Package: ggvegan (via r-universe)

October 19, 2026

Type Package

Title 'ggplot2' Plots for the 'vegan' Package

Version 0.2.1.9

Depends R (>= 4.1.0)

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

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

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Encoding UTF-8

Config/testthat/edition 3

Language en-GB

URL https://github.com/gavinsimpson/ggvegan, https://gavinsimpson.github.io/ggvegan/

BugReports https://github.com/gavinsimpson/ggvegan/issues

Config/pak/sysreqs libicu-dev

Repository https://gavinsimpson.r-universe.dev

Date/Publication 2026-05-22 08:22:13 UTC

RemoteUrl https://github.com/gavinsimpson/ggvegan

RemoteRef HEAD

RemoteSha 476b0e77168b5e0626af70232148c9e3cf38db1e

Contents

add_biplot_arrows . . . 3
add_biplot_centroids . . . 3
add_spp_site_scores . . . 4
arrow_mul . . . 4
autolayer.envfit . . . 5
autolayer.ordisurf . . . 6
autoplot.anosim . . . 7
autoplot.cca . . . 9
autoplot.dbrda . . . 10
autoplot.decorana . . . 12
autoplot.envfit . . . 13
autoplot.fisherfit . . . 14
autoplot.isomap . . . 15
autoplot.metaMDS . . . 17
autoplot.permustats . . . 18
autoplot.poolaccum . . . 19
autoplot.prc . . . 21
autoplot.prestonfit . . . 22
autoplot.radfit . . . 23
autoplot.rda . . . 25
autoplot.vegan_pco . . . 27
check_user_layers . . . 28
fortify.anosim . . . 29
fortify.cca . . . 30
fortify.dbrda . . . 31
fortify.decorana . . . 32
fortify.envfit . . . 33
fortify.fisherfit . . . 34
fortify.isomap . . . 35
fortify.metaMDS . . . 36
fortify.permustats . . . 37
fortify.poolaccum . . . 38
fortify.prc . . . 39
fortify.prestonfit . . . 40
fortify.radfit . . . 41
fortify.rda . . . 42
fortify.renyiaccum . . . 44
fortify.vegan_pco . . . 45
geom_ordi_arrow . . . 46
geom_ordi_axis . . . 47
geom_ordi_label . . . 48
geom_ordi_point . . . 49
geom_ordi_repel . . . 50
geom_ordi_text . . . 51
get_dimension_names . . . 52
label_fun . . . 53

layer_draw_list 53
ordiggplot 54
scores_length 56
StatVectorfit 56
valid_layers 59

Index **60**

`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

`autolayer.envfit` *Add envfit Results to an ordiggplot Graph*

Description

Function adds layers of fitted vectors and centroids of factor levels in an `ordiggplot()` graph.

Usage

```
## S3 method for class 'envfit'
autolayer(
  object,
  text = TRUE,
  box = FALSE,
  arrow.mul = 1,
  arrow.params = list(),
  text.params = list(),
  ...
)
```

Arguments

`object`	`vegan::envfit()` result object
`text`	add text to plot (logical).
`box`	write text on a non-transparent label (logical).
`arrow.mul`	arrow multiplier.
`arrow.params, text.params`	List of additional parameters to `ggplot2::geom_segment()` for arrows, and to text labels of both arrows and centroids of factor levels (`ggplot2::geom_text()`, `ggplot2::geom_label()`).
`...`	Other parameters passed to all graphical functions `geom_ordi_arrow()`, `geom_ordi_text()` and `geom_ordi_label()`.

Details

Function adds a previosly fitted `vegan::envfit()` model. This does not adapt to changes in the `ordiggplot()` object, for instance in `scaling` or `axes`, but it must re-fitted for a changed model.

Value

Returns ggplot2 layers `geom_segment` for arrows and `geom_text` or `geom_label` for their text labels (when appropriate), and another layer of `geom_text` or `geom_label` for the centroids of factor leves (when appropriate). The `score` type is called `envfit`.

