

# Package ‘tactile’

July 27, 2025

**Title** New and Extended Plots, Methods, and Panel Functions for  
'lattice'

**Version** 0.2.2

**Description** Extensions to 'lattice', providing new high-level  
functions, methods for existing functions, panel functions, and a theme.

**Depends** R (>= 3.4.0), lattice

**Imports** grDevices, grid, gridExtra, latticeExtra, MASS, RColorBrewer,  
stats, utils

**Suggests** covr, forecast, knitr, rmarkdown, spelling, testthat, zoo

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**URL** <https://github.com/jolars/tactile>

**BugReports** <https://github.com/jolars/tactile/issues>

**VignetteBuilder** knitr

**Language** en-US

**NeedsCompilation** no

**Author** Johan Larsson [aut, cre] (ORCID:  
<<https://orcid.org/0000-0002-4029-5945>>),  
Deepayan Sarkar [ctb, cph] (lattice),  
Brian Ripley [ctb] (stats::plot.acf)

**Maintainer** Johan Larsson <johanlarsson@outlook.com>

**Repository** CRAN

**Date/Publication** 2025-07-27 21:20:02 UTC

**Contents**

|                                      |           |
|--------------------------------------|-----------|
| bubbleplot . . . . .                 | 2         |
| bwplot2 . . . . .                    | 4         |
| diag.panel.splom.density . . . . .   | 5         |
| feldspar . . . . .                   | 6         |
| panel.bubbleplot . . . . .           | 7         |
| panel.ci . . . . .                   | 8         |
| panel.qqmathci . . . . .             | 9         |
| panel.ternaryplot . . . . .          | 11        |
| panel.ternaryplot.clip . . . . .     | 12        |
| panel.ternaryplot.density . . . . .  | 13        |
| panel.ternaryplot.grid . . . . .     | 14        |
| panel.ternaryplot.response . . . . . | 15        |
| panel.ternaryplot.scales . . . . .   | 16        |
| panel.ternaryplot.xyplot . . . . .   | 16        |
| prepanel.ci . . . . .                | 17        |
| qqmath.zoo . . . . .                 | 18        |
| tactile.theme . . . . .              | 19        |
| ternaryplot . . . . .                | 20        |
| xyplot.acf . . . . .                 | 22        |
| xyplot.Arima . . . . .               | 23        |
| xyplot.forecast . . . . .            | 24        |
| xyplot.lm . . . . .                  | 25        |
| <b>Index</b>                         | <b>27</b> |

---

|            |                    |
|------------|--------------------|
| bubbleplot | <i>Bubbleplots</i> |
|------------|--------------------|

---

**Description**

Draws bubbleblots – trivariate plots where the third dimension is mapped to the size of the points drawn on the screen.

**Usage**

```
bubbleplot(x, data = NULL, ...)

## S3 method for class 'formula'
bubbleplot(
  x,
  data = NULL,
  maxsize = 3,
  bubblekey = TRUE,
  panel = panel.bubbleplot,
  groups = NULL,
  subset = TRUE,
```

```

    drop.unused.levels = lattice.getOption("drop.unused.levels"),
    ...,
    outer,
    allow.multiple
)

```

## Arguments

|                                 |                                                                                                                                                                                                                                                                                                                                                                                                         |
|---------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>                  | A formula of the form $z \sim x * y$ , where <code>x</code> and <code>y</code> have the usual interpretation in trellis graphics (see <a href="#">lattice::xyplot()</a> ) and <code>z</code> is mapped to the size of bubbles.                                                                                                                                                                          |
| <code>data</code>               | A data.frame, list or environment wherein the formula and groups arguments can be evaluated.                                                                                                                                                                                                                                                                                                            |
| <code>...</code>                | Further arguments to pass to <a href="#">lattice::xyplot()</a> .                                                                                                                                                                                                                                                                                                                                        |
| <code>maxsize</code>            | Maximum size (in cex) for the bubbles.                                                                                                                                                                                                                                                                                                                                                                  |
| <code>bubblekey</code>          | Set to TRUE to draw an informative legend about the bubbles. Uses <a href="#">lattice::draw.key()</a> . See the <b>key</b> section of the documentation in <a href="#">lattice::xyplot()</a> . If both <code>auto.key</code> and <code>bubblekey</code> are given and their space arguments (positions) conflict, <code>bubblekey</code> will silently override the position of <code>auto.key</code> . |
| <code>panel</code>              | See <a href="#">lattice::xyplot()</a> . Here, we are passing an additional variable, <code>z</code> , which is then used in <a href="#">panel.bubbleplot()</a> .                                                                                                                                                                                                                                        |
| <code>groups</code>             | See <a href="#">lattice::xyplot()</a>                                                                                                                                                                                                                                                                                                                                                                   |
| <code>subset</code>             | See <a href="#">lattice::xyplot()</a>                                                                                                                                                                                                                                                                                                                                                                   |
| <code>drop.unused.levels</code> | See <a href="#">lattice::xyplot()</a>                                                                                                                                                                                                                                                                                                                                                                   |
| <code>outer</code>              | Ignored.                                                                                                                                                                                                                                                                                                                                                                                                |
| <code>allow.multiple</code>     | Ignored.                                                                                                                                                                                                                                                                                                                                                                                                |

## Value

An object of class "trellis". The [update](#) method can be used to update components of the object and the [print](#) method (usually called by default) will plot it on an appropriate plotting device.

## Author(s)

Johan Larsson

## Examples

```

bubbleplot(displacement ~ horsepower * weight, groups = cylinder, data = mtcars, auto.key = TRUE)
bubbleplot(displacement ~ horsepower * miles.per.gallon | factor(cylinder), groups = gear, data = mtcars,
           auto.key = TRUE)

```

---

bwplot2

*An extended box and whiskers plot*


---

## Description

An extended version of `lattice::bwplot()`. The only modification is to group and stack box plots if groups is provided.

