Package 'dppca'

June 4, 2026

Title Differentially Private Principal Component Analysis Visualization

Version 0.1.0

Description Provides tools for differentially private principal component analysis (PCA) visualization. It includes functions for estimating private principal component directions, constructing private scree and proportion of variance explained summaries, and visualizing two-dimensional PCA score summaries using additive and sparse histogram mechanisms. Group-wise score visualizations and an interactive 'shiny' app are also provided. Private principal component directions are based on Kim and Jung (2025) <doi:10.1002/sam.70053>. Private scree summaries use mechanisms motivated by 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 plot frames use smooth sensitivity quantiles from Nissim, Raskhodnikova and Smith (2007) <doi:10.1145/1250790.1250803>. Private score histograms use additive and sparse histogram ideas from 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 dplyr, ggplot2, graphics, grDevices, patchwork, rARPACK, Rdpack, rlang, stats, VGAM

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

NeedsCompilation no

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

Maintainer Yejin Jo <yejinjo0220@gmail.com>

Repository CRAN

Date/Publication 2026-06-04 12:00:07 UTC

Contents

adult . . . 2
clipped_control . . . 3
dppca_app . . . 4
dp_pc_dir . . . 5
dp_score . . . 7
dp_score_group . . . 10
dp_score_plot . . . 12
dp_score_plot_group . . . 13
dp_scree . . . 15
dp_scree_plot . . . 18
gau . . . 21
gau_g . . . 21
huber_control . . . 22
pmwm_control . . . 24

Index **26**

`adult` *Adult numeric data*

Description

A numerical subset of the Adult dataset from the UCI Machine Learning Repository. The original Adult dataset is based on data extracted from the 1994 United States Census database.

Usage

```
adult
```

Format

A data frame with 32,561 rows and 5 columns:

age Age of the individual.

education_num Number of years of education.

capital_gain Capital gain.

capital_loss Capital loss.

hours_per_week Number of working hours per week.

Details

This package dataset retains five numerical variables: `age`, `education_num`, `capital_gain`, `capital_loss`, and `hours_per_week`. These selected numerical variables contain no missing values in the original data file used here. The resulting dataset contains 32,561 observations.

Since the variables have substantially different units and scales, scaling is recommended before applying PCA-based methods.

Source

UCI Machine Learning Repository, Adult dataset. The Adult dataset is licensed under the Creative Commons Attribution 4.0 International (CC BY 4.0) license.

References

Becker B, Kohavi R (1996). "Adult." UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5XW20, https://archive.ics.uci.edu/dataset/2/adult.

`clipped_control` *Control options for clipped scree estimation*

Description

Creates a control list for the clipped-mean scree estimator used by `dp_scree()` and `dp_scree_plot()` when `method = "clipped"`.

Usage

```
clipped_control(C_clip)
```

Arguments

`C_clip`	Positive clipping threshold for squared principal component scores. This value has no default because it should be chosen according to the scale of the data.

Details

The clipped method estimates each scree value by clipping the squared scores at `C_clip` and then applying a sensitivity-calibrated Gaussian mechanism (Dwork and Roth 2014). Larger values of `C_clip` reduce clipping bias but increase the sensitivity, and therefore the scale of the privacy noise.

Value

A list of control options 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. `dp_scree_plot()` 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 app provides a graphical interface for `dp_scree_plot()` and `dp_score_plot()`, including private scree plot methods and histogram-based private score plot methods.

Usage

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

Arguments

`X`	Optional numeric matrix or data frame. If supplied, the app opens with this data as the initial dataset.
`group`	Optional group labels. This can be either a vector of length `nrow(X)` or a single column name in `X`.

Details

The app can be opened with built-in example datasets or with a user-supplied dataset. If `X` is supplied, the app starts with `X` as the initial dataset. If `group` is supplied, the score plot can use the group labels for coloring. The `group` argument can be either a vector of length `nrow(X)` or the name of a column in `X`. When `group` is a column name, that column is used as group labels and is removed from the PCA feature matrix.

Value

No return value. This function opens a Shiny application.

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_pc_dir` *Estimate principal component directions*

Description

This function returns the principal component directions used by the plotting and estimation functions in this package. By default, it computes the usual non-private directions from the sample covariance matrix. If `g_dppca = TRUE`, it computes private directions using the spherical-transformation version of g-DPPCA proposed by Kim and Jung (2025): it adds Gaussian noise to the spherical Kendall matrix and then extracts its leading eigenvectors.

