

Package: cassowaryr (via r-universe)

October 18, 2026

Title Compute Scagnostics on Pairs of Numeric Variables in a Data Set

Version 2.0.21

Description Computes a range of scatterplot diagnostics (scagnostics) on pairs of numerical variables in a data set. A range of scagnostics, including graph and association-based scagnostics described by Leland Wilkinson and Graham Wills (2008) <[doi:10.1198/106186008X320465](https://doi.org/10.1198/106186008X320465)> and association-based scagnostics described by Katrin Grimm (2016, ISBN:978-3-8439-3092-5) can be computed. Summary and plotting functions are provided.

License GPL-3

Encoding UTF-8

LazyData true

URL <https://github.com/numbats/cassowaryr>

BugReports <https://github.com/numbats/cassowaryr/issues>

Depends R (>= 4.0.0)

Imports igraph, alphahull (>= 2.5), splancs, interp (>= 1.1-6), energy, dplyr, ggplot2, progress, tibble, stats, tidyselect, hexbin

Suggests rmarkdown, knitr, mgcv, GGally, tidyr, testthat (>= 3.0.0), covr

VignetteBuilder knitr

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.0.0

Config/pak/sysreqs libglpk-dev libgsl0-dev libxml2-dev

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

Date/Publication 2026-09-17 22:39:27 UTC

RemoteUrl <https://github.com/numbats/cassowaryr>

RemoteRef HEAD

RemoteSha 590e4342576c128fc1155ac022574d616d71559b

Contents

anscombe_tidy	2
calc_scags	3
calc_scags_wide	4
datasaurus_dozen	5
diagnose_outliers	6
draw_functions	7
features	8
numbat	9
pk	9
sc_clumpy	10
sc_clumpy_r	11
sc_clumpy2	12
sc_convex	13
sc_dcor	14
sc_grid	15
sc_monotonic	16
sc_outlying	17
sc_skewed	18
sc_skinny	19
sc_sparse	20
sc_sparse2	21
sc_splines	22
sc_striated	23
sc_stringy05	24
sc_stringy06	25
sc_stripped	26
scree	27
top_functions	28
Index	30

anscombe_tidy

Data from Anscombe's famous example in tidy format

Description

All variables and pairs of variables have same summary statistics but are very different data, as can be seen by visualisation.

Format

A tibble with 44 observations and 3 variables

set label of the data set, each set has 11 observations

x variable for horizontal axis

y variable for vertical axis

Description

This function allows you to calculate a large number of scagnostics quickly and efficiently. While the individual scagnostic calculation functions (`sc_`) are good for looking at a single scagnostic, it is inefficient when computing more than one scagnostic. This is because the `sc_` functions recompute the graph object for each plot and scagnostic pair, even if the graph objects are unchanged. Additionally, typing the functions over and over again quickly becomes tedious. For this reason, we have the `calc_scags` function that will reuse the same graph object for all scagnostics.

Usage

```
calc_scags(
  x,
  y,
  scags = c("outlying", "stringy", "striated", "clumpy", "sparse", "skewed", "convex",
    "skinny", "monotonic"),
  out.rm = TRUE,
  binner = "hex",
  alpha = "rahman"
)
```

Arguments

<code>x, y</code>	numeric vectors
<code>scags</code>	collection of strings matching names of scagnostics to calculate: outlying, stringy, striated, grid, striped, clumpy, clumpy2, sparse, skewed, convex, skinny, monotonic, splines, dcor
<code>out.rm</code>	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
<code>binner</code>	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner
<code>alpha</code>	character, numeric, or function. Controls the alpha radius. Valid character values are: • "rahman" (default): Rahman's MST-based middle-50% alpha • "q90": 90th percentile of MST edge lengths • "omega": graph-theoretic scagnostics alpha Alternatively: • a numeric value giving a fixed alpha • a function with no arguments that returns a single numeric alpha

Value

A data frame with all selected scagnostic values for a particular x, y pair.