## Usage

```
bwplot2(x, data = NULL, ...)

## S3 method for class 'formula'
bwplot2(
  x,
  data = NULL,
  allow.multiple = is.null(groups) || outer,
  outer = FALSE,
  auto.key = FALSE,
  groups = NULL,
  drop.unused.levels = lattice.getOption("drop.unused.levels"),
  ...,
  subset = TRUE
)

## S3 method for class 'numeric'
bwplot2(x, data = NULL, xlab = deparse(substitute(x)), ...)
```

## Arguments

|                    |                                                         |
|--------------------|---------------------------------------------------------|
| x                  | see <code>lattice::bwplot()</code>                      |
| data               | see <code>lattice::bwplot()</code>                      |
| ...                | arguments passed down to <code>lattice::bwplot()</code> |
| allow.multiple     | see <code>lattice::bwplot()</code>                      |
| outer              | see <code>lattice::bwplot()</code>                      |
| auto.key           | see <code>lattice::bwplot()</code>                      |
| groups             | see <code>lattice::bwplot()</code>                      |
| drop.unused.levels | see <code>lattice::bwplot()</code>                      |
| subset             | see <code>lattice::bwplot()</code>                      |
| xlab               | see <code>lattice::bwplot()</code>                      |

## Value

An object of class "trellis". The `update` method can be used to update components of the object and the `print` method (usually called by default) will plot it on an appropriate plotting device.

**Examples**

```
bwplot2(variety ~ yield,
        groups = site,
        data = barley,
        par.settings = tactile.theme())
```

---

diag.panel.splom.density

*Diagonal Density Panels*


---

**Description**

Plots univariate density estimates to be used in a `lattice::splom()` call with the `diag.panel` argument.

**Usage**

```
diag.panel.splom.density(
  x,
  bw = "nrd0",
  adjust = 1,
  kernel = "gaussian",
  weights = NULL,
  n = 512,
  ...
)
```

**Arguments**

- |        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x      | data vector corresponding to that row / column (which will be the same for diagonal 'panels').                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| bw     | <p>the smoothing bandwidth to be used. The kernels are scaled such that this is the standard deviation of the smoothing kernel. (Note this differs from the reference books cited below.)</p> <p>bw can also be a character string giving a rule to choose the bandwidth. See <a href="#">bw.nrd</a>.</p> <p>The default, "nrd0", has remained the default for historical and compatibility reasons, rather than as a general recommendation, where e.g., "SJ" would rather fit, see also Venables and Ripley (2002).</p> <p>The specified (or computed) value of bw is multiplied by adjust.</p> |
| adjust | the bandwidth used is actually <code>adjust*bw</code> . This makes it easy to specify values like 'half the default' bandwidth.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| kernel | the smoothing kernel to be used. See <a href="#">stats::density()</a> for options.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

|                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>weights</code> | numeric vector of non-negative observation weights, hence of same length as <code>x</code> . The default <code>NULL</code> is equivalent to <code>weights = rep(1/nx, nx)</code> where <code>nx</code> is the length of (the finite entries of) <code>x[]</code> . If <code>na.rm = TRUE</code> and there are <code>NA</code> 's in <code>x</code> , they <i>and</i> the corresponding weights are removed before computations. In that case, when the original weights have summed to one, they are re-scaled to keep doing so.<br><br>Note that weights are <i>not</i> taken into account for automatic bandwidth rules, i.e., when <code>bw</code> is a string. When the weights are proportional to true counts <code>cn</code> , <code>density(x = rep(x, cn))</code> may be used instead of <code>weights</code> . |
| <code>n</code>       | the number of equally spaced points at which the density is to be estimated. When <code>n &gt; 512</code> , it is rounded up to a power of 2 during the calculations (as <code>fft</code> is used) and the final result is interpolated by <code>approx</code> . So it almost always makes sense to specify <code>n</code> as a power of two.                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| <code>...</code>     | Further arguments passed on to <code>lattice::diag.panel.splom()</code> and <code>lattice::panel.lines()</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

**See Also**

`lattice::diag.panel.splom()`, `lattice::splom()`, `stats::density()`.

**Examples**

```
splom(~ iris[1:4],
      data = iris,
      diag.panel = diag.panel.splom.density,
      pscales = 0
)
```

---

feldspar

*Ternary feldspar experiments and thermodynamic models*


---

**Description**

A data set that has been manually transcribed from Table 5 of Elkins and Grove's *Ternary feldspar experiments and thermodynamic models*.

**Usage**

```
feldspar
```

**Format**

A data frame of 40 rows and 7 columns:

**Experiment** The ID of the experiment

**Feldspar** Coexisting feldspars, *Alkali* or *Plagioclase*

**Or** Proportion of orthoclase

**An** Proportion of anorthite

**Ab** Proportion of albite  
**Temperature** Temperature of the reaction (degrees centigrade)  
**Pressure** Pressure of the reaction (bars)

**Abstract**

This paper reports the results of 20 experiments in which mixes of two or three feldspars were reacted to produce coexisting plagioclase feldspar (PF) and alkali feldspar (AF). Starting materials with similar bulk compositions were prepared using different combinations of two and three minerals, and experiments were designed to produce similar AF and PF minerals in the experimental products from different starting binary and ternary compositions. The coexisting AF and PF compositions produced as products define compositional fields that are elongate parallel to the ternary solvus. In 11 experiments reaction was sufficient to product fields of coexisting AF and PF, or AF, PF, and melt with a bulk composition close to that of the starting mixture. In six experiments significant reaction occurred in the form of reaction rim overgrowths on seeds of the starting materials. Three experiments produced AF, PF, and melt from a natural granite starting material. A two-feldspar thermometer is presented in which temperature is constrained by equilibria among all three components - Albite, Orthoclase, and Anorthite - in coexisting ternary feldspars.