Usage

```
dp_pc_dir(
  X,
  k,
  center = TRUE,
  standardize = FALSE,
  g_dppca = FALSE,
  eps = NULL,
  delta = NULL,
  cpp.option = FALSE
)
```

Arguments

`X`	A numeric matrix or data frame. Rows correspond to observations and columns correspond to variables.
`k`	Number of principal component directions to return. Must be an integer between `1` and the number of columns in `X`.
`center`	A logical value indicating whether to center the columns of `X` before computing principal component directions. The default is `TRUE`.

standardize	A logical value indicating whether to scale the columns of X by their sample standard deviations after optional centering. The default is FALSE.
g_dppca	Whether to compute private principal component directions using the spherical Kendall mechanism based on the g-DPPCA method. If FALSE, the usual non-private directions are computed from the sample covariance matrix. The default is FALSE.
eps	Positive number defining the epsilon privacy parameter for private principal component directions. Required when g_dppca = TRUE.
delta	Number in (0, 1) defining the delta privacy parameter for private principal component directions. Required when g_dppca = TRUE.
cpp.option	A logical value reserved for a future C++ implementation of the spherical Kendall matrix. Currently only FALSE is supported.

Details

The non-private option computes leading eigenvectors of the sample covariance matrix of the pre-processed data. The private option is based on the spherical Kendall mechanism of Kim and Jung (2025): it first forms the spherical Kendall matrix from pairwise normalized differences, adds symmetric Gaussian noise, and then computes leading eigenvectors. The final eigenvector matrix is re-orthonormalized by QR decomposition. For a detailed procedure and mathematical formulations, refer https://yejinjo0220.github.io/dppca/articles/pc_direction.

Value

A numeric matrix with ncol(X) rows and k columns. The columns are orthonormal principal component directions.

References

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.

Examples

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

# Use a small subset to keep the example fast.
X <- gau[1:200, ]

# Non-private principal component directions
V <- dp_pc_dir(X, k = 2)
head(V)


# Private principal component directions
set.seed(123)
V_private <- dp_pc_dir(
  X,
```

```
  k = 2,
  g_dppca = TRUE,
  eps = 2,
  delta = 1e-3
)
head(V_private)
```

---

dp_score *Differentially private score histograms*

Description

This function computes two-dimensional principal component scores and returns differentially private histogram estimates on the score space. It returns the score coordinates, the plotting frame, the non-private histogram, and the requested private histogram estimates.

Usage