See Also

calc_scags_wide

Examples

```
# Calculate selected scagnostics on a single pair
calc_scags(anscombe$x1, anscombe$y1, scags=c("monotonic", "outlying"))

# Compute on long form data, or subsets
# defined by a categorical variable
require(dplyr)
datasaurus_dozen |>
  group_by(dataset) |>
  summarise(calc_scags(x,y, scags=c("monotonic", "outlying", "convex")))
```

calc_scags_wide	<i>Compute scagnostics on all possible scatter plots for the given data</i>
-----------------	---

Description

It is quite common to have data in a wide format that is not suitable to feed into the calc_scags function that would need a long format. To save users time and energy we also provide a wide version of the calc_scags function. This function will compute all selected scagnostics for every pair wise set of variables in the data frame.

Usage

```
calc_scags_wide(
  all_data,
  scags = c("outlying", "stringy", "striated", "clumpy", "sparse", "skewed", "convex",
    "skinny", "monotonic"),
  out.rm = TRUE,
  binner = "hex",
  alpha = "rahman"
)
```

Arguments

all_data	tibble of wide multivariate data
scags	collection of strings matching names of scagnostics to calculate: outlying, stringy, striated, grid, striped, clumpy, clumpy2, sparse, skewed, convex, skinny, monotonic, splines, dcor

out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner
alpha	character, numeric, or function. Controls the alpha radius. Valid character values are: • "rahman" (default): Rahman's MST-based middle-50% alpha • "q90": 90th percentile of MST edge lengths • "omega": graph-theoretic scagnostics alpha Alternatively: • a numeric value giving a fixed alpha • a function with no arguments that returns a single numeric alpha

Value

A data frame that gives the scagnostic scores for every possible pair of variables.

See Also

calc_scags

Examples

```
# Calculate selected scagnostics
data(pk)
calc_scags_wide(pk[,2:5], scags=c("outlying", "monotonic"))
```

datasaurus_dozen	<i>datasaurus_dozen data</i>
------------------	------------------------------

Description

From the datasauRus package. A modern update of Anscombe. All plots have same x and y mean, variance and correlation, but look different visually.

All variables and pairs of variables have same summary statistics but are very different data, as can be seen by visualisation.

Format

A tibble with 1,846 observations and 3 variables

dataset label of data set

x variable for horizontal axis

y variable for vertical axis

A tibble with 142 observations and 26 variables

away_x, away_y x and y variables for away data

bullseye_x, bullseye_y x and y variables for bullseye data

circle_x, circle_y x and y variables for circle data

dino_x, dino_y x and y variables for dino data

dots_x, dots_y x and y variables for dots data

h_lines_x, h_lines_y x and y variables for h_lines data

high_lines_x, high_lines_y x and y variables for high_lines data

slant_down_x, slant_down_y x and y variables for slant_down data

slant_up_x, slant_up_y x and y variables for slant_up data

star_x, star_y x and y variables for star data

v_lines_x, v_lines_y x and y variables for v_lines data

wide_lines_x, wide_lines_y x and y variables for wide_lines data

star_x, star_y x and y variables for star data

x_shape_x, x_shape_y x and y variables for x_shape data

diagnose_outliers	<i>Diagnose outlier removal for one variable pair</i>
-------------------	---

Description

Identifies which observations are kept or removed by the outlier removal step for a chosen pair of numeric variables.

Usage

```
diagnose_outliers(data, x, y)
```

Arguments

data A data frame or tibble.

x, y Two numeric columns to use for the outlier removal diagnostic.

Value

A data frame with the same rows and columns as data, plus an outlier_status column. Values are "Kept" or "Removed".

Examples

```
data <- data.frame(
  id = 1:10,
  var1 = c(1, 2, 3, 4, 5, 6, 7, 20, 21, 22),
  var2 = c(1, 2, 3, 4, 5, 6, 7, 20, -20, 22)
)

diagnose_outliers(data, var1, var2)
```

draw_functions

Drawing the graph objects

Description

These functions will draw the graph objects that are used to compute the scagnostics. They are useful for debugging and seeing the impact of parameter adjustments, alpha binning, or outlier removal. You can draw the MST, convex hull, and alpha hull with each respective draw_* function.

