

# Package ‘bullseye’

January 28, 2025

**Title** Visualising Multiple Pairwise Variable Correlations and Other Scores

**Version** 0.1.2

## Description

We provide a tidy data structure and visualisations for multiple or grouped variable correlations, general association measures scagnostics and other pairwise scores suitable for numerical, ordinal and nominal variables.

Supported measures include distance correlation, maximal information, ace correlation, Kendall's tau, and polychoric correlation.

**License** MIT + file LICENSE

**Depends** R (>= 4.1.0)

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**Imports** stats, dplyr, cli, rlang, ggplot2, labeling, ggiraph, DendSer, tidy, polycor

**Suggests** DescTools, acepack, energy, linkspotter, minerva, scagnostics, testthat (>= 3.0.0), knitr, rmarkdown, correlation, palmerpenguins, kableExtra, openintro, corplot

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**URL** <https://cbhurley.github.io/bullseye/>,  
<https://github.com/cbhurley/bullseye>

**NeedsCompilation** no

**Author** Amit Chinwan [aut],  
Catherine Hurley [aut, cre]

**Maintainer** Catherine Hurley <catherine.hurley@mu.ie>

**Repository** CRAN

**Date/Publication** 2025-01-28 10:40:06 UTC

## Contents

ace_cor . . . . .	2
as.matrix.pairwise . . . . .	3
pairwise . . . . .	4
pairwise_by . . . . .	5
pairwise_multi . . . . .	5
pairwise_scores . . . . .	6
pair_ace . . . . .	7
pair_cancor . . . . .	8
pair_chi . . . . .	9
pair_control . . . . .	9
pair_cor . . . . .	10
pair_dcor . . . . .	11
pair_gkGamma . . . . .	12
pair_gkTau . . . . .	13
pair_methods . . . . .	13
pair_mine . . . . .	14
pair_nmi . . . . .	15
pair_polychor . . . . .	15
pair_polyserial . . . . .	16
pair_scagnostics . . . . .	17
pair_tauA . . . . .	18
pair_tauB . . . . .	18
pair_tauC . . . . .	19
pair_tauW . . . . .	20
pair_uncertainty . . . . .	21
plot.pairwise . . . . .	21
plot_pairwise . . . . .	22
plot_pairwise_linear . . . . .	23
<b>Index</b>	<b>25</b>

---

ace_cor	<i>Calculates ace based transformations and correlation, handling missing values and factors.</i>
---------	---

---

### Description

Calculates ace based transformations and correlation, handling missing values and factors.

### Usage

```
ace_cor(x, y, handle.na = TRUE)
```

**Arguments**

x                    a numeric vector or factor  
y                    a numeric vector or factor  
handle.na           If TRUE uses pairwise complete observations.

**Value**

result of acepack::ace

**Examples**

```
ace_cor(iris$Sepal.Length, iris$Species)
```

---

as.matrix.pairwise     *Converts a pairwise to a symmetric matrix. Uses the first entry for each (x,y) pair.*

---

**Description**

Converts a pairwise to a symmetric matrix. Uses the first entry for each (x,y) pair.

**Usage**

```
## S3 method for class 'pairwise'  
as.matrix(x, ...)
```

**Arguments**

x                    An object of class pairwise  
...                  other arguments

**Value**

A symmetric matrix

---

pairwise	<i>A generic function to create a data structure for every variable pair in a dataset</i>
----------	---

---

### Description

Creates a data structure for every variable pair in a dataset.

### Usage

```
pairwise(x, score = NA_character_, pair_type = NA_character_)

## S3 method for class 'matrix'
pairwise(x, score = NA_character_, pair_type = NA_character_)

## S3 method for class 'data.frame'
pairwise(x, score = NA_character_, pair_type = NA_character_)

## S3 method for class 'easycorrelation'
pairwise(x, score = NA_character_, pair_type = NA_character_)

as.pairwise(x, score = NA_character_, pair_type = NA_character_)
```

### Arguments

x	A dataframe or symmetric matrix.
score	a character string indicating the value of association, either "nn", "fn", "ff".
pair_type	a character string specifying the type of variable pair.

