Package ‘ggvegan’

May 8, 2026

Type Package

Title 'ggplot2' Plots for the 'vegan' Package

Version 0.2.1

Depends R (>= 4.1.0)

Imports vegan (>= 2.3-0), dplyr, tibble, vctrs, ggplot2, grid, tidyr, ggrepel, generics

Suggests testthat (>= 3.0.0), vdiffr (>= 1.0.0), permute, withr

Description Functions to produce 'ggplot2'-based plots of objects produced by functions in the 'vegan' package. Provides 'fortify()', 'autoplot()', and 'tidy()' methods for many of 'vegan''s functions. The aim of 'ggvegan' is to make it easier to work within the 'tidyverse' with 'vegan'.

License GPL-2

ByteCompile true

RoxygenNote 7.3.3

Encoding UTF-8

Config/testthat/edition 3

Language en-GB

NeedsCompilation no

Author Gavin L. Simpson [aut, cre] (ORCID: <`https://orcid.org/0000-0002-9084-8413`>), Jari Oksanen [aut] (ORCID: <`https://orcid.org/0000-0001-7102-9626`>), Didzis Elferts [ctb] (ORCID: <`https://orcid.org/0000-0002-9401-1231`>)

Maintainer Gavin L. Simpson <`ucfagls@gmail.com`>

Repository CRAN

Date/Publication 2026-02-27 20:30:02 UTC

Contents

`add_biplot_arrows` *Adds a biplot arrow layer to an existing plot*

Description

Adds a biplot arrow layer to an existing plot

Usage

```
add_biplot_arrows(object, plt, vars, arrow.col = "navy")
```

Arguments

`object`	an ordination object.
`plt`	a ggplot object.
`vars`	character; length 2 vector of dimension names.
`arrow.col`	colour specification for biplot arrows and their labels.

`add_biplot_centroids` *Adds a biplot arrow layer to an existing plot*

Description

Adds a biplot arrow layer to an existing plot

Usage

```
add_biplot_centroids(object, plt, vars, arrow.col = "navy")
```

Arguments

`object`	an ordination object.
`plt`	a ggplot object.
`vars`	character; length 2 vector of dimension names.
`arrow.col`	colour specification for biplot arrows and their labels.

`add_spp_site_scores` *Adds species and site score layers to an existing plot*

Description

Adds species and site score layers to an existing plot

Usage

```
add_spp_site_scores(object, plt, vars, geom, draw_list, arrows)
```

Arguments

`object`	an ordination object.
`plt`	a ggplot object.
`vars`	character; length 2 vector of dimension names.
`geom`	character; vector of length 1 or 2 indicating which geom will be used for the species or site scores.
`draw_list`	logical; vector of types of scores indicating which are available and requested for plotting.
`arrows`	logical; length 1 vector indicating if species scores should be drawn using arrows.

`arrow_mul` *Scale Vectors to Data*

Description

Scale vector arrows to `fill` proportion of the data.

Usage

```
arrow_mul(arrows, data, at = c(0, 0), fill = 0.75)
```

Arguments

`arrows`	a two-column matrix-like object containing coordinates for the arrows/vectors on x and y axes.
`data`	a two-column matrix-like object containing coordinates of the data on the x and y axes.
`at`	numeric vector of length 2; location of the origin of the arrows.
`fill`	numeric; what proportion of the range of the data to fill

Value

a numeric multiplier that will scale the arrows

Author(s)

Gavin L. Simpson

`autoplot.anosim` *ggplot-based plot for objects of class* `"anosim"`

Description

Produces a multi-layer ggplot object representing the output of objects produced by `vegan::anosim()`.

Usage

```
## S3 method for class 'anosim'
autoplot(
  object,
  notch = TRUE,
  varwidth = TRUE,
  xlab = NULL,
  ylab = NULL,
  title = "Analysis of similarities",
  subtitle = NULL,
  caption = bquote(R == .(r) * "," ~ P == .(p)),
  ...
)
```

Arguments

`object`	an object of class `"anosim"`, the result of a call to `vegan::anosim()`.
`notch`	logical; make notched (default) or standard box plot?
`varwidth`	logical; make box width proportional to the square-root of the number of observations in the group (default)?
`xlab`	character; label for the x-axis.
`ylab`	character; label for the y-axis.
`title`	character; title for the plot.
`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`...`	additional arguments passed to other methods.

Value

A ggplot object.

Author(s)

Didzis Elferts. Modifications by Gavin L. Simpson.

Examples

```
library("vegan")

data(dune)
data(dune.env)
dune.dist <- vegdist(dune)
dune.ano <- with(dune.env, anosim(dune.dist, Management))

autoplot(dune.ano, notch = FALSE)
```

`autoplot.cca` *ggplot-based plot for objects of class* `"cca"`

Description

Produces a multi-layer ggplot object representing the output of objects produced by `vegan::cca()` or `vegan::capscale()`.

Usage

```
## S3 method for class 'cca'
autoplot(
  object,
  axes = c(1, 2),
  geom = c("point", "text"),
  layers = c("species", "sites", "biplot", "centroids"),
  legend.position = "right",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ylab = NULL,
  xlab = NULL,
  arrow.col = "navy",
  ...
)
```

Arguments

`object`	an object of class `"cca"`, the result of a call to `vegan::cca()` or `vegan::capscale()`.
`axes`	numeric; which axes to plot, given as a vector of length 2.
`geom`	character; which geom to use for the species (variables) and sites (samples) layers. A vector of length 2; if a vector of length 1, `geom` is extended to the required length.

`layers`	character; which scores to plot as layers
`legend.position`	character or two-element numeric vector; where to position the legend. See `ggplot2::theme()` for details. Use `"none"` to not draw the legend.
`title`	character; subtitle for the plot.
`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`ylab`	character; label for the y-axis.
`xlab`	character; label for the x-axis.
`arrow.col`	colour specification for biplot arrows and their labels.
`...`	Additional arguments passed to the `fortify()` method.

