

Package: riemannianStats (via r-universe)

July 17, 2026

Type Package

Title Riemannian Statistics for Dimensionality Reduction and Visualization

Version 0.1.1

Description Provides tools for applying statistical methods on Riemannian manifolds using local geometry derived from Uniform Manifold Approximation and Projection (UMAP). The package enables dimensionality reduction, visualization, and analysis of complex data through Riemannian versions of principal component analysis and related multivariate methods. Methods are based on McInnes et al. (2018) <[doi:10.21105/joss.00861](https://doi.org/10.21105/joss.00861)>.

License BSD_3_clause + file LICENSE

Encoding UTF-8

Language en-US

Depends R (>= 4.1)

Imports rlang, ggplot2, ggrepel, grid, uwot

Suggests scatterplot3d, plotly, testthat (>= 3.0.0), knitr, rmarkdown

Config/testthat/edition 3

VignetteBuilder knitr, rmarkdown

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Oldemar Rodríguez Rojas [aut, cre], Jennifer Lobo Vásquez [aut]

Maintainer Oldemar Rodríguez Rojas <oldemar.rodriguez@ucr.ac.cr>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-07-17 13:20:02 UTC

RemoteUrl <https://github.com/cran/riemannianStats>

RemoteRef HEAD

RemoteSha 6d07b90afef39922bb11ad550958e9820dff087c

Contents

riem.biplot	2
riem.center.data	4
riem.cor	5
riem.cov	6
riem.diff	6
riem.dist	7
riem.ind.coord	8
riem.inertia	9
riem.mean.index	10
riem.norm	11
riem.plot	11
riem.plot.3d	12
riem.rho	14
riem.similarities.umap	15
riem.var.coord	16
Index	18

riem.biplot	<i>Plot a Riemannian Biplot</i>
-------------	---------------------------------

Description

Generates a biplot showing individuals on the principal plane and variables as arrows projected onto the same component space.

Usage

```
riem.biplot(  
  data,  
  components,  
  correlations,  
  clusters = NULL,  
  explained.inertia = 0,  
  title = "",  
  var.scale = NULL,  
  point.size = 2,  
  alpha = 1,  
  show.ind.labels = TRUE,  
  show.var.labels = TRUE,  
  ind.label.size = 3,  
  var.label.size = 3,  
  arrow.size = 0.4,  
  var.color = "red",  
  interactive = FALSE  
)
```

Arguments

<code>data</code>	A data frame containing the original data. Row names are used as individual labels and column names are used as variable labels.
<code>components</code>	A matrix or data frame containing at least two components. The first two columns are used to represent individuals on the principal plane.
<code>correlations</code>	A matrix or data frame containing the correlations between variables and components. The first two columns are used to draw the variable arrows.
<code>clusters</code>	Optional vector of cluster labels used to color individuals according to the cluster they belong to.
<code>explained.inertia</code>	Numeric percentage of inertia explained by the plotted components. Defaults to '0'.
<code>title</code>	Optional character string added above the default title.
<code>var.scale</code>	Numeric scaling factor for variable arrows. Defaults to 'NULL', which automatically scales arrows to the individual plane.
<code>point.size</code>	Numeric size of the points. Defaults to '2'.
<code>alpha</code>	Numeric transparency value between 0 and 1. Defaults to '1'.
<code>show.ind.labels</code>	Logical. If 'TRUE', displays individual labels. Defaults to 'TRUE'.
<code>show.var.labels</code>	Logical. If 'TRUE', displays variable labels. Defaults to 'TRUE'.
<code>ind.label.size</code>	Numeric size of individual labels. Defaults to '3'.
<code>var.label.size</code>	Numeric size of variable labels. Defaults to '3'.
<code>arrow.size</code>	Numeric line width for variable arrows. Defaults to '0.4'.
<code>var.color</code>	Character string indicating the color of variable arrows and variable labels. Defaults to "red".
<code>interactive</code>	Logical. If 'TRUE', returns an interactive 'plotly' plot. If 'FALSE', returns a static 'ggplot2' plot. Defaults to 'FALSE'.