### Value

A `tbl_df` of class `pairwise` for pairs of variables with a column value for the score value, `score` for a type of association value and `pair_type` for the type of variable pair.

### Methods (by class)

- `pairwise(matrix)`: pairwise method
- `pairwise(data.frame)`: pairwise method
- `pairwise(easycorrelation)`: pairwise method

### Functions

- `as.pairwise()`: Same as `pairwise`

### Examples

```
pairwise(cor(iris[,1:4]), score="pearson")
pairwise(iris)
```

---

pairwise_by	<i>Constructs a pairwise result for each level of a by variable.</i>
-------------	--

---

**Description**

Constructs a pairwise result for each level of a by variable.

**Usage**

```
pairwise_by(d, by, pair_fun, ungrouped = TRUE)
```

**Arguments**

d	a dataframe
by	a character string for the name of the conditioning variable.
pair_fun	A function returning a pairwise from a dataset.
ungrouped	If TRUE calculates the ungrouped score in addition to grouped scores.

**Value**

tibble of class "pairwise"

**Examples**

```
pairwise_by(iris, by="Species", pair_cor)
```

---

pairwise_multi	<i>Calculates multiple scores</i>
----------------	-----------------------------------

---

**Description**

Calculates multiple scores for every variable pair in a dataset.

**Usage**

```
pairwise_multi(  
  d,  
  scores = c("pair_cor", "pair_dcor", "pair_mine", "pair_ace", "pair_cancor", "pair_nmi",  
            "pair_uncertainty", "pair_chi"),  
  handle.na = TRUE  
)
```

**Arguments**

d	dataframe
scores	a vector naming functions returning a pairwise from a dataset.
handle.na	If TRUE uses pairwise complete observations to calculate pairwise score, otherwise NAs not handled.

**Value**

tibble of class "pairwise"

**Examples**

```
iris1 <- iris
iris1$Sepal.Length <- cut(iris1$Sepal.Length,3)
pairwise_multi(iris1)
```

---

pairwise_scores	<i>Calculates scores or conditional scores for a dataset</i>
-----------------	--

---

**Description**

Calculates scores for every variable pair in a dataset when by is NULL. If by is a name of a variable in the dataset, conditional scores for every variable pair at different levels of the grouping variable are calculated.

**Usage**

```
pairwise_scores(
  d,
  by = NULL,
  ungrouped = TRUE,
  control = pair_control(),
  handle.na = TRUE
)
```

**Arguments**

d	a dataframe
by	a character string for the name of the conditioning variable. Set to NULL by default.
ungrouped	Ignored if by is NULL. If TRUE calculates the ungrouped score in addition to grouped scores.
control	a list for the measures to be calculated for different variable types. The default is <code>pair_control()</code> which calculates Pearson's correlation if the variable pair is numeric, canonical correlation for factor or mixed pairs, and polychoric correlation for two ordered factors.
handle.na	If TRUE uses pairwise complete observations to calculate measure of association.

**Details**

Returns a pairwise tibble structure.

**Value**

A tibble with class pairwise.

**Examples**

```

iris_sc <- pairwise_scores(iris)
iris_sc <- pairwise_scores(iris, control=pair_control(nnargs= c(method="spearman")))
iris_sc <- pairwise_scores(iris, control=pair_control(fn="pair_ace"))

#Lots of numerical measures
iris_sc <- pairwise_scores(iris, control=pair_control(nn="pairwise_multi", fn=NULL))
iris_sc <- pairwise_scores(iris,
  control=pair_control(nn="pairwise_multi", nnargs="pair_cor", fn=NULL))
#conditional measures
cond_iris <- pairwise_scores(iris, by = "Species")
cond_iris_wo <- pairwise_scores(iris, by = "Species",ungrouped=FALSE) # without overall
iris_sc <- pairwise_scores(iris, control=pair_control(nn="pairwise_multi", fn=NULL))
iris_sc <- pairwise_scores(iris, by = "Species",control=pair_control(nn="pairwise_multi", fn=NULL))

#scagnostics
sc <- pairwise_scores(iris, control=pair_control(nn="pair_scagnostics", fn=NULL)) # ignore fn pairs
sc <- pairwise_scores(iris, by = "Species",
  control=pair_control(nn="pair_scagnostics", fn=NULL)) # ignore fn pairs

```

---

pair\_ace

*Alternating conditional expectations correlation*

---

**Description**

Calculates the maximal correlation coefficient from alternating conditional expectations algorithm for every variable pair in a dataset.