Author(s)

Jari Oksanen

Examples

```
library(vegan)
library(ggplot2)
data(mite, mite.env, package = "vegan")
mod <- cca(mite)
## you must have same scaling in envfit() and ordiggplot()
ef <- envfit(mod ~ Shrub+Topo+WatrCont+SubsDens, mite.env,
  scaling = "sites")
ordiggplot(mod, scaling="sites") +
  geom_ordi_axis() +
  geom_ordi_point("sites") +
  autolayer(ef, arrow.mul=1.3, col="navy", box=TRUE,
    text.params=list(mapping=aes(fontface="bold")))
```

`autolayer.ordisurf` *Add ordisurf Result as Contours to an Ordination Graph*

Description

Function adds `vegan::ordisurf()` results to an graph.

Usage

```
## S3 method for class 'ordisurf'
autolayer(
  object,
  fill = FALSE,
  contour.params = list(),
  fill.params = list(),
  ...
)
```

Arguments

`object`	`vegan::ordisurf()` result object.
`fill`	Use filled ordination space where raster colour match the fitted values of the surface (logical).

`contour.params`, `fill.params`
: Arguments passed to `ggplot2::geom_contour()` or to `ggplot2::geom_raster()`, respectively.

`...`
: Other arguments passed both to `ggplot2::geom_contour()` and `ggplot2::geom_raster()`.

Details

Function draws contours of fitted `vegan::ordisurf()` surface and (optionally) a raster of the surface using `ggplot2::geom_contour()` and `ggplot2::geom_raster()`.

Function uses gridded values of ordination plane saved within the `vegan::ordisurf()` result instead of scores data from an ordination object. Therefore its results can be added to any ordination graph, also from `autoplot` methods. However, you must be careful with using exactly the same axes and scaling in `vegan::ordisurf()` object and ordination.

Surface `fill` is non-transparent and will paint over all previous layers. Filled surface should be used before other visible layers, or fill should be made transparent with `alpha` in argument `fill.params` passed to `ggplot2::geom_raster()`.

Value

Returns ggplot2 layers `geom_contour` and `geom_raster` (optionally).

Author(s)

Jari Oksanen

Examples

```
library(vegan)
library(ggplot2)
data(mite, mite.env, package="vegan")
mod <- cca(mite)
surf <- ordisurf(mod ~ WatrCont, mite.env, plot = FALSE)
ordiggplot(mod) +
  geom_ordi_point("sites", size = mite.env$WatrCont/100) +
  autolayer(surf) +
  autolayer(envfit(mod ~ WatrCont, mite.env, permutations=0),
    arrow.mul = 1.3)
```

`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.radfit` *Autoplot graphics for vegan rank-abundance models*

Description

ggplot2 graphics for Rank-Abundance Distribution models fitted with **vegan** functions `vegan::radfit()` or produced with `vegan::as.rad()`.

Usage

```
## S3 method for class 'radfit'
autoplot(
  object,
  facet = TRUE,
  point.params = list(),
  line.params = list(),
  ...
)

## S3 method for class 'radfit.frame'
autoplot(object, point.params = list(), line.params = list(), ...)

## S3 method for class 'radline'
autoplot(object, point.params = list(), line.params = list(), ...)

## S3 method for class 'rad'
autoplot(object, point.params = list(), line.params = list(), ...)

## S3 method for class 'rad.frame'
autoplot(object, point.params = list(), highlight = NULL, ...)
```

Arguments

`object`	Result object from `radfit`.
`facet`	Draw each fitted model to a separate facet or (if `FALSE`) all fitted lines to a single graph.
`point.params, line.params`	Parameters to modify points or lines (passed to `geom_point` and `geom_line`).
`...`	Additional arguments passed to the functions.
`highlight`	Names of species that should be highlighted as coloured points.

Details

The `ggplot2::autoplot()` function draws graphics which are **ggplot2** alternatives for **lattice** graphics in **vegan**. In addition, there are functions for `vegan::as.rad()` results which do not have dedicated graphics in**vegan**.

