Package 'maplegend'

June 30, 2025

Title Legends for Maps

Version 0.3.0

Description Create legends for maps and other graphics. Thematic maps need to be accompanied by legible legends to be fully comprehensible. This package offers a wide range of legends useful for cartography, some of which may also be useful for other types of graphics.

License GPL-3

Depends R (>= 4.1.0)

Imports graphics, grDevices

Suggests tinytest, covr

Encoding UTF-8

RoxygenNote 7.3.2

Language en-US

URL https://github.com/riatelab/maplegend/

BugReports https://github.com/riatelab/maplegend/issues/

NeedsCompilation no

Author Timothée Giraud [cre, aut] (ORCID: <https://orcid.org/0000-0002-1932-3323>)

Maintainer Timothée Giraud <timothee.giraud@cnrs.fr>

Repository CRAN

Date/Publication 2025-06-30 21:20:06 UTC

Contents

leg 2
leg_comp 8
leg_draw 10

Index **12**

leg *Plot a single map legend*

Description

Plot different types of legend. The "type" argument defines the legend type. Please note that some arguments are available for all types of legend and some others are only relevant for specific legend types (see Details).

Usage

```
leg(
  type,
  val,
  pos = "left",
  pal = "Inferno",
  alpha = NULL,
  col = "tomato4",
  inches = 0.3,
  symbol = "circle",
  self_adjust = FALSE,
  lwd = 0.7,
  border = "#333333",
  pch = seq_along(val),
  cex = rep(1, length(val)),
  title = "Legend Title",
  title_cex = 0.8 * size,
  val_cex = 0.6 * size,
  val_rnd = 0,
  col_na = "white",
  cex_na = 1,
  pch_na = 4,
  no_data = FALSE,
  no_data_txt = "No Data",
  box_border = "#333333",
  box_cex = c(1, 1),
  horiz = FALSE,
  frame_border = fg,
  frame = FALSE,
  bg = "#f7f7f7",
  fg = "#333333",
  size = 1,
  return_bbox = FALSE,
  mar = par("mar"),
  adj = c(0, 0)
)
```

Arguments

type	type of legend: • **prop** for proportional symbols, • **choro** for choropleth maps, • **cont** for continuous maps (e.g. raster), • **typo** for typology maps, • **symb** for symbols maps, • **prop_line** for proportional lines maps, • **grad_line** for graduated lines maps.
val	vector of value(s) (for "prop" and "prop_line", at least c(min, max) for "cont"), vector of categories (for "symb" and "typo"), break labels (for "choro" and "grad_line").
pos	position of the legend. It can be one of 'topleft', 'top', 'topright', 'right', 'bottomright', 'bottom','bottomleft', 'left', 'interactive' or a vector of two coordinates in map units (c(x, y)).
pal	a color palette name or a vector of colors
alpha	opacity, in the range [0,1]
col	color of the symbols (for "prop") or color of the lines (for "prop_line" and "grad_line")
inches	size of the largest symbol (radius for circles, half width for squares) in inches
symbol	type of symbols, 'circle' or 'square'
self_adjust	if TRUE values are self-adjusted to keep min, max and intermediate rounded values
lwd	width(s) of the symbols borders (for "prop" and "symb"), width of the largest line (for "prop_line"), vector of line width (for "grad_line")
border	symbol border color(s)
pch	type(s) of the symbols (0:25)
cex	size(s) of the symbols
title	title of the legend
title_cex	size of the legend title
val_cex	size of the values in the legend
val_rnd	number of decimal places of the values in the legend
col_na	color for missing values
cex_na	size of the symbols for missing values
pch_na	type of the symbols for missing values
no_data	if TRUE a "missing value" box is plotted
no_data_txt	label for missing values
box_border	border color of legend boxes
box_cex	width and height size expansion of boxes, (or offset between circles for "prop" legends with horiz = TRUE)

horiz	if TRUE plot an horizontal legend
frame_border	border color of the frame
frame	if TRUE the legend is plotted within a frame
bg	background color of the legend
fg	foreground color of the legend
size	size of the legend; 2 means two times bigger
return_bbox	return only bounding box of the legend. No legend is plotted.
mar	plot margins
adj	adjust the postion of the legend in x and y directions.

Details

Some arguments are available for all types of legend: `val`, `pos`, `title`, `title_cex`, `val_cex`, `frame`, `bg`, `fg`, `size`, `adj`, `alpha`, `return_bbox` and `mar`).