Details

The biplot combines two types of information: the position of the individuals on the first two Riemannian components and the contribution of the variables represented as arrows. This allows the user to explore both the structure of the observations and the relationship between variables and components.

If `interactive = TRUE`, the static 'ggplot2' object is converted into an interactive 'plotly' object. In that case, hovering over the points displays information about the individuals, while hovering over the variable arrows or labels displays information about the corresponding variables.

Value

If `interactive = FALSE`, a 'ggplot' object. If `interactive = TRUE`, a 'plotly' object.

Examples

```
# Static biplot
data <- iris[, 1:4]
components <- prcomp(data, scale. = TRUE)$x
correlations <- cor(data, components[, 1:2])
riem.biplot(
  data = data,
  components = components,
  correlations = correlations,
  show.ind.labels = FALSE
)

# Interactive biplot
riem.biplot(
  data = data,
  components = components,
  correlations = correlations,
  show.ind.labels = FALSE,
  interactive = TRUE
)
```

riem.center.data

Center Data Around the Riemannian Mean Observation

Description

Centers a dataset with respect to the Riemannian mean observation, defined as the row with the smallest total UMAP-based distance to all other rows.

Usage

```
riem.center.data(data, rho, umap.distance.matrix)
```

Arguments

data	A numeric data frame or matrix.
rho	A square numeric Rho matrix.
umap.distance.matrix	A square numeric UMAP-based distance matrix.

Value

A numeric matrix of Riemannian-centered data.

Examples

```
data <- iris[1:5, 1:4]
rho <- matrix(1, nrow = 5, ncol = 5)
dist.matrix <- as.matrix(dist(data))
centered <- riem.center.data(data, rho, dist.matrix)
head(centered)
```

riem.cor

*Calculate the Riemannian Correlation Matrix***Description**

Calculates the Riemannian correlation matrix from a numeric data frame or matrix, a Rho matrix, and a UMAP-based distance matrix.

Usage

```
riem.cor(data, rho, umap.distance.matrix)
```

Arguments

data	A numeric data frame or matrix where rows are observations and columns are variables.
rho	A numeric Rho matrix.
umap.distance.matrix	A numeric UMAP-based distance matrix.

Value

A numeric Riemannian correlation matrix.

Examples

```
data <- iris[, 1:4]
similarities <- riem.similarities.umap(data, n.neighbors = 5)
rho <- riem.rho(similarities)
riemannian.diff <- riem.diff(data = data, rho = rho)
distance.matrix <- riem.dist(riemannian.diff)

correlation.matrix <- riem.cor(
  data = data,
  rho = rho,
  umap.distance.matrix = distance.matrix
)
```

riem.cov	<i>Calculate a Riemannian Covariance Matrix</i>
----------	---

Description

Calculates a covariance matrix using Riemannian-centered differences with respect to the Riemannian mean observation.

Usage

```
riem.cov(data, rho, umap.distance.matrix, unbiased = FALSE)
```

Arguments

data	A numeric data frame or matrix.
rho	A square numeric Rho matrix.
umap.distance.matrix	A square numeric UMAP-based distance matrix.
unbiased	Logical. If 'TRUE', divides by 'n - 1'; otherwise divides by 'n'. Defaults to 'FALSE'.

Value

A numeric Riemannian covariance matrix.

Examples

```
data <- iris[1:10, 1:4]
similarities <- diag(nrow(data))
rho <- riem.rho(similarities)
diffs <- riem.diff(data, rho)
dist.matrix <- riem.dist(diffs)
riem.cov(data, rho, dist.matrix)
```

riem.diff	<i>Calculate Riemannian Vector Differences</i>
-----------	--

Description

Calculates weighted pairwise vector differences between all rows of a numeric dataset using the Rho matrix.

Usage

```
riem.diff(data, rho)
```

Arguments

data	A numeric data frame or matrix where rows are observations and columns are variables.
rho	A square numeric matrix with the same number of rows as ‘data’.