```
dp_score(
  X,
  eps,
  delta,
  bins,
  method = c("add", "sparse"),
  center = TRUE,
  standardize = FALSE,
  g_dppca = FALSE,
  cpp.option = FALSE,
  axes = c(1, 2)
)
```

Arguments

X	A numeric matrix or data frame. Rows correspond to observations and columns correspond to variables.
eps	Positive number defining the total epsilon privacy parameter.
delta	Number in (0, 1) defining the total delta privacy parameter.
bins	Integer vector of length 2 defining the number of histogram bins along the first and second score axes, respectively.
method	Character vector specifying which private histogram methods to compute. Use "add" for the additive Gaussian histogram and "sparse" for the sparse thresholded histogram. The default is c("add", "sparse").
center	A logical value indicating whether to center the columns of X before computing principal component directions. The default is TRUE.

standardize	A logical value indicating whether to scale the columns of X by their sample standard deviations after optional centering. The default is FALSE.
g_dppca	A logical value indicating whether to use private principal component directions. The default is FALSE. See dp_pc_dir() for details.
cpp.option	A logical value passed to dp_pc_dir() when g_dppca = TRUE. The default is FALSE.
axes	Integer vector of length 2 specifying the principal components used to construct the score coordinates. The default is c(1, 2).

Details

Let v_a and v_b be the principal component directions selected by axes = c(a, b) for some $1 \leq a < b \leq ncol(X)$. After preprocessing, the score point for ith observation is $s_i = (x_i^\top v_a, x_i^\top v_b)$. A non-private score plot would display the points $s_1, \ldots, s_n$ directly. This function instead summarizes their empirical distribution by a two-dimensional histogram and releases private versions of the histogram for the visualization.

The plotting frame is constructed privately from the score coordinates. The frame center is estimated by coordinate-wise private medians, and the frame radius is estimated by the private 0.99 quantile of the Euclidean distances from this private center. The resulting private radius is inflated by a fixed factor and used to form a square plotting frame. The private frame is computed using a smooth-sensitivity based quantile mechanism (Nissim et al. 2007).

The private histogram is computed on the rectangular grid defined by the private frame and the bin counts in bins. Under row-level adjacency, changing one observation can increase one bin count by one and decrease another by one, giving ℓ_1 sensitivity at most 2 and ℓ_2 sensitivity at most $\sqrt{2}$ for the count vector.

Two private histogram mechanisms are supported:

- "add" constructs an additive differentially private histogram by adding Gaussian noise to all bin counts, clipping negative noisy counts to zero, and normalizing the result. This additive-noise approach is commonly used for private histograms; see Wasserman and Zhou (2010).
- "sparse" constructs a sparse differentially private histogram for settings where many bins are empty. It perturbs only nonzero empirical bin proportions and keeps bins whose noisy values exceed a stability threshold, following the stability-based private histogram idea of Karwa and Vadhan (2018).

The privacy parameters are allocated across the privacy-consuming steps. If g_dppca = FALSE, half of eps and delta is used for private frame construction and half for the private histogram. If g_dppca = TRUE, the parameters are split equally among private direction estimation, private frame construction, and private histogram release.

For a detailed procedure and mathematical formulations, refer https://yejinjo0220.github.io/dppca/articles/dp_score.

Value

A list with components:

score An $n \times 2$ matrix containing the PC scores for the two selected axes.

frame A list with components `xlim` and `ylim`.

none Data frame for the non-private empirical histogram.

add Data frame for the additive Gaussian private histogram, or `NULL` if not requested.

sparse Data frame for the sparse private histogram, or `NULL` if not requested.

method Character vector of private histogram methods used.

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.

Nissim K, Raskhodnikova S, Smith A (2007). "Smooth Sensitivity and Sampling in Private Data Analysis." In *STOC'07: Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, 75–84. ISBN 9781595936318, doi:10.1145/1250790.1250803.

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

`dp_score_plot()` for plotting the output of this function. `dp_score_group()` and `dp_score_plot_group()` for group-wise score histograms. `dp_pc_dir()` for private principal component direction estimation.

Examples

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

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

# Compute private two-dimensional PCA scores using the additive histogram method.
set.seed(123)
score_gau <- dp_score(
  X,
  eps = 2,
  delta = 1e-3,
  method = "add",
  bins = c(10, 10)
)

head(score_gau$score)
head(score_gau$add)
```

dp_score_group *Group-wise differentially private score histograms*

Description

This function computes two-dimensional principal component scores and releases group-wise differentially private histograms on a common score frame and grid. It is useful when observations have group labels and the low-dimensional score distribution should be compared across groups.

Usage

```
dp_score_group(
  X,
  group,
  eps,
  delta,
  bins,
  method = c("add", "sparse"),
  center = TRUE,
  standardize = FALSE,
  g_dppca = FALSE,
  cpp.option = FALSE,
  axes = c(1, 2)
)
```

Arguments

X	A matrix or data frame where rows correspond to observations and columns correspond to variables. X can additionally include a named column representing the group label for each observation.
group	Group labels. This can be a vector of length nrow(X) or a single column name in X. If a column name is supplied, that column is used as the group label and removed from the feature matrix.
eps	Positive number defining the total epsilon privacy parameter.
delta	Number in (0, 1) defining the total delta privacy parameter.
bins	Integer vector of length 2 defining the number of histogram bins along the first and second score axes, respectively.
method	Character vector specifying which private histogram methods to compute. Use "add" for the additive Gaussian histogram and "sparse" for the sparse thresholded histogram. The default is c("add", "sparse").
center	A logical value indicating whether to center the columns of X before computing principal component directions. The default is TRUE.
standardize	A logical value indicating whether to scale the columns of X by their sample standard deviations after optional centering. The default is FALSE.

g_dppca	A logical value indicating whether to use private principal component directions. The default is FALSE. See dp_pc_dir() for details.
cpp.option	A logical value passed to dp_pc_dir() when g_dppca = TRUE. The default is FALSE.
axes	Integer vector of length 2 specifying the principal components used to construct the score coordinates. The default is c(1, 2).

