legend {graphics} R Documentation

Add Legends to Plots

Description

This function can be used to add legends to plots. Note that a call to the function `locator(1)` can be used in place of the `x` and `y` arguments.

Usage

```
legend(x, y = NULL, legend, fill = NULL, col = par("col"),
       border = "black", lty, lwd, pch,
       angle = 45, density = NULL, bty = "o", bg = par("bg"),
       box.lwd = par("lwd"), box.lty = par("lty"), box.col = par("fg"),
       pt.bg = NA, cex = 1, pt.cex = cex, pt.lwd = lwd,
       xjust = 0, yjust = 1, x.intersp = 1, y.intersp = 1,
       adj = c(0, 0.5), text.width = NULL, text.col = par("col"),
       text.font = NULL, merge = do.lines && has.pch, trace = FALSE,
       plot = TRUE, ncol = 1, horiz = FALSE, title = NULL,
       inset = 0, xpd, title.col = text.col, title.adj = 0.5,
       seg.len = 2)
```

Arguments

`x, y`
: the x and y co-ordinates to be used to position the legend. They can be specified by keyword or in any way which is accepted by `xy.coords`: See ‘Details’.

`legend`
: a character or expression vector of length ≥ *1* to appear in the legend. Other objects will be coerced by `as.graphicsAnnot`.

`fill`
: if specified, this argument will cause boxes filled with the specified colors (or shaded in the specified colors) to appear beside the legend text.

`col`
: the color of points or lines appearing in the legend.

`border`
: the border color for the boxes (used only if `fill` is specified).

`lty, lwd`
: the line types and widths for lines appearing in the legend. One of these two *must* be specified for line drawing.

`pch`
: the plotting symbols appearing in the legend, as numeric vector or a vector of 1-character strings (see `points`). Unlike `points`, this can all be specified as a single multi-character string. *Must* be specified for symbol drawing.

`angle`
: angle of shading lines.

`density` the density of shading lines, if numeric and positive. If `NULL` or negative or `NA` color filling is assumed.

`bty` the type of box to be drawn around the legend. The allowed values are `"o"` (the default) and `"n"`.

`bg` the background color for the legend box. (Note that this is only used if `bty != "n"`.)

`box.lty`, `box.lwd`, `box.col` the line type, width and color for the legend box (if `bty = "o"`).

`pt.bg` the background color for the `points`, corresponding to its argument `bg`.

`cex` character expansion factor **relative** to current `par("cex")`. Used for text, and provides the default for `pt.cex` and `title.cex`.

`pt.cex` expansion factor(s) for the points.

`pt.lwd` line width for the points, defaults to the one for lines, or if that is not set, to `par("lwd")`.

`xjust` how the legend is to be justified relative to the legend x location. A value of 0 means left justified, 0.5 means centered and 1 means right justified.

`yjust` the same as `xjust` for the legend y location.

`x.intersp` character interspacing factor for horizontal (x) spacing.

`y.intersp` the same for vertical (y) line distances.

`adj` numeric of length 1 or 2; the string adjustment for legend text. Useful for y-adjustment when `labels` are plotmath expressions.

`text.width` the width of the legend text in x (`"user"`) coordinates. (Should be positive even for a reversed x axis.) Defaults to the proper value computed by `strwidth(legend)`.

`text.col` the color used for the legend text.

`text.font` the font used for the legend text, see `text`.

`merge` logical; if `TRUE`, merge points and lines but not filled boxes. Defaults to `TRUE` if there are points and lines.

`trace` logical; if `TRUE`, shows how `legend` does all its magical computations.

`plot` logical. If `FALSE`, nothing is plotted but the sizes are returned.

`ncol` the number of columns in which to set the legend items (default is 1, a vertical legend).

`horiz` logical; if `TRUE`, set the legend horizontally rather than vertically (specifying `horiz` overrides the `ncol` specification).

`title` a character string or length-one expression giving a title to be placed at the top of the legend. Other objects will be coerced by `as.graphicsAnnot`.

`inset` inset distance(s) from the margins as a fraction of the plot region when legend is placed by keyword.

`xpd` if supplied, a value of the graphical parameter `xpd` to be used while the legend is being drawn.

`title.col` color for `title`.

`title.adj` horizontal adjustment for `title`: see the help for `par("adj")`.

`seg.len` the length of lines drawn to illustrate `lty` and/or `lwd` (in units of character widths).

