

Package: bbsvg (via r-universe)

July 24, 2026

Type Package

Title Create svg graphs for typical blackboard figures that illustrate economic models

Version 0.1

Date 2017-05-27

Author Sebastian Kranz

Maintainer Sebastian Kranz <sebastian.kranz@uni-ulm.de>

Description Create svg graphs for typical blackboard figures that illustrate economic models

License GPL >= 2.0

Depends dplyr, tidyr, restorepoint, codeUtils, stringtools, rmdtools, whisker, RColorBrewer, dplyrExtras, latexsvg, digest

Suggests rsvg

RoxygenNote 7.3.3

Config/pak/sysreqs cmake make libicu-dev libuv1-dev zlib1g-dev

Repository <https://skranz.r-universe.dev>

Date/Publication 2026-07-24 12:17:32 UTC

RemoteUrl <https://github.com/skranz/bbsvg>

RemoteRef master

RemoteSha ac447a2e2d4d76ef7c8253de0ea8e28d42c2b85f

Contents

EconCurves-package	3
bb_area	4
bb_area_above_curve	5
bb_area_below_curve	6
bb_area_beside_curve	7
bb_area_left_of_curve	8
bb_area_rect	9

bb_area_right_of_curve	10
bb_arrow	11
bb_compute_curve	12
bb_compute_obj	13
bb_compute_objs	14
bb_compute_slopecurve	14
bb_curve	15
bb_data	17
bb_hide_object	17
bb_hline	18
bb_isoquant	19
bb_margins	20
bb_object	20
bb_pane	21
bb_period	23
bb_point	24
bb_segment	25
bb_series	26
bb_series_tooltip_bars	28
bb_set_data	29
bb_slopecurve	30
bb_svg_css	31
bb_tangent	32
bb_text	33
bb_to_pdf	34
bb_to_png	35
bb_to_ps	36
bb_to_svg	36
bb_var	37
bb_vline	38
bb_xaxis	39
bb_xmarker	40
bb_xtick	41
bb_yaxis	42
bb_ymarker	44
bb_ytick	45
geom.curves.intersections	46
geom.to.geom.pos	46
point.to.geom.pos	46
two.point.lines.intersections	47

EconCurves-package	<i>What the package does (short line) ~~ package title ~~</i>
--------------------	---

Description

More about what it does (maybe more than one line) ~~ A concise (1-5 lines) description of the package ~~

Details

Package:	EconCurves
Type:	Package
Version:	1.0
Date:	2015-01-12
License:	What license is it under?

~~ An overview of how to use the package, including the most important functions ~~

Author(s)

Who wrote it

Maintainer: Who to complain to <yourfault@somewhere.net> ~~ The author and/or maintainer of the package ~~

References

~~ Literature or other references for background information ~~

See Also

~~ Optional links to other man pages, e.g. ~~ <pkg> ~~

Examples

~~ simple examples of the most important functions ~~

bb_area	<i>Add a polygonal area</i>
---------	-----------------------------

Description

Add a polygonal area

Usage

```
bb_area(  
    bb,  
    x,  
    y,  
    fill = "#8888ff",  
    alpha = 0.3,  
    stroke = "none",  
    style = nlist(fill = fill, `fill-opacity` = alpha, stroke = stroke, ...),  
    level = -10,  
    ...,  
    id = random.string(),  
    tooltip = NULL  
)
```

Arguments

bb	A ‘bb’ graphic object.
x, y	Vectors containing the polygon coordinates.
fill	Fill color.
alpha	Fill opacity.
stroke	Stroke color.
style	A list of SVG style properties.
level	Drawing level; lower values are drawn first.
...	Additional SVG style properties.
id	A unique object identifier.
tooltip	Optional tooltip text.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_area(x = c(1, 1, 3), y = c(1, 3, 1))

## End(Not run)
```

bb_area_above_curve *Add an area above a curve*

Description

Add an area above a curve

Usage

```
bb_area_above_curve(
  bb,
  eq = NULL,
  curve.id = NULL,
  fill = "#8888ff",
  alpha = 0.3,
  stroke = "none",
  style = nlist(fill = fill, `fill-opacity` = alpha, stroke = stroke, ...),
  level = -10,
  ...,
  id = random.string(),
  tooltip = NULL
)
```

Arguments

bb	A ‘bb’ graphic object.
eq	Optional curve equation as a character string.
curve.id	Identifier of an existing curve. Used when ‘eq’ is ‘NULL’.
fill	Fill color.
alpha	Fill opacity.
stroke	Stroke color.
style	A list of SVG style properties.
level	Drawing level; lower values are drawn first.
...	Additional arguments passed to [bb_curve()].
id	A unique object identifier.
tooltip	Optional tooltip text.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_area_above_curve(eq = "y_ = x_")

## End(Not run)
```

bb_area_below_curve	<i>Add an area below a curve</i>
---------------------	----------------------------------

Description

Add an area below a curve

Usage

```
bb_area_below_curve(
  bb,
  eq = NULL,
  curve.id = NULL,
  fill = "#8888ff",
  alpha = 0.3,
  stroke = "none",
  style = nlist(fill = fill, `fill-opacity` = alpha, stroke = stroke, ...),
  level = -10,
  ...,
  id = random.string(),
  tooltip = NULL
)
```