Relevant arguments for each specific legend types:

- `leg(type = "prop", val, inches, symbol, col, lwd, border, val_rnd, self_adjust, horiz)`
- `leg(type = "choro", val, pal, val_rnd, col_na, no_data, no_data_txt, box_border, horiz)`
- `leg(type = "cont", val, pal, val_rnd, col_na, no_data, no_data_txt, box_border, horiz)`
- `leg(type = "typo", val, pal, col_na, no_data, no_data_txt, box_border)`
- `leg(type = "symb", val, pal, pch, cex, lwd, pch_na, cex_na, col_na, no_data, no_data_txt)`
- `leg(type = "prop_line", val, col, lwd, val_rnd)`
- `leg(type = "grad_line", val, col, lwd, val_rnd)`

Value

No value is returned, a legend is displayed.

Examples

```
# minimal example
plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
box()
leg(type = "prop", val = c(10, 50, 100), pos = "topleft")
leg(type = "choro", val = c(10, 20, 30, 40, 50), pos = "bottomleft")
leg(type = "typo", val = c("A", "B", "C"), pos = "top")
leg(type = "symb", val = c("A", "B", "C"), pos = "topright")
leg(type = "prop_line", val = c(5, 50, 100), pos = "bottom", lwd = 20)
leg(
  type = "grad_line", val = c(1, 4, 10, 15), pos = "bottomright",
  lwd = c(1, 5, 10)
)

plot.new()
```

```
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
leg(type = "prop", val = c(10, 50, 100), pos = "topleft", horiz = TRUE)
leg(type = "choro", val = c(10, 20, 30, 40, 50), pos = "left", horiz = TRUE)
leg(
  type = "cont", val = c(10, 20, 30, 40, 50), pos = "bottomleft",
  horiz = TRUE
)
leg(
  type = "cont", val = c(10, 20, 30, 40, 50), pos = "topright",
  horiz = FALSE
)
box()

# full example
plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
leg(
  type = "prop",
  val = c(5, 100),
  pos = "top",
  inches = .4,
  symbol = "circle",
  col = "#940000",
  lwd = 1,
  border = "#9494ff",
  val_rnd = 1,
  self_adjust = TRUE,
  title = "Proportional Symbols",
  title_cex = 1,
  val_cex = .8,
  bg = "grey10",
  fg = "yellow",
  frame = TRUE
)

plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
leg(
  type = "choro",
  alpha = 1,
  val = c(10, 20, 30, 40, 50),
  pos = "top",
  pal = c("#7F000D", "#B56C6F", "#DBBABB", "#F1F1F1"),
  val_rnd = 2,
  col_na = "grey",
  no_data = TRUE,
  no_data_txt = "No data",
  box_border = "cornsilk",
  box_cex = c(2, 1),
  title = "Choropleth (sequential)"
)

plot.new()
```

```
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
leg(
  type = "typo",
  val = c("A", "B", "C"),
  pos = "top",
  pal = "Dynamic",
  col_na = "grey",
  no_data = TRUE,
  no_data_txt = "No data",
  box_cex = c(1.2, 1),
  title = "Typology (categories)"
)

plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
leg(
  type = "symb",
  val = c("A", "B", "C"),
  pos = "top",
  pch = 21:23,
  cex = c(4, 4, 2),
  pal = "Inferno",
  lwd = 2,
  border = "red",
  col_na = "grey",
  pch_na = 3,
  cex_na = 1,
  no_data = TRUE,
  no_data_txt = "No data",
  title = "Symbols"
)

plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
leg(
  type = "cont",
  val = c(1, 2, 3, 4, 5),
  pos = "top",
  pal = "Inferno",
  alpha = .7,
  val_rnd = 2,
  horiz = TRUE,
  box_cex = c(2, 1),
  title = "Continuous"
)

plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
leg(
  type = "prop_line",
  val = c(54, 505, 1025),
  pos = "top",
  lwd = 15,
```

```
  col = "green",
  val_rnd = -1,
  box_cex = c(2, .5),
  title = "Proportional Lines",
  bg = "black",
  fg = "white",
  frame = TRUE
)

plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
leg(
  type = "grad_line",
  val = c(1.25, 4.07, 10.001, 15.071),
  pos = "top",
  lwd = c(1, 7, 15),
  col = "#C130ff",
  val_rnd = 3,
  box_cex = c(2, 1),
  title = "Graduated Lines"
)

# Positions
plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
leg(
  type = "prop", val = c(10,60, 100), pos = "bottomleft", adj = c(0, 2),
  title = "adj = c(0, 2)", frame = TRUE
)
leg(
  type = "choro", val = c(10, 50, 100), pos = "bottomright",
  adj = c(0, 4), title = "adj = c(0, 4)", frame = TRUE
)
leg(
  type = "prop", val = c(10, 50, 100), pos = "topleft",
  adj = c(0, -4), title = "adj = c(0, -4)"
)
box()
mtext(
  text = "A text on 1 line", side = 1, adj = .01,
  line = -1, cex = 1
)
mtext(
  text = "A text\non 2 lines", side = 1, adj = .99,
  line = -1, cex = 1
)
mtext(
  text = "A large text on 1 line", side = 3, adj = .01,
  line = -2, cex = 2
)
```