Details

Arguments `x`, `y`, `legend` are interpreted in a non-standard way to allow the coordinates to be specified *via* one or two arguments. If `legend` is missing and `y` is not numeric, it is assumed that the second argument is intended to be `legend` and that the first argument specifies the coordinates.

The coordinates can be specified in any way which is accepted by `xy.coords`. If this gives the coordinates of one point, it is used as the top-left coordinate of the rectangle containing the legend. If it gives the coordinates of two points, these specify opposite corners of the rectangle (either pair of corners, in any order).

The location may also be specified by setting `x` to a single keyword from the list `"bottomright"`, `"bottom"`, `"bottomleft"`, `"left"`, `"topleft"`, `"top"`, `"topright"`, `"right"` and `"center"`. This places the legend on the inside of the plot frame at the given location. Partial argument matching is used. The optional `inset` argument specifies how far the legend is inset from the plot margins. If a single value is

given, it is used for both margins; if two values are given, the first is used for `x`- distance, the second for `y`-distance.

Attribute arguments such as `col`, `pch`, `lty`, etc, are recycled if necessary: `merge` is not. Set entries of `lty` to `0` or set entries of `lwd` to `NA` to suppress lines in corresponding legend entries; set `pch` values to `NA` to suppress points.

Points are drawn *after* lines in order that they can cover the line with their background color `pt.bg`, if applicable.

See the examples for how to right-justify labels.

Since they are not used for Unicode code points, values `-31:-1` are silently omitted, as are `NA` and `""` values.

Value

A list with list components

`rect`
: a list with components

  `w, h`
  : positive numbers giving **w**idth and **h**eight of the legend's box.

  `left, top`
  : x and y coordinates of upper left corner of the box.

`text`
: a list with components

  `x, y`
  : numeric vectors of length `length(legend)`, giving the x and y coordinates of the legend's text(s).

returned invisibly.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

Murrell, P. (2005) *R Graphics*. Chapman & Hall/CRC Press.

See Also

`plot`, `barplot` which uses `legend()`, and `text` for more examples of math expressions.

