Package 'cardinalR'

April 16, 2024

Type Package

Title Collection of Data Structures

Version 0.1.1

Description A collection of simple simulation datasets designed for generating Nonlinear Dimension Reduction representations techniques such as t-distributed Stochastic Neighbor Embedding, and Uniform Manifold Approximation and Projection. These datasets serve as a valuable resource for understanding the reliability of Nonlinear Dimension Reduction representations in various contexts.

License MIT + file LICENSE

URL https://github.com/JayaniLakshika/cardinalR

BugReports https://github.com/JayaniLakshika/cardinalR/issues

Depends R (>= 3.5.0)

Imports purrr, stats

Suggests knitr, langevitour, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.3.1

NeedsCompilation no

Author Jayani P.G. Lakshika [aut, cre] (<https://orcid.org/0000-0002-6265-6481>), Dianne Cook [aut] (<https://orcid.org/0000-0002-3813-7155>), Paul Harrison [aut] (<https://orcid.org/0000-0002-3980-268X>), Michael Lydeamore [aut] (<https://orcid.org/0000-0001-6515-827X>), Thiyanga S. Talagala [aut] (<https://orcid.org/0000-0002-0656-9789>)

Maintainer Jayani P.G. Lakshika <jayanilakshika76@gmail.com>

Repository CRAN

Date/Publication 2024-04-16 09:00:06 UTC

R topics documented:

cell_cycle *Generate Cell Cycle Data with Noise*

Description

This function generates a cell cycle dataset with added noise dimensions.

Usage

```
cell_cycle(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the cell cycle data with added noise.

Examples

```
set.seed(20240412)
cell_cycle_data <- cell_cycle(
  n = 300, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

clust_diff_shapes *Generate Clusters with Different Shapes*

Description

This function generates clusters with different shapes, including both Gaussian and non-Gaussian clusters.

Usage

```
clust_diff_shapes(
  n,
  num_gau_clust,
  num_non_gau_clust,
  clust_sd_gau,
  clust_sd_non_gau,
  num_dims,
  a,
  b
)
```

Arguments

n	The total number of data points to be generated.
num_gau_clust	The number of Gaussian clusters to generate.
num_non_gau_clust	The number of non-Gaussian clusters to generate.
clust_sd_gau	The standard deviation for the Gaussian clusters.
clust_sd_non_gau	The standard deviation for the non-Gaussian clusters.
num_dims	The number of dimensions for the data points.
a	The scaling factor for the non-Gaussian cluster shape.
b	The translation factor for the non-Gaussian cluster shape.

Value

A matrix containing the generated clusters with different shapes.

Examples

```
# Generate clusters with default parameters
set.seed(20240412)
data <- clust_diff_shapes(
  n = 300, num_gau_clust = 4,
  num_non_gau_clust = 2, clust_sd_gau = 0.05, clust_sd_non_gau = 0.1,
  num_dims = 7, a = 2, b = 4
)
```

clust_diff_shapes_pts *Generate Clusters with Different Shapes and Different Number of Points*

Description

This function generates clusters with different shapes, including both Gaussian and non-Gaussian clusters, with different numbers of points in each cluster.

Usage

```
clust_diff_shapes_pts(
  clust_size_vec,
  num_gau_clust,
  num_non_gau_clust,
  clust_sd_gau,
  clust_sd_non_gau,
  num_dims,
  a,
  b
)
```

Arguments

clust_size_vec	A vector specifying the number of points for each cluster.
num_gau_clust	The number of Gaussian clusters to generate.
num_non_gau_clust	The number of non-Gaussian clusters to generate.
clust_sd_gau	The standard deviation for the Gaussian clusters.
clust_sd_non_gau	The standard deviation for the non-Gaussian clusters.
num_dims	The number of dimensions for the data points.
a	The scaling factor for the non-Gaussian cluster shape.
b	The translation factor for the non-Gaussian cluster shape.

Value

A matrix containing the generated clusters with different shapes and different numbers of points.

Examples

```
# Generate clusters with default parameters
set.seed(20240412)
data <- clust_diff_shapes_pts(
  clust_size_vec = c(50, 50, 50, 50, 100, 100),
  num_gau_clust = 4,
```

```
  num_non_gau_clust = 2, clust_sd_gau = 0.05, clust_sd_non_gau = 0.1,
  num_dims = 7, a = 2, b = 4
)
```

`conic_spiral_3d` *Generate data points along a conic spiral curve with optional noise.*

Description

This function generates data points along a conic spiral curve with optional noise.

Usage

```
conic_spiral_3d(n, num_noise, min_n, max_n)
```

Arguments

`n`	Total number of data points to generate.
`num_noise`	Number of additional noise dimensions to add to the data.
`min_n`	Minimum value for the noise added to the data.
`max_n`	Maximum value for the noise added to the data.

Value

A matrix containing the generated data points with or without added noise.

Examples

```
set.seed(20240412)
conic_spiral_3d(n = 100, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

`conic_spiral_3d_row` *Generate points on a conic spiral in 3D space.*

Description

This function generates points on a conic spiral in 3D space.

Usage

```
conic_spiral_3d_row(a, b, c, w)
```

Arguments

a	Final radius of the cone.
b	Height of the object.
c	Inner radius.
w	Number of spirals.

Value

A matrix containing the generated points on the conic spiral.

Examples

```
set.seed(20240412)
conic_spiral_3d_row(1, 2, 0.5, 3)
```

---

cube_3d *Generate a 3D cube with optional noise.*

Description

This function generates a 3D cube along with optional noise.

Usage

```
cube_3d(num_dims, num_noise, min_n, max_n)
```

Arguments

num_dims	Number of effective dimensions (default is 3 for a 3D cube).
num_noise	Number of additional noise dimensions to add to the data.
min_n	Minimum value for the noise added to the data.
max_n	Maximum value for the noise added to the data.

Value

A list containing the generated data matrix and the sample size.

Examples

```
set.seed(20240412)
cube_3d(num_dims = 3, num_noise = 2, min_n = -0.01, max_n = 0.01)
```

curvy_branch *Generate Curvy Branching Clusters with Noise*

Description

This function generates data with curvy branching clusters along with added noise.

Usage

```
curvy_branch(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate curvy branching clusters with noise with custom parameters
set.seed(20240412)
data <- curvy_branch(n = 200, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

curvy_branch_clust *Generate Curvy Branching Cluster Data*

Description

This function generates curvy branching cluster data with three clusters of different shapes.

