

Package: isopleuros (via r-universe)

November 25, 2024

Title Ternary Plots

Version 1.3.0

Maintainer Nicolas Frerebeau

<nicolas.frerebeau@u-bordeaux-montaigne.fr>

Description Ternary plots made simple. This package allows to create ternary plots using 'graphics'. It provides functions to display the data in the ternary space, to add or tune graphical elements and to display statistical summaries. It also includes common ternary diagrams which are useful for the archaeologist (e.g. soil texture charts, ceramic phase diagram).

License GPL (>= 3)

URL <https://packages.tesselle.org/isopleuros/>,
<https://github.com/tesselle/isopleuros>

BugReports <https://github.com/tesselle/isopleuros/issues>

Depends R (>= 3.5)

Imports graphics, grDevices, methods, stats, utils

Suggests interp, rsvg, svglite, tinysnapshot, tinytest

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Collate 'AllGenerics.R' 'coordinates.R' 'data.R'
'isopleuros-internal.R' 'isopleuros-package.R'
'ternary_arrows.R' 'ternary_axes.R' 'ternary_box.R'
'ternary_contour.R' 'ternary_crosshairs.R' 'ternary_density.R'
'ternary_ellipse.R' 'ternary_grid.R' 'ternary_hull.R'
'ternary_image.R' 'ternary_labels.R' 'ternary_lines.R'
'ternary_mean.R' 'ternary_pairs.R' 'ternary_pca.R'
'ternary_plot.R' 'ternary_points.R' 'ternary_polygon.R'
'ternary_segments.R' 'ternary_text.R' 'ternary_title.R'
'ternary_window.R' 'triangle_phase.R' 'triangle_soil.R' 'zzz.R'

NeedsCompilation no

Author Nicolas Frerebeau [aut, cre]

(<https://orcid.org/0000-0001-5759-4944>), Brice Lebrun [ctb]

(<https://orcid.org/0000-0001-7503-8685>), Logo designer),

Université Bordeaux Montaigne [fnd], CNRS [fnd]

Repository CRAN

Date/Publication 2024-11-22 14:30:07 UTC

Contents

arctic	2
boxite	3
lava	4
ternary_arrows	4
ternary_axis	5
ternary_box	7
ternary_contour	8
ternary_crosshairs	10
ternary_density	11
ternary_ellipse	13
ternary_grid	14
ternary_hull	16
ternary_image	17
ternary_labels	18
ternary_lines	19
ternary_mean	20
ternary_pairs	21
ternary_pca	22
ternary_plot	23
ternary_points	26
ternary_polygon	27
ternary_segments	28
ternary_text	29
ternary_title	31
triangle_phase_cas	32
triangle_soil	33
Index	35

arctic

Arctic Lake Sediments Compositions

Description

Sand, silt, clay compositions of 39 sediment samples at different water depths in an Arctic lake.

Usage

arctic

Format

A `data.frame` with 4 variables:

sand

silt

clay

depth Water depth (m).

Source

Aitchison, J. (1986). *The Statistical Analysis of Compositional Data*. London: Chapman and Hall.
[doi:10.1007/9789400941090](https://doi.org/10.1007/9789400941090).

See Also

Other datasets: [boxite](#), [lava](#)

boxite

Boxite Compositions

Description

Compositions of 25 specimens of boxite.

Usage

boxite

Format

A `data.frame` with 5 variables:

A albite.

B blandite.

C cornite.

D daubite.

E endite.

Source

Aitchison, J. (1986). *The Statistical Analysis of Compositional Data*. London: Chapman and Hall.
[doi:10.1007/9789400941090](https://doi.org/10.1007/9789400941090).

See Also

Other datasets: [arctic](#), [lava](#)

lava

Skye Lavas Compositions

Description

AFM compositions of 23 aphyric Skye lavas.

Usage

lava

Format

A [data.frame](#) with 3 variables:

A Na₂O + K₂O (percent).

F Fe₂O₃ (percent).

M MgO (percent).

Source

Aitchison, J. (1986). *The Statistical Analysis of Compositional Data*. London: Chapman and Hall.
[doi:10.1007/9789400941090](https://doi.org/10.1007/9789400941090).

See Also

Other datasets: [arctic](#), [boxite](#)

ternary_arrows

Add Arrows to a Ternary Plot

Description

Draw arrows between pairs of points.

Usage

```
ternary_arrows(x0, y0, z0, ...)
```

```
## S4 method for signature 'numeric,numeric,numeric'
ternary_arrows(x0, y0, z0, x1 = x0, y1 = y0, z1 = z0, ...)
```

Arguments

`x0, y0, z0` A **numeric** vector giving the x, y and z ternary coordinates of points from which to draw.

...

`x1, y1, z1` A **numeric** vector giving the x, y and z ternary coordinates of points to which to draw.

Value

`ternary_arrows()` is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

[graphics::arrows\(\)](#)

Other geometries: [ternary_crosshairs\(\)](#), [ternary_image\(\)](#), [ternary_labels\(\)](#), [ternary_lines\(\)](#), [ternary_points\(\)](#), [ternary_polygon\(\)](#), [ternary_segments\(\)](#), [ternary_text\(\)](#)

Examples

```
## Add arrows
ternary_plot(NULL, panel.first = ternary_grid())
ternary_arrows(x0 = 40, y0 = 20, z0 = 40,
               x1 = 20, y1 = 40, z1 = 40)
```

ternary_axis

Add an Axis to a Ternary Plot

Description

Adds an axis to the current plot.

Usage

```
ternary_axis(
  side,
  at = NULL,
  labels = TRUE,
  tick = TRUE,
  center = getOption("isopleuros.center"),
  scale = getOption("isopleuros.scale"),
  font = NA,
  lty = "solid",
  lwd = 1,
```

```

    lwd.ticks = lwd,
    col = NULL,
    col.ticks = NULL,
    ...
)

```

Arguments

side	An integer specifying which side of the plot the axis is to be drawn on. The axis is placed as follows: 1=below, 2=right and 3=left.
at	A numeric vector giving the points at which tick-marks are to be drawn.
labels	A logical scalar specifying whether (numerical) annotations are to be made at the tickmarks, or a character vector of labels to be placed at the tickpoints. If this is not logical, at should also be supplied and of the same length.
tick	A logical scalar: should tickmarks and an axis line be drawn?
center	A numeric vector giving the center. If NULL (the default), data are assumed not centered.
scale	A numeric vector giving the scale factor. If NULL (the default), data are assumed not scaled.
font	font for text. Defaults to <code>par("font.axis")</code> .
lty	A character string or numeric value specifying the line type for both the axis line and the tick marks.
lwd, lwd.ticks	A non-negative numeric value specifying the line widths for the axis line and the tick marks.
col, col.ticks	Colors for the axis line and the tick marks respectively. Defaults to <code>par("col.axis")</code> .
...	Other graphical parameters may also be passed as arguments to this function, particularly, <code>cex.axis</code> , <code>col.axis</code> and <code>font.axis</code> for axis annotation.