Details

TODO

Value

Returns a ggplot object.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")
data(dune)
data(dune.env)

sol <- cca(dune ~ A1 + Management, data = dune.env)
autoplot(sol)
```

`autoplot.dbrda` *ggplot-based plot for objects of class* `"dbrda"`

Description

Produces a multi-layer ggplot object representing the output of objects produced by `vegan::dbrda()`.

Usage

```
## S3 method for class 'dbrda'
autoplot(
  object,
  axes = c(1, 2),
  geom = c("point", "text"),
  layers = c("sites", "biplot", "centroids"),
  legend.position = "none",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ylab = NULL,
  xlab = NULL,
  const = NULL,
  arrow.col = "navy",
  ...
)
```

Arguments

`object`	an object of class `"dbrda"`, the result of a call to `vegan::dbrda()`
`axes`	numeric; which axes to plot, given as a vector of length 2.
`geom`	character; which geom to use for the site (sample) scores. One of `"point"`, or `"text"`.
`layers`	character; which scores to plot as layers
`legend.position`	character or two-element numeric vector; where to position the legend. See `ggplot2::theme()` for details. Use `"none"` to not draw the legend.
`title`	character; subtitle for the plot.
`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`ylab`	character; label for the y-axis.
`xlab`	character; label for the x-axis.
`const`	General scaling constant to `dbrda` scores. See `vegan::scores.rda()` for details.
`arrow.col`	colour specification for biplot arrows and their labels.
`...`	Additional arguments passed to the `fortify()` method.

Details

TODO

Value

Returns a ggplot object.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(dune, dune.env)

dune_dbrda <- dbrda(
  dune ~ A1 + Moisture + Use + Management,
  data = dune.env
)
autoplot(dune_dbrda)
```

`autoplot.decorana` *ggplot-based plot for objects of class* `"decorana"`

Description

Produces a multi-layer ggplot object representing the output of objects produced by `vegan::decorana()`.

Usage

```
## S3 method for class 'decorana'
autoplot(
  object,
  axes = c(1, 2),
  geom = c("point", "text"),
  layers = c("sites", "species"),
  legend.position = "right",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ylab,
  xlab,
  ...
)
```

Arguments

`object`	an object of class `"decorana"`, the result of a call to `vegan::decorana()`.
`axes`	numeric; which axes to plot, given as a vector of length 2.
`geom`	character; which geom to use for the species (variables) and sites (samples) layers. A vector of length 2; if a vector of length 1, `geom` is extended to the required length.
`layers`	character; which scores to plot as layers

`legend.position`	character or two-element numeric vector; where to position the legend. See `ggplot2::theme()` for details. Use `"none"` to not draw the legend.
`title`	character; subtitle for the plot.
`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`ylab`	character; label for the y-axis.
`xlab`	character; label for the x-axis.
`...`	Additional arguments passed to `fortify.decorana()`.

Details

TODO

Value

Returns a ggplot object.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")
data(dune)

sol <- decorana(dune)
autoplot(sol)
autoplot(sol, layers = "species", geom = "text")
```

`autoplot.envfit` *ggplot-based plot for* `envfit` *objects*

Description

Produces a multi-layer ggplot object representing the output of objects produced by `vegan::envfit()`.

Usage

```
## S3 method for class 'envfit'
autoplot(
  object,
  geom = c("label", "text", "label_repel", "text_repel"),
  line.col = "black",
  xlab = NULL,
  ylab = NULL,
```

```
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ...
)
```

Arguments

`object`	an object of class `"envfit"`, the result of a call to `vegan::envfit()`.
`geom`	character; which geom to use to label vectors and factor centroids.
`line.col`	colour with which to draw vectors.
`xlab`	character; label for the x-axis.
`ylab`	character; label for the y-axis.
`title`	character; subtitle for the plot.
`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`...`	additional arguments passed to `ggplot2::fortify()`.

Value

A ggplot object.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")
data(varespec, varechem)
ord1 <- metaMDS(varespec)
fit1 <- envfit(ord1, varechem, perm = 199)

autoplot(fit1, geom = 'label_repel')

data(dune, dune.env)
ord2 <- cca(dune)
fit2 <- envfit(ord2 ~ Moisture + A1, dune.env, perm = 199)

autoplot(fit2)
```

`autoplot.fisherfit` *Plot Fisher's log-series*

Description

Draws a bar plot of species rank abundance with Fisher's log-series superimposed.

Usage

```
## S3 method for class 'fisherfit'
autoplot(
  object,
  show.fitted = TRUE,
  xlab = "Abundance",
  ylab = "Number of Species",
  title = "Fisher's log-series distribution",
  bar.col = NA,
  bar.fill = "grey35",
  line.col = "red",
  linewidth = 0.7,
  subtitle = NULL,
  caption = NULL,
  ...
)
```

Arguments

`object`	an object of class `vegan::fisherfit()`.
`show.fitted`	logical; should the estimated distribution also be plotted?
`xlab`	character; label for the x axis.
`ylab`	character; label for the y axis.
`title`	character; subtitle for the plot.
`bar.col`	colour for the bar outlines. The default, `NA`, does not draw outlines around bars.
`bar.fill`	fill colour for the bars.
`line.col`	colour for Fisher's log-series curve.
`linewidth`	numeric; linewidth aesthetic for the log-series curve.
`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`...`	additional arguments passed to other methods.

Value

A ggplot object.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(BCI)
mod <- fisherfit(BCI[5,])
autoplot(mod)
```

`autoplot.isomap` *ggplot-based plot for objects of class* `"isomap"`

Description

Produces a multi-layer ggplot object representing the output of objects produced by `vegan::isomap()`.

Usage

```
## S3 method for class 'isomap'
autoplot(
  object,
  axes = c(1, 2),
  geom = c("point", "text"),
  network = TRUE,
  line.col = "grey85",
  linewidth = 0.7,
  xlab = NULL,
  ylab = NULL,
  title = "Isometric feature mapping",
  subtitle = NULL,
  caption = NULL,
  ...
)
```

Arguments

`object`	an object of class `"isomap"`, the result of a call to `vegan::isomap()`.
`axes`	numeric; which axes to plot, given as a vector of length 2.
`geom`	character; which geom to use for the MDS scores layer.
`network`	logical; should the edges of the ISOMAP network be drawn?
`line.col`	colour with which to draw the network edges.
`linewidth`	numeric; linewidth aesthetic for the log-series curve.
`xlab`	character; label for the x-axis.

`ylab`	character; label for the y-axis.
`title`	character; subtitle for the plot.
`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`...`	additional arguments passed to other methods.

Value

A ggplot object.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(BCI)
dis <- vegdist(BCI)
ord <- isomap(dis, k = 3)

autoplot(ord)

autoplot(ord, geom = "text")
```

`autoplot.metaMDS` *ggplot-based plot for objects of class* `"metaMDS"`

Description

Produces a multi-layer ggplot object representing the output of objects produced by `vegan::metaMDS()`.