Usage

```
curvy_branch_clust(n, clust_vec, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
clust_vec	A vector specifying the number of points for each cluster. If not provided, the n is divided equally among the clusters.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate curvy branching cluster data with custom parameters
set.seed(20240412)
data <- curvy_branch_clust(
  n = 300, clust_vec = c(100, 150, 50),
  num_noise = 2, min_n = -0.05, max_n = 0.05
)
```

---

curvy_branch_clust_bkg

Generate Curvy Branching Cluster Data with Background Noise

Description

This function generates data with four clusters, two of which follow a curvilinear pattern and the other two are distributed randomly.

Usage

```
curvy_branch_clust_bkg(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate curvy branching cluster data with background noise with custom parameters
set.seed(20240412)
data <- curvy_branch_clust_bkg(
  n = 400, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

`curvy_cycle` *Generate Curvy Cell Cycle Data with Noise*

Description

This function generates a curvy cell cycle dataset with added noise dimensions.

Usage

```
curvy_cycle(n, num_noise, min_n, max_n)
```

Arguments

`n`	The total number of samples to generate.
`num_noise`	The number of additional noise dimensions to add to the data.
`min_n`	The minimum value for the noise dimensions.
`max_n`	The maximum value for the noise dimensions.

Value

A matrix containing the curvy cell cycle data with added noise.

Examples

```
set.seed(20240412)
curvy_cell_cycle_data <- curvy_cycle(
  n = 300, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

curvy_tree *Generate Curvy Tree Data with Noise*

Description

This function generates a dataset representing a curvy tree structure, with added noise.

Usage

```
curvy_tree(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the curvy tree data with added noise.

Examples

```
set.seed(20240412)
tree_data <- curvy_tree(n = 300, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

curv_2d *Generate points on a curvilinear 2D manifold*

Description

This function generates points on a curvilinear 2D manifold based on a nonlinear equation.

Usage

```
curv_2d(n, num_noise, min_n, max_n)
```

Arguments

n	The number of points to generate.
num_noise	The number of noise dimensions to add to the generated points.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the generated points on the curvilinear 2D manifold.

Examples

```
set.seed(20240412)
curvilinear_points <- curv_2d(
  n = 100, num_noise = 2, min_n = -0.01,
  max_n = 0.01
)
```

`diff_sphere` *Generate data representing small spheres within a larger encompassing sphere with added noise.*

Description

This function generates data points representing small spheres within a larger encompassing sphere and adds noise to the data if specified.

Usage

```
diff_sphere(n, num_noise, min_n, max_n)
```

Arguments

`n`	Total number of data points to generate, should be a multiple of 13.
`num_noise`	Number of additional noise dimensions to add to the data.
`min_n`	Minimum value for the noise added to the data.
`max_n`	Maximum value for the noise added to the data.

Value

A matrix containing the generated data points with or without added noise.

Examples

```
set.seed(20240412)
diff_sphere(
  n = 390, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

dini_surface_3d *Generate points sampled from the Dini surface with optional noise.*

Description

This function generates points sampled from the Dini surface along with optional noise.

Usage

```
dini_surface_3d(n, num_noise, min_n, max_n)
```

Arguments

n	Total number of data points to generate.
num_noise	Number of additional noise dimensions to add to the data.
min_n	Minimum value for the noise added to the data.
max_n	Maximum value for the noise added to the data.

Value

A matrix containing the generated data points with or without added noise.

Examples

```
set.seed(20240412)
dini_surface_3d(n = 100, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

dini_surface_3d_row *Generate points on a Dini's surface.*

Description

This function generates points on a Dini's surface.

Usage

```
dini_surface_3d_row(a = 1, b = 1)
```

Arguments

a	Outer radius of the surface.
b	Space between loops.

Value

A matrix containing the generated points on the surface.

Examples

```
set.seed(20240412)
dini_surface_3d_row(a = 1, b = 1)
```

`eight_branch` *Generate Eight Branching Data with Noise*

Description

This function generates a dataset representing eight branching patterns, with added noise.

Usage

```
eight_branch(n, num_noise, min_n, max_n)
```

Arguments

`n`	The total number of samples to generate.
`num_noise`	The number of additional noise dimensions to add to the data.
`min_n`	The minimum value for the noise dimensions.
`max_n`	The maximum value for the noise dimensions.

Value

A matrix containing the eight branching data with added noise.

Examples

```
set.seed(20240412)
branching_data <- eight_branch(
  n = 400, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

four_branch *Generate Four-Branching Data with Noise*

Description

This function generates a dataset representing four branches with added noise.

Usage

```
four_branch(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the four-branching data with added noise.

Examples

```
set.seed(20240412)
four_branching_data <- four_branch(
  n = 400, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

four_long_clust *Generate Four Different Long Clusters with Noise*

Description

This function generates a dataset consisting of four different long clusters with added noise.

Usage

```
four_long_clust(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the four different long clusters with added noise.