**Source**

Elkins LT, Grove TL. Ternary feldspar experiments and thermodynamic models. American Mineralogist. 1990;75(5-6):544-59.

---

|                  |                                        |
|------------------|----------------------------------------|
| panel.bubbleplot | <i>Panel Function for Bubble Plots</i> |
|------------------|----------------------------------------|

---

**Description**

Panel Function for Bubble Plots

**Usage**

panel.bubbleplot(x, y, z, groups = NULL, subscripts, cex = NULL, ...)

**Arguments**

|            |                                                                                                              |
|------------|--------------------------------------------------------------------------------------------------------------|
| x, y       | variables to be plotted in the scatterplot                                                                   |
| z          | A numeric vector that areas of circles will be mapped to.                                                    |
| groups     | Grouping variable (see <a href="#">lattice::xyplot()</a> ).                                                  |
| subscripts | A vector of indexes to specify which observation to plot. Normally does not need to be provided by the user. |
| cex        | Is used internally and user settings will be ignored.                                                        |
| ...        | Further arguments to pass to <a href="#">lattice::panel.xyplot()</a> .                                       |

**Value**

Plots a layer inside a panel of a lattice plot.

panel.ci

*Panel function for confidence interval***Description**

Panel function for confidence interval

**Usage**

```

panel.ci(
  x,
  y,
  lower,
  upper,
  groups = NULL,
  subscripts,
  col,
  fill = if (is.null(groups)) plot.line$col else superpose.line$col,
  alpha = 0.15,
  lty = 0,
  lwd = if (is.null(groups)) plot.line$lwd else superpose.line$lwd,
  grid = FALSE,
  ...,
  col.line = if (is.null(groups)) plot.line$col else superpose.line$col
)

```

**Arguments**

|            |                                                                                                                                                                                                                                                                                                                                                                  |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x, y       | variables to be plotted in the scatterplot                                                                                                                                                                                                                                                                                                                       |
| lower      | lower confidence limits                                                                                                                                                                                                                                                                                                                                          |
| upper      | upper confidence limits                                                                                                                                                                                                                                                                                                                                          |
| groups     | an optional grouping variable. If present, <a href="#">panel.superpose</a> will be used instead to display each subgroup                                                                                                                                                                                                                                         |
| subscripts | see <a href="#">lattice::xyplot()</a>                                                                                                                                                                                                                                                                                                                            |
| col        | line color                                                                                                                                                                                                                                                                                                                                                       |
| fill       | fill color                                                                                                                                                                                                                                                                                                                                                       |
| alpha      | opacity for the fill                                                                                                                                                                                                                                                                                                                                             |
| lty        | line type                                                                                                                                                                                                                                                                                                                                                        |
| lwd        | line width                                                                                                                                                                                                                                                                                                                                                       |
| grid       | A logical flag, character string, or list specifying whether and how a background grid should be drawn. This provides the same functionality as type="g", but is the preferred alternative as the effect type="g" is conceptually different from that of other type values (which are all data-dependent). Using the grid argument also allows more flexibility. |

Most generally, `grid` can be a list of arguments to be supplied to `panel.grid`, which is called with those arguments. Three shortcuts are available:

`TRUE`: roughly equivalent to `list(h = -1, v = -1)`

`"h"`: roughly equivalent to `list(h = -1, v = 0)`

`"v"`: roughly equivalent to `list(h = 0, v = -1)`

No grid is drawn if `grid = FALSE`.

... Extra arguments, if any, for `panel.xyplot`. Usually passed on as graphical parameters to low level plotting functions, or to the panel functions performing smoothing, if applicable.

`col.line` line color. Supersedes `col` if both are specified.

## Examples

```
mod <- lm(Petal.Width ~ Petal.Length * Species, data = iris)
newdat <- expand.grid(
  Petal.Length = seq(1, 7, by = 0.1),
  Species = c("setosa", "versicolor", "virginica")
)
pred <- predict(mod, newdat, interval = "confidence")
dd <- cbind(newdat, pred)

xyplot(
  fit ~ Petal.Length,
  groups = Species, data = dd,
  prepanel = prepanel.ci, auto.key = list(lines = TRUE, points = FALSE),
  ylab = "Petal Width",
  xlab = "Petal Length",
  lower = dd$lwr, upper = dd$upr, type = "l",
  panel = function(...) {
    panel.ci(..., alpha = 0.15, grid = TRUE)
    panel.xyplot(...)
  }
)
```

---

panel.qqmathci

*Q-Q Diagram Confidence Intervals Panels*

---

## Description

Panel function to go along with `lattice::qqmath()` and `lattice::panel.qqmathline()`. Adds filled confidence bands to the Q-Q-plot.

**Usage**

```

panel.qqmathci(
  x,
  y = x,
  distribution = qnorm,
  probs = c(0.25, 0.75),
  qtype = 7,
  groups = NULL,
  ci = 0.95,
  alpha = 0.25,
  col = trellis.par.get("plot.line")$col,
  ...,
  col.line
)

```

**Arguments**

|                           |                                                                                                                                                 |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>            | The original sample, possibly reduced to a fewer number of quantiles, as determined by the <code>f.value</code> argument to <code>qqmath</code> |
| <code>y</code>            | an alias for <code>x</code> for backwards compatibility                                                                                         |
| <code>distribution</code> | quantile function for reference theoretical distribution.                                                                                       |
| <code>probs</code>        | numeric vector of length two, representing probabilities. Corresponding quantile pairs define the line drawn.                                   |
| <code>qtype</code>        | the type of quantile computation used in <a href="#">quantile</a>                                                                               |
| <code>groups</code>       | optional grouping variable. If non-null, a line will be drawn for each group.                                                                   |
| <code>ci</code>           | Confidence level                                                                                                                                |
| <code>alpha</code>        | Alpha level for the color fill                                                                                                                  |
| <code>col</code>          | Color fill for the confidence bands.                                                                                                            |
| <code>...</code>          | Arguments passed to <a href="#">lattice::panel.superpose()</a> and <a href="#">lattice::panel.polygon()</a>                                     |
| <code>col.line</code>     | Color fill for the confidence bands. Is used internally by <a href="#">lattice::panel.superpose()</a> and should generally not be changed.      |

**Details**

The function tries to figure out the density function counterpart to that provided in the argument `distribution` by regular expressions.