Usage

```
## S3 method for class 'metaMDS'
autoplot(
  object,
  geom = c("point", "text"),
  layers = c("species", "sites"),
  legend.position = "right",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ylab,
  xlab,
  ...
)
```

Arguments

`object`	an object of class `"metaMDS"`, the result of a call to `vegan::metaMDS()`.
`geom`	character; which geom to use for the species (variables) and sites (samples) layers. A vector of length 2; if a vector of length 1, `geom` is extended to the required length.
`layers`	character; which scores to plot as layers
`legend.position`	character or two-element numeric vector; where to position the legend. See `ggplot2::theme()` for details. Use `"none"` to not draw the legend.
`title`	character; subtitle for the plot.
`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`ylab`	character; label for the y-axis.
`xlab`	character; label for the x-axis.
`...`	Additional arguments passed to `\link{fortify.metaMDS}`.

Details

TODO

Value

Returns a ggplot object.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(dune)

sol <- metaMDS(dune)
autoplot(sol)
```

`autoplot.permustats` *Autoplot Graphics for vegan permustats Objects*

Description

Alternatives for **lattice** graphics functions `vegan::densityplot.permustats()`, `vegan::densityplot.permustats()` and `vegan::boxplot.permustats()`.

Usage

```
## S3 method for class 'permustats'
autoplot(
  object,
  plot = c("box", "violin", "density", "qqnorm"),
  scale = FALSE,
  facet = FALSE,
  gg.params = list(),
  ...
)
```

Arguments

`object`	object from `vegan::permustats()`.
`plot`	character; the type of plot, or a geom from **ggplot2**.
`scale`	logical; use standardized effect sizes (SES)?
`facet`	logical; should the plot be faceted by `term`?
`gg.params`	list; arguments passed to function drawing the box-like object. Depending on argument `plot` the parameters are passed to `ggplot2::geom_boxplot()`, `ggplot2::geom_violin()`, `ggplot2::geom_density()` or `ggplot2::geom_qq()`.
`...`	Other parameters passed to functions (ignored).

Details

Function `fortify()` returns a data frame with variables `permutations` (numeric) and `term` (factor labelling the permutation). The result of `fortify()` can be used to custom build diagnostic plots. `autoplot()` provides basic plots of `vegan::permustats()` objects with limited flexibility.

Value

Returns a ggplot object.

Examples

```
library("vegan")

data(dune, dune.env)
mod <- cca(dune ~ A1 + Management + Moisture, dune.env)
(ano <- anova(mod, by = "onedf"))
pstat <- permustats(ano)
head(fortify(pstat))
autoplot(pstat, "box")


if (requireNamespace("ggplot2")) {
  library("ggplot2")

  # avoid overplotting x-axis text
  autoplot(pstat, "violin") +
    scale_x_discrete(guide = guide_axis(n.dodge = 2))

  autoplot(pstat, "qqnorm", facet = TRUE) +
    geom_qq_line()
}

autoplot(pstat, "density", facet = TRUE)
```

`autoplot.poolaccum` *ggplot-based plot for objects of class* "poolaccum"

Description

Produces a multi-layer ggplot object representing the output of objects produced by `vegan::poolaccum()`.

Usage

```
## S3 method for class 'poolaccum'
autoplot(
  object,
  facet = TRUE,
  ribbon = facet,
  ncol = NULL,
  ribbon.alpha = 0.3,
  xlab = "Size",
  ylab = "Richness",
  title = "Accumulated species richness",
  subtitle = NULL,
  caption = NULL,
  ...
)
```

Arguments

`object`	an object of class `"poolaccum"`, the result of a call to `vegan::poolaccum()`.
`facet`	logical; should extrapolated richness estimators be shown in separate facets?
`ribbon`	logical; show the quantile-based uncertainty interval? Uses `ggplot2::geom_ribbon()` for plot.
`ncol`	numeric; if faceting the plot, how many columns to use. Default arguments will result in three columns.
`ribbon.alpha`	numeric; alpha transparency used for the uncertainty interval. Passed to the `alpha` aesthetic of `ggplot2::geom_ribbon()`.
`xlab`	character; label for the x axis.
`ylab`	character; label for the y axis.
`title`	character; title for the plot.
`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`...`	additional arguments passed to `fortify.poolaccum()`.

Value

A ggplot object.

Author(s)

Didzis Elferts & Gavin L. Simpson

Examples

```
library("vegan")

data(BCI)
pool <- poolaccum(BCI)

autoplot(pool)

## Turn off facetting; turns off ribbon too
autoplot(pool, facet = FALSE)
```

`autoplot.prc` *ggplot-based plot for objects of class* `"prc"`

Description

Produces a multi-layer ggplot object representing the output of objects produced by `vegan::prc()`.

Usage

```
## S3 method for class 'prc'
autoplot(
  object,
  select,
  xlab,
  ylab,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  legend.position = "top",
  ...
)
```

Arguments

`object`	an object inheriting from class `"prc"`, the result of a call to `vegan::prc()`.
`select`	a logical vector where `TRUE` selects and `FALSE` deselects species. Alternatively a numeric vector that contains the indices selecting species. Note that these are with respect to the original species matrix, **not** the fortified object.
`xlab`	character; label for the x-axis
`ylab`	character; label for the y-axis
`title`	character; subtitle for the plot
`subtitle`	character; subtitle for the plot
`caption`	character; caption for the plot
`legend.position`	character; position for the legend grob. See argument `legend.position` in function `ggplot2::theme()`.
`...`	Additional arguments passed to `\link{fortify.prc}`.

Details

TODO

Value

Returns a ggplot object.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(pyrifos)
week <- gl(11, 12, labels=c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24))
dose <- factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11))

## Fit PRC model
mod <- prc(pyrifos, dose, week)

## plot
want <- colSums(pyrifos) >= 300
autoplot(mod, select = want)
```

`autoplot.prestonfit` *Plot Preston's log-normal model of species abundance*

Description

Draws a bar plot of species rank abundance with Preston's log-normal model superimposed.

Usage

```
## S3 method for class 'prestonfit'
autoplot(
  object,
  show.fitted = TRUE,
  xlab = "Abundance",
  ylab = "Number of Species",
  title = "Preston's lognormal distribution",
  subtitle = NULL,
  caption = NULL,
  bar.col = NA,
  bar.fill = "grey35",
  line.col = "red",
  linewidth = 0.7,
  ...
)
```

Arguments

`object`	an object of class `vegan::prestonfit()`.
`show.fitted`	logical; should the estimated distribution also be plotted?
`xlab`	character; label for the x axis.
`ylab`	character; label for the y axis.
`title`	character; title for the plot.

`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`bar.col`	colour for the bar outlines. The default, ‘NA“, does not draw outlines around bars.
`bar.fill`	fill colour for the bars.
`line.col`	colour for Preston’s log-normal curve.
`linewidth`	numeric; size aesthetic for the log-normal curve.
`...`	additional arguments passed to other methods.