Usage

```
draw_alphahull(
  x,
  y,
  out.rm = TRUE,
  binner = "hex",
  alpha = "rahman",
  fill = FALSE
)

draw_mst(x, y, out.rm = TRUE, binner = "hex")

draw_convexhull(x, y, out.rm = TRUE, binner = "hex", fill = FALSE)
```

Arguments

x, y	numeric vectors
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells)

	• NULL: no binning (use raw points) • a function: user-defined binner
alpha	character, numeric, or function. Controls the alpha radius. Valid character values are: • "rahman" (default): Rahman's MST-based middle-50% alpha • "q90": 90th percentile of MST edge lengths • "omega": graph-theoretic scagnostics alpha Alternatively: • a numeric value giving a fixed alpha • a function with no arguments that returns a single numeric alpha
fill	set to TRUE if you want the polygon filled

Value

A ggplot object that shows the respective graph object

Examples

```
require(dplyr)
require(ggplot2)
require(alphahull)

cl <- features |> filter(feature == "clusters")

# draw the alpha hull
draw_alphahull(cl$x, cl$y)

# draw the MST
draw_mst(cl$x, cl$y)

# draw the convex hull
draw_convexhull(cl$x, cl$y)

# You can utilise these functions to see the impact of parameter changes
draw_alphahull(cl$x, cl$y, alpha = "omega")
```

features	<i>Simulated data with special features</i>
----------	---

Description

Simulated data with common features that might be seen in 2D data. Variable are feature, x, y.

Format

A tibble with 1,013 observations and 3 variables, and 15 different patterns

- feature** label of data set
- x** variable for horizontal axis
- y** variable for vertical axis

numbat	<i>A toy data set with a numbat shape hidden among noise variables</i>
--------	--

Description

There are 7 variables (x1-x7) and 2,100 observations. Variables 4 and 7 have the numbat. The rest are noise. Group A has the numbat, and group B is all noise.

pk	<i>Parkinsons data from UCI machine learning archive</i>
----	--

Description

Biomedical voice measurements from 31 people, 23 with Parkinson's disease (PD). Each column in the table is a particular voice measure, and each row corresponds one of 195 voice recording from these individuals ("name" column). The main aim of the data is to discriminate healthy people from those with PD, according to "status" column which is set to 0 for healthy and 1 for PD.

Format

A tibble with 1,013 observations and 3 variables

name ASCII subject name and recording number

MDVP:Fo(Hz) Average vocal fundamental frequency

MDVP:Fhi(Hz) Maximum vocal fundamental frequency

MDVP:Flo(Hz) Minimum vocal fundamental frequency

MDVP:Jitter,MDVP:Jitter(Abs),MDVP:RAP,MDVP:PPQ,Jitter:DDP Several measures of variation in fundamental frequency

MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5,MDVP:APQ,Shimmer:DDA Several measures of variation in amplitude

NHR,HNR Two measures of ratio of noise to tonal components in the voice

status Health status of the subject (one) - Parkinson's, (zero) - healthy

RPDE,D2 Two nonlinear dynamical complexity measures

DFA Signal fractal scaling exponent

spread1,spread2,PPE Three nonlinear measures of fundamental frequency variation

Details

The data is available at [The UCI Machine Learning Repository](#) in ASCII CSV format. The rows of the CSV file contain an instance corresponding to one voice recording. There are around six recordings per patient, the name of the patient is identified in the first column.

The data are originally analysed in: Max A. Little, Patrick E. McSharry, Eric J. Hunter, Lorraine O. Ramig (2008), 'Suitability of dysphonia measurements for telemonitoring of Parkinson's disease', IEEE Transactions on Biomedical Engineering.

sc_clumpy

*Compute clumpy scagnostic measure using MST***Description**

This measure is used to detect clustering and is calculated through an iterative process. It was first defined in Graph Theoretic Scagnostics, Wilkinson, et al. (2005). First an edge J is selected and removed from the MST. From the two spanning trees that are created by this break, we select the largest edge from the smaller tree (K). The length of this edge (K) is compared to the removed edge (J) giving a clumpy measure for this edge. This process is repeated for every edge in the MST and the final clumpy measure is the maximum of this value over all edges.

Usage

```
sc_clumpy(x, y, out.rm = TRUE, binner = "hex")
```

Arguments

x, y	numeric vectors
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner

Value

A numeric object that gives the plot's clumpy score.