**Usage**

```
pair_ace(d, handle.na = TRUE, ...)
```

**Arguments**

d	A dataframe
handle.na	If TRUE uses pairwise complete observations, otherwise NAs not handled.
...	other arguments

**Details**

The maximal correlation is calculated using alternating conditional expectations algorithm which find the transformations of variables such that the squared correlation is maximised. The `ace` function from `acepack` package is used for the calculation.

**Value**

A tibble of class `pairwise` with a maximal correlation from the alternating conditional expectations algorithm for every variable pair

**References**

Breiman, Leo, and Jerome H. Friedman. "Estimating optimal transformations for multiple regression and correlation." *Journal of the American statistical Association* 80.391 (1985): 580-598.

**Examples**

```
pair_ace(iris)
```

---

pair_cancor	<i>Canonical correlation</i>
-------------	------------------------------

---

**Description**

Calculates canonical correlation for every variable pair in a dataset.

**Usage**

```
pair_cancor(d, handle.na = TRUE, ...)
```

**Arguments**

<code>d</code>	A dataframe
<code>handle.na</code>	If TRUE uses pairwise complete observations to calculate correlation coefficient, otherwise NAs not handled.
<code>...</code>	other arguments

**Value**

A tibble of class `pairwise` with canonical correlation for every numeric or factor or mixed variable pair

**Examples**

```
pair_cancor(iris)
```



---

pair_chi	<i>Pearson's Contingency Coefficient for association between factors.</i>
----------	---

---

### Description

Calculates Pearson's Contingency coefficient for every factor variable pair in a dataset.

### Usage

```
pair_chi(d, handle.na = TRUE, ...)
```

### Arguments

d	A dataframe
handle.na	ignored. Pairwise complete observations are used automatically.
...	other arguments

### Details

The Pearson's contingency coefficient is calculated using [ContCoef](#). NAs are automatically handled by pairwise omit.

### Value

A tibble of class `pairwise` with calculated Pearson's contingency coefficient for every factor variable pair, or NULL if there are not at least two factor variables

### Examples

```
pair_chi(iris)
```

---

pair_control	<i>Default scores calculated by pairwise_scores</i>
--------------	---

---

### Description

Gives a list specifying the function to be used for two numeric (nn) variables, two factors (ff), two ordinals (oo) and for a factor-numeric pair (fn).

**Usage**

```
pair_control(
  nn = "pair_cor",
  oo = "pair_polychor",
  ff = "pair_cancor",
  fn = "pair_cancor",
  nnargs = NULL,
  ooargs = NULL,
  ffargs = NULL,
  fnargs = NULL
)
```

**Arguments**

nn	function for numeric pairs of variables, should return object of class pairwise. Use NULL to ignore numeric pairs.
oo	function for ordered factor pairs of variables, should return object of class pairwise. Use NULL to ignore ordered factor pairs.
ff	function for factor pairs of variables (not ordered), should return object of class pairwise. Use NULL to ignore factor-factor pairs.
fn	function for factor-numeric pairs of variables, should return object of class pairwise. Use NULL to ignore factor-numeric pairs.
nnargs	other arguments for the nn function
ooargs	other arguments for the oo function
ffargs	other arguments for the ff function
fnargs	other arguments for the fn function

**Value**

list

---

pair\_cor

*Pearson, Spearman or Kendall correlation*

---

**Description**

Calculates one of either pearson, spearman or kendall correlation for every numeric variable pair in a dataset.

**Usage**

```
pair_cor(d, method = "pearson", handle.na = TRUE, ...)
```

**Arguments**

d	A dataframe
method	A character string for the correlation coefficient to be calculated. Either "pearson" (default), "spearman", or "kendall". If the value is "all", then all three correlations are calculated.
handle.na	If TRUE uses pairwise complete observations to calculate correlation coefficient, otherwise NAs not handled.
...	other arguments

**Value**

A tibble of class `pairwise` with calculated association value for every numeric variable pair, or NULL if there are not at least two numeric variables

**See Also**

See [pair\\_methods](#) for other score options.