Value

Returns a ggplot object.

Author(s)

Jari Oksanen

Examples

```
library(vegan)
library(ggplot2)
data(mite)
m1 <- radfit(mite[1, ])

## With logarithmic y-axis (default) Pre-emption model is a line
autoplot(m1) +
  labs(title="log-Abundance: Pre-emption model is a line")

## With log-log scale, Zipf model is a line
autoplot(m1) +
  scale_x_log10() +
  labs(title="log-log Scale: Zipf model is a line")

## Show only the best model
autoplot(m1, pick = "AIC")

## Show selected models in one frame
autoplot(m1, pick = c("Z","M","L"), facet=FALSE)

## plot best models for several sites
m <- radfit(mite[1:12,])
autoplot(m) +
  labs(title = "Model Selection AIC (Default)")

## use BIC and reorder sites by their diversity
autoplot(m, pick="BIC", order.by = diversity(mite[1:12,])) +
   labs(title="Model Selection BIC, Ordered by Increasing Diversity")

## Plot RAD models without fits highlighting most abundant species in the
## whole data.
m0 <- as.rad(mite[1:12,])
dominants <- names(sort(colSums(mite), decreasing = TRUE))[1:6]
autoplot(m0, highlight = dominants)
```

`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* `"metaMDS"` *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.radfit` *Fortify results of ranked abundance distribution models*

Description

Prepares a fortified version of results from `vegan::radfit()` or `vegan::as.rad()`.

Usage

```
## S3 method for class 'radfit'
fortify(model, data, pick = NULL, ...)

## S3 method for class 'radfit.frame'
fortify(model, data, pick = "AIC", order.by = NULL, ...)

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

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

## S3 method for class 'rad.frame'
fortify(model, data, order.by = NULL, ...)

## S3 method for class 'radfit'
tidy(x, data, pick = NULL, ...)

## S3 method for class 'radfit.frame'
tidy(x, data, pick = "AIC", order.by = NULL, ...)

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

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

## S3 method for class 'rad.frame'
tidy(x, data, order.by = NULL, ...)
```

Arguments

model	model or other R object to convert to data frame
data	original dataset, if needed
pick	Pick or several models. Allowed values are "AIC" and "BIC" for selecting the best model by AIC or BIC, or (a vector of) model names that can be abbreviated. The default returns all fitted models.
...	Arguments passed to methods.
order.by	A vector used for ordering site panels.
x	An object to be converted into a tidy tibble::tibble().

Value

A data frame or tibble which in all cases has columns species (character), for species names, with rank and abundance for data, and may also have columns site (factor) for site names when several sites were analysed, fit of fitted abundances when evaluated, and model (factor) when several models were fitted.

Author(s)

Jari Oksanen

---

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 an Arrow Layer to an ordiggplot Graph.*

Description

Function adds layers of arrows (`ggplot2::geom_segment()`) and (optionally) their text labels (`ggplot2::geom_text()` or `ggplot2::geom_label()`) to an `ordiggplot()` graph. Typically these arrows are `biplot` or `regression` scores, but the function allows drawing species arrows for PCA biplots, or adding fitted environmental vectors of `vegan::envfit()`.

Usage

```
geom_ordi_arrow(
  score,
  data,
  text = TRUE,
  box = FALSE,
  arrow.mul = 1,
  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`. This can be a vegan::envfit result object which is used to draw arrows of fitted vectors.
`text`	Add text labels to the plot (logical).
`box`	Draw a non-transparent box behind the text (logical).
`arrow.mul`	Arrow multiplier when arrows end points are given in `data`. Ignored for `biplot` and `regression` scores or `species` scores which are already scaled in constrained ordination methods when calling `ordiggplot()`.
`arrow.params, text.params`	Parameters to modify arrows or their text labels.
`...`	other arguments passed to `ggplot2::geom_segment()`, `ggplot2::geom_label()` and `ggplot2::geom_text()`

Value

Returns ggplot2 layers `geom_segment` for arrows, and `geom_text` or `geom_label` (optionally) for their names.

Author(s)

Jari Oksanen

See Also

Underlying functions are ggplot2::geom_segment(), ggplot2::geom_text() and ggplot2::geom_label().

Examples