Examples

```
require(ggplot2)
require(dplyr)

# plot the feature
ggplot(features, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~feature, ncol = 5, scales = "free")

# calculate using tidy code
features |>
  group_by(feature) |>
  summarise(clumpy = sc_clumpy(x,y))

# using two vectors
```

```

x <- datasaurus_dozen_wide$slant_up_x
y <- datasaurus_dozen_wide$slant_up_y

# plot it
ggplot() +
  geom_point(aes(x = x, y = y))

# calculate using vectors
sc_clumpy(x, y)

```

sc_clumpy_r

*Compute robust clumpy scagnostic measure using MST***Description**

A computation for clumpy that is supposed to make the measure more robust to changes in binning. The scagnostic is defined in Improving the Robustness of Scagnostics, Wang, et al. (2020).

Usage

```
sc_clumpy_r(x, y, out.rm = TRUE, binner = "hex")
```

Arguments

x, y	numeric vectors
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner

Value

A numeric object that gives the plot's robust clumpy score.

Examples

```

require(ggplot2)
require(dplyr)
ggplot(features, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~feature, ncol = 5, scales = "free")
features |> group_by(feature) |> summarise(clumpy = sc_clumpy_r(x,y))
sc_clumpy_r(datasaurus_dozen_wide$away_x, datasaurus_dozen_wide$away_y)

```

sc_clumpy2

Compute adjusted clumpy measure using MST

Description

This measure is defined in the cassowaryr paper by Mason, et al. (2025). It is an alternative measure for clumpiness. It is the ratio of the between cluster edges and the within cluster edges. It is a good alternative measure to clumpy when binning is removed as a pre-processing step.

Usage

```
sc_clumpy2(x, y, out.rm = TRUE, binner = "hex")
```

Arguments

x, y	numeric vectors
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner

Value

A numeric object that gives the plot's adjusted clumpy score.

Examples

```
require(ggplot2)
require(dplyr)

# plot features
ggplot(features, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~feature, ncol = 5, scales = "free")

# calculate clumpy2 on all features
features |>
  group_by(feature) |>
  summarise(clumpy2 = sc_clumpy2(x,y))

sc_clumpy2(datasaurus_dozen_wide$dots_x, datasaurus_dozen_wide$dots_y)

data <- features |> filter(feature == "clusters")
x <- data$x
y <- data$y
```

```
# calculate using vectors
sc_clumpy2(x, y)
```

sc_convex

Compute convex scagnostic measure

Description

A measure of how convex the shape of the data is. It was first defined in Graph Theoretic Scagnostics, Wilkinson, et al. (2005). Computed as the ratio between the area of the alpha hull and convex hull. Unlike the other scagnostic measures, a high value on convex does not correlate to an interesting scatter plot, rather it usually indicates a lack of relationship between the two variables.

Usage

```
sc_convex(x, y, alpha = "rahman", out.rm = TRUE, binner = "hex")
```

Arguments

x, y	numeric vectors
alpha	character, numeric, or function. Controls the alpha radius. Valid character values are: • "rahman" (default): Rahman's MST-based middle-50% alpha • "q90": 90th percentile of MST edge lengths • "omega": graph-theoretic scagnostics alpha Alternatively: • a numeric value giving a fixed alpha • a function with no arguments that returns a single numeric alpha
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner

Value

A numeric object that gives the plot's convex score.

Examples

```
require(ggplot2)
require(dplyr)
ggplot(features, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~feature, ncol = 5, scales = "free")
features |> group_by(feature) |> summarise(convex = sc_convex(x,y))
sc_convex(datasaurus_dozen_wide$away_x, datasaurus_dozen_wide$away_y)
```

sc_dcor

Distance correlation index.

Description

The distance correlation between X and Y defined by Székely, et al. in Measuring and testing dependence by correlation of distances. The measure was suggested as an association scagnostic in Katrin Grimm's PhD thesis (2016). Distance correlation is a measure of non-linear dependence which is 0 if and only if the two variables are independent. It is computed using an ANOVA like calculation on the pairwise distances between observations.

Usage

```
sc_dcor(x, y)
```

Arguments

<code>x</code>	numeric vector
<code>y</code>	numeric vector

Value

A "numeric" object that gives the plot's dcor score.