Examples

```
## Run the example in '?matplot' or the following:
leg.txt <- c("Setosa     Petals", "Setosa     Sepals",
```

```
             "Versicolor Petals", "Versicolor Sepals")
y.leg <- c(4.5, 3, 2.1, 1.4, .7)
cexv  <- c(1.2, 1, 4/5, 2/3, 1/2)
matplot(c(1, 8), c(0, 4.5), type = "n", xlab = "Length", ylab = "Width",
        main = "Petal and Sepal Dimensions in Iris Blossoms")
for (i in seq(cexv)) {
  text  (1, y.leg[i] - 0.1, paste("cex=", formatC(cexv[i])), cex = 0.8, adj = 0)
  legend(3, y.leg[i], leg.txt, pch = "sSvV", col = c(1, 3), cex = cexv[i])
}

## 'merge = TRUE' for merging lines & points:
x <- seq(-pi, pi, len = 65)
plot(x, sin(x), type = "l", ylim = c(-1.2, 1.8), col = 3, lty = 2)
points(x, cos(x), pch = 3, col = 4)
lines(x, tan(x), type = "b", lty = 1, pch = 4, col = 6)
title("legend(..., lty = c(2, -1, 1), pch = c(NA, 3, 4), merge = TRUE)",
      cex.main = 1.1)
legend(-1, 1.9, c("sin", "cos", "tan"), col = c(3, 4, 6),
       text.col = "green4", lty = c(2, -1, 1), pch = c(NA, 3, 4),
       merge = TRUE, bg = "gray90")

## right-justifying a set of labels: thanks to Uwe Ligges
x <- 1:5; y1 <- 1/x; y2 <- 2/x
plot(rep(x, 2), c(y1, y2), type = "n", xlab = "x", ylab = "y")
lines(x, y1); lines(x, y2, lty = 2)
temp <- legend("topright", legend = c(" ", " "),
               text.width = strwidth("1,000,000"),
               lty = 1:2, xjust = 1, yjust = 1,
               title = "Line Types")
text(temp$rect$left + temp$rect$w, temp$text$y,
     c("1,000", "1,000,000"), pos = 2)


##--- log scaled Examples ------------------------------
leg.txt <- c("a one", "a two")

par(mfrow = c(2, 2))
for(ll in c("","x","y","xy")) {
  plot(2:10, log = ll, main = paste0("log = '", ll, "'"))
  abline(1, 1)
  lines(2:3, 3:4, col = 2)
  points(2, 2, col = 3)
  rect(2, 3, 3, 2, col = 4)
  text(c(3,3), 2:3, c("rect(2,3,3,2, col=4)",
                      "text(c(3,3),2:3,\"c(rect(...)\")"), adj = c(0, 0.3))
  legend(list(x = 2,y = 8), legend = leg.txt, col = 2:3, pch = 1:2,
         lty = 1, merge = TRUE)   #, trace = TRUE)
}
par(mfrow = c(1,1))

##-- Math expressions:  ------------------------------
x <- seq(-pi, pi, len = 65)
plot(x, sin(x), type = "l", col = 2, xlab = expression(phi),
     ylab = expression(f(phi)))
abline(h = -1:1, v = pi/2*(-6:6), col = "gray90")
lines(x, cos(x), col = 3, lty = 2)
ex.cs1 <- expression(plain(sin) * phi,  paste("cos", phi))  # 2 ways
utils::str(legend(-3, .9, ex.cs1, lty = 1:2, plot = FALSE,
           adj = c(0, 0.6)))  # adj y !
legend(-3, 0.9, ex.cs1, lty = 1:2, col = 2:3,  adj = c(0, 0.6))

require(stats)
x <- rexp(100, rate = .5)
hist(x, main = "Mean and Median of a Skewed Distribution")
abline(v = mean(x),   col = 2, lty = 2, lwd = 2)
```

```
abline(v = median(x), col = 3, lty = 3, lwd = 2)
ex12 <- expression(bar(x) == sum(over(x[i], n), i == 1, n),
                   hat(x) == median(x[i], i == 1, n))
utils::str(legend(4.1, 30, ex12, col = 2:3, lty = 2:3, lwd = 2))

## 'Filled' boxes -- for more, see example(plot.factor)
op <- par(bg = "white") # to get an opaque box for the legend
plot(cut(weight, 3) ~ group, data = PlantGrowth, col = NULL,
     density = 16*(1:3))
par(op)

## Using 'ncol' :
x <- 0:64/64
matplot(x, outer(x, 1:7, function(x, k) sin(k * pi * x)),
        type = "o", col = 1:7, ylim = c(-1, 1.5), pch = "*")
op <- par(bg = "antiquewhite1")
legend(0, 1.5, paste("sin(", 1:7, "pi * x)"), col = 1:7, lty = 1:7,
       pch = "*", ncol = 4, cex = 0.8)
legend(.8,1.2, paste("sin(", 1:7, "pi * x)"), col = 1:7, lty = 1:7,
       pch = "*", cex = 0.8)
legend(0, -.1, paste("sin(", 1:4, "pi * x)"), col = 1:4, lty = 1:4,
       ncol = 2, cex = 0.8)
legend(0, -.4, paste("sin(", 5:7, "pi * x)"), col = 4:6,  pch = 24,
       ncol = 2, cex = 1.5, lwd = 2, pt.bg = "pink", pt.cex = 1:3)
par(op)

## point covering line :
y <- sin(3*pi*x)
plot(x, y, type = "l", col = "blue",
    main = "points with bg & legend(*, pt.bg)")
points(x, y, pch = 21, bg = "white")
legend(.4,1, "sin(c x)", pch = 21, pt.bg = "white", lty = 1, col = "blue")

## legends with titles at different locations
plot(x, y, type = "n")
legend("bottomright", "(x,y)", pch = 1, title = "bottomright")
legend("bottom", "(x,y)", pch = 1, title = "bottom")
legend("bottomleft", "(x,y)", pch = 1, title = "bottomleft")
legend("left", "(x,y)", pch = 1, title = "left")
legend("topleft", "(x,y)", pch = 1, title = "topleft, inset = .05",
       inset = .05)
legend("top", "(x,y)", pch = 1, title = "top")
legend("topright", "(x,y)", pch = 1, title = "topright, inset = .02",
       inset = .02)
legend("right", "(x,y)", pch = 1, title = "right")
legend("center", "(x,y)", pch = 1, title = "center")

# using text.font (and text.col):
op <- par(mfrow = c(2, 2), mar = rep(2.1, 4))
c6 <- terrain.colors(10)[1:6]
for(i in 1:4) {
   plot(1, type = "n", axes = FALSE, ann = FALSE); title(paste("text.font =",i))
   legend("top", legend = LETTERS[1:6], col = c6,
          ncol = 2, cex = 2, lwd = 3, text.font = i, text.col = c6)
}
par(op)
```

[Package *graphics* version 3.3.0 Index]