Value

`ternary_axis()` is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

Other graphical elements: [ternary_box\(\)](#), [ternary_grid\(\)](#), [ternary_pairs\(\)](#), [ternary_plot\(\)](#), [ternary_title\(\)](#)

Examples

```

## Add axis
ternary_plot(NULL, axes = FALSE)
ternary_axis(side = 1, col = "red")
ternary_axis(side = 2, col = "blue")

```

```
ternary_axis(side = 3, col = "green")

## Add box and grid
ternary_plot(NULL, axes = FALSE)
ternary_box(lty = "dashed", col = "red")
ternary_grid(lty.primary = "dotted")
```

ternary_box*Draw a Box around a Ternary Plot*

Description

Draw a Box around a Ternary Plot

Usage

```
ternary_box(lty = "solid", ...)
```

Arguments

lty A [character](#) string or [numeric](#) value specifying the line type of the box.
... Other [graphical parameters](#) may also be passed as arguments to this function, particularly, `col` or `lwd`.

Value

`ternary_box()` is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

Other graphical elements: [ternary_axis\(\)](#), [ternary_grid\(\)](#), [ternary_pairs\(\)](#), [ternary_plot\(\)](#), [ternary_title\(\)](#)

Examples

```
## Add axis
ternary_plot(NULL, axes = FALSE)
ternary_axis(side = 1, col = "red")
ternary_axis(side = 2, col = "blue")
ternary_axis(side = 3, col = "green")

## Add box and grid
ternary_plot(NULL, axes = FALSE)
ternary_box(lty = "dashed", col = "red")
ternary_grid(lty.primary = "dotted")
```

ternary_contour	<i>Contour Lines</i>
-----------------	----------------------

Description

Computes and draws contour lines.

Usage

```
ternary_contour(x, y, z, ...)
```

```
## S4 method for signature 'numeric,numeric,numeric'
ternary_contour(
  x,
  y,
  z,
  value,
  n = 50,
  nlevels = 10,
  levels = pretty(range(value, na.rm = TRUE), nlevels),
  ilr = TRUE,
  method = "linear",
  extrapolate = FALSE,
  palette = function(i) grDevices::hcl.colors(i, "YlOrRd", rev = TRUE),
  ...
)
```

```
## S4 method for signature 'ANY,missing,missing'
ternary_contour(
  x,
  value,
  n = 50,
  nlevels = 10,
  levels = pretty(range(value, na.rm = TRUE), nlevels),
  ilr = TRUE,
  method = "linear",
  extrapolate = FALSE,
  palette = function(i) grDevices::hcl.colors(i, "YlOrRd", rev = TRUE),
  ...
)
```

Arguments

x, y, z	A numeric vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see grDevices::xyz.coords()).
...	Further arguments to be passed to ternary_lines() .

value	A numeric vector giving the values to be plotted.
n	A length-one numeric specifying the number of grid points.
nlevels	A length-one numeric vector specifying the number of contour levels desired. Only used if levels is NULL.
levels	A numeric vector of levels at which to draw contour lines.
ilr	A logical scalar: should interpolation be computed in ILR space? If FALSE, interpolation is computed in Cartesian space.
method	A character string: specifying the method for interpolation (see <code>interp::interp()</code>).
extrapolate	A logical scalar: should extrapolation be used outside of the convex hull determined by the data points (see <code>interp::interp()</code>)?
palette	A color palette function that takes a single integer argument (the number of levels) and returns a vector of colors.

Details

Contour are computed from a bivariate interpolation onto a grid, after an isometric log ratio transformation of the original data.

Value

`ternary_contour()` is called it for its side-effects.

Invisibly returns a **list** with elements `levels` (the contour levels) and `colors` (the contour colors) that can be used for a legend.

Note

The **interp** package needs to be installed on your machine.

Author(s)

N. Frerebeau

See Also

`interp::interp()`, `grDevices::contourLines()`

Other statistics: `ternary_density()`, `ternary_ellipse()`, `ternary_hull()`, `ternary_mean()`, `ternary_pca()`

Examples

```
## Add density
## Data from Aitchison 1986
ternary_plot(arctic, panel.first = ternary_grid())
levels <- ternary_contour(arctic, value = arctic$depth, n = 100, nlevels = 10)

## Add a legend
legend_image <- grDevices::as.raster(rev(levels$colors))
graphics::rasterImage(legend_image, 0.85, 0.75, 0.9, 1)
graphics::text(x = 0.9, y = c(0.75, 1), labels = range(levels$levels), pos = 4)
```

ternary_crosshairs *Add Cross-Hairs to a Ternary Plot*

Description

Draw lines that intersect at a point.

Usage

```
ternary_crosshairs(x, y, z, ...)

## S4 method for signature 'numeric,numeric,numeric'
ternary_crosshairs(x, y, z, x_mark = TRUE, y_mark = TRUE, z_mark = TRUE, ...)

## S4 method for signature 'ANY,missing,missing'
ternary_crosshairs(x, x_mark = TRUE, y_mark = TRUE, z_mark = TRUE, ...)
```

Arguments

`x, y, z` A [numeric](#) vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see [grDevices::xyz.coords\(\)](#)).

`...` Further graphical parameters (see [graphics::par\(\)](#)) may also be supplied as arguments, particularly, line type, `lty`, line width, `lwd` and color, `col`. Also the line characteristics `lend`, `ljoin` and `lmitre`.

`x_mark, y_mark, z_mark` A [logical](#) scalar: should the x, y or z axis component be drawn?