Examples

```
set.seed(20240412)
four_diff_long_clusters <- four_long_clust(
  n = 200, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

---

four_long_clust_bkg *Generate Four Long Clusters with Background Noise*

Description

This function generates data with four long clusters along with background noise.

Usage

```
four_long_clust_bkg(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate four long clusters with background noise with custom parameters
set.seed(20240412)
data <- four_long_clust_bkg(
  n = 400, num_noise = 4, min_n = -0.05,
  max_n = 0.05
)
```

gau_clust *Generate synthetic data with Gaussian clusters*

Description

Generate Gaussian Clusters

Usage

```
gau_clust(
  n,
  num_clust,
  mean_matrix,
  var_vec,
  num_dims,
  num_noise,
  min_n,
  max_n
)
```

Arguments

n	The total number of data points to be generated.
num_clust	The number of clusters to generate.
mean_matrix	A matrix where each row represents the mean vector for a cluster.
var_vec	A vector specifying the variance for each cluster.
num_dims	The number of effective dimensions for the data points.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Details

This function generates Gaussian clusters with specified parameters.

Value

A matrix containing the generated Gaussian clusters.

Examples

```
set.seed(20240412)
gau_clust(
  n = 300, num_clust = 5,
  mean_matrix = rbind(
```

```
    c(1, 0, 0, 0), c(0, 1, 0, 0), c(0, 0, 1, 0),
    c(0, 0, 0, 1), c(0, 0, 0, 0)
  ), var_vec = c(0.05, 0.05, 0.05, 0.05, 0.05),
  num_dims = 4, num_noise = 2, min_n = -0.05, max_n = 0.05
)
```

`gau_clust_diff` *Generate Gaussian Clusters with Different Points*

Description

This function generates Gaussian clusters with different numbers of points per cluster.

Usage

```
gau_clust_diff(
  clust_size_vec,
  num_clust,
  mean_matrix,
  var_vec,
  num_dims,
  num_noise,
  min_n,
  max_n
)
```

Arguments

`clust_size_vec`	A vector specifying the number of points in each cluster.
`num_clust`	The number of clusters to generate.
`mean_matrix`	A matrix where each row represents the mean vector for a cluster.
`var_vec`	A vector specifying the variance for each cluster.
`num_dims`	The number of effective dimensions for the data points.
`num_noise`	The number of additional noise dimensions to be generated.
`min_n`	The minimum value for the noise added to the data points.
`max_n`	The maximum value for the noise added to the data points.

Value

A matrix containing the generated Gaussian clusters with different points.

Examples

```
# Generate Gaussian clusters with custom parameters
set.seed(20240412)
data <- gau_clust_diff(
  clust_size_vec = c(50, 100, 200, 50),
  num_clust = 4, mean_matrix =
    rbind(
      c(1, 0, 0, 0, 0, 0), c(0, 1, 0, 0, 0, 0),
      c(0, 0, 1, 0, 0, 0), c(0, 0, 0, 1, 0, 0)
    ),
  var_vec = c(0.02, 0.05, 0.06, 0.1),
  num_dims = 6, num_noise = 4,
  min_n = -0.05, max_n = 0.05
)
```

---

gau_curvy_clust *Generate Cluster and Curvilinear Data with Noise*

Description

This function generates data with two clusters, one following a curvilinear pattern and the other distributed randomly.

Usage

```
gau_curvy_clust(n, clust_size_vec, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
clust_size_vec	A vector specifying the number of points for each cluster. If not provided, the n is divided equally between the two clusters.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate cluster and curvilinear data with custom parameters
set.seed(20240412)
data <- gau_curvy_clust(
  n = 300, clust_size_vec = c(100, 200), num_noise = 3,
  min_n = -0.05, max_n = 0.05
)
```

---

gau_curvy_clust_bkg *Generate Clusters and Curvilinear Data with Noise*

Description

This function generates data with clusters and curvilinear patterns along with added background noise.

Usage

```
gau_curvy_clust_bkg(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate clusters and curvilinear data with noise with custom parameters
set.seed(20240412)
data <- gau_curvy_clust_bkg(
  n = 260, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

gen_bkg_noise *Generate Background Noise Data*

Description

This function generates background noise data with specified parameters such as the number of samples, number of dimensions, mean, and standard deviation.

Usage

```
gen_bkg_noise(n, num_dims, mean, sd)
```

Arguments

n	Number of samples to generate.
num_dims	Number of dimensions (columns) of the data.
mean	Mean of the normal distribution used to generate noise (default is 0).
sd	Standard deviation of the normal distribution used to generate noise (default is 1).

Value

A matrix containing the generated background noise data, with n rows and num_dims columns.

Examples

```
# Generate background noise with custom mean and standard deviation
set.seed(20240412)
gen_bkg_noise(n = 50, num_dims = 3, mean = 5, sd = 2)
```

gen_noise_dims *Generate Random Noise Dimensions*

Description

This function generates random noise dimensions to be added to the coordinates of a sphere.

Usage

```
gen_noise_dims(n, num_noise, min_n, max_n)
```

Arguments

n	The number of observations for which to generate noise dimensions.
num_noise	The number of noise dimensions to generate.
min_n	The minimum value for the random noise.
max_n	The maximum value for the random noise.

Value

A matrix containing the generated random noise dimensions.

Examples

```
# Generate random noise dimensions with 3 dimensions, minimum value -1, and maximum value 1
set.seed(20240412)
gen_noise_dims(n = 50, num_noise = 3, min_n = -0.01, max_n = 0.01)
```

---

mirror_scurves *Generate Mirror S-curve Datasets with Noise*

Description

This function generates mirror S-curve datasets with added noise dimensions.

Usage

```
mirror_scurves(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate (should be divisible by 2).
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the combined mirror S-curve datasets with added noise.

Examples

```
set.seed(20240412)
mirror_s_curve_data <- mirror_scurves(
  n = 200, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

mobius_5d *Generate a 5-D Mobius Strip*

Description

This function generates a dataset representing a 5-dimensional Mobius strip.