Details

The score directions, plotting frame, and histogram grid are shared across all groups. For each group g, the group-specific count in bin B_k is $c_k^{(g)} = \sum_i 1\{s_i \in B_k, g_i = g\}$. Private histograms are then computed separately for each group on the common grid. Because the groups form a partition of the rows, the group-wise histograms use the same histogram privacy parameters for each group by parallel composition.

Value

A list with components:

score An $n \times 2$ matrix containing the PC scores for the two selected axes.

frame A list with components xlim and ylim.

groups A named list of group-specific histogram outputs.

method Character vector of private histogram methods used.

See Also

dp_score_plot_group() for plotting group-wise score histograms. dp_score() for pooled score histograms.

Examples

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

# Compute private grouped PCA scores.
set.seed(123)
score_gau_g <- dp_score_group(
  gau_g,
  group = "group",
  eps = 3,
  delta = 1e-3,
  bins = c(8, 8)
)

head(score_gau_g$score)
head(score_gau_g$groups$group1$add)
```

dp_score_plot *Plot differentially private score histograms*

Description

This function computes and visualizes two-dimensional principal component score histograms, including the original scatter plot, the non-private empirical histogram, and one or more differentially private histogram estimates. It is a plotting wrapper around `dp_score()` and returns both the computed score output and `ggplot` objects.

Usage

```
dp_score_plot(
  X,
  eps,
  delta,
  bins,
  method = c("add", "sparse"),
  center = TRUE,
  standardize = FALSE,
  g_dppca = FALSE,
  cpp.option = FALSE,
  axes = c(1, 2)
)
```

Arguments

`X`	A numeric matrix or data frame. Rows correspond to observations and columns correspond to variables.
`eps`	Positive number defining the total `epsilon` privacy parameter.
`delta`	Number in `(0, 1)` defining the total `delta` privacy parameter.
`bins`	Integer vector of length 2 defining the number of histogram bins along the first and second score axes, respectively.
`method`	Character vector specifying which private histogram methods to compute. Use `"add"` for the additive Gaussian histogram and `"sparse"` for the sparse thresholded histogram. The default is `c("add", "sparse")`.
`center`	A logical value indicating whether to center the columns of `X` before computing principal component directions. The default is `TRUE`.
`standardize`	A logical value indicating whether to scale the columns of `X` by their sample standard deviations after optional centering. The default is `FALSE`.
`g_dppca`	A logical value indicating whether to use private principal component directions. The default is `FALSE`. See `dp_pc_dir()` for details.
`cpp.option`	A logical value passed to `dp_pc_dir()` when `g_dppca = TRUE`. The default is `FALSE`.
`axes`	Integer vector of length 2 specifying the principal components used to construct the score coordinates. The default is `c(1, 2)`.

Value

A list with components:

score The output of `dp_score()`.

plot A list containing the scatter plot, histogram panels, and the combined patchwork plot.

See Also

`dp_score()` for computing score histograms without plotting. `dp_score_plot_group()` for group-wise score histogram plots.

Examples

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

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

# Draw a private score plot using the additive histogram method.
set.seed(123)
score_plot <- dp_score_plot(
  X,
  eps = 3,
  delta = 1e-3,
  bins = c(8, 8),
  method = "add"
)
score_plot$plot$add

# Draw score plots for all available histogram methods.
set.seed(123)
score_plot <- dp_score_plot(
  X,
  eps = 3,
  delta = 1e-3,
  bins = c(8, 8)
)
score_plot$plot$all
```

---

dp_score_plot_group *Plot group-wise differentially private score histograms*

Description

This function computes and visualizes group-wise differentially private score histograms. It is a plotting wrapper around `dp_score_group()` and returns both the computed group-wise score output and `ggplot` objects.

Usage

```
dp_score_plot_group(
  X,
  group,
  eps,
  delta,
  bins,
  center = TRUE,
  standardize = FALSE,
  g_dppca = FALSE,
  cpp.option = FALSE,
  axes = c(1, 2),
  method = c("add", "sparse")
)
```