**Value**

Augments a trellis plot panel, such as that created by [lattice::qqmath\(\)](#), with confidence levels.

**Author(s)**

Johan Larsson.

**See Also**

[lattice::panel.qqmathline\(\)](#), [lattice::qqmath\(\)](#), and [lattice::panel.qqmath\(\)](#).

**Examples**

```
qqmath(~ height | voice.part, aspect = "xy", data = singer,
  prepanel = prepanel.qqmathline,
  panel = function(x, ...) {
    panel.qqmathci(x, ...)
    panel.qqmathline(x, ...)
    panel.qqmath(x, ...)
  })
```

---

|                   |                                         |
|-------------------|-----------------------------------------|
| panel.ternaryplot | <i>Panel Function for Ternary Plots</i> |
|-------------------|-----------------------------------------|

---

**Description**

Panel Function for Ternary Plots

**Usage**

```
panel.ternaryplot(
  x,
  y,
  z,
  subscripts,
  response = NULL,
  density = FALSE,
  region = density || !is.null(response),
  contour = density || !is.null(response),
  labels = !is.null(response),
  points = TRUE,
  grid = TRUE,
  density_breaks = NULL,
  xlab,
  ylab,
  zlab,
  xlab.default,
  ylab.default,
  zlab.default,
  ...
)
```

**Arguments**

|                |                                                                                         |
|----------------|-----------------------------------------------------------------------------------------|
| x              | Numeric vector                                                                          |
| y              | Numeric vector                                                                          |
| z              | Numeric vector                                                                          |
| subscripts     | See <a href="#">lattice::panel.xyplot()</a> .                                           |
| response       | An optional response variable                                                           |
| density        | Compute two-dimensional density estimates via <a href="#">MASS::kde2d()</a> .           |
| region         | Fill density or response estimates with a color gradient.                               |
| contour        | Draw contour lines for density and response estimates.                                  |
| labels         | Label contour lines.                                                                    |
| points         | Draw points (via <a href="#">panel.ternaryplot.xyplot()</a> ).                          |
| grid           | Draw a reference grid.                                                                  |
| density_breaks | Breaks for the density plot if used (see <a href="#">panel.ternaryplot.density()</a> ). |
| xlab           | X axis label (the left dimension)                                                       |
| ylab           | Y axis label (the right dimension)                                                      |
| zlab           | Z axis label (the top dimension)                                                        |
| xlab.default   | Internal argument                                                                       |
| ylab.default   | Internal argument                                                                       |
| zlab.default   | Internal argument                                                                       |
| ...            | Arguments passed down to subsequent panel functions.                                    |

**Value**

Plots a layer inside a panel of a lattice plot.

**See Also**

The building blocks of this function are available as the separate panel functions [panel.ternaryplot.xyplot\(\)](#), [panel.ternaryplot.grid\(\)](#), [panel.ternaryplot.scales\(\)](#), [panel.ternaryplot.clip\(\)](#), [panel.ternaryplot.respo](#) and [panel.ternaryplot.density\(\)](#) in case the user would like to get complete control of the customization.

---

panel.ternaryplot.clip

*Plot Region Clipping for Ternary Plots*

---

**Description**

Plot Region Clipping for Ternary Plots

**Usage**

```
panel.ternaryplot.clip(
  xl = current.panel.limits()$x,
  yl = current.panel.limits()$y,
  border = "transparent",
  col = if (background$col == "transparent") "#FFFFFF" else background$col
)
```

**Arguments**

|        |               |
|--------|---------------|
| xl     | X axis limits |
| yl     | Y axis limits |
| border | Border color  |
| col    | Polygon fill  |

**Value**

Plots a layer inside a panel of a lattice plot.

---

panel.ternaryplot.density

*Two-Dimensional Density Estimation for Ternary Plots*

---

**Description**

Two-Dimensional Density Estimation for Ternary Plots

**Usage**

```
panel.ternaryplot.density(
  x,
  y,
  z,
  subscripts,
  n = 100,
  region = TRUE,
  contour = FALSE,
  labels = isTRUE(contour),
  density_breaks = NULL,
  ...
)
```

**Arguments**

|                |                                                                                                                                                                     |
|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x              | Numeric vector                                                                                                                                                      |
| y              | Numeric vector                                                                                                                                                      |
| z              | Numeric vector                                                                                                                                                      |
| subscripts     | See <a href="#">lattice::panel.xyplot()</a> .                                                                                                                       |
| n              | Number of grid points in each direction. Can be scalar or a length-2 integer vector.                                                                                |
| region         | Fill density or response estimates with a color gradient.                                                                                                           |
| contour        | Draw contour lines for density and response estimates.                                                                                                              |
| labels         | Label contour lines.                                                                                                                                                |
| density_breaks | Breaks for the density plot if used (see <a href="#">panel.ternaryplot.density()</a> ).                                                                             |
| ...            | Arguments that will be passed on to <a href="#">lattice::panel.lines()</a> , <a href="#">lattice::panel.polygon()</a> , and <a href="#">lattice::panel.text()</a> . |

**Value**

Plots a layer inside a panel of a lattice plot.