Value

`ternary_crosshairs()` is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

Other geometries: [ternary_arrows\(\)](#), [ternary_image\(\)](#), [ternary_labels\(\)](#), [ternary_lines\(\)](#), [ternary_points\(\)](#), [ternary_polygon\(\)](#), [ternary_segments\(\)](#), [ternary_text\(\)](#)

Examples

```
## Add cross-hairs
## Data from Aitchison 1986
ternary_plot(lava, panel.first = ternary_grid())
ternary_crosshairs(lava)
```

```
ternary_plot(lava, panel.first = ternary_grid())
ternary_crosshairs(lava, y_mark = FALSE, z_mark = FALSE, col = "red")

ternary_plot(lava, panel.first = ternary_grid())
ternary_crosshairs(lava, x_mark = FALSE, z_mark = FALSE, col = "green")

ternary_plot(lava, panel.first = ternary_grid())
ternary_crosshairs(lava, x_mark = FALSE, y_mark = FALSE, col = "blue")
```

ternary_density	<i>Density Contour Lines</i>
-----------------	------------------------------

Description

Computes and draws density contour lines.

Usage

```
ternary_density(x, y, z, ...)
```

```
## S4 method for signature 'numeric,numeric,numeric'
```

```
ternary_density(
  x,
  y,
  z,
  h = NULL,
  n = 25,
  nlevels = 10,
  levels = NULL,
  palette = function(i) grDevices::hcl.colors(i, "YlOrRd", rev = TRUE),
  ...
)
```

```
## S4 method for signature 'ANY,missing,missing'
```

```
ternary_density(
  x,
  h = NULL,
  n = 25,
  nlevels = 10,
  levels = NULL,
  palette = function(i) grDevices::hcl.colors(i, "YlOrRd", rev = TRUE),
  ...
)
```

Arguments

`x, y, z` A [numeric](#) vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see [grDevices::xyz.coords\(\)](#)).

...	Further arguments to be passed to ternary_lines() .
h	A length-one numeric vector giving the bandwidth.
n	A length-one numeric specifying the number of grid points.
nlevels	A length-one numeric vector specifying the number of contour levels desired. Only used if <code>levels</code> is NULL.
levels	A numeric vector of levels at which to draw contour lines.
palette	A color palette function that takes a single integer argument (the number of levels) and returns a vector of colors.

Details

Two-dimensional kernel density estimation with an axis-aligned bivariate normal kernel. Normal kernel is evaluated on a square grid, after an isometric log ratio transformation of the original data.

Value

`ternary_density()` is called it for its side-effects.

Invisibly returns a [list](#) with elements `levels` (the contour levels) and `colors` (the contour colors) that can be used for a legend.

Note

This must be considered as experimental and subject to major changes in a future release.

Author(s)

N. Frerebeau

Source

Two-dimensional kernel density estimation is adapted from [MASS::kde2d\(\)](#).

See Also

[grDevices::contourLines\(\)](#)

Other statistics: [ternary_contour\(\)](#), [ternary_ellipse\(\)](#), [ternary_hull\(\)](#), [ternary_mean\(\)](#), [ternary_pca\(\)](#)

Examples

```
## Add density
## Data from Aitchison 1986
ternary_plot(lava, panel.first = ternary_grid())
levels <- ternary_density(lava, n = 500, nlevels = 10)

## Add a legend
legend_image <- grDevices::as.raster(rev(levels$colors))
graphics::rasterImage(legend_image, 0.85, 0.75, 0.9, 1)
graphics::text(x = 0.9, y = c(0.75, 1), labels = range(levels$levels), pos = 4)
```

ternary_ellipse	<i>Add an Ellipse to a Ternary Plot</i>
-----------------	---

Description

Computes and draws a confidence/tolerance ellipse.

Usage

```
ternary_ellipse(x, y, z, ...)

ternary_confidence(x, y, z, ...)

ternary_tolerance(x, y, z, ...)

## S4 method for signature 'numeric,numeric,numeric'
ternary_ellipse(x, y, z, radius = 1, ...)

## S4 method for signature 'ANY,missing,missing'
ternary_ellipse(x, radius = 1, ...)

## S4 method for signature 'numeric,numeric,numeric'
ternary_confidence(x, y, z, level = 0.95, ...)

## S4 method for signature 'ANY,missing,missing'
ternary_confidence(x, level = 0.95, ...)

## S4 method for signature 'numeric,numeric,numeric'
ternary_tolerance(x, y, z, level = 0.95, ...)

## S4 method for signature 'ANY,missing,missing'
ternary_tolerance(x, level = 0.95, ...)
```

Arguments

x, y, z	A numeric vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see grDevices::xyz.coords()).
...	Further arguments to be passed to graphics::polygon() .
radius	A numeric vector specifying the scaling of the half-diameters.
level	A numeric vector specifying the confidence/tolerance level.

Details

Ellipse coordinates are computed after an isometric log ratio transformation of the original data.

Value

ternary_ellipse() is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

[graphics::polygon\(\)](#)

Other statistics: [ternary_contour\(\)](#), [ternary_density\(\)](#), [ternary_hull\(\)](#), [ternary_mean\(\)](#), [ternary_pca\(\)](#)

Examples

```
## Ellipses
## Data from Aitchison 1986
ternary_plot(lava, panel.first = ternary_grid(5, 10))
ternary_tolerance(lava, level = 0.95, border = "blue", lty = 2)
ternary_confidence(lava, level = 0.95, border = "red", lty = 3)
```

ternary_grid

Add Grid to a Ternary Plot

Description

Adds a triangular grid to an existing plot.

Usage

```
ternary_grid(
  primary = NULL,
  secondary = NULL,
  center = getOption("isopleuros.center"),
  scale = getOption("isopleuros.scale"),
  col.primary = "darkgray",
  col.secondary = "lightgray",
  lty.primary = "dashed",
  lty.secondary = "dotted",
  lwd.primary = 1,
  lwd.secondary = lwd.primary
)
```

Arguments

primary	An integer specifying the number of cells of the primary grid in x, y and z direction.
secondary	An integer specifying the number of cells of the secondary grid in x, y and z direction.
center	A numeric vector giving the center. If NULL (the default), data are assumed not centered.
scale	A numeric vector giving the scale factor. If NULL (the default), data are assumed not scaled.
col.primary, col.secondary	A character string specifying the color of the grid lines.
lty.primary, lty.secondary	A character string or numeric value specifying the line type of the grid lines.
lwd.primary, lwd.secondary	A non-negative numeric value specifying the line width of the grid lines.