Usage

```
mobius_5d(n, num_noise, min_n, max_n)
```

Arguments

n	The number of points to generate for the Mobius strip.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the generated Mobius strip.

Examples

```
set.seed(20240412)
mobius_data <- mobius_5d(n = 100, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

mobius_5d_row *Generate a Single Row for a 5-D Mobius Strip*

Description

This function generates a single row of data representing a point on a 5-dimensional Mobius strip.

Usage

```
mobius_5d_row()
```

Value

A vector containing the coordinates of the point on the Mobius strip.

Examples

```
set.seed(20240412)
mobius_row <- mobius_5d_row()
```

mobius_clust *Generate Mobius Cluster with Noise*

Description

This function generates a dataset consisting of a mobius cluster with added noise.

Usage

```
mobius_clust(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the mobius cluster with added noise.

Examples

```
mobius_cluster <- mobius_clust(
  n = 200, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

mobius_clust_data *Mobius clust dataset with noise dimensions*

Description

The ‘mobius_clust_data‘ dataset contains a 3-dimensional Mobius and Gaussian cluster with added noise dimensions. Each data point is represented by five dimensions (x1 to x5).

Usage

```
data(mobius_clust_data)
```

Format

A data frame with 500 rows and 5 columns:

x1, x2, x3, x4, x5 High-dimensional coordinates

Source

This dataset is generated for illustrative purposes.

Examples

```
# Load the mobius_clust_data dataset
data(mobius_clust_data)

# Display the first few rows of the dataset
head(mobius_clust_data)
```

`mobius_clust_tsne_param1` *tSNE embedding for mobius_clust_data dataset which with noise dimensions tSNE parameters set to perplexity: 15.*

Description

The 'mobius_clust_tsne_param1' dataset contains the tSNE (t-distributed Stochastic Neighbor Embedding) embeddings of a five-dimensional mobius_clust_data. Each data point is represented by two tSNE coordinates (emb1 and emb2).

Usage

```
data(mobius_clust_tsne_param1)
```

Format

'mobius_clust_tsne_param1' A data frame with 500 rows and 2 columns:

emb1 Numeric, first tSNE 2D embeddings.

emb2 Numeric, second tSNE 2D embeddings.

Source

This dataset is generated for illustrative purposes.

Examples

```
# Load the mobius_clust_tsne_param1 dataset
data(mobius_clust_tsne_param1)

# Display the first few rows of the dataset
head(mobius_clust_tsne_param1)
```

mobius_clust_tsne_param2 *tSNE embedding for mobius_clust_data dataset which with noise dimensions tSNE parameters set to perplexity: 30.*

Description

The 'mobius_clust_tsne_param2' dataset contains the tSNE (t-distributed Stochastic Neighbor Embedding) embeddings of a five-dimensional mobius_clust_data. Each data point is represented by two tSNE coordinates (emb1 and emb2).

Usage

```
data(mobius_clust_tsne_param2)
```

Format

'mobius_clust_tsne_param2' A data frame with 500 rows and 2 columns:

emb1 Numeric, first tSNE 2D embeddings.

emb2 Numeric, second tSNE 2D embeddings.

Source

This dataset is generated for illustrative purposes.

Examples

```
# Load the mobius_clust_tsne_param2 dataset
data(mobius_clust_tsne_param2)

# Display the first few rows of the dataset
head(mobius_clust_tsne_param2)
```

mobius_clust_tsne_param3 *tSNE embedding for mobius_clust_data dataset which with noise dimensions tSNE parameters set to perplexity: 5.*

Description

The 'mobius_clust_tsne_param3' dataset contains the tSNE (t-distributed Stochastic Neighbor Embedding) embeddings of a five-dimensional mobius_clust_data. Each data point is represented by two tSNE coordinates (emb1 and emb2).

Usage

```
data(mobius_clust_tsne_param3)
```

Format

'mobius_clust_tsne_param3' A data frame with 500 rows and 2 columns:

emb1 Numeric, first tSNE 2D embeddings.

emb2 Numeric, second tSNE 2D embeddings.

Source

This dataset is generated for illustrative purposes.

Examples

```
# Load the mobius_clust_tsne_param1 dataset
data(mobius_clust_tsne_param3)

# Display the first few rows of the dataset
head(mobius_clust_tsne_param3)
```

---

mobius_clust_umap_param1

UMAP embedding for mobius_clust_data dataset which with noise dimensions UMAP parameters set to n-neigbors: 15 and min-dist: 0.1.

Description

The 'mobius_clust_umap_param1' dataset contains the UMAP (Uniform Manifold Approximation and Projection) embeddings of a five-dimensional mobius_clust_data. Each data point is represented by two UMAP coordinates (emb1 and emb2).

Usage

```
data(mobius_clust_umap_param1)
```

Format

'mobius_clust_umap_param1' A data frame with 500 rows and 2 columns:

emb1 Numeric, first UMAP 2D embeddings.

emb2 Numeric, second UMAP 2D embeddings.

Source

This dataset is generated for illustrative purposes.

Examples

```
# Load the mobius_clust_umap_param1 dataset
data(mobius_clust_umap_param1)

# Display the first few rows of the dataset
head(mobius_clust_umap_param1)
```

`mobius_clust_umap_param2`

UMAP embedding for mobius_clust_data dataset which with noise dimensions UMAP parameters set to n-neigbors: 30 and min-dist: 0.08.

Description

The 'mobius_clust_umap_param2' dataset contains the UMAP (Uniform Manifold Approximation and Projection) embeddings of a five-dimensional mobius_clust_data. Each data point is represented by two UMAP coordinates (emb1 and emb2).

Usage