---

panel.ternaryplot.grid

*Reference Grid for Ternary Plot*

---

**Description**

Reference Grid for Ternary Plot

**Usage**

```
panel.ternaryplot.grid(
  at = seq.int(0, 1, by = 0.2),
  alpha = reference.line$alpha,
  col = reference.line$col,
  lty = reference.line$lty,
  lwd = reference.line$lwd
)
```

**Arguments**

|       |                                   |
|-------|-----------------------------------|
| at    | Where to draw the reference lines |
| alpha | Alpha                             |
| col   | Color                             |
| lty   | Line type                         |
| lwd   | Line weight                       |

**Value**

Plots a layer inside a panel of a lattice plot.

---

panel.ternaryplot.response

*Response Panels for Ternary Plots*

---

**Description**

Response Panels for Ternary Plots

**Usage**

```
panel.ternaryplot.response(
  x,
  y,
  z,
  subscripts,
  response,
  region = TRUE,
  contour = TRUE,
  labels = isTRUE(contour),
  fun = c("glm", "lm"),
  formula = response ~ poly(x, y),
  ...
)
```

**Arguments**

|            |                                                                                                                                                    |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------|
| x          | Numeric vector                                                                                                                                     |
| y          | Numeric vector                                                                                                                                     |
| z          | Numeric vector                                                                                                                                     |
| subscripts | See <a href="#">lattice::panel.xyplot()</a> .                                                                                                      |
| response   | An optional response variable                                                                                                                      |
| region     | Fill density or response estimates with a color gradient.                                                                                          |
| contour    | Draw contour lines for density and response estimates.                                                                                             |
| labels     | Label contour lines.                                                                                                                               |
| fun        | Function to apply to the response variable.                                                                                                        |
| formula    | Formula for the function.                                                                                                                          |
| ...        | Arguments passed on to <a href="#">lattice::panel.lines()</a> , <a href="#">lattice::panel.polygon()</a> , <a href="#">lattice::panel.text()</a> . |

**Value**

Plots a layer inside a panel of a lattice plot.

---

```
panel.ternaryplot.scales
```

*Axes and Labels for Ternary Plots*

---

### Description

Axes and Labels for Ternary Plots

### Usage

```
panel.ternaryplot.scales(
  xlab,
  ylab,
  zlab,
  xlab.default,
  ylab.default,
  zlab.default,
  at = seq.int(0, 1, by = 0.2),
  ...
)
```

### Arguments

|                  |                                                                                                                                                |
|------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| xlab, ylab, zlab | Labels, have to be lists. Typically the user will not manipulate these, but instead control this via arguments to <code>cloud</code> directly. |
| xlab.default     | for internal use                                                                                                                               |
| ylab.default     | for internal use                                                                                                                               |
| zlab.default     | for internal use                                                                                                                               |
| at               | Where to draw tick marks.                                                                                                                      |
| ...              | Currently ignored.                                                                                                                             |

### Value

Plots a layer inside a panel of a lattice plot.

---

```
panel.ternaryplot.xyplot
```

*Ternary Plot Wrapper for `lattice::xyplot`*

---

### Description

This mainly exists to enable users to string together their own ternary plot functions.

**Usage**

```
panel.ternaryplot.xyplot(x, y, z, subscripts, ...)
```

**Arguments**

|            |                                                                           |
|------------|---------------------------------------------------------------------------|
| x          | Numeric vector of values in the original space                            |
| y          | Numeric vector of values in the original space                            |
| z          | Numeric vector of values in the original space                            |
| subscripts | see <a href="#">lattice::xyplot()</a> .                                   |
| ...        | Arguments that are passed on to <a href="#">lattice::panel.xyplot()</a> . |

**Value**

Plots a layer inside a panel of a lattice plot.

---

|             |                            |
|-------------|----------------------------|
| prepanel.ci | <i>Prepanel for ciplot</i> |
|-------------|----------------------------|

---

**Description**

Prepanel for ciplot

**Usage**

```
prepanel.ci(x, y, lower, upper, subscripts, groups = NULL, ...)
```

**Arguments**

|                    |                                                                                                                       |
|--------------------|-----------------------------------------------------------------------------------------------------------------------|
| x, y               | x and y values, numeric or factor                                                                                     |
| lower              | lower confidence limits                                                                                               |
| upper              | upper confidence limits                                                                                               |
| groups, subscripts | See <a href="#">xyplot</a> . Whenever appropriate, calculations are done separately for each group and then combined. |
| ...                | other arguments, usually ignored                                                                                      |

**Examples**

```
mod <- lm(Petal.Width ~ Petal.Length * Species, data = iris)
newdat <- expand.grid(
  Petal.Length = seq(1, 7, by = 0.1),
  Species = c("setosa", "versicolor", "virginica")
)
pred <- predict(mod, newdat, interval = "confidence")
dd <- cbind(newdat, pred)
```

```

xyplot(
  fit ~ Petal.Length,
  groups = Species, data = dd,
  prepanel = prepanel.ci,
  ylab = "Petal Width",
  xlab = "Petal Length",
  lower = dd$lwr, upper = dd$upr, alpha = 0.3,
  panel = function(...) {
    panel.ci(..., grid = TRUE)
    panel.xyplot(type = "l", ...)
  }
)

```

---

qqmath.zoo

*Q-Q Plots for Zoo Objects*


---

### Description

Draw quantile-Quantile plots of a sample against a theoretical distribution, possibly conditioned on other variables.

