_group*)*, 25
summary.dp_scree *(*dp_scree*)*, 13

vowel, 35