```
data(mobius_clust_umap_param2)
```

Format

'mobius_clust_umap_param2' A data frame with 500 rows and 2 columns:

emb1 Numeric, first UMAP 2D embeddings.

emb2 Numeric, second UMAP 2D embeddings.

Source

This dataset is generated for illustrative purposes.

Examples

```
# Load the mobius_clust_umap_param2 dataset
data(mobius_clust_umap_param2)

# Display the first few rows of the dataset
head(mobius_clust_umap_param2)
```

mobius_clust_umap_param3 *UMAP embedding for mobius_clust_data dataset which with noise dimensions UMAP parameters set to n-neigbors: 5 and min-dist: 0.9.*

Description

The 'mobius_clust_umap_param3' dataset contains the UMAP (Uniform Manifold Approximation and Projection) embeddings of a five-dimensional mobius_clust_data. Each data point is represented by two UMAP coordinates (emb1 and emb2).

Usage

```
data(mobius_clust_umap_param3)
```

Format

'mobius_clust_umap_param3' A data frame with 500 rows and 2 columns:

emb1 Numeric, first UMAP 2D embeddings.

emb2 Numeric, second UMAP 2D embeddings.

Source

This dataset is generated for illustrative purposes.

Examples

```
# Load the mobius_clust_umap_param3 dataset
data(mobius_clust_umap_param3)

# Display the first few rows of the dataset
head(mobius_clust_umap_param3)
```

nonlinear_2d *Generate points on a nonlinear 2D manifold*

Description

This function generates points on a nonlinear 2D manifold based on a given equation.

Usage

```
nonlinear_2d(n, num_noise, min_n, max_n)
```

Arguments

n	The number of points to generate.
num_noise	The number of noise dimensions to add to the generated points.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the generated points on the nonlinear 2D manifold.

Examples

```
set.seed(20240412)
nonlinear_points <- nonlinear_2d(
  n = 100, num_noise = 2, min_n = -0.01,
  max_n = 0.01
)
```

---

nonlinear_connect *Generate Nonlinear Connected Data with Noise*

Description

This function generates a dataset representing nonlinear connected clusters with added noise.

Usage

```
nonlinear_connect(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the nonlinear connected data with noise.

Examples

```
set.seed(20240412)
nonlinear_connect <- nonlinear_connect(
  n = 400, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

nonlinear_mirror *Generate Nonlinear Mirror Data with Noise*

Description

This function generates a dataset representing two mirror-image clusters with added noise.

Usage

```
nonlinear_mirror(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the nonlinear mirror data with noise.

Examples

```
set.seed(20240412)
nonlinear_mirror <- nonlinear_mirror(
  n = 400, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

one_doublet *Generate Doublets with Noise*

Description

This function generates data with one set of doublets (pairs of clusters) along with added background noise.

Usage

```
one_doublet(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate doublets with noise with custom parameters
set.seed(20240412)
data <- one_doublet(n = 220, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

---

one_doublet_bkg *Generate Doublets with Background Noise*

Description

This function generates data with doublets (pairs of clusters) along with added background noise.

Usage

```
one_doublet_bkg(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate doublets with background noise with custom parameters
set.seed(20240412)
data <- one_doublet_bkg(n = 250, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

one_doublet_diff_patterns

Generate Doublets with Different Pattern Clusters and Noise

Description

This function generates data with one set of doublets (pairs of clusters) having different patterns, along with added background noise.

Usage

```
one_doublet_diff_patterns(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate doublets with different pattern clusters and noise with custom parameters
set.seed(20240412)
data <- one_doublet_diff_patterns(
  n = 280, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

one_doublet_diff_var_clust

Generate Doublets with Different Variance Clusters and Noise

Description

This function generates data with one set of doublets (pairs of clusters) having clusters with different variance, along with added background noise.

Usage

```
one_doublet_diff_var_clust(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate doublets with different variance clusters and noise with custom parameters
set.seed(20240412)
data <- one_doublet_diff_var_clust(
  n = 260, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

---

one_doublet_four_clusts

Generate Doublets with Four Clusters and Noise

Description

This function generates data with one set of doublets (pairs of clusters) containing four clusters, along with added background noise.

Usage

```
one_doublet_four_clusts(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate doublets with four clusters and noise with custom parameters
set.seed(20240412)
data <- one_doublet_four_clusts(
  n = 440, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

`one_grid` *Generate Grid Data with Noise*

Description

This function generates a grid dataset with specified grid points along the x and y axes, and optionally adds noise dimensions.

Usage

```
one_grid(nx, ny, num_noise, min_n, max_n)
```

Arguments

`nx`	The number of grid points along the x axis.
`ny`	The number of grid points along the y axis.
`num_noise`	The number of additional noise dimensions to add to the data.
`min_n`	The minimum value for the noise dimensions.
`max_n`	The maximum value for the noise dimensions.

Value

A matrix containing the grid data with added noise.

Examples

```
set.seed(20240412)
one_grid <- one_grid(nx = 10, ny = 10, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

one_grid_bkg *Generate One Grid with Different Values and Background Noise*

Description

This function generates a grid dataset with different values and background noise.

Usage

```
one_grid_bkg(n_value, num_noise, min_n, max_n)
```

Arguments

n_value	The number of grid points along each axis for the grids.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A list containing the one grid datasets with background noise and the sample size.

Examples

```
set.seed(20240412)
one_grid_bkg <- one_grid_bkg(
  n_value = 10, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

plane *Generate points on a plane in 2D space*

Description

This function generates points on a plane in 3D space based on the provided coefficients, intercepts, and ranges for the parameters.