**Examples**

```
pair_cor(iris)
pair_cor(iris, method="kendall")
pair_cor(iris, method="spearman")
pair_cor(iris, method="all")
```

---

pair_dcor	<i>Distance correlation</i>
-----------	-----------------------------

---

**Description**

Calculates distance correlation for every numeric variable pair in a dataset.

**Usage**

```
pair_dcor(d, handle.na = TRUE, ...)
```

**Arguments**

d	A dataframe
handle.na	If TRUE uses pairwise complete observations to calculate distance correlation, otherwise NAs not handled.
...	other arguments

**Details**

The distance correlation is calculated using [dcor2d](#) from energy package

**Value**

A tibble of class `pairwise` with distance correlation for every numeric variable pair, or NULL if there are not at least two numeric variables

**Examples**

```
pair_dcor(iris)
```

---

```
pair_gkGamma
```

*Goodman Kruskal's Gamma for association between ordinal factors.*

---

**Description**

Calculates Goodman Kruskal's Gamma coefficient for every factor variable pair in a dataset.

**Usage**

```
pair_gkGamma(d, handle.na = TRUE, ...)
```

**Arguments**

<code>d</code>	A dataframe
<code>handle.na</code>	ignored. Pairwise complete observations are used automatically.
<code>...</code>	other arguments

**Details**

The Goodman Kruskal's Gamma coefficient is calculated using `GoodmanKruskalGamma` function from the `DescTools` package. Assumes factor levels are in the given order. NAs are automatically handled by `pairwise omit`.

**Value**

A tibble of class `pairwise` with factor variable pairs and Goodman Kruskal's Gamma coefficient, or NULL if there are not at least two factor variables

**Examples**

```
pair_gkGamma(iris)
```

---

pair_gkTau	<i>Goodman Kruskal's Tau for association between ordinal factors.</i>
------------	---

---

**Description**

Calculates Goodman Kruskal's Tau coefficient for every factor variable pair in a dataset.

**Usage**

```
pair_gkTau(d, handle.na = TRUE, ...)
```

**Arguments**

d	A dataframe
handle.na	ignored. Pairwise complete observations are used automatically.
...	other arguments

**Details**

The Goodman Kruskal's Tau coefficient is calculated using [GoodmanKruskalTau](#) function from the DescTools package. Assumes factor levels are in the given order. NAs are automatically handled by pairwise omit.

**Value**

A tibble of class pairwise with Goodman Kruskal's Tau for every factor variable pair, or NULL if there are not at least two factor variables

**Examples**

```
pair_gkTau(iris)
```

---

pair_methods	<i>Pairwise score functions available in the package</i>
--------------	--

---

**Description**

A tibble of score functions along with the types of variable pairs these functions can be applied to. It also contains information regarding the packages used to calculate scores and the range of the values calculated.

**Usage**

```
pair_methods
```

**Format**

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 17 rows and 7 columns.

**Value**

tibble

**Examples**

```
pair_methods
```

---

pair_mine	<i>MINE family values</i>
-----------	---------------------------

---

**Description**

Calculates MINE family values for every numeric variable pair in a dataset.

**Usage**

```
pair_mine(d, method = "MIC", handle.na = TRUE, ...)
```

**Arguments**

<code>d</code>	A dataframe
<code>method</code>	character vector for the MINE value to be calculated. Subset of "MIC", "MAS", "MEV", "MCN", "MICR2", "GMIC", "TIC"
<code>handle.na</code>	If TRUE uses pairwise complete observations to calculate score, otherwise NAs not handled.
<code>...</code>	other arguments

**Details**

The values are calculated using [mine](#) from `minerva`

**Value**

A tibble of class `pairwise` with scores for numeric variable pairs, or NULL if there are not at least two numeric variables

**References**

Reshef, David N., et al. "Detecting novel associations in large data sets." *science* 334.6062 (2011): 1518-1524

**Examples**

```
pair_mine(iris)
pair_mine(iris, method="MAS")
```

---

pair_nmi	<i>Normalized mutual information</i>
----------	--------------------------------------

---

**Description**

Calculates normalized mutual information for every numeric or factor or mixed variable pair in a dataset.