Value

ternary_grid() is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

Other graphical elements: [ternary_axis\(\)](#), [ternary_box\(\)](#), [ternary_pairs\(\)](#), [ternary_plot\(\)](#), [ternary_title\(\)](#)

Examples

```
## Data from Aitchison 1986
ternary_plot(lava, center = FALSE, scale = FALSE, col = "red", pch = 16)
ternary_grid(5)

## Center
z <- ternary_plot(lava, center = TRUE, col = "blue", pch = 16)
ternary_grid(5, center = z$center)

## Center and scale
z <- ternary_plot(lava, center = TRUE, scale = TRUE, col = "green", pch = 16)
ternary_grid(5, center = z$center, scale = z$scale)
```

ternary_hull

Convex Hull of a Set of Points

Description

Computes and draws the convex hull of the set of points specified.

Usage

```
ternary_hull(x, y, z, ...)

## S4 method for signature 'numeric,numeric,numeric'
ternary_hull(x, y, z, center = FALSE, scale = FALSE, ...)

## S4 method for signature 'ANY,missing,missing'
ternary_hull(x, center = FALSE, scale = FALSE, ...)
```

Arguments

x, y, z	A numeric vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see grDevices::xyz.coords()).
...	Further arguments to be passed to graphics::polygon() .
center	A logical scalar specifying whether the data should be centered, or a numeric vector giving the center.
scale	A logical scalar specifying whether the data should be scaled, or a numeric vector giving the scale factor.

Value

ternary_hull() is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

[grDevices::chull\(\)](#), [graphics::polygon\(\)](#)
 Other statistics: [ternary_contour\(\)](#), [ternary_density\(\)](#), [ternary_ellipse\(\)](#), [ternary_mean\(\)](#), [ternary_pca\(\)](#)

Examples

```
## Convex hull
## Data from Aitchison 1986
ternary_plot(lava, panel.first = ternary_grid(5, 10))
ternary_hull(lava, border = "red")
```

ternary_image	<i>Display a Color Image</i>
---------------	------------------------------

Description

Creates a grid of colored triangles with colors corresponding to the output of a function.

Usage

```
ternary_image(f, ...)  
  
## S4 method for signature 'function'  
ternary_image(f, n = 48, palette = NULL, ...)
```

Arguments

f	A function that takes three arguments (x, y and z coordinates) and returns a numeric vector.
...	Further parameters to be passed to f.
n	A length-one integer vector specifying the maximum number of tiles on each axis.
palette	A function that takes a single numeric vector (the output of f) as argument and returns a vector of color. If NULL, the default color scheme will be used. If FALSE, the output of f is used as colors.

Value

ternary_image() is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

Other geometries: [ternary_arrows\(\)](#), [ternary_crosshairs\(\)](#), [ternary_labels\(\)](#), [ternary_lines\(\)](#), [ternary_points\(\)](#), [ternary_polygon\(\)](#), [ternary_segments\(\)](#), [ternary_text\(\)](#)

Examples

```
## RGB  
ternary_plot(NULL, xlab = "Red", ylab = "Green", zlab = "Blue")  
ternary_image(f = rgb, n = 20, palette = FALSE)
```

ternary_labels	<i>Non-Overlapping Text Labels</i>
----------------	------------------------------------

Description

Optimize the location of text labels to minimize overplotting text.

Usage

```
ternary_labels(x, y, z, ...)

## S4 method for signature 'numeric,numeric,numeric'
ternary_labels(
  x,
  y,
  z,
  center = FALSE,
  scale = FALSE,
  labels = seq_along(x),
  type = c("text", "shadow"),
  ...
)

## S4 method for signature 'ANY,missing,missing'
ternary_labels(x, center = FALSE, scale = FALSE, labels = seq_along(x$x), ...)
```

Arguments

x, y, z	A numeric vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see grDevices::xyz.coords()).
...	Further graphical parameters (see graphics::par()) may also be supplied as arguments, particularly, character expansion, cex and color, col.
center	A logical scalar specifying whether the data should be centered, or a numeric vector giving the center.
scale	A logical scalar specifying whether the data should be scaled, or a numeric vector giving the scale factor.
labels	A character vector or expression specifying the text to be written.
type	A character string specifying the shape of the field. It must be one of "text" or "shadow". Any unambiguous substring can be given.

Value

ternary_labels() is called it for its side-effects.

Author(s)

N. Frerebeau

See Also[graphics::text\(\)](#)Other geometries: [ternary_arrows\(\)](#), [ternary_crosshairs\(\)](#), [ternary_image\(\)](#), [ternary_lines\(\)](#), [ternary_points\(\)](#), [ternary_polygon\(\)](#), [ternary_segments\(\)](#), [ternary_text\(\)](#)**Examples**

```
## Compositional data
coda <- data.frame(
  X = c(41.0, 40, 39.0),
  Y = c(19.5, 20, 20.5),
  Z = c(39.5, 40, 40.5)
)

## Add text
ternary_plot(NULL, panel.first = ternary_grid())
ternary_points(coda)
ternary_labels(coda, labels = c("A", "B", "C"))
```

ternary_lines

*Add Connected Line Segments to a Ternary Plot***Description**

Add Connected Line Segments to a Ternary Plot

Usage

```
ternary_lines(x, y, z, ...)

## S4 method for signature 'numeric,numeric,numeric'
ternary_lines(x, y, z, type = "l", ...)

## S4 method for signature 'ANY,missing,missing'
ternary_lines(x, type = "l", ...)
```

Arguments

x, y, z A **numeric** vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see [grDevices::xyz.coords\(\)](#)).

... Further graphical parameters (see [graphics::par\(\)](#)) may also be supplied as arguments, particularly, line type, lty, line width, lwd, color, col and for type = "b", pch. Also the line characteristics lend, ljoin and lmitre.

type A [character](#) string indicating the type of plotting; actually any of the types as in [graphics::plot.default\(\)](#).

Value

`ternary_lines()` is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

[graphics::lines\(\)](#)

Other geometries: [ternary_arrows\(\)](#), [ternary_crosshairs\(\)](#), [ternary_image\(\)](#), [ternary_labels\(\)](#), [ternary_points\(\)](#), [ternary_polygon\(\)](#), [ternary_segments\(\)](#), [ternary_text\(\)](#)

Examples

```
## Compositional data
coda <- data.frame(
  X = c(20, 60, 20, 20),
  Y = c(20, 20, 60, 40),
  Z = c(60, 20, 20, 40)
)

## Add lines
ternary_plot(NULL, panel.first = ternary_grid())
ternary_lines(coda, col = "red", lwd = 2)
```

ternary_mean

Compositional Mean

Description

Computes and draws the closed geometric mean of the set of points specified.