Arguments

X	A matrix or data frame where rows correspond to observations and columns correspond to variables. X can additionally include a named column representing the group label for each observation.
group	Group labels. This can be a vector of length nrow(X) or a single column name in X. If a column name is supplied, that column is used as the group label and removed from the feature matrix.
eps	Positive number defining the total epsilon privacy parameter.
delta	Number in (0, 1) defining the total delta privacy parameter.
bins	Integer vector of length 2 defining the number of histogram bins along the first and second score axes, respectively.
center	A logical value indicating whether to center the columns of X before computing principal component directions. The default is TRUE.
standardize	A logical value indicating whether to scale the columns of X by their sample standard deviations after optional centering. The default is FALSE.
g_dppca	A logical value indicating whether to use private principal component directions. The default is FALSE. See dp_pc_dir() for details.
cpp.option	A logical value passed to dp_pc_dir() when g_dppca = TRUE. The default is FALSE.
axes	Integer vector of length 2 specifying the principal components used to construct the score coordinates. The default is c(1, 2).
method	Character vector specifying which private histogram methods to compute. Use "add" for the additive Gaussian histogram and "sparse" for the sparse thresholded histogram. The default is c("add", "sparse").

Value

A list with components:

score The output of dp_score_group().

plot A list containing group-wise histogram plots.

group_colors Named vector of colors used for the groups.

See Also

dp_score_group() for computing group-wise score histograms without plotting. dp_score_plot() for pooled score histogram plots.

Examples

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

# Draw a private grouped score plot.
set.seed(123)
score_plot_gau_g <- dp_score_plot_group(
  gau_g,
  group = "group",
  eps = 3,
  delta = 1e-3,
  bins = c(8, 8)
)

score_plot_gau_g$plot$all
```

---

dp_scree *Differentially private scree values*

Description

This function computes estimates of scree values, eigenvalues of covariance matrix, for principal component analysis, including both the usual non-private estimates and differentially private estimates. The private estimates are computed as the private mean of the squared principal component scores. See Details for the estimating equations and method-specific construction.

Usage

```
dp_scree(
  X,
  k,
  method = c("clipped", "pmwm", "huber"),
  control = NULL,
  eps,
  delta,
  center = TRUE,
  standardize = FALSE,
  g_dppca = FALSE,
  cpp.option = FALSE,
  mono = TRUE
)
```

Arguments

X	A numeric matrix or data frame. Rows correspond to observations and columns correspond to variables.
k	Positive integer defining the number of leading principal components to estimate. Must be an integer between 1 and the number of columns in X.
method	Scree value estimation method or methods. One or more of "clipped", "pmwm", or "huber". If omitted, "clipped" is used.
control	Optional method-specific control list created by clipped_control(), pmwm_control(), or huber_control(). When multiple methods are requested, use a named list with method names.
eps	Positive number defining the total epsilon privacy parameter. If g_dppca = TRUE, it is split between private direction estimation and private scree estimation.
delta	Number in (0, 1) defining the total delta privacy parameter. If g_dppca = TRUE, it is split between private direction estimation and private scree estimation.
center	A logical value indicating whether to center the columns of X before computing principal component directions. The default is TRUE.
standardize	A logical value indicating whether to scale the columns of X by their sample standard deviations after optional centering. The default is FALSE.
g_dppca	A logical value indicating whether to use private principal component directions for scree estimation. The default is FALSE. See dp_pc_dir() for details.
cpp.option	A logical value passed to dp_pc_dir() when g_dppca = TRUE. The default is FALSE. When g_dppca = TRUE, dp_pc_dir() is called with arguments eps = eps / 2 and delta = delta / 2.
mono	A logical value indicating whether to apply monotone post-processing to the vector of private scree values. The default is TRUE.

Details

Let X denote the preprocessed data matrix and let v_l be the lth principal component direction. The lth score vector is $z_l = Xv_l$. The corresponding sample scree value can be written as

$$\hat{\lambda}_l = v_l^\top \widehat{\Sigma} v_l = \frac{1}{n-1}\sum_{i=1}^{n} z_{il}^2 = \frac{n}{n-1}\left(\frac{1}{n}\sum_{i=1}^{n} w_{il}\right), \qquad w_{il} = z_{il}^2.$$

Therefore, each scree value is estimated by privately estimating the mean of $w_{1l}, \ldots, w_{nl}$ and multiplying by $n/(n-1)$.

The supported methods differ in how this private mean is estimated:

- "clipped" clips the squared scores $w_{i\ell}$ at C_clip and then applies the Gaussian mechanism (Dwork and Roth 2014). This is the simplest option but depends directly on the clipping threshold.
- "pmwm" uses the private modified winsorized mean approach of Ramsay and Spicker (2025), adapted from the accompanying Python implementation into R. It privately estimates tail cutoffs, winsorizes the squared scores $w_{i\ell}$, and releases a noisy winsorized mean.

- "huber" uses a Huber-type private robust mean estimator based on noisy gradient descent, following Yu et al. (2024).

The argument g_dppca controls how the principal component directions are obtained. If g_dppca = FALSE, the directions are computed non-privately as an eigenvector of sample covariance, and the full privacy parameters eps and delta are used for private scree value estimation. If g_dppca = TRUE, the directions are computed privately using dp_pc_dir(). In that case, the privacy parameters are split equally: dp_pc_dir() receives eps = eps / 2 and delta = delta / 2 for private direction estimation, and the remaining eps / 2 and delta / 2 are used for private scree value estimation. When mono = TRUE, the final monotone adjustment is a post-processing step and does not change the privacy guarantee.

For a detailed procedure and mathematical formulations, refer https://yejinjo0220.github.io/dppca/articles/dp_scree.

Value

If one method is requested, a list with components:

- method: scree value estimation method.
- scree_np: non-private scree estimates.
- pve_np: non-private proportions of variance explained.
- scree: differentially private scree value estimates.
- pve: differentially private proportions of variance explained.

If multiple methods are requested, a named list of method-specific results is returned.

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.

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_pc_dir() for 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, ]

# Estimate the private scree values using the clipped mean method.
set.seed(123)
dp_scree(
  X,
  k = 2,
  method = "clipped",
  control = clipped_control(C_clip = 3),
  eps = 2,
  delta = 1e-3
)

# Other scree methods can be used by changing `method` and `control`, e.g.,
# method = "pmwm",
# control = pmwm_control(a = 0, b = 50, trim_const = 10, eta = 0.01)
#
# method = "huber",
# control = huber_control(k_min_m2 = -10, k_max_m2 = 10, m2_frac = 1 / 4)
```