Value

A ggplot object.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(BCI)
pfit <- prestonfit(colSums(BCI))
autoplot(pfit)
```

`autoplot.rda` *ggplot-based plot for objects of class* `'rda'`

Description

Produces a multi-layer ggplot object representing the output of objects produced by `vegan::rda()`.

Usage

```
## S3 method for class 'rda'
autoplot(
  object,
  axes = c(1, 2),
  geom = c("point", "text"),
  layers = c("species", "sites", "biplot", "centroids"),
  arrows = TRUE,
  legend.position = "none",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ylab = NULL,
```

```
    xlab = NULL,
    const = NULL,
    arrow.col = "navy",
    ...
)
```

Arguments

`object`	an object of class `"rda"`, the result of a call to `vegan::rda()`
`axes`	numeric; which axes to plot, given as a vector of length 2.
`geom`	character; which geom to use for the species (variables) and sites (samples) layers. A vector of length 2; if a vector of length 1, `geom` is extended to the required length.
`layers`	character; which scores to plot as layers
`arrows`	logical; represent species (variables) using vectors?
`legend.position`	character or two-element numeric vector; where to position the legend. See `ggplot2::theme()` for details. Use `"none"` to not draw the legend.
`title`	character; subtitle for the plot.
`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`ylab`	character; label for the y-axis.
`xlab`	character; label for the x-axis.
`const`	General scaling constant to `rda` scores. See `vegan::scores.rda()` for details.
`arrow.col`	colour specification for biplot arrows and their labels.
`...`	Additional arguments passed to the `fortify()` method.

Details

TODO

Value

Returns a ggplot object.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(dune)

pca <- rda(dune)
```

```
autoplot(pca)

## Just the species scores
autoplot(pca, layers = "species")
```

`autoplot.vegan_pco` *ggplot-based plot for objects of class* `"vegan_pco"`

Description

Produces a multi-layer ggplot object representing the output of objects produced by `vegan::pco()`.

Usage

```
## S3 method for class 'vegan_pco'
autoplot(
  object,
  geom = "point",
  legend.position = "right",
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ylab = NULL,
  xlab = NULL,
  ...
)
```

Arguments

`object`	an object of class `"vegan_pco"`, the result of a call to `vegan::pco()`.
`geom`	character; which geom to use for the site (sample) scores. One of `"point"`, or `"text"`.
`legend.position`	character or two-element numeric vector; where to position the legend. See `ggplot2::theme()` for details. Use `"none"` to not draw the legend.
`title`	character; subtitle for the plot.
`subtitle`	character; subtitle for the plot.
`caption`	character; caption for the plot.
`ylab`	character; label for the y-axis.
`xlab`	character; label for the x-axis.
`...`	Additional arguments passed to the `fortify()` method.

Details

TODO

Value

Returns a ggplot object.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(dune)

sol <- pco(dune)
autoplot(sol)
```

`check_user_layers` *Check user-supplied layers against list of valid layers*

Description

Check user-supplied layers against list of valid layers

Usage

```
check_user_layers(user, valid, message = FALSE)
```

Arguments

`user`	character; vector of user supplied layer names.
`valid`	character; vector of valid layer names.
`message`	logical; should a message be raised in the case of invalid user-supplied layer names.

`fortify.anosim` *Fortify analysis of similarities (ANOSIM) results*

Description

Prepares a fortified version of results from `vegan::anosim()` objects.

Usage

```
## S3 method for class 'anosim'
fortify(model, data, ...)

## S3 method for class 'anosim'
tidy(x, data, ...)
```

Arguments

`model, x`	an object of class `vegan::anosim()`.
`data`	original data set. Currently ignored.
`...`	additional arguments for other methods. Currently ignored.

Value

A data frame with columns `Rank` and `Class` with ranks of dissimilarity.

Author(s)

Original implementation by Didzis Elferts. Modification to tibbles by Gavin L. Simpson.

Examples

```
library("vegan")

library("ggplot2")
data(dune, dune.env)
dune.dist <- vegdist(dune)
dune.ano <- with(dune.env, anosim(dune.dist, Management))

df <- fortify(dune.ano)

ggplot(df, aes(x = class, y = rank)) +
    geom_boxplot(notch = FALSE, varwidth = TRUE)
```

`fortify.cca` *Fortify a* `"cca"` *object.*

Description

Fortifies an object of class `"cca"` to produce a data frame of the selected axis scores in long format, suitable for plotting with `ggplot2::ggplot()`.

Usage

```
## S3 method for class 'cca'
fortify(model, data, axes = 1:6, layers = c("sp", "wa", "lc", "bp", "cn"), ...)

## S3 method for class 'cca'
tidy(x, data, axes = 1:6, layers = c("sp", "wa", "lc", "bp", "cn"), ...)
```

Arguments

`model, x`	an object of class `"cca"`, the result of a call to `vegan::cca()`, `vegan::rda()`, or `vegan::capscale()`.
`data`	currently ignored.
`axes`	numeric; which axes to extract scores for.
`layers`	character; the scores to extract in the fortified object. Passed to `display` in the respective `vegan::scores()` method.
`...`	additional arguments passed to `vegan::scores.cca()`.

Details

TODO

Value

A data frame (tibble) in long format containing the ordination scores. The first two components are `score` (the type of score in each row) and `label` (the text label to use on plots for this row). The remaining columns are the extracted ordination axis scores.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(dune)
data(dune.env)

sol <- cca(dune ~ A1 + Management, data = dune.env)
head(fortify(sol))
```

`fortify.dbrda` *Fortify a* `"dbrda"` *object.*

Description

Fortifies an object of class `"dbrda"` to produce a data frame of the selected axis scores in long format, suitable for plotting with `ggplot2::ggplot()`.

Usage

```
## S3 method for class 'dbrda'
fortify(
  model,
  data = NULL,
  axes = 1:6,
  layers = c("wa", "lc", "bp", "reg", "cn"),
  const = NULL,
  ...
)

## S3 method for class 'dbrda'
tidy(
  x,
  data = NULL,
  axes = 1:6,
  layers = c("wa", "lc", "bp", "reg", "cn"),
  const = NULL,
  ...
)
```

Arguments

`model, x`	an object of class `"dbrda"`, the result of a call to `vegan::dbrda()`.
`data`	currently ignored.
`axes`	numeric; which axes to extract scores for.
`layers`	character; the scores to extract in the fortified object. Passed to `display` in the respective `vegan::scores()` method.
`const`	NULL; General scaling constant to RDA scores. See `vegan::scores.rda()` for the details.
`...`	additional arguments passed to `vegan::scores.rda()`.