Examples

```
require(ggplot2)
require(tidyr)
require(dplyr)
data(anscombe)
anscombe_tidy <- anscombe |>
pivot_longer(cols = everything(),
  names_to = c(".value", "set"),
  names_pattern = "(.)(.)")
ggplot(anscombe_tidy, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~set, ncol=2, scales = "free")
sc_dcor(anscombe$x1, anscombe$y1)
sc_dcor(anscombe$x2, anscombe$y2)
sc_dcor(anscombe$x3, anscombe$y3)
```

```
sc_dcor(anscombe$x4, anscombe$y4)
```

sc_grid

Compute the grid scanostic measure using MST

Description

The grid scagnostic as defined in Adam Rahman's PhD thesis (2018). The scagnosti identifies grid-like structures by counting the number of 90 and 180 degree angles in the MST. This measure can be used as an effective alternative to striated when computing scagnostics without binning.

Usage

```
sc_grid(x, y, epsilon = 0.01, out.rm = TRUE, binner = "hex")
```

```
sc_striated2(x, y, epsilon = 0.01, out.rm = TRUE, binner = "hex")
```

Arguments

x, y	numeric vectors
epsilon	the error tolerance allowed when deciding if the MST angles are at a right angle or not
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner

Value

A numeric object that gives the plot's grid score.

Examples

```
require(ggplot2)
require(dplyr)
ggplot(features, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~feature, ncol = 5, scales = "free")
features |> group_by(feature) |>
  summarise(grid1 = sc_grid(x,y),
            grid2 = sc_grid(x,y, epsilon=0.05))
sc_striated2(datasaurus_dozen_wide$away_x, datasaurus_dozen_wide$away_y)
```

`sc_monotonic`*Measure of Spearman Correlation*

Description

The measure of Spearman correlation calculated using the stats package cor function with method='spearman'.

Usage

```
sc_monotonic(x, y)
```

Arguments

x	numeric vector
y	numeric vector

Value

A "numeric" object that gives the plot's monotonic score.

See Also

`stats::cor`

Examples

```
require(ggplot2)
require(tidyr)
require(dplyr)
data(anscombe)
anscombe_tidy <- anscombe |>
pivot_longer(cols = everything(),
  names_to = c(".value", "set"),
  names_pattern = "(.)(.)")
ggplot(anscombe_tidy, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~set, ncol=2, scales = "free")
sc_monotonic(anscombe$x1, anscombe$y1)
sc_monotonic(anscombe$x2, anscombe$y2)
sc_monotonic(anscombe$x3, anscombe$y3)
sc_monotonic(anscombe$x4, anscombe$y4)
```

sc_outlying

Compute outlying scagnostic measure using MST

Description

A measure of proportion and severity of outliers in the dataset. It was first defined in Graph Theoretic Scagnostics, Wilkinson, et al. (2005). It is calculated by comparing the edge lengths of the outlying points in the MST with the total length of all the edges in the MST.

Usage

```
sc_outlying(x, y, binner = "hex")
```

Arguments

x, y	numeric vectors
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner

Value

A numeric object that gives the plot's outlying score.

Examples

```
require(ggplot2)
require(dplyr)

# plot the feature
ggplot(features, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~feature, ncol = 5, scales = "free")

# calculate using tidy code
features |>
  group_by(feature) |>
  summarise(outlying = sc_outlying(x,y))

# using two vectors
x <- datasaurus_dozen_wide$away_x
y <- datasaurus_dozen_wide$away_y

# plot it
ggplot() +
  geom_point(aes(x = x, y = y))
```

```
# calculate scag
sc_outlying(x, y)
```

sc_skewed

Compute skewed scagnostic measure using MST

Description

A measure of skewness in the edge lengths of the MST (not in the distribution of the data). It was first defined in Graph Theoretic Scagnostics, Wilkinson, et al. (2005). It is the ratio between the 90th to 50th percentile range and the 10th to 90th percentile range.

Usage

```
sc_skewed(x, y, out.rm = TRUE, binner = "hex")
```

Arguments

x, y	numeric vectors
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner

Value

A numeric object that gives the plot's skewed score.

Examples

```
require(ggplot2)
require(dplyr)

# plot the feature
ggplot(features, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~feature, ncol = 5, scales = "free")

# calculate using tidy code
features |>
  group_by(feature) |>
  summarise(skewed = sc_skewed(x,y))

# using two vectors
```

```

x <- datasaurus_dozen_wide$away_x
y <- datasaurus_dozen_wide$away_y

# plot it
ggplot() +
  geom_point(aes(x = x, y = y))

# calculate using vectors
sc_skewed(x, y)

```

sc_skinny

*Compute skinny scagnostic measure***Description**

A measure of how “thin” the shape of the data is. It was first defined in Graph Theoretic Scagnostics, Wilkinson, et al. (2005). It is calculated as the ratio between the area and perimeter of the alpha hull with some normalisation such that 0 correspond to a perfect circle and values close to 1 indicate a skinny polygon.