---

dp_scree_plot *Plot differentially private scree estimates*

Description

This function computes and visualizes scree curves for principal component analysis, including the usual non-private curve and one or more differentially private estimates. It is a plotting wrapper around `dp_scree()` and returns a ggplot object.

Usage

```
dp_scree_plot(
  X,
  k,
  method = c("clipped", "pmwm", "huber"),
  control = NULL,
  eps,
  delta,
  center = TRUE,
  standardize = FALSE,
  g_dppca = FALSE,
  cpp.option = FALSE,
  mono = TRUE,
  type = c("pve", "scree")
)
```

Arguments

X	A numeric matrix or data frame. Rows correspond to observations and columns correspond to variables.
k	Positive integer defining the number of leading principal components to estimate. Must be an integer between 1 and the number of columns in X.
method	Scree estimation method or methods to plot. One or more of "clipped", "pmwm", or "huber". If omitted, "clipped" is used.
control	Optional method-specific control list, or a named list of control lists when multiple methods are requested. Use clipped_control(), pmwm_control(), and huber_control().
eps	Positive number defining the total epsilon privacy parameter. If g_dppca = TRUE, it is split between private direction estimation and private scree estimation.
delta	Number in (0, 1) defining the total delta privacy parameter. If g_dppca = TRUE, it is split between private direction estimation and private scree estimation.
center	A logical value indicating whether to center the columns of X before computing principal component directions. The default is TRUE.
standardize	A logical value indicating whether to scale the columns of X by their sample standard deviations after optional centering. The default is FALSE.
g_dppca	A logical value indicating whether to use private principal component directions for scree estimation. The default is FALSE. See dp_pc_dir() for details.
cpp.option	A logical value passed to dp_pc_dir() when g_dppca = TRUE. The default is FALSE. When g_dppca = TRUE, dp_pc_dir() is called with arguments eps = eps / 2 and delta = delta / 2.
mono	A logical value indicating whether to apply monotone post-processing to the private scree vector. The default is TRUE.
type	Quantity to plot. Use "pve" to plot proportions of variance explained and "scree" to plot raw scree values. The default is "pve".

Details

This function is a plotting wrapper around dp_scree(). For each requested method, it computes a private scree estimate and overlays it with the corresponding non-private curve. When type = "pve", the plotted quantity is the proportion of variance explained (PVE); when type = "scree", the raw scree values are shown.

To plot multiple methods, pass a character vector to method. If a method requires tuning parameters, pass control as a named list, for example control = list(clipped = clipped_control(), pmwm = pmwm_control(), huber = huber_control()).