```
library(vegan)
data(varespec, varechem, package="vegan")
## PCA biplot
mod <- rda(varespec)
ordiggplot(mod, scaling = "sites", corr = TRUE) +
  geom_ordi_axis() +
  geom_ordi_point("sites") +
  geom_ordi_arrow("species")
## RDA
mod <- rda(varespec ~ N + K + Al + Baresoil, varechem)
ordiggplot(mod) +
  geom_ordi_axis() +
  geom_ordi_point("sites") +
  geom_ordi_arrow("biplot", box = TRUE)
```

---

geom_ordi_axis *Crosshair for Axes in Eigenvector Methods.*

Description

The origin (coordinates 0,0) has a special interpretation in eigenvector methods: sites, species or variables at the centroid are average cases, and the point is more exceptional the further it is from the origin. Therefore crosshair of axes through the origin should be added to the ordination graph with eigenvector methods. Often it is forgotten, though. See the Example in geom_ordi_arrow() where the length of PCA biplot arrow shows how exceptional the species is in ordination.

Usage

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

Arguments

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

Value

Returns ggplot2 layers geom_hline and geom_vline.

Author(s)

Jari Oksanen

See Also

The underlying functions are ggplot2::geom_hline() and ggplot2::geom_vline().

Examples

```
library(vegan)
library(ggplot2)
data(dune, package = "vegan")
mod <- cca(dune)
## simple "autoplot" using ggplot2::geom_text
ordiggplot(mod) +
  geom_ordi_axis() +
  geom_text()
```

---

geom_ordi_label *Add a Label Layer to an ordiggplot Graph*

Description

Function adds "labels" or text in a box with non-transparent background using ggplot2::geom_label() to an ordiggplot() graph. These can help in congested plot since uppermost labels are readable (but cover lower ones), or to emphasize text. Some **ggvegan** functions call ggplot2::geom_label() with argument box = TRUE.

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. This can be a vegan::envfit() result object which is used to plot centroids of factor levels.
...	other arguments passed to ggplot2::geom_label()

Value

Returns a ggplot2 layer geom_label.

Author(s)

Jari Oksanen

See Also

ggplot2::geom_label().

Examples

```
library(vegan)
library(ggplot2)
data(dune, dune.env, package = "vegan")
mod <- cca(dune ~ Moisture, dune.env)
ordiggplot(mod) +
  geom_ordi_axis() +
  geom_ordi_point("sites") +
  geom_ordi_label("species", size = 3,
    mapping = aes(fontface = "italic")) +
  geom_ordi_label("centroids", size = 5,
    mapping = aes(fontface = "bold"))
```

---

geom_ordi_point *Add a Point Layer to an ordiggplot Graph*

Description

Function adds a point layer using `ggplot2::geom_point()` to an `ordiggplot()` graph.

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 `geom_point`.

Author(s)

Jari Oksanen

See Also

`ggplot2::geom_point()`.

Examples

```
library(vegan)
library(ggplot2)
data(mite, mite.env, package = "vegan")
mod <- metaMDS(mite, trace = 0)
surf <- vegan::ordisurf(mod ~ WatrCont, mite.env, plot = FALSE)
ef <- vegan::envfit(mod ~ WatrCont + Topo, mite.env)
## Change sizes and colours of points
ordiggplot(mod) +
  geom_ordi_point("sites",
    mapping=aes(colour = mite.env$Topo, size=mite.env$WatrCont/10)) +
  autolayer(surf) +
  autolayer(ef, box = TRUE)
```

---

geom_ordi_repel *Add Points and their Replled Labels to an ordiggplot Graph*

Description

Function adds points to the exact position of the ordination score with a label, but labels are repelled from each other to avoid over-plotting. In very congested areas, the labels are completely omitted.

Usage