Value

A three-dimensional numeric array. Entry ‘[i, j,]’ contains ‘rho[i, j] * (data[i,] - data[j,])’.

Examples

```
data <- iris[1:5, 1:4]
rho <- matrix(1, nrow = 5, ncol = 5)
diffs <- riem.diff(data, rho)
dim(diffs)
```

riem.dist

*Calculate the UMAP-Based Distance Matrix***Description**

Calculates a pairwise distance matrix from a tensor of Riemannian vector differences.

Usage

```
riem.dist(riemannian.diff)
```

Arguments

riemannian.diff	A three-dimensional array where entry ‘[i, j,]’ contains the weighted vector difference between observations ‘i’ and ‘j’.
-----------------	--

Value

A square numeric distance matrix.

Examples

```
data <- iris[1:5, 1:4]
rho <- matrix(1, nrow = 5, ncol = 5)
diffs <- riem.diff(data, rho)
riem.dist(diffs)
```

riem.ind.coord	<i>Calculate Riemannian Principal Components</i>
----------------	--

Description

Performs Riemannian principal component analysis from a numeric data frame or matrix, a Riemannian correlation matrix, a Rho matrix, and a UMAP-based distance matrix.

Usage

```
riem.ind.coord(
  data,
  correlation.matrix,
  rho,
  umap.distance.matrix,
  sign.convention = c("largest.negative", "largest.positive", "none")
)
```

Arguments

<code>data</code>	A numeric data frame or matrix where rows are observations and columns are variables.
<code>correlation.matrix</code>	A square numeric correlation matrix.
<code>rho</code>	A numeric Rho matrix.
<code>umap.distance.matrix</code>	A numeric UMAP-based distance matrix.
<code>sign.convention</code>	Character string indicating how to orient the signs of the eigenvectors. Options are "largest.negative", "largest.positive", and "none". Defaults to "largest.negative".

Details

The sign of principal components is arbitrary. This function uses a sign convention to make component orientations reproducible.

Value

A numeric matrix of Riemannian principal components.

Examples

```
data <- iris[, 1:4]
similarities <- riem.similarities.umap(data, n.neighbors = 5)
rho <- riem.rho(similarities)
riemannian.diff <- riem.diff(data, rho = rho)
distance.matrix <- riem.dist(riemannian.diff)
```

```

correlation.matrix <- riem.cor(
  data = data,
  rho = rho,
  umap.distance.matrix = distance.matrix
)

components <- riem.ind.coord(
  data = data,
  correlation.matrix = correlation.matrix,
  rho = rho,
  umap.distance.matrix = distance.matrix
)

head(components)

```

riem.inertia

Calculate PCA Inertia for Two Components

Description

Calculates the proportion of total inertia, or explained variance, associated with two selected principal components from a correlation matrix.

Usage

```
riem.inertia(correlation.matrix, component1, component2)
```

Arguments

correlation.matrix	A square numeric correlation matrix.
component1	Integer index of the first selected component.
component2	Integer index of the second selected component.

Details

In R, component indices start at 1. Therefore, the first principal component is ‘component1 = 1’, not ‘component1 = 0’.

Value

A numeric value between 0 and 1.

Examples

```

data <- iris[, 1:4]
similarities <- riem.similarities.umap(data, n.neighbors = 5)
rho <- riem.rho(similarities)
riemannian.diff <- riem.diff(data, rho = rho)
distance.matrix <- riem.dist(riemannian.diff)

correlation.matrix <- riem.cor(
  data = data,
  rho = rho,
  umap.distance.matrix = distance.matrix
)

components <- riem.ind.coord(
  data = data,
  correlation.matrix = correlation.matrix,
  rho = rho,
  umap.distance.matrix = distance.matrix
)
riem.inertia(correlation.matrix, component1 = 1, component2 = 2)

```

riem.mean.index

Find the Riemannian Mean Observation Index

Description

Returns the index of the observation with the smallest total distance to all other observations.