For the estimating equations, privacy-budget allocation, and method-specific construction, see dp_scree().

Value

Invisibly returns a list with components:

- nonprivate: non-private scree and PVE values.
- results: method-specific dp_scree() outputs used in the plot.

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.

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_pc_dir()` for principal component direction estimation. `dp_scree()` for computing non-private and differentially private scree estimates. `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[1:200, ]

# Draw a private scree plot using the clipped mean method.
set.seed(123)
dp_scree_plot(
  X,
  k = 5,
  method = "clipped",
  control = clipped_control(C_clip = 3),
  eps = 3,
  delta = 1e-3
)

# Multiple scree methods can be overlaid by passing a vector to `method`
# and a named list to `control`, for example:
# dp_scree_plot(
#   X,
#   k = 5,
#   method = c("clipped", "pmwm", "huber"),
#   control = list(
#     clipped = clipped_control(C_clip = 3),
#     pmwm = pmwm_control(a = 0, b = 50, trim_const = 10, eta = 0.01),
#     huber = huber_control(k_min_m2 = -10, k_max_m2 = 10, m2_frac = 1 / 4)
#   ),
#   eps = 3,
#   delta = 1e-3
# )
```

gau *Five Gaussian clusters*

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 5,000 rows and 20 numerical variables. The variables are named `V1`, `V2`, ..., `V20`.

Details

The dataset contains 5,000 observations in 20 dimensions. It consists of five groups, with 1,000 observations in each group. The data were generated with seed 123. For each group $g = 1, \ldots, 5$, observations were 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 clusters with group labels*

Description

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

Usage