```
geom_ordi_repel(
  score,
  data,
  box = FALSE,
  points = TRUE,
  point.params = list(),
  text.params = list(),
  ...
)
```

Arguments

score	Ordination score added to the plot.
data	Alternative data to the function that will be used instead of score. This function does *not* handle vegan::envfit() results.
box	Draw a non-transparent box behind the text (logical).
points	Draw points (logical).
point.params, text.params	Parameters to modify points or their repelled text labels.
...	other arguments passed to ggrepel::geom_text_repel() and ggrepel::geom_label_repel().

Value

Returns ggrepel layers `geom_text_repel` or `geom_label_repel` and ggplot2 layer `geom_point` (optionally).

Author(s)

Jari Oksanen

Examples

```
library(vegan)
data(mite, mite.env, package = "vegan")
mod <- cca(mite)
ordiggplot(mod) +
  geom_ordi_axis() +
  geom_ordi_point("sites") +
  geom_ordi_repel("species",
    text.params = list(size=3, fontface = "italic"))
```

---

geom_ordi_text *Add a Text Layer to an ordiggplot Graph*

Description

Function adds a text layer to an `ordiggplot()` graph using `ggplot2::geom_text()`.

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`. This can be a `vegan::envfit()` result object which is used to plot centroids of factor levels.
`...`	other arguments passed to `ggplot2::geom_text()`

Value

Returns a ggplot2 layer `geom_text`.

Author(s)

Jari Oksanen

See Also

ggplot2::geom_text().

Examples

```
library(vegan)
library(ggplot2)
data(dune, package = "vegan")
mod <- metaMDS(dune, trace = 0)
ordiggplot(mod) +
  geom_ordi_text("sites", size=3) +
  geom_ordi_text("species", mapping=aes(fontface="italic"))
```

---

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_* functions 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), legend.position = "right", ...)

ggscores(score)
```

Arguments

model	An ordination result object with a compatible fortify() meethods (typically from **vegan**).
axes	Two axes to be plotted
legend.position	Legend position: see ggplot2::theme() for details. Use "none" to not draw the legend. Some ggplot2::theme() can ignore this argument.
...	Parameters passed to fortify functions to extract ordination scores. You can define scores arguments such as scaling.
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, and it may be difficult to add new layers to the graph.

In contrast, function ordiggplot() only sets up an ordination plot, but allows you to add layers to the graph one by one with full flexibility of the **ggplot2** functions. It initializes the **ggplot2** graph with idiom ggplot(data, mapping) which allows other functions to use the ordination data and axis mapping in adding new layers. The included mapping specifies coordinates x and y and variables label for text labels and a score-type specific colour. Any **ggplot2** function that can use this mapping will work automatically. Support function ggscores selects a slice of ordination scores that can be used as a data argument in the geom function. To plot the site scores of ordination result mod as points, you can use idiom ordiggplot(mod) + geom_point(data = ggscores("sites")).

There are specific functions to ease adding layers to an ordination graph. See the documentation of geom_ordi_arrow(), geom_ordi_label(), geom_ordi_point(), geom_ordi_repel(), geom_ordi_text(). These correspond to similarly named geom_* functions. For instance, geom_ordi_point adds very little to geom_point, and you can use all arguments of the standard geom function. These

functions are of type `geom_ordi_*(score, ...)` which is similar to `geom_*(data = ggscores(score), ...)`. Some functions were adapted to typical ordination data and return compound geometries. For instance, `geom_ordi_arrow()` returns layers `geom_segment` for the arrows, and `geom_text` for their name labels. In addition, there are functions to add previously fitted results of `vegan::envfit()` and `vegan::ordisurf()` (see `autolayer.envfit()`, `autolayer.ordisurf()`).

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. The `ordiggplot` skeleton sets up `data` used in plotting, and you should define axis scaling, axes *etc* in the `ordiggplot` call and they will be used in all added layers.

