plot {graphics} R Documentation

Generic X-Y Plotting

Description

Generic function for plotting of **R** objects. For more details about the graphical parameter arguments, see `par`.

For simple scatter plots, `plot.default` will be used. However, there are `plot` methods for many **R** objects, including `function`s, `data.frame`s, `density` objects, etc. Use `methods(plot)` and the documentation for these.

Usage

```
plot(x, y, ...)
```

Arguments

`x`

the coordinates of points in the plot. Alternatively, a single plotting structure, function or *any **R** object with a `plot` method* can be provided.

`y`

the y coordinates of points in the plot, *optional* if `x` is an appropriate structure.

`...`

Arguments to be passed to methods, such as graphical parameters (see `par`). Many methods will accept the following arguments:

`type`

what type of plot should be drawn. Possible types are

- `"p"` for **p**oints,
- `"l"` for **l**ines,
- `"b"` for **b**oth,
- `"c"` for the lines part alone of `"b"`,
- `"o"` for both '**o**verplotted',
- `"h"` for '**h**istogram' like (or 'high-density') vertical lines,
- `"s"` for stair **s**teps,
- `"S"` for other **s**teps, see 'Details' below,
- `"n"` for no plotting.

All other `type`s give a warning or an error; using, e.g., `type = "punkte"` being equivalent to `type`

`= "p"` for S compatibility. Note that some methods, e.g. `plot.factor`, do not accept this.

`main`

an overall title for the plot: see `title`.

`sub`

a sub title for the plot: see `title`.

`xlab`

a title for the x axis: see `title`.

`ylab`

a title for the y axis: see `title`.

`asp`

the *y/x* aspect ratio, see `plot.window`.

Details

The two step types differ in their x-y preference: Going from *(x1,y1)* to *(x2,y2)* with *x1 < x2*, `type = "s"` moves first horizontal, then vertical, whereas `type = "S"` moves the other way around.

See Also

`plot.default`, `plot.formula` and other methods; `points`, `lines`, `par`. For thousands of points, consider using `smoothScatter()` instead of `plot()`.

For X-Y-Z plotting see `contour`, `persp` and `image`.

Examples

```
require(stats) # for lowess, rpois, rnorm
plot(cars)
lines(lowess(cars))

plot(sin, -pi, 2*pi) # see ?plot.function

## Discrete Distribution Plot:
plot(table(rpois(100, 5)), type = "h", col = "red", lwd = 10,
     main = "rpois(100, lambda = 5)")

## Simple quantiles/ECDF, see ecdf() {library(stats)} for a better one:
plot(x <- sort(rnorm(47)), type = "s", main = "plot(x, type = \"s\")")
points(x, cex = .5, col = "dark red")
```

[Package *graphics* version 3.2.2 Index]