```
gau_g
```

Format

A data frame with 5,000 rows and 21 columns. The first 20 columns, named `V1`, `V2`, ..., `V20`, are the numerical variables from `gau`. The last column, `group`, is a group label with values `group1`, `group2`, `group3`, `group4`, and `group5`.

Details

The dataset contains the same 5,000 observations and 20 numerical variables as `gau`, together with a group label column. There are five groups, and each group contains 1,000 observations.

Source

Generated by the authors of the package.

`huber_control` *Control options for Huber scree estimation*

Description

Creates a control list for the Huber-type private scree estimator used by `dp_scree()` and `dp_scree_plot()` when `method = "huber"`.

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`
: Integers defining the lower and upper dyadic bin indices used in the private second-moment scale step. The histogram searches over scale levels 2^k for $k_{\min} \leq k \leq k_{\max}$. These values have no defaults because they should be chosen according to the scale of the data.

`m2_frac`
: Number in `(0, 1)` defining the fraction of the Huber scree privacy parameters allocated to the private second-moment scale step. This value has no default.

`mu0`
: Numeric initial value for Huber noisy gradient descent. The default is `0`.

`eta0`
: Positive number defining the fixed step size for Huber noisy gradient descent. The default is `1`.

`T` Optional positive integer defining the number of noisy gradient descent iterations. If `NULL`, the implementation uses $\lceil \log n \rceil$, where n is the number of observations.

`M` Optional positive integer defining the number of blocks used in the private second-moment scale step. If `NULL`, the implementation uses $\lfloor \sqrt{n}/2 \rfloor$, where n is the number of observations.

Details

The Huber method estimates the mean of squared principal component scores by noisy gradient descent on the Huber loss. It follows the Huber-type private robust mean approach of Yu et al. (2024).

The method first privately estimates a scale proxy for the squared scores, denoted by m_2. This scale proxy is then used to choose the Huber robustification level for noisy gradient descent. The parameters `k_min_m2`, `k_max_m2`, and `m2_frac` control this private scale-proxy step, while `mu0`, `eta0`, `T`, and `M` control the subsequent noisy gradient descent routine.

The dyadic indices `k_min_m2` and `k_max_m2` define the search range for the private histogram used to estimate m_2. The histogram searches over candidate scale levels 2^k satisfying $k_{\min} \leq k \leq k_{\max}$. Because this range depends on the scale of the squared scores, these arguments are intentionally not given defaults.

The argument `m2_frac` determines how the Huber scree privacy parameters are split between the private scale-proxy step and the noisy gradient descent step. If $(\epsilon_{\text{scree}}, \delta_{\text{scree}})$ denotes the privacy parameters available for Huber scree estimation, then $m2_frac \cdot (\epsilon_{\text{scree}}, \delta_{\text{scree}})$ is used to privately estimate m_2, while $(1 - m2_frac) \cdot (\epsilon_{\text{scree}}, \delta_{\text{scree}})$ is used for Huber noisy gradient descent.

The remaining parameters have default values. The default `mu0 = 0` is the initial value for noisy gradient descent, and the default `eta0 = 1` is the fixed step size. If `T = NULL`, the number of noisy gradient descent iterations is chosen as $\lceil \log n \rceil$. If `M = NULL`, the number of blocks used in the private estimator of m_2 is chosen as $\lfloor \sqrt{n}/2 \rfloor$.

Value

A list of control options 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. `dp_scree_plot()` for plotting scree estimates.

Examples

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

`pmwm_control` *Control options for private modified winsorized scree estimation*

Description

Creates a control list for the private modified winsorized mean scree estimator used by `dp_scree()` and `dp_scree_plot()` when `method = "pmwm"`.

Usage

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

Arguments

`a, b`	Finite lower and upper search bounds supplied to the private quantile routine. The private lower and upper clipping cutoffs are searched within this range. These values have no defaults because they should be chosen on the scale of squared principal component scores.
`trim_const`	Positive number controlling the baseline clipping level in the practical clipping proportion. This value has no default.
`eta`	Nonnegative number controlling the expected contamination level in the practical clipping proportion. This value has no default.
`beta`	Positive number greater than `1` defining the log-binning base used by the private quantile routine. The default is `1.001`.
`split_mode`	A logical value indicating whether to split the sample into quantile-estimation and mean-estimation subsets. The default is `TRUE`.

Details

The PMWM method privately estimates lower and upper tail cutoffs, winsorizes the squared scores to those cutoffs, and then releases a noisy winsorized mean. It is based on the private modified winsorized mean of Ramsay and Spicker (2025).

The implementation used here is an R adaptation of the publicly available Python implementation accompanying Ramsay and Spicker (2025). The adaptation is used for scree estimation by applying the PMWM estimator to squared principal component scores.

The PMWM scree estimator uses additional control parameters for private quantile estimation and winsorization. The parameter `beta` determines the spacing of the geometric search grid used by the private quantile estimator and must satisfy $\beta > 1$. Smaller values of `beta` give a finer grid but may increase computation.

The bounds `a` and `b` define the lower and upper search range supplied to the private quantile routine. The private lower and upper winsorization cutoffs are searched within this range. These bounds should be chosen on the scale of the squared principal component scores.

The parameters `trim_const` and `eta` determine the practical clipping proportion used by the modified winsorized mean. If n_q denotes the number of observations used for private quantile estimation, the clipping proportion is

$$p = \min\left\{\max\left(\frac{\text{trim\_const}}{n_q}, \eta\right), 0.49\right\}.$$

Here, `trim_const / n_q` controls the baseline clipping level, while `eta` gives a lower bound reflecting the expected contamination level.

If `split_mode = TRUE`, the sample is split into two parts: one part is used for private quantile estimation and the other part is used for the winsorized mean step. If `split_mode = FALSE`, all observations are used in both steps.

The parameters `a`, `b`, `trim_const`, and `eta` are intentionally not given defaults. They are data- and robustness-dependent choices and should be set deliberately by the user.

Value

A list of control options 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.

See Also

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

Examples

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

Index

∗ **datasets**
adult, 2
gau, 21
gau_g, 21

adult, 2

clipped_control, 3
clipped_control(), *16, 17*, *19, 20*

dp_pc_dir, 5
dp_pc_dir(), *8, 9*, *11, 12*, *14*, *16, 17*, *19, 20*
dp_score, 7
dp_score(), *11–13*
dp_score_group, 10
dp_score_group(), *9*, *13–15*
dp_score_plot, 12
dp_score_plot(), *9*, *15*
dp_score_plot_group, 13
dp_score_plot_group(), *9*, *11*, *13*
dp_scree, 15
dp_scree(), *3, 4*, *18–20*, *22–25*
dp_scree_plot, 18
dp_scree_plot(), *3, 4*, *22–25*
dppca_app, 4

gau, *21*, 21, *22*
gau_g, 21

huber_control, 22
huber_control(), *16, 17*, *19, 20*

pmwm_control, 24
pmwm_control(), *16, 17*, *19, 20*