Usage

```
sc_skinny(x, y, alpha = "rahman", out.rm = TRUE, binner = "hex")
```

Arguments

x, y	numeric vectors
alpha	character, numeric, or function. Controls the alpha radius. Valid character values are: • "rahman" (default): Rahman's MST-based middle-50% alpha • "q90": 90th percentile of MST edge lengths • "omega": graph-theoretic scagnostics alpha Alternatively: • a numeric value giving a fixed alpha • a function with no arguments that returns a single numeric alpha
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner

Value

A numeric object that gives the plot's skinny score.

Examples

```
require(ggplot2)
require(dplyr)
ggplot(features, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~feature, ncol = 5, scales = "free")
features |> group_by(feature) |> summarise(skinny = sc_skinny(x,y))
sc_skinny(datasaurus_dozen_wide$away_x, datasaurus_dozen_wide$away_y)
```

sc_sparse

*Compute sparse scagnostic measure using MST***Description**

Identifies if the data is confined to a small number of locations on the plane. It was first defined in Scagnostics Distributions by Wilkinson & Wills (2008). It is calculated as the 90th percentile of MST edge lengths

Usage

```
sc_sparse(x, y, out.rm = TRUE, binner = "hex")
```

Arguments

x, y	numeric vectors
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner

Value

A numeric object that gives the plot's sparse score.

Examples

```
require(ggplot2)
require(dplyr)

# plot the feature
ggplot(features, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~feature, ncol = 5, scales = "free")
```

```

# calculate using tidy code
features |>
  group_by(feature) |>
  summarise(sparse = sc_sparse(x,y))

# using two vectors
x <- datasaurus_dozen_wide$dots_x
y <- datasaurus_dozen_wide$dots_y

# plot it
ggplot() +
  geom_point(aes(x = x, y = y))

# calculate using vectors
sc_sparse(x, y)

```

sc_sparse2

Compute adjusted sparse measure using the alpha hull

Description

The sparse2 measure created for cassowaryr. The measure calculates the sparsity of the plot as $1 - \text{area}(\text{ahull})$.

Usage

```
sc_sparse2(x, y, out.rm = TRUE, binner = "hex")
```

Arguments

x, y	numeric vectors
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation. Can be: "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until ≤ 250 nonempty cells) - NULL: no binning (use raw points) - a function: user-defined binner

Value

A numeric object that gives the plot's adjusted sparse score.

Examples

```
require(ggplot2)
require(tidyr)
require(dplyr)
data(anscombe_tidy)
ggplot(anscombe_tidy, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~set, ncol=2, scales = "free")
sc_sparse2(anscombe$x1, anscombe$y1)
```

sc_splines

Spline based index.

Description

Measures the functional non-linear dependence by fitting a penalised splines model on X using Y, and on Y using X. The measure was defined as an association scagnostic in Katrin Grimm's PhD thesis (2016) The variance of the residuals are scaled down by the axis so they are comparable, and finally the maximum is taken. Therefore the value will be closer to 1 if either relationship can be decently explained by a splines model.

Usage

```
sc_splines(x, y)
```

Arguments

x	numeric vector
y	numeric vector

Value

A "numeric" object that gives the plot's splines score.

Examples

```
require(ggplot2)
require(tidyr)
require(dplyr)
data(anscombe)
anscombe_tidy <- anscombe |>
  pivot_longer(cols = everything(),
    names_to = c(".value", "set"),
    names_pattern = "(.)(.)")
ggplot(anscombe_tidy, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~set, ncol=2, scales = "free")
```

```

sc_splines(anscombe$x1, anscombe$y1)
sc_splines(anscombe$x2, anscombe$y2)
sc_splines(anscombe$x3, anscombe$y3)

```

sc_striated

Compute striated scagnostic measure using MST

Description

This measure identifies features such as discreteness by finding parallel lines. It was first defined in Graph Theoretic Scagnostics, Wilkinson, et al. (2005). It is calculated by counting the proportion of vertices with only two edges that have an inner angle approximately between 135 and 220 degrees.