leg_comp *Compose a map legend*

Description

Compose a map legend with several elements. The "type" argument defines the legend type. Please note that some arguments are available for all types of legend and some others are only relevant for specific legend types.

Usage

```
leg_comp(
  leg,
  type,
  val,
  pal = "Inferno",
  alpha = NULL,
  col = "tomato4",
  inches = 0.3,
  symbol = "circle",
  self_adjust = FALSE,
  lwd = 0.7,
  border = "#333333",
  pch = 1:seq_along(val),
  cex = rep(1, length(val)),
  title = "Legend Title",
  val_rnd = 0,
  col_na = "white",
  cex_na = 1,
  pch_na = 4,
  no_data = FALSE,
  no_data_txt = "No Data",
  box_border = "333333",
  box_cex = c(1, 1),
  horiz = FALSE
)
```

Arguments

leg	legend object
type	type of legend: • **prop** for proportional symbols, • **choro** for choropleth maps, • **cont** for continuous maps, • **typo** for typology maps, • **symb** for symbols maps,

- **prop_line** for proportional lines maps,
- **grad_line** for graduated lines maps.

val	vector of value(s) (for "prop" and "prop_line", at least c(min, max) for "cont"), vector of categories (for "symb" and "typo"), break labels (for "choro" and "grad_line").
pal	a color palette name or a vector of colors
alpha	opacity, in the range [0,1]
col	color of the symbols (for "prop") or color of the lines (for "prop_line" and "grad_line")
inches	size of the largest symbol (radius for circles, half width for squares) in inches
symbol	type of symbols, 'circle' or 'square'
self_adjust	if TRUE values are self-adjusted to keep min, max and intermediate rounded values
lwd	width(s) of the symbols borders (for "prop" and "symb"), width of the largest line (for "prop_line"), vector of line width (for "grad_line")
border	symbol border color(s)
pch	type(s) of the symbols (0:25)
cex	size(s) of the symbols
title	title of the legend
val_rnd	number of decimal places of the values in the legend
col_na	color for missing values
cex_na	size of the symbols for missing values
pch_na	type of the symbols for missing values
no_data	if TRUE a "missing value" box is plotted
no_data_txt	label for missing values
box_border	border color of legend boxes
box_cex	width and height size expansion of boxes, (or offset between circles for "prop" legends with horiz = TRUE)
horiz	if TRUE plot an horizontal legend

Value

A list of legends parameters is returned.

Examples

```
# minimal example
plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
box()
leg_comp(type = "prop", val = c(10, 50, 100)) |>
  leg_comp(type = "typo", val = c("A", "B", "C")) |>
  leg_draw()
```

leg_draw *Plot a composed map legend*

Description

Draw a map legend with several elements.

Usage

```
leg_draw(
  x,
  pos = "bottomright",
  size = 1,
  bg = "#f7f7f7",
  fg = "#333333",
  frame = TRUE,
  frame_border = fg,
  title_cex = 0.8 * size,
  val_cex = 0.6 * size,
  adj = c(0, 0),
  mar = par("mar")
)
```

Arguments

x	list of legend parameters
pos	position of the legend. . It can be one of 'topleft', 'topright', 'right', 'bottomright', 'bottomleft' or 'left',
size	size of the legend; 2 means two times bigger
bg	background color of the legend
fg	foreground color of the legend
frame	if TRUE the legend is plotted within a frame
frame_border	border color of the frame
title_cex	size of the legend title
val_cex	size of the values in the legend
adj	adjust the postion of the legend in x and y directions.
mar	plot margins

Value

A composed legend is plotted. Nothing is returned.

Examples

```
# minimal example
plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1), asp = 1)
box()
leg_comp(type = "prop", val = c(10, 50, 100)) |>
  leg_comp(type = "typo", val = c("A", "B", "C")) |>
  leg_draw(pos = "topright", bg = "lightblue")
```

Index

leg, 2
leg_comp, 8
leg_draw, 10