**Usage**

```
pair_nmi(d, handle.na = TRUE, ...)
```

**Arguments**

d	A dataframe
handle.na	If TRUE uses pairwise complete observations to calculate normalized mutual information, otherwise NAs not handled.
...	other arguments

**Details**

The normalized mutual information is calculated using [maxNMI](#) from linkpotter package

**Value**

A tibble of class pairwise

**Examples**

```
if (requireNamespace("linkspotter", quietly = TRUE)) {  
  pair_nmi(iris)  
}
```

---

pair_polychor	<i>Polychoric correlation</i>
---------------	-------------------------------

---

**Description**

Calculates Polychoric correlation using for every factor variable pair in a dataset.

**Usage**

```
pair_polychor(d, handle.na = TRUE, ...)
```

**Arguments**

d	A dataframe
handle.na	ignored. Pairwise complete observations are used automatically.
...	other arguments

**Details**

The polychoric correlation is calculated using the [polychor](#) function from the polycor package, and assumes factor levels are in the given order. NAs are automatically handled by pairwise omit.

**Value**

A tibble of class pairwise with polychoric correlation for factor pairs, or NULL if there are not at least two factor variables

**Examples**

```
pair_polychor(iris)
```

---

pair_polyserial	<i>Polyserial correlation</i>
-----------------	-------------------------------

---

**Description**

Calculates Polyserial correlation using for every factor-numeric variable pair in a dataset.

**Usage**

```
pair_polyserial(d, handle.na = TRUE, ...)
```

**Arguments**

d	A dataframe
handle.na	ignored. Pairwise complete observations are used automatically.
...	other arguments

**Details**

The polyserial correlation is calculated using the [polyserial](#) function from the polycor package, and assumes factor levels are in the given order. NAs are automatically handled by pairwise omit.

**Value**

A tibble of class pairwise with polyserial correlation for factor-numeric pairs, or NULL if there are not at least one such pair.

**Examples**

```
pair_polyserial(iris)
```



---

pair\_scagnostics      *Graph-theoretic scagnostics values*

---

### Description

Calculates scagnostic values for every numeric variable pair in a dataset.

### Usage

```
pair_scagnostics(  
  d,  
  scagnostic = c("Outlying", "Skewed", "Clumpy", "Sparse", "Striated", "Convex",  
    "Skinny", "Stringy", "Monotonic"),  
  handle.na = TRUE,  
  ...  
)
```

### Arguments

d	A dataframe
scagnostic	a character vector for the scagnostic to be calculated. Subset of "Outlying", "Stringy", "Striated", "Clumpy", "Sparse", "Skewed", "Convex", "Skinny" or "Monotonic"
handle.na	If TRUE uses pairwise complete observations.
...	other arguments

### Details

The scagnostic values are calculated using [scagnostics](#) function from the `scagnostics` package.

### Value

A tibble of class `pairwise` with scagnostic values for every numeric variable pair, or NULL if there are not at least two numeric variables

### References

Wilkinson, Leland, Anushka Anand, and Robert Grossman. "Graph-theoretic scagnostics." Information Visualization, IEEE Symposium on. IEEE Computer Society, 2005

### Examples

```
pair_scagnostics(iris)
```

---

pair_tauA	<i>Kendall's tau A for association between ordinal factors.</i>
-----------	---

---

**Description**

Calculates Kendall's tau A for every factor variable pair in a dataset.

**Usage**

```
pair_tauA(d, handle.na = TRUE, ...)
```

**Arguments**

d	A dataframe
handle.na	ignored. Pairwise complete observations are used automatically.
...	other arguments

**Details**

Calculated using [KendallTauA](#). Assumes factor levels are in the given order. NAs are automatically handled by pairwise omit.