Usage

```
ternary_mean(x, y, z, ...)
```

S4 method for signature 'numeric,numeric,numeric'

```
ternary_mean(x, y, z, ...)
```

S4 method for signature 'ANY,missing,missing'

```
ternary_mean(x, y, z, ...)
```

Arguments

`x, y, z` A **numeric** vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see [grDevices::xyz.coords\(\)](#)).

`...` Further arguments to be passed to [graphics::points\(\)](#).

Value

`ternary_mean()` is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

Other statistics: [ternary_contour\(\)](#), [ternary_density\(\)](#), [ternary_ellipse\(\)](#), [ternary_hull\(\)](#), [ternary_pca\(\)](#)

Examples

```
## Mean
## Data from Aitchison 1986
ternary_plot(lava, panel.first = ternary_grid())
ternary_mean(lava, pch = 16, col = "red")
ternary_confidence(lava, level = 0.95, border = "red", lty = 1)
```

ternary_pairs

Ternary Plot Matrices

Description

Produces a matrix of ternary plots.

Usage

```
ternary_pairs(x, ...)
```

S4 method for signature 'matrix'

```
ternary_pairs(x, margin = NULL, ...)
```

S4 method for signature 'data.frame'

```
ternary_pairs(x, margin = NULL, ...)
```

Arguments

x	A matrix or a data.frame . Columns are converted to numeric in the same way that data.matrix() does.
...	Further graphical parameters .
margin	A character string or an integer giving the index of the column to be used as the third part of the ternary plots. If NULL (the default), marginal compositions will be used (i.e. the geometric mean of the non-selected parts).

Value

`ternary_pairs()` is called it for its side-effects: it results in a graphic being displayed. Invisibly returns x.

Author(s)

N. Frerebeau

See Also

Other graphical elements: [ternary_axis\(\)](#), [ternary_box\(\)](#), [ternary_grid\(\)](#), [ternary_plot\(\)](#), [ternary_title\(\)](#)

Examples

```
## Data from Aitchison 1986
## Ternary plots with marginal compositions
ternary_pairs(boxite)

## Ternary plots with endite
ternary_pairs(boxite, margin = "E")
```

ternary_pca

Principal Component Analysis

Description

Computes and draws principal component.

Usage

```
ternary_pca(x, y, z, ...)
```

```
## S4 method for signature 'numeric,numeric,numeric'
ternary_pca(x, y, z, axis = 1, ...)
```

```
## S4 method for signature 'ANY,missing,missing'
ternary_pca(x, axis = 1, ...)
```

Arguments

`x, y, z` A **numeric** vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see [grDevices::xyz.coords\(\)](#)).

`...` Further arguments to be passed to [graphics::lines\(\)](#).

`axis` An **integer** specifying the dimension to be plotted.

Value

`ternary_pca()` is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

Other statistics: [ternary_contour\(\)](#), [ternary_density\(\)](#), [ternary_ellipse\(\)](#), [ternary_hull\(\)](#), [ternary_mean\(\)](#)

Examples

```
## PCA
## Data from Aitchison 1986
ternary_plot(lava, panel.first = ternary_grid())
ternary_pca(lava, axis = 1, col = "red", lty = 2)
```

ternary_plot

Ternary Plot

Description

Produces a ternary plot.

Usage

```
ternary_plot(x, y, z, ...)

## S4 method for signature 'numeric,numeric,numeric'
ternary_plot(
  x,
  y,
  z,
  center = FALSE,
  scale = FALSE,
  xlim = NULL,
  ylim = NULL,
```

```

    zlim = NULL,
    xlab = NULL,
    ylab = NULL,
    zlab = NULL,
    main = NULL,
    sub = NULL,
    ann = graphics::par("ann"),
    axes = TRUE,
    frame.plot = axes,
    panel.first = NULL,
    panel.last = NULL,
    ...
)

## S4 method for signature 'ANY,missing,missing'
ternary_plot(
  x,
  xlim = NULL,
  ylim = NULL,
  zlim = NULL,
  xlab = NULL,
  ylab = NULL,
  zlab = NULL,
  main = NULL,
  sub = NULL,
  ann = graphics::par("ann"),
  axes = TRUE,
  frame.plot = axes,
  panel.first = NULL,
  panel.last = NULL,
  ...
)

```

Arguments

x, y, z	A numeric vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see grDevices::xyz.coords()).
...	Other graphical parameters may also be passed as arguments to this function.
center	A logical scalar: should the data be centered?
scale	A logical scalar: should the data be scaled?
xlim	A length-three numeric vector giving the x limits in the range [0, 1].
ylim	A length-three numeric vector giving the y limits in the range [0, 1].
zlim	A length-three numeric vector giving the z limits in the range [0, 1].
xlab, ylab, zlab	A character string giving a label for the x, y and z axes.
main	A character string giving a main title for the plot.

sub	A character string giving a subtitle for the plot.
ann	A logical scalar: should the default annotation (title and x, y and z axis labels) appear on the plot?
axes	A logical scalar: should axes be drawn on the plot?
frame.plot	A logical scalar: should a box be drawn around the plot?
panel.first	An expression to be evaluated after the plot axes are set up but before any plotting takes place. This can be useful for drawing background grids.
panel.last	An expression to be evaluated after plotting has taken place but before the axes, title and box are added.

Value

`ternary_plot()` is called it for its side-effects: it results in a graphic being displayed. Invisibly returns a [list](#) with the components:

x	A numeric vector of x values.
y	A numeric vector of y values.
z	A numeric vector of z values.
center	A numeric vector giving the center.
scale	A numeric vector giving the scale factor.