Usage

```
riem.mean.index(umap.distance.matrix)
```

Arguments

```
umap.distance.matrix
```

A square numeric distance matrix.

Value

An integer index. In R, indexing starts at 1.

Examples

```

d <- as.matrix(dist(iris[1:5, 1:4]))
riem.mean.index(d)

```

riem.norm	<i>Calculate the Euclidean Norm</i>
-----------	-------------------------------------

Description

Calculates the Euclidean norm of a vector or the Euclidean distance between two vectors.

Usage

```
riem.norm(x, y = NULL)
```

Arguments

x	Numeric vector.
y	Optional numeric vector. If supplied, the norm of 'x - y' is computed.

Value

A single numeric value.

Examples

```
riem.norm(c(3, 4))
riem.norm(c(1, 2), c(4, 6))
```

riem.plot	<i>Plot Riemannian Results</i>
-----------	--------------------------------

Description

Plots either the principal plane for individuals or the correlation circle for variables.

Usage

```
riem.plot(
  data,
  choix = c("ind", "var"),
  components = NULL,
  correlations = NULL,
  clusters = NULL,
  explained.inertia = 0,
  title = "",
  ...
)
```

Arguments

<code>data</code>	A numeric data frame or matrix.
<code>choix</code>	Character. Either <code>"ind"</code> for the principal plane or <code>"var"</code> for the correlation circle.
<code>components</code>	A matrix or data frame containing the coordinates of the individuals on the components. Required when <code>'choix = "ind"'</code> .
<code>correlations</code>	A matrix or data frame containing the correlations between variables and components. Required when <code>'choix = "var"'</code> .
<code>clusters</code>	Optional vector of cluster labels used to color individuals according to the cluster they belong to. Used only when <code>'choix = "ind"'</code> .
<code>explained.inertia</code>	Numeric percentage of inertia explained by the plotted components. Defaults to <code>'0'</code> .
<code>title</code>	Optional character string added above the default title.
<code>...</code>	Additional arguments passed to the selected plotting function.

Value

A `'ggplot'` object.

Examples

```
data <- iris[, 1:4]
components <- prcomp(data, scale. = TRUE)$x
correlations <- cor(data, components[, 1:2])

riem.plot(
  data = data,
  choix = "ind",
  components = components
)

riem.plot(
  data = data,
  choix = "var",
  correlations = correlations
)
```

riem.plot.3d

Plot a Three-Dimensional Scatter Plot

Description

Creates an interactive three-dimensional scatter plot from selected columns of a data frame. Points can optionally be colored according to a cluster column.

Usage

```
riem.plot.3d(
  data,
  x.col,
  y.col,
  z.col,
  cluster.col = NULL,
  label.col = NULL,
  title = "",
  explained.inertia = 0,
  point.size = 5,
  alpha = 1
)
```

Arguments

<code>data</code>	A data frame containing the variables to plot.
<code>x.col</code>	Character string with the name of the x-axis column.
<code>y.col</code>	Character string with the name of the y-axis column.
<code>z.col</code>	Character string with the name of the z-axis column.
<code>cluster.col</code>	Optional character string with the name of the cluster column. If provided, points are colored according to the cluster they belong to. Defaults to 'NULL'.
<code>label.col</code>	Optional character string with the name of the column used as individual labels. If 'NULL', row names are used. Defaults to 'NULL'.
<code>title</code>	Optional character string added above the default title.
<code>explained.inertia</code>	Numeric percentage of inertia explained by the plotted components. Defaults to '0'.
<code>point.size</code>	Numeric size of the markers. Defaults to '5'.
<code>alpha</code>	Numeric transparency value between 0 and 1. Defaults to '1'.

Details

This function requires the optional package 'plotly'.

Value

A 'plotly' htmlwidget.