Usage

```
sc_striated(x, y, out.rm = TRUE, binner = "hex")
```

Arguments

x, y	numeric vectors
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner

Value

A numeric object that gives the plot's striated score.

Examples

```

require(ggplot2)
require(dplyr)

# plot the features data
ggplot(features, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~feature, ncol = 5, scales = "free")

# calculate using tidy code
features |>
  group_by(feature) |>
  summarise(striated = sc_striated(x,y)) |>
  arrange(striated)

```

```
# using just vectors of points
x <- datasaurus_dozen_wide$v_lines_x
y <- datasaurus_dozen_wide$v_lines_y

# plot it
ggplot() +
  geom_point(aes(x = x, y = y))

# calculate scagnostic
sc_striated(x, y)
```

sc_stringy05

Compute the stringy05 scagnostic measure

Description

Computes the Stringy05 measure as defined in *Graph-Theoretic Scagnostics* (Wilkinson et al., 2005).

Usage

```
sc_stringy05(x, y, out.rm = TRUE, binner = "hex", rescale = FALSE)

## Default S3 method:
sc_stringy05(x, y, out.rm = TRUE, binner = "hex", rescale = FALSE)

sc_stringy2(x, y, out.rm = TRUE, binner = "hex")
```

Arguments

x, y	numeric vectors
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner
rescale	Logical. If TRUE, apply the Gaussian-noise rescaling. Defaults to FALSE.

Details

Stringy05 is the weighted diameter of the minimum spanning tree divided by the total length of its edges.

When `rescale = TRUE`, the index is rescaled using a sample-size-dependent Gaussian-noise threshold:

$$l(n) = 0.05 + 3.86 / \sqrt{n}$$

The rescaled index is:

$$\max(0, (I - l(n)) / (1 - l(n)))$$

Value

A numeric value giving the original or rescaled Stringy05 score.

Examples

```
x <- datasaurus_dozen_wide$star_x
y <- datasaurus_dozen_wide$star_y

sc_stringy05(x, y)
sc_stringy05(x, y, rescale = TRUE)
```

sc_stringy06

Compute the stringy06 scagnostic measure using the MST

Description

This measure identifies a “stringy” shape with no branches, such as a thin line of data. The `stringy06` function is defined in *High-Dimensional Visual Analytics: Interactive Exploration Guided by Pair-wise Views of Point Distributions* (Wilkinson et al., 2006). It is calculated using the minimum spanning tree (MST) by comparing the number of vertices with degree two to the total number of vertices, dropping those of degree one.

Usage

```
sc_stringy06(x, y, out.rm = TRUE, binner = "hex")

sc_stringy(x, y, out.rm = TRUE, binner = "hex")
```

Arguments

<code>x, y</code>	numeric vectors
<code>out.rm</code>	logical; if <code>TRUE</code> , iteratively trim large MST edges, If <code>FALSE</code> the scagnostics will be computed on the entire data set with no outlier removal.
<code>binner</code>	an optional function that bins the <code>x</code> and <code>y</code> vectors prior to triangulation Can be:

- "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until ≤ 250 nonempty cells)
- NULL: no binning (use raw points)
- a function: user-defined binner

Details

The name "stringy06" is used to distinguish this version from earlier definition of the stringy measure.

Value

A numeric object that gives the plot's stringy score.

Examples

```
require(ggplot2)
require(dplyr)

# plot the features data
ggplot(features, aes(x=x, y=y)) +
  geom_point() +
  facet_wrap(~feature, ncol = 5, scales = "free")

# calculate using tidy code
features |>
  group_by(feature) |>
  summarise(stringy = sc_stringy06(x,y))

# using just vectors of points
x <- datasaurus_dozen_wide$star_x
y <- datasaurus_dozen_wide$star_y

# plot it
ggplot() +
  geom_point(aes(x = x, y = y))

# calculate using vectors
sc_stringy06(x, y)
```

sc_stripped

Measure of Discreteness

Description

This metric computes the 1-(ratio between the number of unique values to total data values) on number of rotations of the data, and returns the smallest value. If this value is large it means that there are only a few unique data values, and hence the distribution is discrete

Usage

```
sc_stripped(x, y)
```

Arguments

x	numeric vector
y	numeric vector

Value

double

Examples

```
data("datasaurus_dozen_wide")
sc_stripped(datasaurus_dozen_wide$v_lines_x,
            datasaurus_dozen_wide$v_lines_y)
sc_stripped(datasaurus_dozen_wide$dino_x,
            datasaurus_dozen_wide$dino_y)
```

scree

Pre-processing to generate scagnostic measures

Description

This function performs the pre-processing requires to calculate the scagnostic measures. This includes the binning, outlier removal, and calculation of the alpha value.