Author(s)

N. Frerebeau

See Also

Other graphical elements: [ternary_axis\(\)](#), [ternary_box\(\)](#), [ternary_grid\(\)](#), [ternary_pairs\(\)](#), [ternary_title\(\)](#)

Examples

```
## Blank plot
ternary_plot(NULL)

## Compositional data
coda <- data.frame(
  X = c(20, 60, 20, 1/3),
  Y = c(20, 20, 60, 1/3),
  Z = c(60, 20, 20, 1/3)
)

## Ternary plot
ternary_plot(coda, pch = 16, col = "red")

## Add a grid
ternary_plot(coda, panel.first = ternary_grid(5, 10))
```

```
## Zoom
ternary_plot(coda, ylim = c(0, 0.4, 0), zlim = c(0, 0, 0.4),
             panel.first = ternary_grid())
ternary_plot(coda, xlim = c(0, 0.4, 0), zlim = c(0, 0, 0.4),
             panel.first = ternary_grid())
ternary_plot(coda, xlim = c(0.4, 0, 0), ylim = c(0, 0.4, 0),
             panel.first = ternary_grid())

## Color according to a supplementary variable
## Data from Aitchison 1986
col <- grDevices::colorRampPalette(c("red", "blue"))(nrow(arctic))
ternary_plot(arctic, panel.first = ternary_grid(), pch = 16, col = col)
```

ternary_points	<i>Add Points to a Ternary Plot</i>
----------------	-------------------------------------

Description

Add Points to a Ternary Plot

Usage

```
ternary_points(x, y, z, ...)

## S4 method for signature 'numeric,numeric,numeric'
ternary_points(x, y, z, center = FALSE, scale = FALSE, type = "p", ...)

## S4 method for signature 'ANY,missing,missing'
ternary_points(x, center = FALSE, scale = FALSE, type = "p", ...)
```

Arguments

x, y, z	A numeric vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see grDevices::xyz.coords()).
...	Further graphical parameters (see graphics::par()) may also be supplied as arguments, particularly, plotting character, pch, character expansion, cex and color, col.
center	A logical scalar specifying whether the data should be centered, or a numeric vector giving the center.
scale	A logical scalar specifying whether the data should be scaled, or a numeric vector giving the scale factor.
type	A character string indicating the type of plotting; actually any of the types as in graphics::plot.default() .

Value

ternary_points() is called it for its side-effects. Invisibly returns a [list](#) with the components:

```
x      A numeric vector of x values.
y      A numeric vector of y values.
z      A numeric vector of z values.
center A numeric vector giving the center.
scale  A numeric vector giving the scale factor.
```

Author(s)

N. Frerebeau

See Also

[graphics::points\(\)](#)

Other geometries: [ternary_arrows\(\)](#), [ternary_crosshairs\(\)](#), [ternary_image\(\)](#), [ternary_labels\(\)](#), [ternary_lines\(\)](#), [ternary_polygon\(\)](#), [ternary_segments\(\)](#), [ternary_text\(\)](#)

Examples

```
## Add points
## Data from Aitchison 1986
ternary_plot(NULL, panel.first = ternary_grid())
ternary_points(lava, col = "red", pch = 16)

## Center and scale
ternary_plot(NULL, axes = FALSE, frame.plot = TRUE)
ternary_points(lava, col = "red", pch = 16)
ternary_points(lava, center = TRUE, col = "blue", pch = 16)
ternary_points(lava, center = TRUE, scale = TRUE, col = "green", pch = 16)
```

ternary_polygon	<i>Polygon Drawing</i>
-----------------	------------------------

Description

Draws the polygons whose vertices are given in x, y and z.

Usage

```
ternary_polygon(x, y, z, ...)
```

```
## S4 method for signature 'numeric,numeric,numeric'
ternary_polygon(x, y, z, ...)
```

```
## S4 method for signature 'ANY,missing,missing'
ternary_polygon(x, y, z, ...)
```

Arguments

`x, y, z` A `numeric` vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see `grDevices::xyz.coords()`).

`...` Further arguments to be passed to `graphics::polygon()`.

Value

`ternary_polygon()` is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

`graphics::polygon()`

Other geometries: `ternary_arrows()`, `ternary_crosshairs()`, `ternary_image()`, `ternary_labels()`, `ternary_lines()`, `ternary_points()`, `ternary_segments()`, `ternary_text()`

Examples

```
## Compositional data
coda <- data.frame(
  X = c(20, 60, 20),
  Y = c(20, 20, 60),
  Z = c(60, 20, 20)
)

## Add a polygon
ternary_plot(NULL, panel.first = ternary_grid())
ternary_polygon(coda, density = 5, border = "red")
```

ternary_segments

Add Line Segments to a Ternary Plot

Description

Draw line segments between pairs of points.

Usage

```
ternary_segments(x0, y0, z0, ...)
```

```
## S4 method for signature 'numeric,numeric,numeric'
ternary_segments(x0, y0, z0, x1 = x0, y1 = y0, z1 = z0, ...)
```

Arguments

<code>x0, y0, z0</code>	A numeric vector giving the x, y and z ternary coordinates of points from which to draw.
<code>...</code>	Further graphical parameters (see <code>graphics::par()</code>) may also be supplied as arguments, particularly, line type, <code>lty</code> , line width, <code>lwd</code> and color, <code>col</code> . Also the line characteristics <code>lend</code> , <code>ljoin</code> and <code>lmitre</code> .
<code>x1, y1, z1</code>	A numeric vector giving the x, y and z ternary coordinates of points to which to draw.

Value

`ternary_segments()` is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

[graphics::segments\(\)](#)

Other geometries: [ternary_arrows\(\)](#), [ternary_crosshairs\(\)](#), [ternary_image\(\)](#), [ternary_labels\(\)](#), [ternary_lines\(\)](#), [ternary_points\(\)](#), [ternary_polygon\(\)](#), [ternary_text\(\)](#)

Examples

```
## Add segments
ternary_plot(NULL, panel.first = ternary_grid())
ternary_segments(x0 = 40, y0 = 20, z0 = 40,
                x1 = 20, y1 = 40, z1 = 40)
```

ternary_text

Add Text to a Ternary Plot

Description

Draws the strings given in the vector `labels` at the coordinates given by `x`, `y` and `z`.

Usage

```
ternary_text(x, y, z, ...)

## S4 method for signature 'numeric,numeric,numeric'
ternary_text(
  x,
  y,
  z,
  center = FALSE,
```

```

    scale = FALSE,
    labels = seq_along(x),
    ...
  )

## S4 method for signature 'ANY,missing,missing'
ternary_text(x, center = FALSE, scale = FALSE, labels = seq_along(x$x), ...)

```

Arguments

<code>x, y, z</code>	A numeric vector giving the x, y and z ternary coordinates of a set of points. If y and z are missing, an attempt is made to interpret x in a suitable way (see grDevices::xyz.coords()).
<code>...</code>	Further arguments to be passed to graphics::text() .
<code>center</code>	A logical scalar specifying whether the data should be centered, or a numeric vector giving the center.
<code>scale</code>	A logical scalar specifying whether the data should be scaled, or a numeric vector giving the scale factor.
<code>labels</code>	A character vector or expression specifying the text to be written.

Value

`ternary_text()` is called for its side-effects.