Usage

```
plane(
  n,
  coef_x1,
  coef_x2,
  coef_y1,
  coef_y2,
  intercept_x,
  intercept_y,
  u_min,
  u_max,
  v_min,
  v_max,
  num_noise,
  min_n,
  max_n
)
```

Arguments

n	The number of points to generate.
coef_x1	The coefficient of the first parameter in the x-dimension equation.
coef_x2	The coefficient of the second parameter in the x-dimension equation.
coef_y1	The coefficient of the first parameter in the y-dimension equation.
coef_y2	The coefficient of the second parameter in the y-dimension equation.
intercept_x	The intercept for the x-dimension equation.
intercept_y	The intercept for the y-dimension equation.
u_min	The minimum value for the first parameter (u) range.
u_max	The maximum value for the first parameter (u) range.
v_min	The minimum value for the second parameter (v) range.
v_max	The maximum value for the second parameter (v) range.
num_noise	The number of noise dimensions to add to the generated points.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the generated points on the plane.

Examples

```
set.seed(20240412)
plane_points <- plane(
  n = 100, coef_x1 = 1, coef_x2 = 1,
  coef_y1 = -1, coef_y2 = 1, intercept_x = -10,
```

```
  intercept_y = 8, u_min = 10, u_max = 30, v_min = 10, v_max = 20,
  num_noise = 2, min_n = -0.05, max_n = 0.05
)
```

---

plane_2d_hole *Generate 2D Plane with Hole and Noise*

Description

This function generates a dataset representing a 2D plane with a hole in the middle, with added noise.

Usage

```
plane_2d_hole(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A list containing the 2D plane data with a hole and the sample size.

Examples

```
set.seed(20240412)
plane_data <- plane_2d_hole(
  n = 100, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

roman_surface_3d *Generate data points on a Roman surface with optional noise.*

Description

This function generates data points on a Roman surface with optional noise.

Usage

```
roman_surface_3d(n, num_noise, min_n, max_n)
```

Arguments

n	Total number of data points to generate.
num_noise	Number of additional noise dimensions to add to the data.
min_n	Minimum value for the noise added to the data.
max_n	Maximum value for the noise added to the data.

Value

A matrix containing the generated data points with or without added noise.

Examples

```
set.seed(20240412)
roman_surface_3d(n = 100, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

roman_surface_3d_row *Generate points on a Roman surface in 3D space.*

Description

This function generates points on a Roman surface in 3D space.

Usage

```
roman_surface_3d_row(a = 1)
```

Arguments

a	Maximum radius of the object.

Value

A matrix containing the generated points on the Roman surface in 3D space.

Examples

```
set.seed(20240412)
roman_surface_3d_row(a = 1)
```

`scurve` *Generate S-curve Data*

Description

This function generates S-curve data, which is a commonly used dataset for testing and visualizing dimensionality reduction algorithms.

Usage

```
scurve(n, num_noise, min_n, max_n)
```

Arguments

`n`	The number of samples to generate.
`num_noise`	The number of additional noise dimensions to add to the data.
`min_n`	The minimum value for the noise dimensions.
`max_n`	The maximum value for the noise dimensions.

Value

A matrix containing the generated S-curve data.

Examples

```
set.seed(20240412)
s_curve_data <- scurve(
  n = 100, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

scurve_hole *Generate S-curve Data with a Hole*

Description

This function generates S-curve data with a hole by filtering out samples that are not close to a specified anchor point.

Usage

```
scurve_hole(n, num_noise, min_n, max_n)
```

Arguments

n	The number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the generated S-curve data with a hole.

Examples

```
set.seed(20240412)
s_curve_hole_data <- scurve_hole(
  n = 100, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

seven_branch *Generate Seven-Branching Data with Noise*

Description

This function generates a dataset representing seven branches with added noise.

Usage

```
seven_branch(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the seven-branching data with added noise.

Examples

```
set.seed(20240412)
seven_branching_data <- seven_branch(
  n = 210, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

---

sine_curve *Generate Sine Curve Data with Noise*

Description

This function generates a dataset representing a sine curve with added noise.

Usage

```
sine_curve(n, num_noise, min_n, max_n)
```

Arguments

n	The number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the sine curve data with noise.

Examples

```
set.seed(20240412)
sine_curve <- sine_curve(n = 100, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

sphere *Generate Coordinates for a Sphere*

Description

This function generates the coordinates for a sphere in three-dimensional space.

Usage

```
sphere(radius, resolution, num_noise, min_n, max_n)
```

Arguments

radius	The radius of the sphere.
resolution	The number of points used to approximate the surface of the sphere.
num_noise	The number of additional noise dimensions to add to the coordinates.
min_n	The minimum value for the random noise added to the coordinates.
max_n	The maximum value for the random noise added to the coordinates.

Value

A matrix containing the Cartesian coordinates of the points on the sphere.

Examples

```
# Generate coordinates for a sphere with radius 1 and resolution 20
set.seed(20240412)
sphere(
  radius = 1, resolution = 20, num_noise = 3, min_n = -0.05,
  max_n = 0.05
)
```

spiral_3d *Generate a spiral dataset with optional noise.*

Description

This function generates a dataset arranged in a spiral pattern with optional noise.

Usage

```
spiral_3d(n, num_dims, num_noise, min_n, max_n)
```

Arguments

n	Total number of data points to generate.
num_dims	Number of effective dimensions for each data point.
num_noise	Number of additional noise dimensions to add to the data.
min_n	Minimum value for the noise added to the data.
max_n	Maximum value for the noise added to the data.

Value

A matrix containing the generated data points with or without added noise.

Examples

```
set.seed(20240412)
spiral_3d(n = 100, num_dims = 10, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

---

swiss_roll *Generate Swiss Roll Data*

Description

This function generates data points in the shape of a Swiss roll.

Usage

```
swiss_roll(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated Swiss roll data points.