**Value**

A tibble of class pairwise with factor pairs, or NULL if there are not at least two factor variables

**Examples**

```
d <- data.frame(x=rnorm(20),
                y=factor(sample(3,20, replace=TRUE)),
                z=factor(sample(2,20, replace=TRUE)))
pair_tauA(d)
```

---

pair_tauB	<i>Kendall's tau B for association between ordinal factors.</i>
-----------	---

---

**Description**

Calculates Kendall's tau B every factor variable pair in a dataset.

**Usage**

```
pair_tauB(d, handle.na = TRUE, ...)
```

**Arguments**

d	A dataframe
handle.na	ignored. Pairwise complete observations are used automatically.
...	other arguments

**Details**

Calculated using [KendallTauB](#). Assumes factor levels are in the given order. NAs are automatically handled by pairwise omit.

**Value**

A tibble of class `pairwise` with factor pairs, or `NULL` if there are not at least two factor variables

**Examples**

```
d <- data.frame(x=rnorm(20),
                y=factor(sample(3,20, replace=TRUE)),
                z=factor(sample(2,20, replace=TRUE)))
pair_tauB(d)
```

---

pair\_tauC

*Stuarts's tau C for association between ordinal factors.*

---

**Description**

Calculates Stuarts's tau C every factor variable pair in a dataset.

**Usage**

```
pair_tauC(d, handle.na = TRUE, ...)
```

**Arguments**

d	A dataframe
handle.na	ignored. Pairwise complete observations are used automatically.
...	other arguments

**Details**

Calculated using [StuartTauC](#). Assumes factor levels are in the given order. NAs are automatically handled by pairwise omit.

**Value**

A tibble of class `pairwise` with factor pairs, or `NULL` if there are not at least two factor variables

### Examples

```
d <- data.frame(x=rnorm(20),
               y=factor(sample(3,20, replace=TRUE)),
               z=factor(sample(2,20, replace=TRUE)))
pair_tauC(d)
```

---

pair\_tauW

*Kendall's W for association between ordinal factors.*

---

### Description

Calculates Kendall's tau W every factor variable pair in a dataset.

### Usage

```
pair_tauW(d, handle.na = TRUE, ...)
```

### Arguments

d	A dataframe
handle.na	ignored. Pairwise complete observations are used automatically.
...	other arguments

### Details

Calculated using [KendallW](#). Assumes factor levels are in the given order. NAs are automatically handled by pairwise omit.

### Value

A tibble of class pairwise with factor pairs, or NULL if there are not at least two factor variables

### Examples

```
d <- data.frame(x=rnorm(20),
               y=factor(sample(3,20, replace=TRUE)),
               z=factor(sample(2,20, replace=TRUE)))
pair_tauW(d)
```

---

pair_uncertainty	<i>Uncertainty coefficient for association between factors.</i>
------------------	---

---

**Description**

Calculates uncertainty coefficient for every factor variable pair in a dataset.

**Usage**

```
pair_uncertainty(d, handle.na = TRUE, ...)
```

**Arguments**

d	A dataframe
handle.na	ignored. Pairwise complete observations are used automatically.
...	other arguments

**Details**

The Uncertainty coefficient is calculated using [UncertCoef](#) function from the DescTools package.

**Value**

A tibble of class pairwise with every factor variable pair and uncertainty coefficient value, or NULL if there are not at least two factor variables

**Examples**

```
pair_uncertainty(iris)
```

---

plot.pairwise	<i>Plot method for class pairwise.</i>
---------------	--

---

**Description**

Plot method for class pairwise.

**Usage**

```
## S3 method for class 'pairwise'
plot(x, type = c("matrix", "linear"), ...)
```

**Arguments**

x	An object of class pairwise
type	If "matrix", calls plot_pairwise, if "linear" calls plot_pairwise_linear
...	further arguments to plot_pairwise or plot_pairwise_linear

**Value**

a plot

**Examples**

```
plot(pairwise_scores(iris))
```

---

plot\_pairwise

*Pairwise plot in a matrix layout*

---

**Description**

Plots multiple pairwise variable scores in a matrix layout.