Author(s)

N. Frerebeau

See Also

[graphics::text\(\)](#)

Other geometries: [ternary_arrows\(\)](#), [ternary_crosshairs\(\)](#), [ternary_image\(\)](#), [ternary_labels\(\)](#), [ternary_lines\(\)](#), [ternary_points\(\)](#), [ternary_polygon\(\)](#), [ternary_segments\(\)](#)

Examples

```

## Compositional data
coda <- data.frame(
  X = c(20, 60, 20),
  Y = c(20, 20, 60),
  Z = c(60, 20, 20)
)

## Add text
ternary_plot(NULL, panel.first = ternary_grid())
ternary_text(coda, labels = c("A", "B", "C"), col = "red", cex = 2)

```

ternary_title	<i>Ternary Plot Annotation</i>
---------------	--------------------------------

Description

Ternary Plot Annotation

Usage

```
ternary_title(  
  main = NULL,  
  sub = NULL,  
  xlab = NULL,  
  ylab = NULL,  
  zlab = NULL,  
  line = NA,  
  outer = FALSE,  
  ...  
)
```

Arguments

main	A character string specifying the main title (on top).
sub	A character string specifying the sub-title (at bottom).
xlab, ylab, zlab	A character string giving a label for the x, y and z axes.
line	Specifying a value for line overrides the default placement of labels, and places them this many lines outwards from the plot edge.
outer	A logical scalar: should the titles be placed in the outer margins of the plot?
...	Other graphical parameters may also be passed as arguments to this function, particularly, <code>font.main</code> , <code>cex.main</code> , <code>col.main</code> and <code>font.sub</code> , <code>cex.sub</code> , <code>col.sub</code> for title annotation; <code>font.lab</code> , <code>cex.lab</code> and <code>col.lab</code> for axis label.

Value

`ternary_title()` is called it for its side-effects.

Author(s)

N. Frerebeau

See Also

Other graphical elements: [ternary_axis\(\)](#), [ternary_box\(\)](#), [ternary_grid\(\)](#), [ternary_pairs\(\)](#), [ternary_plot\(\)](#)

Examples

```
## Add title
ternary_plot(NULL, main = "Main title", sub = "Subtitle",
             xlab = "A", ylab = "B", zlab = "C")

ternary_plot(NULL, ann = FALSE)
ternary_title(main = "Main title", sub = "Subtitle",
             xlab = "A", ylab = "B", zlab = "C")
```

triangle_phase_cas *Ceramic Phase Diagram*

Description

Ceramic Phase Diagram

Usage

```
triangle_phase_cas(labels = TRUE, symbol = FALSE, mol = FALSE, ...)

triangle_phase_ceramic(labels = TRUE, symbol = FALSE, mol = FALSE, ...)
```

Arguments

labels	A logical scalar: should labels be displayed?
symbol	A logical scalar: should symbol be used instead of full labels? Only used if labels is TRUE.
mol	A logical scalar: should molarity be used instead of molar mass?
...	Further arguments to be passed to graphics::polygon() .

Author(s)

N. Frerebeau

See Also

Other charts: [triangle_soil](#)

Examples

```
## Ceramic phase diagram
ternary_plot(NULL, xlab = "CaO", ylab = "Al2O3", zlab = "SiO2")
triangle_phase_ceramic(symbol = TRUE, mol = TRUE, pch = 16)

ternary_plot(NULL, xlab = "CaO", ylab = "Al2O3", zlab = "SiO2")
triangle_phase_ceramic(symbol = TRUE, mol = FALSE, pch = 16)

## CAS diagram
```

```
ternary_plot(NULL, axes = FALSE, ann = FALSE, frame.plot = TRUE)
triangle_phase_cas(mol = FALSE, pch = 16)
```

triangle_soil	<i>Soil Texture Triangle</i>
---------------	------------------------------

Description

Soil Texture Triangle

Usage

```
triangle_soil_hypres(labels = TRUE, symbol = FALSE, ...)
triangle_soil_folk(labels = TRUE, symbol = FALSE, ...)
triangle_soil_shepard(labels = TRUE, symbol = FALSE, ...)
triangle_soil_usda(labels = TRUE, symbol = FALSE, ...)
```

Arguments

labels	A logical scalar: should labels be displayed?
symbol	A logical scalar: should symbol be used instead of full labels? Only used if labels is TRUE.
...	Further arguments to be passed to graphics::polygon() .

Author(s)

N. Frerebeau

See Also

Other charts: [triangle_phase_cas\(\)](#)

Examples

```
## HYPRES soil texture
ternary_plot(NULL, xlab = "sand", ylab = "silt", zlab = "clay")
triangle_soil_hypres()

## USDA (1951) soil texture
ternary_plot(NULL, xlab = "sand", ylab = "silt", zlab = "clay")
triangle_soil_usda(symbol = TRUE)

## Folk (1954) soil texture
ternary_plot(NULL, xlab = "sand", ylab = "silt", zlab = "clay")
triangle_soil_folk(symbol = TRUE)
```

```
## Shepard (1954) soil texture  
ternary_plot(NULL, xlab = "sand", ylab = "silt", zlab = "clay")  
triangle_soil_shepard()
```

Index

- * **charts**
 - triangle_phase_cas, 32
 - triangle_soil, 33
- * **datasets**
 - arctic, 2
 - boxite, 3
 - lava, 4
- * **geometries**
 - ternary_arrows, 4
 - ternary_crosshairs, 10
 - ternary_image, 17
 - ternary_labels, 18
 - ternary_lines, 19
 - ternary_points, 26
 - ternary_polygon, 27
 - ternary_segments, 28
 - ternary_text, 29
- * **graphical elements**
 - ternary_axis, 5
 - ternary_box, 7
 - ternary_grid, 14
 - ternary_pairs, 21
 - ternary_plot, 23
 - ternary_title, 31
- * **statistics**
 - ternary_contour, 8
 - ternary_density, 11
 - ternary_ellipse, 13
 - ternary_hull, 16
 - ternary_mean, 20
 - ternary_pca, 22
- arctic, 2, 4
- boxite, 3, 3, 4
- character, 6, 7, 9, 15, 18, 20, 22, 24–26, 30, 31
- data.frame, 3, 4, 22
- data.matrix(), 22
- expression, 18, 30
- function, 9, 12, 17
- graphical parameters, 6, 7, 22, 24, 31
- graphics::arrows(), 5
- graphics::lines(), 20, 23
- graphics::par(), 10, 18, 19, 26, 29
- graphics::plot.default(), 20, 26
- graphics::points(), 21, 27
- graphics::polygon(), 13, 14, 16, 28, 32, 33
- graphics::segments(), 29
- graphics::text(), 19, 30
- grDevices::chull(), 16
- grDevices::contourLines(), 9, 12
- grDevices::xyz.coords(), 8, 10, 11, 13, 16, 18, 19, 21, 23, 24, 26, 28, 30
- integer, 6, 15, 17, 22, 23
- interp::interp(), 9
- lava, 3, 4, 4
- list, 9, 12, 25, 27
- logical, 6, 9, 10, 16, 18, 24–26, 30–33
- MASS::kde2d(), 12
- matrix, 22
- numeric, 5–13, 15, 16, 18, 19, 21, 23–30
- ternary_arrows, 4, 10, 17, 19, 20, 27–30
- ternary_arrows, numeric, numeric, numeric-method (ternary_arrows), 4
- ternary_arrows-method (ternary_arrows), 4
- ternary_axis, 5, 7, 15, 22, 25, 31
- ternary_box, 6, 7, 15, 22, 25, 31
- ternary_confidence (ternary_ellipse), 13