Examples

```
# Generate Swiss roll data with noise with custom parameters
set.seed(20240412)
data <- swiss_roll(n = 200, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

three_circulars *Generate Three Circular Clusters with Noise*

Description

This function generates three circular clusters in 4D space with added noise dimensions.

Usage

```
three_circulars(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the three circular clusters with added noise.

Examples

```
set.seed(20240412)
circular_clusters_data <- three_circulars(
  n = 300, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

three_clust_diff_dist *Generate three clusters of data points with optional noise.*

Description

This function generates three clusters of data points along with optional noise.

Usage

```
three_clust_diff_dist(n, num_dims, num_noise, min_n, max_n)
```

Arguments

n	Total number of data points to generate, should be a multiple of three.
num_dims	Number of dimensions for each data point.
num_noise	Number of additional noise dimensions to add to the data.
min_n	Minimum value for the noise added to the data.
max_n	Maximum value for the noise added to the data.

Value

A matrix containing the generated data points with or without added noise.

Examples

```
set.seed(20240412)
three_clust_diff_dist(
  n = 150, num_dims = 7, num_noise = 4, min_n = -0.05,
  max_n = 0.05
)
```

---

three_clust_mirror *Generate Three Cluster Mirror with Noise*

Description

This function generates data with three clusters forming a mirror image, along with added noise.

Usage

```
three_clust_mirror(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate three cluster mirror with noise with custom parameters
set.seed(20240412)
data <- three_clust_mirror(n = 300, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

three_diff_linear *Generate Three Different Linear Data with Noise*

Description

This function generates a dataset consisting of three different linear patterns with added noise.

Usage

```
three_diff_linear(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the three different linear data with added noise.

Examples

```
set.seed(20240412)
three_diff_linear <- three_diff_linear(
  n = 150, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

three_doublets *Generate Doublets with Three Clusters and Noise*

Description

This function generates data with three sets of doublets (pairs of clusters) along with added background noise.

Usage

```
three_doublets(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate doublets with three clusters and noise with custom parameters
set.seed(20240412)
data <- three_doublets(
  n = 420, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

---

three_grid *Generate Three Grids with Noise*

Description

This function generates three grid datasets with noise dimensions.

Usage

```
three_grid(n_value, num_noise, min_n, max_n)
```

Arguments

n_value	The number of grid points along the x and y axes for each grid.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A list containing three grid datasets with added noise and the sample size of each dataset.

Examples

```
set.seed(20240412)
three_grids <- three_grid(
  n_value = 19, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

three_long_clust *Generate Three Linear Clusters with Noise*

Description

This function generates data with three linear clusters, along with added noise.

Usage

```
three_long_clust(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate three linear clusters with noise with custom parameters
set.seed(20240412)
data <- three_long_clust(n = 300, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

three_nonlinear *Generate Three Nonlinear Clusters with Noise*

Description

This function generates data with three nonlinear clusters, along with added noise.

Usage

```
three_nonlinear(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate three nonlinear clusters with noise with custom parameters
set.seed(20240412)
data <- three_nonlinear(n = 300, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

---

torus_3d *Generate a torus-shaped dataset with optional noise.*

Description

This function generates a torus-shaped dataset along with optional noise.

Usage

```
torus_3d(n, num_noise, min_n, max_n)
```

Arguments

n	Total number of data points to generate.
num_noise	Number of additional noise dimensions to add to the data.
min_n	Minimum value for the noise added to the data.
max_n	Maximum value for the noise added to the data.

Value

A matrix containing the generated torus-shaped data points with or without added noise.

Examples

```
set.seed(20240412)
torus_3d(n = 100, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

torus_3d_row *Generate a row of data points for a 3D torus.*

Description

This function generates a row of data points for a 3D torus with given radii.

Usage

```
torus_3d_row(radius)
```

Arguments

radius	A numeric vector containing the radii of the torus, from largest to smallest.

Value

A vector representing a row of data points for the 3D torus.

Examples

```
set.seed(20240412)
torus_3d_row(c(2, 1))
```

tree *Generate Tree-like Data with Noise*

Description

This function generates a dataset representing a tree-like structure, with added noise.

Usage

```
tree(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the tree-like data with added noise.

Examples

```
set.seed(20240412)
tree_data <- tree(n = 300, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

`tri_3d` *Generate Triangular 3D Datasets with Noise*

Description

This function generates triangular 3D datasets with added noise dimensions.

Usage

```
tri_3d(n, num_noise, min_n, max_n)
```

Arguments

`n`	The total number of samples to generate.
`num_noise`	The number of additional noise dimensions to add to the data.
`min_n`	The minimum value for the noise dimensions.
`max_n`	The maximum value for the noise dimensions.

Value

A matrix containing the triangular 3D datasets with added noise.

Examples

```
set.seed(20240412)
triangular_3d_data <- tri_3d(
  n = 100, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

tri_plane_bkg *Generate Triangular Plane with Background Noise*

Description

This function generates a triangular plane dataset with background noise dimensions.

Usage

```
tri_plane_bkg(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the triangular plane dataset with background noise.

Examples

```
set.seed(20240412)
triangular_plane_data <- tri_plane_bkg(
  n = 216,
  num_noise = 2, min_n = -0.05, max_n = 0.05
)
```

two_circulars *Generate Linked Data*

Description

This function generates linked data points.

Usage

```
two_circulars(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated. Should be a product of two.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated linked data points.

Examples

```
# Generate linked data with noise with custom parameters
set.seed(20240412)
data <- two_circulars(n = 200, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

`two_curvilinear` *Generate Two Curvilinear Data with Noise*

Description

This function generates a dataset representing two curvilinear clusters with added noise.

Usage

```
two_curvilinear(n, num_noise, min_n, max_n)
```

Arguments

`n`	The total number of samples to generate.
`num_noise`	The number of additional noise dimensions to add to the data.
`min_n`	The minimum value for the noise dimensions.
`max_n`	The maximum value for the noise dimensions.

Value

A matrix containing the two curvilinear data with noise.

Examples

```
set.seed(20240412)
two_curvilinear <- two_curvilinear(
  n = 250, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

two_curvy *Generate Two Curvilinear Clusters with Noise*

Description

This function generates data with two curvilinear clusters along with added noise.