### Usage

```

## S3 method for class 'zoo'
qqmath(
  x,
  data = NULL,
  xlab = "Theoretical quantiles",
  ylab = "Sample quantiles",
  ref = TRUE,
  ci = TRUE,
  ...
)

```

### Arguments

|      |                                                                      |
|------|----------------------------------------------------------------------|
| x    | A zoo object                                                         |
| data | Ignored                                                              |
| xlab | X axis label                                                         |
| ylab | Y axis label                                                         |
| ref  | Plot a reference line via <code>lattice::panel.qqmathline()</code> . |
| ci   | Plot confidence levels via <code>panel.qqmathci()</code> .           |
| ...  | Parameters to pass on to <code>lattice::qqmath()</code> .            |

**Value**

Plots and returns a trellis object.

**Author(s)**

Original by Deepayan Sarkar.

**See Also**

`lattice::qqmath()`, `zoo::zoo()`, `lattice::panel.qqmathline()`.

**Examples**

```
if (require(zoo))
  qqmath(zoo(lh))
```

---

|               |                      |
|---------------|----------------------|
| tactile.theme | <i>Tactile Theme</i> |
|---------------|----------------------|

---

**Description**

A custom theme for lattice that tries to make away with some of the (in this author's opinion) excessive margins that result from the default settings. It also provides a different color theme based partly on `latticeExtra::custom.theme()`.

**Usage**

```
tactile.theme(fontsize = c(12, 8), color = TRUE, ...)
```

**Arguments**

|          |                                                                    |
|----------|--------------------------------------------------------------------|
| fontsize | A vector of two numeric scalars for text and symbols respectively. |
| color    | Colorized theme.                                                   |
| ...      | Additional named options.                                          |

**Details**

The theme currently modifies the default lattice theme so that

- paddings (margins) are minimized,
- axis tick lengths are halved, and
- title size is decreased *slightly*.

**Value**

A list of graphical parameters that for instance could be supplied inside a call to `lattice::xyplot()` or set via `lattice::lattice.options()`.

## Examples

```
xyplot(speed ~ dist, data = cars, par.settings = tactile.theme())
opars <- trellis.par.get()
trellis.par.set(tactile.theme())
show.settings()
trellis.par.set(opars)
```

---

ternaryplot

*Ternary Plot*


---

## Description

A ternary plot is a triangular diagram that displays proportions of three variables. It can be used to map three-dimensional data to a two-dimensional surface with the caveat that the data's original scales are lost (unless it was proportional data to begin with).#'

## Usage

```
ternaryplot(x, data, ...)
```

## S3 method for class 'formula'

```
ternaryplot(
  x,
  data = NULL,
  response = NULL,
  groups = NULL,
  density = FALSE,
  region = density || !is.null(response),
  contour = density || !is.null(response),
  labels = !is.null(response),
  colorkey = region,
  xlab,
  ylab,
  zlab,
  xlim = c(-0.15, 1.15),
  ylim = c(-0.3, 1),
  panel = panel.ternaryplot,
  default.prepanel = lattice.getOption("prepanel.default.xyplot"),
  drop.unused.levels = lattice.getOption("drop.unused.levels"),
  subset = TRUE,
  ...
)
```

## S3 method for class 'data.frame'

```
ternaryplot(x, data = NULL, ...)
```

## S3 method for class 'matrix'

```
ternaryplot(x, data = NULL, ...)
```

**Arguments**

|                                 |                                                                                                                                                                    |
|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>                  | See <b>Methods (by class)</b> .                                                                                                                                    |
| <code>data</code>               | A data frame in which the formula, groups, and conditioning variables are evaluated.                                                                               |
| <code>...</code>                | Arguments that are passed on to other methods, particularly <code>panel.ternaryplot()</code> .                                                                     |
| <code>response</code>           | An optional response variable                                                                                                                                      |
| <code>groups</code>             | A variable or expression to be evaluated in data and used to distinguish groups by varying graphical parameters.                                                   |
| <code>density</code>            | Compute two-dimensional density estimates via <code>MASS::kde2d()</code> .                                                                                         |
| <code>region</code>             | Fill density or response estimates with a color gradient.                                                                                                          |
| <code>contour</code>            | Draw contour lines for density and response estimates.                                                                                                             |
| <code>labels</code>             | Label contour lines.                                                                                                                                               |
| <code>colorkey</code>           | if TRUE automatically computes a colorkey for density or response estimates. Can also be a list (see <code>lattice::levelplot()</code> for details on this).       |
| <code>xlab</code>               | X axis label (the left dimension)                                                                                                                                  |
| <code>ylab</code>               | Y axis label (the right dimension)                                                                                                                                 |
| <code>zlab</code>               | Z axis label (the top dimension)                                                                                                                                   |
| <code>xlim</code>               | X limits for the plot region.                                                                                                                                      |
| <code>ylim</code>               | Y limits for the plot region.                                                                                                                                      |
| <code>panel</code>              | The panel function.                                                                                                                                                |
| <code>default.prepanel</code>   | The default prepanel function.                                                                                                                                     |
| <code>drop.unused.levels</code> | Drop unused conditioning or groups levels.                                                                                                                         |
| <code>subset</code>             | An expression that evaluates to a logical or integer indexing vector. Like groups, it is evaluated in data. Only the resulting rows of data are used for the plot. |

**Value**

An object of class "trellis". The `update` method can be used to update components of the object and the `print` method (usually called by default) will plot it on an appropriate plotting device.

**Methods (by class)**

- `ternaryplot(formula)`: A formula of the form `top ~ left * right`. Variables will be evaluated inside data if provided.
- `ternaryplot(data.frame)`: A data frame for which the first three columns will be mapped to the *left*, *right*, and *top* dimensions of the ternary plot respectively.
- `ternaryplot(matrix)`: A matrix for which the first three columns will be mapped to the *left*, *right*, and *top* dimensions of the ternary plot respectively.