Value

Returns a ggplot object with slot `data` for full ordination data and slot `mapping` with ordination axes mapped to `x` and `y`, `label` to text labels for each row and factor `score` type mapped to `colour`.

Author(s)

Jari Oksanen

See Also

`geom_ordi_arrow()`, `geom_ordi_axis()`, `geom_ordi_label()`, `geom_ordi_point()`, `geom_ordi_repel()`, `geom_ordi_text()`, `autolayer.envfit()`, `autolayer.ordisurf()`.

Examples

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

## data and mapping of the ordiggplot object
ordiggplot(m)@data
ordiggplot(m)@mapping

## use geom_ordi_* functions
ordiggplot(m) + geom_ordi_axis() +
  geom_ordi_point("sites") +
  geom_ordi_repel("species", col = "darkblue",
                  text.params = list(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, corr = TRUE) +
  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()`.

The arrows are fitted to the `data` and `mapping` of `ordiggplot()`, and they will adapt to changes in the parameters of `ordiggplot()`. This is in contrast to similar `autolayer.envfit()` command or to `geom_ordi_arrow(data=<envfit-object>)` which will use a previously fitted `<envfit-object>` and will not not change if the `ordiggplot()` definitions change.

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 layer that containts a StatVectorfit object that is responsible for rendering the fitted vectors in the plot.

Author(s)

Jari Oksanen

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

∗ **datasets**
    StatVectorfit, 56

add_biplot_arrows, 3
add_biplot_centroids, 3
add_spp_site_scores, 4
aes(), *57*
annotation_borders(), *58*
arrow_mul, 4
autolayer.envfit, 5
autolayer.envfit(), *55, 56*
autolayer.ordisurf, 6
autolayer.ordisurf(), *55*
autoplot.anosim, 7
autoplot.cca, 9
autoplot.cca(), *54*
autoplot.dbrda, 10
autoplot.decorana, 12
autoplot.envfit, 13
autoplot.fisherfit, 14
autoplot.isomap, 15
autoplot.metaMDS, 17
autoplot.metaMDS(), *54*
autoplot.permustats, 18
autoplot.poolaccum, 19
autoplot.prc, 21
autoplot.prestonfit, 22
autoplot.rad *(*autoplot.radfit*)*, 23
autoplot.radfit, 23
autoplot.radline *(*autoplot.radfit*)*, 23
autoplot.rda, 25
autoplot.rda(), *54*
autoplot.vegan_pco, 27

check_user_layers, 28

fortify(), *9*, *11*, *26, 27*, *57*
fortify.anosim, 29
fortify.cca, 30
fortify.dbrda, 31
fortify.decorana, 32
fortify.decorana(), *12*
fortify.envfit, 33
fortify.fisherfit, 34
fortify.isomap, 35
fortify.metaMDS, 36
fortify.permustats, 37
fortify.poolaccum, 38
fortify.poolaccum(), *20*
fortify.prc, 39
fortify.prestonfit, 40
fortify.rad *(*fortify.radfit*)*, 41
fortify.radfit, 41
fortify.radline *(*fortify.radfit*)*, 41
fortify.rda, 42
fortify.renyiaccum, 44
fortify.vegan_pco, 45

geom_line, *24*
geom_ordi_arrow, 46
geom_ordi_arrow(), *5*, *47*, *54–56*
geom_ordi_axis, 47
geom_ordi_axis(), *55*
geom_ordi_label, 48
geom_ordi_label(), *5*, *54, 55*
geom_ordi_point, 49
geom_ordi_point(), *54, 55*
geom_ordi_repel, 50
geom_ordi_repel(), *54, 55*
geom_ordi_text, 51
geom_ordi_text(), *5*, *54, 55*
geom_point, *24*
get_dimension_names, 52
ggplot(), *57*
ggplot2::autoplot(), *24*
ggplot2::fortify(), *14*
ggplot2::geom_boxplot(), *19*
ggplot2::geom_contour(), *7*
ggplot2::geom_density(), *19*
ggplot2::geom_hline(), *47, 48*