Examples

```
riem.plot.3d(
  data = iris,
  x.col = "Sepal.Length",
  y.col = "Sepal.Width",
  z.col = "Petal.Length"
)
```

```
riem.plot.3d(  
  data = iris,  
  x.col = "Sepal.Length",  
  y.col = "Sepal.Width",  
  z.col = "Petal.Length",  
  cluster.col = "Species"  
)
```

`riem.rho`*Calculate the Rho Matrix*

Description

Calculates the Rho matrix as one minus the UMAP similarity matrix.

Usage

```
riem.rho(umap.similarities)
```

Arguments

`umap.similarities`

A numeric matrix of UMAP graph similarities.

Details

The Rho matrix is used to weight pairwise vector differences in the Riemannian analysis pipeline.

Value

A numeric matrix computed as `'1 - umap.similarities'`.

Examples

```
similarities <- diag(3)  
riem.rho(similarities)
```

`riem.similarities.umap`*Calculate UMAP Graph Similarities*

Description

Calculates a dense matrix of UMAP graph similarities from the local neighborhood graph of a numeric dataset.

Usage

```
riem.similarities.umap(  
  data,  
  n.neighbors = 3,  
  min.dist = 0.1,  
  metric = "euclidean",  
  sparse = FALSE  
)
```

Arguments

<code>data</code>	A numeric data frame or matrix where rows are observations and columns are variables.
<code>n.neighbors</code>	Integer. Number of nearest neighbors used to construct the local graph. Defaults to '3'.
<code>min.dist</code>	Numeric. Kept for API compatibility with UMAP workflows. The similarity graph itself is driven by local neighborhoods and the distance metric. Defaults to '0.1'.
<code>metric</code>	Character string. Distance metric used by UMAP. Defaults to '"euclidean"'.
<code>sparse</code>	Logical. If 'TRUE', returns the sparse graph returned by 'uwot'; if 'FALSE', returns a dense matrix. Defaults to 'FALSE'.

Details

This function is the R equivalent of extracting 'reducer.graph_' from 'umap-learn' in Python and converting it to a dense matrix.

Value

A numeric matrix or sparse matrix of UMAP similarities.

Examples

```
similarities <- riem.similarities.umap(  
  iris[, 1:4],  
  n.neighbors = 5  
)
```

```
dim(similarities)
```

riem.var.coord	<i>Calculate Riemannian Correlations Between Variables and Components</i>
----------------	---

Description

Calculates Riemannian correlations between the original variables and the first two principal components.

Usage

```
riem.var.coord(data, components, rho, umap.distance.matrix)
```

Arguments

data	A numeric data frame or matrix where rows are observations and columns are variables.
components	A numeric matrix or data frame with at least two columns.
rho	A numeric Rho matrix.
umap.distance.matrix	A numeric UMAP-based distance matrix.

Value

A data frame with columns 'Component.1' and 'Component.2'.

Examples

```
data <- iris[, 1:4]
similarities <- riem.similarities.umap(data, n.neighbors = 5)
rho <- riem.rho(similarities)
riemannian.diff <- riem.diff(data, rho = rho)
distance.matrix <- riem.dist(riemannian.diff)

correlation.matrix <- riem.cor(
  data = data,
  rho = rho,
  umap.distance.matrix = distance.matrix
)

components <- riem.ind.coord(
  data = data,
  correlation.matrix = correlation.matrix,
  rho = rho,
  umap.distance.matrix = distance.matrix
)
```

```
correlations <- riem.var.coord(  
  data = data,  
  components = components,  
  rho = rho,  
  umap.distance.matrix = distance.matrix  
)
```

Index

`riem.biplot`, [2](#)
`riem.center.data`, [4](#)
`riem.cor`, [5](#)
`riem.cov`, [6](#)
`riem.diff`, [6](#)
`riem.dist`, [7](#)
`riem.ind.coord`, [8](#)
`riem.inertia`, [9](#)
`riem.mean.index`, [10](#)
`riem.norm`, [11](#)
`riem.plot`, [11](#)
`riem.plot.3d`, [12](#)
`riem.rho`, [14](#)
`riem.similarities.umap`, [15](#)
`riem.var.coord`, [16](#)