**Examples**

```
ternaryplot(Fertility ~ Agriculture * Catholic, data = swiss)
ternaryplot(Catholic ~ Examination * Education, response = Infant.Mortality,
            data = swiss, contour = FALSE)

ternaryplot(Or ~ An * Ab | Feldspar, data = feldspar)

ternaryplot(Or ~ An * Ab, groups = Feldspar, data = feldspar, density = TRUE)
```

xyplot.acf

*Plot Autocovariance and Autocorrelation Functions***Description**

This is a version of `stats::plot.acf()`.

**Usage**

```
## S3 method for class 'acf'
xyplot(
  x,
  data = NULL,
  ci = 0.95,
  ci_type = c("white", "ma"),
  ci_col = trellis.par.get("add.line")$col,
  ci_lty = 2,
  ...
)
```

**Arguments**

|                      |                                                         |
|----------------------|---------------------------------------------------------|
| <code>x</code>       | An 'acf' object.                                        |
| <code>data</code>    | Ignored                                                 |
| <code>ci</code>      | Confidence level.                                       |
| <code>ci_type</code> | Type of confidence level.                               |
| <code>ci_col</code>  | Line color for the confidence levels.                   |
| <code>ci_lty</code>  | Line type for the confidence levels.                    |
| <code>...</code>     | Arguments passed on to <code>lattice::xyplot()</code> . |

**Value**

Returns and plots a trellis object.

**Author(s)**

Original by Brian Ripley.

**See Also**

[lattice::xyplot\(\)](#), [stats::plot.acf\(\)](#), [stats::acf\(\)](#).

**Examples**

```
z <- ts(matrix(rnorm(400), 100, 4), start = c(1961, 1), frequency = 12)
xyplot(acf(z))
```

---

xyplot.Arima

*Diagnostic Plots for ARIMA Models*


---

**Description**

Diagnostic plots modelled after [stats::tsdiag\(\)](#) with some modifications and corrections of p-values in the Box–Ljung test.

**Usage**

```
## S3 method for class 'Arima'
xyplot(
  x,
  data = NULL,
  which = 1:4,
  lag.max = NULL,
  gof.lag = NULL,
  qq.aspect = "iso",
  na.action = na.pass,
  main = NULL,
  layout = NULL,
  ...
)
```

**Arguments**

|           |                                                                                                                                                                                                                      |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x         | A fitted time-series model of class Arima.                                                                                                                                                                           |
| data      | Ignored                                                                                                                                                                                                              |
| which     | A sequence of integers between 1 and 4, identifying the plots to be shown.                                                                                                                                           |
| lag.max   | Number of lags to compute ACF for.                                                                                                                                                                                   |
| gof.lag   | The maximum number of lags for the Ljung–Box test.                                                                                                                                                                   |
| qq.aspect | Aspect of Q-Q plot (see <a href="#">lattice::qqmath()</a> ).                                                                                                                                                         |
| na.action | Treatment of NAs.                                                                                                                                                                                                    |
| main      | Optional titles for the plots. Can also be TRUE, in which case a default list of titles will be added.                                                                                                               |
| layout    | Either a numeric vector with (columns, rows) to use in the call to <a href="#">gridExtra::grid.arrange()</a> , or a layout matrix which will then be passed as the layout_matrix in <a href="#">grid.arrange()</a> . |
| ...       | Parameters to pass to <a href="#">lattice::xyplot()</a> .                                                                                                                                                            |

**Value**

Plots a lattice plot and returns a trellis object.

**See Also**

[stats::tsdiag\(\)](#), [stats::arima\(\)](#), [lattice::xyplot\(\)](#), [gridExtra::grid.arrange\(\)](#), [stats::Box.test\(\)](#).

**Examples**

```
fit <- arima(lh, order = c(1, 1, 0))
xyplot(fit, layout = c(2, 2))
xyplot(fit, which = c(1:2, 4), layout = rbind(c(1, 1), c(2, 3)))
```

---

xyplot.forecast

---

*Plot Forecasts with Trellis Graphics*


---

**Description**

Plot forecasts from [forecast::forecast\(\)](#). It is built mostly to resemble the [forecast::autoplot.forecast\(\)](#) and [forecast::plot.forecast\(\)](#) functions, but in addition tries to plot the predictions on the original scale.

**Usage**

```
## S3 method for class 'forecast'
xyplot(
  x,
  data = NULL,
  ci = TRUE,
  ci_levels = x$level,
  ci_key = ci,
  ci_pal = hcl(0, 0, 45:100),
  ci_alpha = trellis.par.get("regions")$alpha,
  ...
)
```

**Arguments**

|           |                                                                                                                                                                                                        |
|-----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| x         | An object of class forecast.                                                                                                                                                                           |
| data      | Data of observations left out of the model fit, usually "future" observations.                                                                                                                         |
| ci        | Plot confidence intervals for the predictions.                                                                                                                                                         |
| ci_levels | The prediction levels to plot as a subset of those forecasted in x.                                                                                                                                    |
| ci_key    | Set to TRUE to draw a key automatically or provide a list (if length(ci_levels) > 5 should work with <a href="#">lattice::draw.colorkey()</a> and otherwise with <a href="#">lattice::draw.key()</a> ) |
| ci_pal    | Color palette for the confidence bands.                                                                                                                                                                |
| ci_alpha  | Fill alpha for the confidence interval.                                                                                                                                                                |
| ...       | Arguments passed on to <a href="#">lattice::panel.xyplot()</a> .                                                                                                                                       |

## Details

This function requires the **zoo** package.

## Value

An object of class "trellis". The `update` method can be used to update components of the object and the `print` method (usually called by default) will plot it on an appropriate plotting device.

## See Also

`lattice::panel.xyplot()`, `forecast::forecast()`, `lattice::xyplot.ts()`.