ggplot2::geom_label(), *5*, *46–48*, *53*
ggplot2::geom_point(), *49*
ggplot2::geom_qq(), *19*
ggplot2::geom_raster(), *7*
ggplot2::geom_ribbon(), *20*
ggplot2::geom_segment(), *5*, *46, 47*
ggplot2::geom_text(), *5*, *46, 47*, *51–53*
ggplot2::geom_violin(), *19*
ggplot2::geom_vline(), *47, 48*
ggplot2::ggplot(), *30–32*, *34*, *36*, *39, 40*, *42*, *45*
ggplot2::theme(), *9*, *11, 12*, *17*, *21*, *26, 27*, *54*
ggrepel::geom_label_repel(), *50*, *53*
ggrepel::geom_text_repel(), *50*, *53*
ggscores *(*ordiggplot*)*, 54

label_fun, 53
layer geom, *57*
layer position, *58*
layer_draw_list, 53

ordiggplot, 54
ordiggplot(), *5*, *46*, *48, 49*, *51*, *56*

radfit, *24*

scores_length, 56
stat_vectorfit *(*StatVectorfit*)*, 56
StatVectorfit, 56

tibble::tibble(), *42*
tidy.anosim *(*fortify.anosim*)*, 29
tidy.cca *(*fortify.cca*)*, 30
tidy.dbrda *(*fortify.dbrda*)*, 31
tidy.decorana *(*fortify.decorana*)*, 32
tidy.envfit *(*fortify.envfit*)*, 33
tidy.fisherfit *(*fortify.fisherfit*)*, 34
tidy.isomap *(*fortify.isomap*)*, 35
tidy.metaMDS *(*fortify.metaMDS*)*, 36
tidy.permustats *(*fortify.permustats*)*, 37
tidy.poolaccum *(*fortify.poolaccum*)*, 38
tidy.prc *(*fortify.prc*)*, 39
tidy.prestonfit *(*fortify.prestonfit*)*, 40
tidy.rad *(*fortify.radfit*)*, 41
tidy.radfit *(*fortify.radfit*)*, 41
tidy.radline *(*fortify.radfit*)*, 41
tidy.rda *(*fortify.rda*)*, 42
tidy.renyiaccum *(*fortify.renyiaccum*)*, 44
tidy.vegan_pco *(*fortify.vegan_pco*)*, 45

valid_layers, 59
vegan::anosim(), *7, 8*, *29*
vegan::as.rad(), *23, 24*, *41*
vegan::boxplot.permustats(), *18*
vegan::capscale(), *9*, *30*
vegan::cca(), *9*, *30*
vegan::dbrda(), *10, 11*, *31*
vegan::decorana(), *12*, *32*
vegan::densityplot.permustats(), *18*
vegan::envfit, *46*
vegan::envfit(), *5*, *13*, *33*, *46*, *48*, *50, 51*, *55*
vegan::fisherfit(), *15*, *34*
vegan::isomap(), *15, 16*, *35*
vegan::metaMDS(), *17*, *36*
vegan::ordisurf(), *6, 7*, *55*
vegan::pco(), *27*, *45*
vegan::permustats(), *18, 19*, *37*
vegan::poolaccum(), *19, 20*, *38*
vegan::prc(), *21*, *39*
vegan::prestonfit(), *23*, *40*
vegan::radfit(), *23*, *41*
vegan::rda(), *25, 26*, *30*, *43*
vegan::renyiaccum(), *44*
vegan::scores(), *30, 31*, *36*, *43*
vegan::scores.cca(), *30*, *39*
vegan::scores.decorana(), *32*
vegan::scores.envfit(), *33*
vegan::scores.metaMDS(), *36*
vegan::scores.rda(), *11*, *26*, *31*, *43*, *45*
vegan::summary.poolaccum(), *38*