Details

TODO

Value

A data frame (tibble) in long format containing the ordination scores. The first two components are `score` (the type of score in each row) and `label` (the text label to use on plots for this row). The remaining columns are the extracted ordination axis scores.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(dune)
data(dune.env)

sol <- dbrda(dune ~ A1 + Management, data = dune.env)
head(fortify(sol))
```

`fortify.decorana` *Fortify a* `"decorana"` *object.*

Description

Fortifies an object of class `"decorana"` to produce a data frame of the selected axis scores in long format, suitable for plotting with `ggplot2::ggplot()`.

Usage

```
## S3 method for class 'decorana'
fortify(model, data, axes = 1:4, layers = c("sites", "species"), ...)

## S3 method for class 'decorana'
tidy(x, data, axes = 1:4, layers = c("sites", "species"), ...)
```

Arguments

`model, x`	an object of class `"decorana"`, the result of a call to `vegan::decorana()`.
`data`	currently ignored.
`axes`	numeric; which axis scores are required?
`layers`	character; the scores to extract in the fortified object.
`...`	additional arguments passed to `vegan::scores.decorana()`.

Details

TODO

Value

A data frame in long format containing the ordination scores. The first two components are the axis scores.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(dune)

sol <- decorana(dune)
head(fortify(sol))
head(fortify(sol, layers = "species"))
```

`fortify.envfit` *Fortify method for* `envfit` *objects*

Description

Produces a tidy data frame from the results of an `vegan::envfit()` object.

Usage

```
## S3 method for class 'envfit'
fortify(model, data, ...)

## S3 method for class 'envfit'
tidy(x, data, ...)
```

Arguments

`model, x`	an object of class `envfit`, the result of a call to `vegan::envfit()`.
`data`	additional data to augment the `envfit` results. Currently ignored.
`...`	arguments passed to `vegan::scores.envfit()`.

Value

A data frame with columns `label`, `type`, containing the label for, and whether each row refers to, the fitted vector or factor. Remaining variables are coordinates on the respective ordination axes returned by `vegan::scores.envfit()`.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(varespec, varechem)
ord <- metaMDS(varespec)
fit <- envfit(ord, varechem, perm = 199)

fortify(fit)

data(dune, dune.env)
ord <- ca(dune)
fit <- envfit(ord ~ Moisture + A1, dune.env, perm = 199)

fortify(fit)
```

`fortify.fisherfit` *Tidied species rank abundance data and estimated Fisher's log-series*

Description

Prepares a data frame of results from a `vegan::fisherfit()`) object suitable for plotting with `ggplot2::ggplot()`).

Usage

```
## S3 method for class 'fisherfit'
fortify(model, data, ...)

## S3 method for class 'fisherfit'
tidy(x, data, ...)
```

Arguments

`model, x`	an object of class `vegan::fisherfit()`).
`data`	original data set. Currently ignored.
`...`	other arguments pass to methods. Currently ignored.

Value

A data frame with columns `'Rank'` and `'Abundance'`. Additionally, Fisher's α and the nuisance parameter are returned as attributes `'alpha'` and `'k'` respectively.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(BCI)
mod <- fisherfit(BCI[5,])
head(fortify(mod))
```

`fortify.isomap` *Fortify isometric feature mapping results*

Description

Prepares fortified versions of results from `vegan::isomap()` objects.

Usage

```
## S3 method for class 'isomap'
fortify(model, data = NULL, axes = NULL, what = c("sites", "network"), ...)

## S3 method for class 'isomap'
tidy(x, data = NULL, axes = NULL, what = c("sites", "network"), ...)
```

Arguments

`model, x`	an object of class `vegan::isomap()`.
`data`	data.frame; additional data to be added to the fortified object.
`axes`	numeric; which axes to return. For `what = "network"` this must be of length two only.
`what`	character; what aspect of the results should be fortified? `"sites"` returns to ordination scores from the multidimensional scaling part of the model. `"network"` returns the coordinates for edges joining points.
`...`	additional arguments for other methods. Currently ignored.

Details

Two different objects can be created from the results of an `vegan::isomap()` object. The first is the standard scores representation of fortified ordinations in vegan, which results in a wide data frame where rows contain observations and column the coordinates of observations on the MDS axes. Because ISOMAP also produces a network or sorts, the coordinates of the edges of the network can also be returned in a tidy format using `what = "network"`.

Value

A data frame. For `what = "sites"`, the data frame contains one variable per dimension of the multidimensional scaling embedding of the dissimilarities. Variables are named `"DimX"` with `"X"` being an integer. An additional variable is `label`, containing a label for each observation. For `what = "network"`, the data frame contains four variables containing the coordinates in the chosen MDS axes for the start and end points of the network edges.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

library("ggplot2")
data(BCI)
dis <- vegdist(BCI)
ord <- isomap(dis, k = 3)

df <- fortify(ord, data = data.frame(Richness = specnumber(BCI)))
net <- fortify(ord, what = "network", axes = 1:2)

ggplot(df, aes(x = dim1, y = dim2)) +
    geom_segment(data = net,
                 aes(x = xfrom, y = yfrom,
                     xend = xto, yend = yto),
                 colour = "grey85", size = 0.8) +
    geom_point(aes(size = Richness)) +
    coord_fixed()
```

`fortify.metaMDS` *Fortify a* `"mataMDS"` *object.*

Description

Fortifies an object of class `"metaMDS"` to produce a data frame of the selected axis scores in long format, suitable for plotting with `ggplot2::ggplot()`.

Usage

```
## S3 method for class 'metaMDS'
fortify(model, data, axes = 1:2, layers = c("sites", "species"), ...)

## S3 method for class 'metaMDS'
tidy(x, data, layers = c("sites", "species"), ...)
```

Arguments

`model, x`	an object of class `"metaMDS"`, the result of a call to `vegan::metaMDS()`.
`data`	currently ignored.
`axes`	numeric; which axes to extract scores for.
`layers`	character; the scores to extract in the fortified object. Passed to `display` in the respective `vegan::scores()` method.
`...`	additional arguments passed to `vegan::scores.metaMDS()`. Note you can't use `display`.

Details

TODO

Value

A data frame in long format containing the ordination scores. The first two components are the axis scores.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(dune)

ord <- metaMDS(dune)
head(fortify(ord))
```

`fortify.permustats` *Fortify permutation statistics*

Description

Fortify permutation statistics

Usage

```
## S3 method for class 'permustats'
fortify(model, data, scale = FALSE, ...)

## S3 method for class 'permustats'
tidy(x, data, scale = FALSE, ...)
```

Arguments

`model, x`	an object of created by `vegan::permustats()`.
`data`	original data set. Currently ignored.
`scale`	logical; return standardized effect sizes (SES)?
`...`	Other parameters passed to functions (ignored).