- ternary_confidence, ANY, missing, missing-method (ternary_labels), (ternary_ellipse), 13
- ternary_confidence, numeric, numeric, numeric-method (ternary_ellipse), 13
- ternary_contour, 8, 12, 14, 16, 21, 23
- ternary_contour, ANY, missing, missing-method (ternary_contour), 8
- ternary_contour, numeric, numeric, numeric-method (ternary_contour), 8
- ternary_contour-method (ternary_contour), 8
- ternary_crosshairs, 5, 10, 17, 19, 20, 27–30
- ternary_crosshairs, ANY, missing, missing-method (ternary_crosshairs), 10
- ternary_crosshairs, numeric, numeric, numeric-method (ternary_crosshairs), 10
- ternary_crosshairs-method (ternary_crosshairs), 10
- ternary_density, 9, 11, 14, 16, 21, 23
- ternary_density, ANY, missing, missing-method (ternary_density), 11
- ternary_density, numeric, numeric, numeric-method (ternary_density), 11
- ternary_density-method (ternary_density), 11
- ternary_ellipse, 9, 12, 13, 16, 21, 23
- ternary_ellipse, ANY, missing, missing-method (ternary_ellipse), 13
- ternary_ellipse, numeric, numeric, numeric-method (ternary_ellipse), 13
- ternary_ellipse-method (ternary_ellipse), 13
- ternary_grid, 6, 7, 14, 22, 25, 31
- ternary_hull, 9, 12, 14, 16, 21, 23
- ternary_hull, ANY, missing, missing-method (ternary_hull), 16
- ternary_hull, numeric, numeric, numeric-method (ternary_hull), 16
- ternary_hull-method (ternary_hull), 16
- ternary_image, 5, 10, 17, 19, 20, 27–30
- ternary_image, function-method (ternary_image), 17
- ternary_image-method (ternary_image), 17
- ternary_labels, 5, 10, 17, 18, 20, 27–30
- ternary_labels, ANY, missing, missing-method (ternary_labels), 18
- ternary_labels, numeric, numeric, numeric-method (ternary_labels), 18
- ternary_labels-method (ternary_labels), 18
- ternary_lines, 5, 10, 17, 19, 19, 27–30
- ternary_lines(), 8, 12
- ternary_lines, ANY, missing, missing-method (ternary_lines), 19
- ternary_lines, numeric, numeric, numeric-method (ternary_lines), 19
- ternary_lines-method (ternary_lines), 19
- ternary_mean, 9, 12, 14, 16, 20, 23
- ternary_mean, ANY, missing, missing-method (ternary_mean), 20
- ternary_mean, numeric, numeric, numeric-method (ternary_mean), 20
- ternary_mean-method (ternary_mean), 20
- ternary_pairs, 6, 7, 15, 21, 25, 31
- ternary_pairs, data.frame, missing, missing-method (ternary_pairs), 21
- ternary_pairs, data.frame-method (ternary_pairs), 21
- ternary_pairs, matrix-method (ternary_pairs), 21
- ternary_pairs-method (ternary_pairs), 21
- ternary_pca, 9, 12, 14, 16, 21, 22
- ternary_pca, ANY, missing, missing-method (ternary_pca), 22
- ternary_pca, numeric, numeric, numeric-method (ternary_pca), 22
- ternary_pca-method (ternary_pca), 22
- ternary_plot, 6, 7, 15, 22, 23, 31
- ternary_plot, ANY, missing, missing-method (ternary_plot), 23
- ternary_plot, numeric, numeric, numeric-method (ternary_plot), 23
- ternary_plot-method (ternary_plot), 23
- ternary_points, 5, 10, 17, 19, 20, 26, 28–30
- ternary_points, ANY, missing, missing-method (ternary_points), 26
- ternary_points, numeric, numeric, numeric-method (ternary_points), 26
- ternary_points-method (ternary_points), 26
- ternary_polygon, 5, 10, 17, 19, 20, 27, 27, 29, 30
- ternary_polygon, ANY, missing, missing-method (ternary_polygon), 27
- ternary_polygon, numeric, numeric, numeric-method (ternary_polygon), 27

ternary_polygon-method
 (ternary_polygon), 27

ternary_segments, 5, 10, 17, 19, 20, 27, 28,
 28, 30

ternary_segments,numeric,numeric,numeric-method
 (ternary_segments), 28

ternary_segments-method
 (ternary_segments), 28

ternary_text, 5, 10, 17, 19, 20, 27–29, 29

ternary_text,ANY,missing,missing-method
 (ternary_text), 29

ternary_text,numeric,numeric,numeric-method
 (ternary_text), 29

ternary_text-method (ternary_text), 29

ternary_title, 6, 7, 15, 22, 25, 31

ternary_tolerance (ternary_ellipse), 13

ternary_tolerance,ANY,missing,missing-method
 (ternary_ellipse), 13

ternary_tolerance,numeric,numeric,numeric-method
 (ternary_ellipse), 13

triangle_phase_cas, 32, 33

triangle_phase_ceramic
 (triangle_phase_cas), 32

triangle_soil, 32, 33

triangle_soil_folk (triangle_soil), 33

triangle_soil_hypres (triangle_soil), 33

triangle_soil_shepard (triangle_soil),
 33

triangle_soil_usda (triangle_soil), 33