## Examples

```
if (require(forecast)) {
  train <- window(USAccDeaths, c(1973, 1), c(1977, 12))
  test <- window(USAccDeaths, c(1978, 1), c(1978, 12))
  fit <- arima(train, order = c(0, 1, 1),
               seasonal = list(order = c(0, 1, 1)))
  fcast1 <- forecast(fit, 12)
  xyplot(fcast1, test, grid = TRUE, auto.key = list(corner = c(0, 0.99)),
         ci_key = list(title = "PI Level"))

  # A fan plot
  fcast2 <- forecast(fit, 12, level = seq(0, 95, 10))
  xyplot(fcast2, test, ci_pal = heat.colors(100))
}
```

---

xyplot.lm

---

*Lattice plot diagnostics for lm objects*


---

## Description

Lattice plot diagnostics for `lm` objects, mostly mimicking the behavior of `stats::plot.lm()` but based on `lattice::xyplot()` instead.

## Usage

```
## S3 method for class 'lm'
xyplot(
  x,
  data = NULL,
  which = c(1:3, 5),
  main = FALSE,
  id.n = 3,
  labels.id = names(residuals(x)),
  cex.id = 0.75,
  cook.levels = c(0.5, 1),
```

```

    label.pos = c(4, 2),
    layout = NULL,
    ...
  )

```

### Arguments

|                          |                                                                                                                                                                                                                      |
|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>x</code>           | lm object, typically result of <code>lm</code> or <code>glm</code> .                                                                                                                                                 |
| <code>data</code>        | Only provided for method consistency and is ignored.                                                                                                                                                                 |
| <code>which</code>       | if a subset of the plots is required, specify a subset of the numbers 1:6                                                                                                                                            |
| <code>main</code>        | if TRUE plots default titles. Can also be a list or character vector of length 6.                                                                                                                                    |
| <code>id.n</code>        | number of points to be labelled in each plot, starting with the most extreme.                                                                                                                                        |
| <code>labels.id</code>   | vector of labels, from which the labels for extreme points will be chosen. NULL uses observation numbers.                                                                                                            |
| <code>cex.id</code>      | magnification of point labels.                                                                                                                                                                                       |
| <code>cook.levels</code> | levels of Cook's distance at which to draw contours.                                                                                                                                                                 |
| <code>label.pos</code>   | positioning of labels, for the left half and right half of the graph respectively, for plots 1-3, 5, 6.                                                                                                              |
| <code>layout</code>      | a numeric vector with [columns, rows] to use in the call to <code>gridExtra::grid.arrange()</code> , or a layout matrix which will then be passed as the <code>layout_matrix</code> in <code>grid.arrange()</code> . |
| <code>...</code>         | arguments to be passed to <code>lattice::xyplot()</code> .                                                                                                                                                           |

### Value

A list of trellis objects or a single trellis object.

### Author(s)

Original by John Maindonald and Martin Maechler. Adaptation to lattice by Johan Larsson.

### See Also

`stats::lm()`, `stats::plot.lm()`, `lattice::xyplot()`

### Examples

```

fit <- lm(Sepal.Length ~ Sepal.Width, data = iris)
xyplot(fit)
xyplot(fit, which = 5)

```

# Index

## \* datasets

feldspar, 6

approx, 6

bubbleplot, 2

bw.nrd, 5

bwplot2, 4

diag.panel.splom.density, 5

feldspar, 6

fft, 6

forecast::autoplot.forecast(), 24

forecast::forecast(), 24, 25

forecast::plot.forecast(), 24

glm, 26

gridExtra::grid.arrange(), 23, 24, 26

lattice::bwplot(), 4

lattice::diag.panel.splom(), 6

lattice::draw.colorkey(), 24

lattice::draw.key(), 3, 24

lattice::lattice.options(), 19

lattice::levelplot(), 21

lattice::panel.lines(), 6, 14, 15

lattice::panel.polygon(), 10, 14, 15

lattice::panel.qqmath(), 11

lattice::panel.qqmathline(), 9, 11, 18, 19

lattice::panel.superpose(), 10

lattice::panel.text(), 14, 15

lattice::panel.xyplot(), 7, 12, 14, 15, 17, 24, 25

lattice::qqmath(), 9–11, 18, 19, 23

lattice::splom(), 5, 6

lattice::xyplot(), 3, 7, 8, 17, 19, 22–26

lattice::xyplot.ts(), 25

latticeExtra::custom.theme(), 19

lm, 26

MASS::kde2d(), 12, 21

panel.bubbleplot, 7

panel.bubbleplot(), 3

panel.ci, 8

panel.grid, 9

panel.qqmathci, 9

panel.qqmathci(), 18

panel.superpose, 8

panel.ternaryplot, 11

panel.ternaryplot(), 21

panel.ternaryplot.clip, 12

panel.ternaryplot.clip(), 12

panel.ternaryplot.density, 13

panel.ternaryplot.density(), 12, 14

panel.ternaryplot.grid, 14

panel.ternaryplot.grid(), 12

panel.ternaryplot.response, 15

panel.ternaryplot.response(), 12

panel.ternaryplot.scales, 16

panel.ternaryplot.scales(), 12

panel.ternaryplot.xyplot, 16

panel.ternaryplot.xyplot(), 12

prepanel.ci, 17

print, 3, 4, 21, 25

qqmath.zoo, 18

quantile, 10

stats::acf(), 23

stats::arima(), 24

stats::Box.test(), 24

stats::density(), 5, 6

stats::lm(), 26

stats::plot.acf(), 22, 23

stats::plot.lm(), 25, 26

stats::tsdiag(), 23, 24

tactile.theme, 19

ternaryplot, 20

update, [3](#), [4](#), [21](#), [25](#)

xyplot, [17](#)

xyplot.acf, [22](#)

xyplot.Arima, [23](#)

xyplot.forecast, [24](#)

xyplot.lm, [25](#)

zoo::zoo(), [19](#)