Value

A tibble with columns `permutations`, and `terms` containing the values of tests statistics under the null hypothesis, and a factor labelling the permutation, respectively.

Examples

```
library("vegan")
data(dune, dune.env, package = "vegan")
mod <- adonis2(dune ~ Management + A1, data = dune.env)
## use permustats
perm <- permustats(mod)
```

`fortify.poolaccum` *Fortify extrapolated species richness of a species pool*

Description

Prepares a fortified version of results from `vegan::poolaccum()` objects.

Usage

```
## S3 method for class 'poolaccum'
fortify(model, data, alpha = 0.05, ...)

## S3 method for class 'poolaccum'
tidy(x, data, alpha = 0.05, ...)
```

Arguments

`model, x`	an object of class `vegan::poolaccum()`.
`data`	original data set. Currently ignored.
`alpha`	level of quantiles for envelopes shown (default 0.05).
`...`	additional arguments passed to `vegan::summary.poolaccum()`, notably `display` to control which indices should be computed.

Value

A data frame with columns `index`, `size`, `richness`, `lower`, `upper`, and `std_dev`, containing the richness index, permutation richness estimator, sample size, upper and lower $1 - \alpha$ quantile interval, and standard deviation of permutation estimates, respectively.

Author(s)

Gavin L. Simpson & Didzis Elferts

Examples

```
library("vegan")
library("ggplot2")
data(BCI)
pool <- poolaccum(BCI)
df <- fortify(pool)
df

ggplot(df, aes(x = size, y = richness, colour = index)) +
    geom_ribbon(aes(ymin = lower, ymax = upper, x = size, fill = index),
                alpha = 0.3, inherit.aes = FALSE) +
    geom_line() +
    facet_wrap(~ index)
```

`fortify.prc` *Fortify a* `"prc"` *object*

Description

Fortifies an object of class `"prc"` to produce a data frame of the selected axis scores in long format, suitable for plotting with `ggplot2::ggplot()`.

Usage

```
## S3 method for class 'prc'
fortify(model, data, scaling = "symmetric", axis = 1, ...)

## S3 method for class 'prc'
tidy(x, data, scaling = "symmetric", axis = 1, ...)
```

Arguments

`model, x`	an object of class `"prc"`, the result of a call to `vegan::prc()`.
`data`	currently ignored.
`scaling`	the desired scaling. See `vegan::scores.cca()` for details.
`axis`	numeric; which PRC axis to extract. Default is `axis = 1`, which is the most generally useful choice.
`...`	additional arguments currently ignored.

Details

TODO

Value

A data frame in long format containing the ordination scores. The first three components are the `Time`, `Treatment`, and associated `Response`. The last two components, `score` and `label` are an indicator factor and a label for the rows for use in plotting.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")
data(pyrifos)
week <- gl(11, 12, labels=c(-4, -1, 0.1, 1, 2, 4, 8, 12, 15, 19, 24))
dose <- factor(rep(c(0.1, 0, 0, 0.9, 0, 44, 6, 0.1, 44, 0.9, 0, 6), 11))
ditch <- gl(12, 1, length=132)

pyrifos_prc <- prc(pyrifos, dose, week)
fortify(pyrifos_prc)
```

`fortify.prestonfit` *Tidied species octave abundance data*

Description

Prepares a data frame of results from a `vegan::prestonfit()` object suitable for plotting with `ggplot2::ggplot()`.

Usage

```
## S3 method for class 'prestonfit'
fortify(model, data, ...)

## S3 method for class 'prestonfit'
tidy(x, data, ...)
```

Arguments

`model, x`	an object of class `vegan::prestonfit()`.
`data`	original data set. Currently ignored.
`...`	other arguments pass to methods. Currently ignored.

Value

A data frame with columns `'Octave'`` and `’Abundance’“.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(BCI)
pfit <- prestonfit(colSums(BCI))
fortify(pfit)
```

`fortify.rda` *Fortify a* `"rda"` *object.*

Description

Fortifies an object of class `"rda"` to produce a data frame of the selected axis scores in long format, suitable for plotting with `ggplot2::ggplot()`.

Usage

```
## S3 method for class 'rda'
fortify(
  model,
  data = NULL,
  axes = 1:6,
  layers = c("sp", "wa", "lc", "bp", "cn"),
  const = NULL,
  ...
)

## S3 method for class 'rda'
tidy(
  x,
  data = NULL,
  axes = 1:6,
  layers = c("sp", "wa", "lc", "bp", "reg", "cn"),
  const = NULL,
  ...
)
```

Arguments

`model, x`	an object of class `"rda"`, the result of a call to `vegan::rda()`.
`data`	currently ignored.
`axes`	numeric; which axes to extract scores for.
`layers`	character; the scores to extract in the fortified object. Passed to `display` in the respective `vegan::scores()` method.
`const`	NULL; General scaling constant to RDA scores. See `vegan::scores.rda()` for the details.
`...`	additional arguments passed to `vegan::scores.rda()`.

Details

TODO

Value

A data frame (tibble) in long format containing the ordination scores. The first two components are `score` (the type of score in each row) and `label` (the text label to use on plots for this row). The remaining columns are the extracted ordination axis scores.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(dune)
data(dune.env)

sol <- rda(dune ~ A1 + Management, data = dune.env)
head(fortify(sol))
```

`fortify.renyiaccum` *Fortify Rényi accumulation curves*

Description

Prepares a fortified version of results from `vegan::renyiaccum()` objects.

Usage

```
## S3 method for class 'renyiaccum'
fortify(model, data, ...)

## S3 method for class 'renyiaccum'
tidy(x, data, ...)
```

Arguments

`model, x`	an object of class `vegan::renyiaccum()`.
`data`	original data set. Currently ignored.
`...`	additional arguments passed to other methods. Ignored in this method.

Value

A data frame (tibble) is returned. What is returned depends on how `vegan::renyiaccum()` was called. If `raw` = `FALSE`, then a data frame with columns `site`, `scale`, `mean`, `std_dev`, `min`, `max`, `q2.5`, and `q97.5`, containing the accumulated sites, the Rényi scale, and summary statistics of the Rényi accumulation curves. An additional column `collector` will be present if `collector` = `TRUE` was used in the `vegan::renyiaccum()` call. If `raw` = `TRUE`, then a data frame with columns `site`, `permutation`, `scale`, and `diversity`, containing the `site` and `permutation` identifiers, Rényi scale, and the Rényi diversity, respectively.

Author(s)

Gavin L. Simpson much rewritten from an original by Didzis Elferts.

Examples

```
library("vegan")
library("ggplot2")
data(BCI)
pool <- renyiaccum(BCI)
df <- fortify(pool)
df

ggplot(df, aes(x = site, y = mean)) +
    geom_ribbon(aes(ymin = q2.5, ymax = q97.5, x = site),
                alpha = 0.2, inherit.aes = FALSE) +
    geom_line() +
    facet_wrap(~ scale)
```

`fortify.vegan_pco` *Fortify a* "vegan_pco" *object.*

Description

Fortifies an object of class "vegan_pco" to produce a data frame of the selected axis scores in long format, suitable for plotting with `ggplot2::ggplot()`.

Usage

```
## S3 method for class 'vegan_pco'
fortify(model, data = NULL, axes = 1:6, ...)