Usage

```
two_curvy(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate two curvilinear clusters with noise with custom parameters
set.seed(20240412)
data <- two_curvy(n = 200, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

two_curvy_diff_pts *Generate Two Curvilinear Differentiated Clusters with Noise*

Description

This function generates data with two curvilinear clusters that are differentiated from each other, along with added noise.

Usage

```
two_curvy_diff_pts(cluster_size_vec, num_noise, min_n, max_n)
```

Arguments

cluster_size_vec	A vector specifying the number of points in each cluster.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate two curvilinear differentiated clusters with noise with custom parameters
set.seed(20240412)
data <- two_curvy_diff_pts(
  cluster_size_vec = c(50, 100), num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

---

two_curvy_panckakes *Generate Two Curvy Pancakes with Noise*

Description

This function generates a dataset representing two curvy pancake-shaped clusters with added noise.

Usage

```
two_curvy_panckakes(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the two curvy pancakes data with noise.

Examples

```
set.seed(20240412)
two_curvy_panckakes <- two_curvy_panckakes(
  n = 300, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

---

two_doublets_bkg *Generate Two Doublets with Background Noise*

Description

This function generates data with two doublets along with added background noise.

Usage

```
two_doublets_bkg(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate two doublets with background noise with custom parameters
set.seed(20240412)
data <- two_doublets_bkg(n = 200, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

two_doublets_parallel *Generate Doublets in Parallel with Noise*

Description

This function generates data with two sets of doublets (pairs of clusters) running in parallel, along with added background noise.

Usage

```
two_doublets_parallel(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate doublets in parallel with noise with custom parameters
set.seed(20240412)
data <- two_doublets_parallel(n = 440, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

two_grid *Generate Two Grids with Noise*

Description

This function generates two grid datasets with noise dimensions.

Usage

```
two_grid(n_value, num_noise, min_n, max_n)
```

Arguments

n_value	The number of grid points along the x and y axes for each grid.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A list containing two grid datasets with added noise and the sample size of each dataset.

Examples

```
set.seed(20240412)
two_grids <- two_grid(n_value = 19, num_noise = 2, min_n = -0.05, max_n = 0.05)
```

---

two_grid_comb *Generate One Grid with Different Offset*

Description

This function generates a single grid dataset with a different offset.

Usage

```
two_grid_comb(n_value, num_noise, min_n, max_n)
```

Arguments

n_value	The number of grid points along each axis for the grids.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A list containing the grid dataset with different offsets and the sample size.

Examples

```
set.seed(20240412)
two_grid_comb <- two_grid_comb(
  n_value = 10, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

two_grid_comb_bkg *Generate Two Grids with Background Noise*

Description

This function generates two grid datasets with background noise.

Usage

```
two_grid_comb_bkg(n_value, num_noise, min_n, max_n)
```

Arguments

n_value	The number of grid points along each axis for the grids.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A list containing the two grid datasets with background noise and the sample size.

Examples

```
set.seed(20240412)
two_grid_comb_bkg <- two_grid_comb_bkg(
  n_value = 10, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

two_long_clust *Generate Long Cluster Data*

Description

This function generates a dataset consisting of two long clusters with added noise.

Usage

```
two_long_clust(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate.
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the long cluster data with added noise.

Examples

```
set.seed(20240412)
long_cluster <- two_long_clust(
  n = 200, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

---

two_long_clust_diff *Generate Two Linear Differentiated Clusters with Noise*

Description

This function generates data with two linear clusters that are differentiated from each other, along with added noise.

Usage

```
two_long_clust_diff(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate two linear differentiated clusters with noise with custom parameters
set.seed(20240412)
data <- two_long_clust_diff(
  n = 300, num_noise = 2, min_n = -0.05,
  max_n = 0.05
)
```

two_nonlinear *Generate Two Nonlinear Clusters with Noise*

Description

This function generates data with two nonlinear clusters along with added noise.

Usage

```
two_nonlinear(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the generated data, with each row representing a data point.

Examples

```
# Generate two nonlinear clusters with noise with custom parameters
set.seed(20240412)
data <- two_nonlinear(n = 200, num_noise = 2, min_n = -0.05, max_n = 0.50)
```

two_scurves *Generate Two S-curve Datasets with Noise*

Description

This function generates two S-curve datasets with added noise dimensions.

Usage

```
two_scurves(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of samples to generate (should be divisible by 2).
num_noise	The number of additional noise dimensions to add to the data.
min_n	The minimum value for the noise dimensions.
max_n	The maximum value for the noise dimensions.

Value

A matrix containing the combined S-curve datasets with added noise.

Examples

```
set.seed(20240412)
two_s_curve_data <- two_scurves(
  n = 200, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

---

two_scurve_hole *Generate Two S-Curve Data with Noise*

Description

This function generates two S-curve data with noise.

Usage

```
two_scurve_hole(n, num_noise, min_n, max_n)
```

Arguments

n	The total number of data points to be generated.
num_noise	The number of additional noise dimensions to be generated.
min_n	The minimum value for the noise added to the data points.
max_n	The maximum value for the noise added to the data points.

Value

A matrix containing the two S-curve datasets with added noise.

Examples

```
# Generate two S-curve data with noise with custom parameters
set.seed(20240412)
data <- two_scurve_hole(
  n = 200, num_noise = 2,
  min_n = -0.05, max_n = 0.05
)
```

Index