**Usage**

```
plot_pairwise(
  scores,
  var_order = "seriate_max",
  score_limits = NULL,
  inner_width = 0.5,
  center_level = "all",
  na.value = "grey80",
  interactive = FALSE
)
```

**Arguments**

scores	The scores for the matrix plot. Either of class <code>pairwise</code> or identical in structure to object of class <code>pairwise</code> .
var_order	The variable order to be used. The default <code>NULL</code> means variables in are ordered alphabetically. A value of <code>"seriate_max"</code> means variables are re-ordered to emphasize pairs with maximum absolute scores. A value of <code>"seriate_max_diff"</code> means variables are re-ordered to emphasize pairs with maximum score differences. Otherwise <code>Var_order</code> must be a subset of variables in scores.
score_limits	a numeric vector of length specifying the limits of the scale.
inner_width	A number between 0 and 1 specifying radius of the inner bullseye.
center_level	Specifies which level of group goes into the inner bullseye. Defaults to <code>"all"</code> .
na.value	used for scores with a value of <code>NA</code>
interactive	defaults to <code>FALSE</code>

**Value**

A girafe object if interactive==TRUE, otherwise a ggplot2.

If scores has one value for x,y pair, then a filled circle is drawn with fill representing the score value. If there are multiple values for each x,y pair then the filled circle is split into wedges, with the wedge fill representing the values. If some rows have group=center\_level, then the glyph is drawn as a bullseye.

**Examples**

```
plot_pairwise(pair_cor(iris))
plot_pairwise(pairwise_scores(iris,by="Species"))
```

---

plot\_pairwise\_linear *Pairwise plot in a linear layout*

---

**Description**

Plots the calculated measures of association among different variable pairs for a dataset in a linear layout.

**Usage**

```
plot_pairwise_linear(
  scores,
  pair_order = "seriate_max",
  geom = c("tile", "point"),
  add_lines = FALSE,
  score_limits = NULL,
  na.value = "grey80",
  interactive = FALSE
)
```

**Arguments**

scores	A tibble with the calculated association measures for the matrix plot. Either of class pairwise or identical in structure to object of class pairwise.
pair_order	The variable pair order to be used. The default NULL means pairs are in order of their first appearance in scores. A value of "seriate_max" means pairs are in order of maximum absolute scores. A value of "seriate_max_diff" means pairs are in order of maximum scores difference.
geom	The geom to be used. Should be "point" or "tile".
add_lines	When geom= "point" is used, should the points be connected by lines? Defaults to FALSE.
score_limits	a numeric vector of length specifying the limits of the scale.
na.value	used for geom_tile with a value of NA
interactive	defaults to FALSE

**Value**

A girafe object if `interactive==TRUE`, otherwise a `ggplot2`.

**Examples**

```
plot_pairwise_linear(pairwise_scores(iris))  
plot_pairwise_linear(pairwise_scores(iris,by="Species"))  
plot_pairwise_linear(pairwise_multi(iris), geom="point")
```



# Index

## \* datasets

pair\_methods, 13

ace, 8  
ace\_cor, 2  
as.matrix.pairwise, 3  
as.pairwise (pairwise), 4

ContCoef, 9

dcor2d, 11

GoodmanKruskalGamma, 12  
GoodmanKruskalTau, 13

KendallTauA, 18  
KendallTauB, 19  
KendallW, 20

maxNMI, 15  
mine, 14

pair\_ace, 7  
pair\_cancor, 8  
pair\_chi, 9  
pair\_control, 9  
pair\_control(), 6  
pair\_cor, 10  
pair\_dcor, 11  
pair\_gkGamma, 12  
pair\_gkTau, 13  
pair\_methods, 11, 13  
pair\_mine, 14  
pair\_nmi, 15  
pair\_polychor, 15  
pair\_polyserial, 16  
pair\_scagnostics, 17  
pair\_tauA, 18  
pair\_tauB, 18  
pair\_tauC, 19  
pair\_tauW, 20

pair\_uncertainty, 21  
pairwise, 4  
pairwise\_by, 5  
pairwise\_multi, 5  
pairwise\_scores, 6  
plot.pairwise, 21  
plot\_pairwise, 22  
plot\_pairwise\_linear, 23  
polychor, 16  
polyserial, 16

scagnostics, 17  
StuartTauC, 19

UncertCoef, 21