Usage

```
scree(x, y, out.rm = TRUE, binner = "hex", alpha = "rahman", ...)
```

Arguments

x, y	numeric vectors
out.rm	logical; if TRUE, iteratively trim large MST edges, If FALSE the scagnostics will be computed on the entire data set with no outlier removal.
binner	an optional function that bins the x and y vectors prior to triangulation Can be: • "hex" (default): hexagonal binning following the procedure in the graph-theoretic scagnostics paper (start 40x40, halve until <= 250 nonempty cells) • NULL: no binning (use raw points) • a function: user-defined binner
alpha	character, numeric, or function. Controls the alpha radius. Valid character values are: • "rahman" (default): Rahman's MST-based middle-50% alpha

- "q90": 90th percentile of MST edge lengths
 - "omega": graph-theoretic scagnostics alpha Alternatively:
 - a numeric value giving a fixed alpha
 - a function with no arguments that returns a single numeric alpha
- ... other args

Value

An object of class "scree" that consists of three elements:

- del: the Delauney-Voronoi tessellation from `alphahull::delvor()`
- weights: the lengths of each edge in the Delauney triangulation
- alpha: the radius or alpha value that will be used to generate the alphahull

Examples

```
set.seed(232)

x <- runif(1000)
y <- runif(1000)

# make scree
sc0 <- scree(x,y)
sc1 <- scree(x,y, out.rm = FALSE) # no outlier removal
sc2 <- scree(x, y, binner = NULL) # no hexagonal binning

# see the difference made by binning out out.rm
draw_mst(sc0)
draw_mst(sc1)
draw_mst(sc2)
```

top_functions

Summary computations for scagnostic data

Description

These functions suggests a summary statistic that can be found using the scag calculations provided by `calc_scags`. The `top_pair` function finds the top pair of variables for each of the scagnostics, while `top_scag` finds the highest value scagnostic for each pair of variables. While these computations are relatively straight forward for any R user to compute themselves, including these summary function in the package simultaneously streamlines a common calculation made with the scagnostic data and suggests this summary to new users of the package.

Usage

```
top_pair(scags_data)

top_scag(scags_data)
```

Arguments

scags_data A dataset of scagnostic values that was returned by calc_scags or calc_scags_wide

Value

A data frame. For top_pair, each row will represent a scagnostic with its highest pair. For top_scag, each row will represent a pair of variables with its highest valued scagnostic.

See Also

calc_scags calc_scags_wide

Examples

```
require(dplyr)
# calculate scag data
scag_data <- datasaurus_dozen |>
  group_by(dataset) |>
  summarise(calc_scags(x,y, scags=c("monotonic", "outlying", "convex")))

# Calculate top_pair
scag_data |>
  top_pair()

# Calculate top_scag
scag_data |>
  top_scag()
```

Index

alphahull::delvor(), 28
anscombe_tidy, 2

calc_scags, 3
calc_scags_wide, 4

datasaurus_dozen, 5
datasaurus_dozen_wide
 (datasaurus_dozen), 5
diagnose_outliers, 6
draw_alphahull(draw_functions), 7
draw_convexhull(draw_functions), 7
draw_functions, 7
draw_mst(draw_functions), 7

features, 8

numbat, 9

pk, 9

sc_clumpy, 10
sc_clumpy2, 12
sc_clumpy_r, 11
sc_convex, 13
sc_dcor, 14
sc_grid, 15
sc_monotonic, 16
sc_outlying, 17
sc_skewed, 18
sc_skinny, 19
sc_sparse, 20
sc_sparse2, 21
sc_splines, 22
sc_striated, 23
sc_striated2(sc_grid), 15
sc_stringy(sc_stringy06), 25
sc_stringy05, 24
sc_stringy06, 25
sc_stringy2(sc_stringy05), 24
sc_stripped, 26

scree, 27

top_functions, 28
top_pair(top_functions), 28
top_scag(top_functions), 28