## S3 method for class 'vegan_pco'
tidy(x, data = NULL, axes = 1:6, const = NULL, ...)
```

Arguments

`model, x`	an object of class `"vegan_pco"`, the result of a call to `vegan::pco()`.
`data`	currently ignored.
`axes`	numeric; which axes to extract scores for.
`...`	additional arguments passed to `vegan::scores.rda()`.
`const`	NULL; General scaling constant to RDA scores. See `vegan::scores.rda()` for the details.

Details

TODO

Value

A data frame (tibble) in long format containing the ordination scores. The first two components are `score` (the type of score in each row) and `label` (the text label to use on plots for this row). The remaining columns are the extracted ordination axis scores.

Author(s)

Gavin L. Simpson

Examples

```
library("vegan")

data(dune)

sol <- pco(dune)
head(fortify(sol))
```

`geom_ordi_arrow` *Add a biplot arrow layer to an ordiggplot*

Description

Add a biplot arrow layer to an ordiggplot

Usage

```
geom_ordi_arrow(
  score,
  data,
  text = TRUE,
  box = FALSE,
  arrow.params = list(),
  text.params = list(),
  ...
)
```

Arguments

`score`	Ordination score to be added to the plot.
`data`	Alternative data to the function that will be used instead of `score`.
`text`	Add text labels to the plot.
`box`	Draw a box behind the text (logical).
`arrow.params`, `text.params`	Parameters to modify arrows or their text labels.
`...`	other arguments passed to `ggplot2::geom_segment()`, `ggplot2::geom_label()`, or `ggplot2::geom_text()`

Value

Returns a ggplot2 layer or a list of such layers: a `"LayerInstance"` object that inherits from classes `"Layer"`, `"ggproto"`, and `"gg"`.

`geom_ordi_axis` *Crosshair for axes in eigenvector methods*

Description

Crosshair for axes in eigenvector methods

Usage

```
geom_ordi_axis(lty = 3, ...)
```

Arguments

`lty`	Linetype.
`...`	other arguments passed to `ggplot2::geom_hline()` and `ggplot2::geom_vline()`

Value

Returns a ggplot2 layer or a list of such layers: a `"LayerInstance"` object that inherits from classes `"Layer"`, `"ggproto"`, and `"gg"`.

`geom_ordi_label` *Add a label layer to an ordiggplot*

Description

Add a label layer to an ordiggplot

Usage

```
geom_ordi_label(score, data, ...)
```

Arguments

`score`	Ordination score to be added to the plot.
`data`	Alternative data to the function that will be used instead of `score`.
`...`	other arguments passed to `ggplot2::geom_label()`

Value

Returns a ggplot2 layer or a list of such layers: a `"LayerInstance"` object that inherits from classes `"Layer"`, `"ggproto"`, and `"gg"`.

`geom_ordi_point` *Add a point layer to an ordiggplot*

Description

Add a point layer to an ordiggplot

Usage

```
geom_ordi_point(score, data, ...)
```

Arguments

`score`	Ordination score to be added to the plot.
`data`	Alternative data to the function that will be used instead of `score`.
`...`	other arguments passed to `ggplot2::geom_point()`

Value

Returns a ggplot2 layer or a list of such layers: a `"LayerInstance"` object that inherits from classes `"Layer"`, `"ggproto"`, and `"gg"`.

`geom_ordi_text` *Add a text layer to an ordiggplot*

Description

Add a text layer to an ordiggplot

Usage

```
geom_ordi_text(score, data, ...)
```

Arguments

`score`	Ordination score to be added to the plot.
`data`	Alternative data to the function that will be used instead of `score`.
`...`	other arguments passed to `ggplot2::geom_text()`

Value

Returns a ggplot2 layer or a list of such layers: a `"LayerInstance"` object that inherits from classes `"Layer"`, `"ggproto"`, and `"gg"`.

`get_dimension_names` *Extract the names of the dimensions to plot as a character vector*

Description

Find the character vector of names for the two dimensions of data to be plotted.

Usage

```
get_dimension_names(object)
```

Arguments

`object`	a fortified ordination object.

Value

A length 2 character vector of dimension names.

Author(s)

Gavin L. Simpson

`label_fun` *Adds a label layer using one of a set of common geoms*

Description

Adds labels to a plot using one of `ggplot2::geom_label()`, `ggplot2::geom_text()`, `ggrepel::geom_label_repel()` or `ggrepel::geom_text_repel()`.

Usage

```
label_fun(data, geom = c("label", "text", "label_repel", "text_repel"), vars)
```

Arguments

`data`	data frame; data set to use for the label layer. Must contain a variable label containing the strings to use as labels.
`geom`	character; which geom to use for labelling.
`vars`	character; vector of names of variables to pass to the `x` and `y` aesthetics of the chosen geom.

Author(s)

Gavin L. Simpson

`layer_draw_list` *List of layers to draw for a given vegan object*

Description

List of layers to draw for a given vegan object

Usage

```
layer_draw_list(valid, layers = NULL, available = NULL)
```

Arguments

`valid`	character; vector of valid layer names
`layers`	character; a vector of layer names for `object` that has already been filtered for validity.
`available`	character; what layers are actually available

`ordiggplot` *Create a ggplot Object*

Description

Function `ordiggplot` sets up an ordination graph but draws no result. You can add new graphical elements to this plot with `geom_ordi_*` function of this package, or you can use standard **ggplot2** `geom_*` functions and use `ggscores` as their `data` argument.

Usage

```
ordiggplot(model, axes = c(1, 2), arrow.mul, ...)

ggscores(score)
```

Arguments

`model`	An ordination result object from **vegan**.
`axes`	Two axes to be plotted
`arrow.mul`	Multiplier to arrow length. If missing, the arrow length are adjusted to fit to other scores, but if some score types are not displayed, the arrows may be badly scaled, and manual adjustment can be useful.
`...`	Parameters passed to underlying functions.
`score`	Ordination score to be added to the plot.

Details

The **ggvegan** package has two contrasting approaches to draw ordination plots. The `autoplot` functions (e.g. `autoplot.rda()`, `autoplot.cca()`, and autoplot.metaMDS) draw a complete plot with one command, but the design is hard-coded in the function. However, you can add new elements to the graph.

In contrast, function `ordiggplot()` only sets up an ordination plot, and does not draw anything. It allows you to add layers to the plot one by one with full flexibility of the **ggplot2** functions. There are some specific functions `geom_ordi_*` functions that are similar as similarly named `geom_*` functions. For these you need to give the type of ordination scores to be added, and in addition, you can give any `geom_*` function arguments to modify the plot. Alternatively, you can use any **ggplot2** function and in its `data` argument use `ggscores()` function to select the data elements for the function.

The `ordiggplot()` function extracts results using `fortify()` functions of this package, and it accepts the arguments of those functions. This allows setting, e.g., the scaling of ordination axes.

Value

Returns a ggplot object.

Examples

```
library("vegan")
library("ggplot2")
data(dune, dune.env, varespec, varechem)
m <- cca(dune ~ Management + A1, dune.env)

## use geom_ordi_* functions
ordiggplot(m) + geom_ordi_axis() +
  geom_ordi_point("sites") +
  geom_ordi_text("species", col = "darkblue",
                 mapping = aes(fontface = "italic")) +
  geom_ordi_label("centroids") +
  geom_ordi_arrow("biplot")

## use ggscores + standard geom_* functions
ordiggplot(m, scaling = "sites") +
  geom_point(data = ggscores("sites")) +
  geom_text(data = ggscores("species"),
            mapping = aes(fontface = "italic")) +
  geom_label(data = ggscores("centroids"), fill = "yellow") +
  geom_ordi_arrow("biplot")

## Messy arrow biplot for PCA
m <- rda(dune)
ordiggplot(m) +
  geom_ordi_axis() +
  geom_ordi_point("sites") +
  geom_ordi_arrow("species")
```

`scores_length` *Number of scores*

Description

Returns the number of scores returns in object ‘x“.

Usage

```
scores_length(x)
```

Arguments

`x`	The object whose number of scores is required.

Value

a numeric vector of length 1 with the number of scores.

Author(s)

Gavin L. Simpson

`StatVectorfit` *Add Fitted Vectors to Ordination plots*

Description

Fits arrows to show the direction of fastest increase in continuous environmental variables in ordination space.The arrows are scaled relative to their correlation coefficient, and they can be added to an ordination plot with `geom_ordi_arrow()`.

Usage

```
stat_vectorfit(
  mapping = NULL,
  data = NULL,
  geom = "text",
  position = "identity",
  na.rm = FALSE,
  show.legend = FALSE,
  inherit.aes = TRUE,
  edata = NULL,
  formula = NULL,
  arrow.mul = NULL,
  ...
)
```

Arguments

`mapping`	Set of aesthetic mappings created by `aes()`. If specified and `inherit.aes = TRUE` (the default), it is combined with the default mapping at the top level of the plot. You must supply `mapping` if there is no plot mapping.
`data`	The data to be displayed in this layer. There are three options: If `NULL`, the default, the data is inherited from the plot data as specified in the call to `ggplot()`. A `data.frame`, or other object, will override the plot data. All objects will be fortified to produce a data frame. See `fortify()` for which variables will be created. A `function` will be called with a single argument, the plot data. The return value must be a `data.frame`, and will be used as the layer data. A `function` can be created from a `formula` (e.g. `~ head(.x, 10)`).
`geom`	The geometric object to use to display the data for this layer. When using a `stat_*()` function to construct a layer, the `geom` argument can be used to override the default coupling between stats and geoms. The `geom` argument accepts the following:

	• A `Geom` ggproto subclass, for example `GeomPoint`. • A string naming the geom. To give the geom as a string, strip the function name of the `geom_` prefix. For example, to use `geom_point()`, give the geom as `"point"`. • For more information and other ways to specify the geom, see the layer geom documentation.
`position`	A position adjustment to use on the data for this layer. This can be used in various ways, including to prevent overplotting and improving the display. The `position` argument accepts the following: • The result of calling a position function, such as `position_jitter()`. This method allows for passing extra arguments to the position. • A string naming the position adjustment. To give the position as a string, strip the function name of the `position_` prefix. For example, to use `position_jitter()`, give the position as `"jitter"`. • For more information and other ways to specify the position, see the layer position documentation.
`na.rm`	Remove missing values (Not Yet Implemented).
`show.legend`	logical. Should this layer be included in the legends? `NA`, the default, includes if any aesthetics are mapped. `FALSE` never includes, and `TRUE` always includes. It can also be a named logical vector to finely select the aesthetics to display. To include legend keys for all levels, even when no data exists, use `TRUE`. If `NA`, all levels are shown in legend, but unobserved levels are omitted.
`inherit.aes`	If `FALSE`, overrides the default aesthetics, rather than combining with them. This is most useful for helper functions that define both data and aesthetics and shouldn't inherit behaviour from the default plot specification, e.g. `annotation_borders()`.
`edata`	Environmental data where the continuous variables are found.
`formula`	Formula to select variables from `edata`. If missing, all continuos variables of `edata` are used.
`arrow.mul`	Multiplier to arrow length. If missing, the multiplier is selected automatically so that arrows fit the current graph.
`...`	Other arguments passed to the functions.

Value

Returns a ggplot2 layer or a list of such layers: a `"LayerInstance"` object that inherits from classes `"Layer"`, `"ggproto"`, and `"gg"`.

Examples

```
library("vegan")
library("ggplot2")

data(mite, mite.env)
m <- metaMDS(mite, trace = FALSE, trymax = 100)

## add fitted vectors for continuous variables
```

```
ordiggplot(m) +
  geom_ordi_point("sites") +
  geom_ordi_arrow("sites", stat = "vectorfit", edata = mite.env)

## can be faceted
ordiggplot(m) + geom_ordi_point("sites") +
  geom_ordi_arrow("sites", stat = "vectorfit", edata = mite.env) +
  facet_wrap(mite.env$Topo)
```

`valid_layers` *Valid layers for vegan objects*

Description

Valid layers for vegan objects

Usage

```
valid_layers(object, ...)

## S3 method for class 'rda'
valid_layers(object, ...)

## S3 method for class 'cca'
valid_layers(object, ...)

## S3 method for class 'dbrda'
valid_layers(object, ...)
```

Arguments

`object`	An R object.
`...`	Additional arguments passed to methods.

Value

A character vector of valid ordination layers for the provided `object`.

Index