Arguments

bb	A 'bb' graphic object.
eq	Optional curve equation as a character string.
curve.id	Identifier of an existing curve. Used when 'eq' is 'NULL'.
fill	Fill color.
alpha	Fill opacity.
stroke	Stroke color.
style	A list of SVG style properties.
level	Drawing level; lower values are drawn first.
...	Additional arguments passed to [bb_curve()].
id	A unique object identifier.
tooltip	Optional tooltip text.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_area_beside_curve(eq = "y_ = x_")

## End(Not run)
```

bb_area_beside_curve *Add an area beside a curve*

Description

Shades the part of a plotting pane on a selected side of a curve.

Usage

```
bb_area_beside_curve(
  bb,
  eq = NULL,
  curve.id = NULL,
  fill = "#8888ff",
  alpha = 0.3,
  stroke = "none",
  style = nlist(fill = fill, `fill-opacity` = alpha, stroke = stroke, ...),
  direction = c("above", "below", "left", "right")[1],
  level = -10,
  ...,
  id = random.string(),
  tooltip = NULL,
  xmin = bb$xrange[[1]],
  xmax = bb$xrange[[2]],
  ymin = bb$yrange[[1]],
  ymax = bb$yrange[[2]],
  xrange = c(xmin, xmax),
  yrange = c(ymin, ymax)
)
```

Arguments

bb	A 'bb' graphic object.
eq	Optional curve equation as a character string.
curve.id	Identifier of an existing curve. Used when 'eq' is 'NULL'.
fill	Fill color.

alpha	Fill opacity.
stroke	Stroke color.
style	A list of SVG style properties.
direction	One of "above", "below", "left", or "right".
level	Drawing level; lower values are drawn first.
...	Additional arguments passed to [bb_curve()].
id	A unique object identifier.
tooltip	Optional tooltip text.
xmin, xmax, ymin, ymax	Bounds used to construct the shading region.
xrange, yrange	Two-element coordinate ranges.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_area_beside_curve(eq = "y_ = x_", direction = "below")

## End(Not run)
```

bb_area_left_of_curve *Add an area to the left of a curve*

Description

Add an area to the left of a curve

Usage

```
bb_area_left_of_curve(
  bb,
  eq = NULL,
  curve.id = NULL,
  fill = "#8888ff",
  alpha = 0.3,
  stroke = "none",
  style = nlist(fill = fill, `fill-opacity` = alpha, stroke = stroke, ...),
  level = -10,
  ...,
  id = random.string(),
  tooltip = NULL
)
```

Arguments

bb	A 'bb' graphic object.
eq	Optional curve equation as a character string.
curve.id	Identifier of an existing curve. Used when 'eq' is 'NULL'.
fill	Fill color.
alpha	Fill opacity.
stroke	Stroke color.
style	A list of SVG style properties.
level	Drawing level; lower values are drawn first.
...	Additional arguments passed to [bb_curve()].
id	A unique object identifier.
tooltip	Optional tooltip text.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_area_left_of_curve(eq = "x_ = 2")

## End(Not run)
```

bb_area_rect	<i>Add a rectangular area</i>
--------------	-------------------------------

Description

Add a rectangular area

Usage

```
bb_area_rect(bb, x1, y1, x2, y2, ...)
```

Arguments

bb	A 'bb' graphic object.
x1, y1	Coordinates of one corner.
x2, y2	Coordinates of the opposite corner.
...	Additional arguments passed to [bb_area()].

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_area_rect(1, 1, 3, 2)

## End(Not run)
```

bb_area_right_of_curve

Add an area to the right of a curve

Description

Add an area to the right of a curve

Usage

```
bb_area_right_of_curve(
  bb,
  eq = NULL,
  curve.id = NULL,
  fill = "#8888ff",
  alpha = 0.3,
  stroke = "none",
  style = nlist(fill = fill, `fill-opacity` = alpha, stroke = stroke, ...),
  level = -10,
  ...,
  id = random.string(),
  tooltip = NULL
)
```

Arguments

bb	A 'bb' graphic object.
eq	Optional curve equation as a character string.
curve.id	Identifier of an existing curve. Used when 'eq' is 'NULL'.
fill	Fill color.
alpha	Fill opacity.
stroke	Stroke color.
style	A list of SVG style properties.
level	Drawing level; lower values are drawn first.

...	Additional arguments passed to [bb_curve()].
id	A unique object identifier.
tooltip	Optional tooltip text.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_area_right_of_curve(eq = "x_ = 2")

## End(Not run)
```

bb_arrow	<i>Add an arrow</i>
----------	---------------------

Description

Add an arrow

Usage

```
bb_arrow(
  bb,
  x1 = x,
  x2 = x1,
  y1 = y,
  y2 = y1,
  x,
  y,
  arrow.head = c("end"),
  alpha = NULL,
  color = NULL,
  class = "arrow",
  style = list(stroke = color, `stroke-opacity` = alpha, ...),
  ...,
  id = random.string()
)
```

Arguments

bb	A 'bb' graphic object.
x1, y1	Coordinates of the first endpoint.
x2, y2	Coordinates of the second endpoint.

x, y	Convenience coordinates used by the endpoint defaults.
arrow.head	Which endpoint receives an arrow head.
alpha	Stroke opacity.
color	Stroke color.
class	SVG class name.
style	A list of SVG style properties.
...	Additional SVG style properties.
id	A unique object identifier.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_arrow(x1 = 1, y1 = 1, x2 = 3, y2 = 3)

## End(Not run)
```

bb_compute_curve	<i>Compute a curve's coordinates</i>
------------------	--------------------------------------

Description

Compute a curve's coordinates

Usage

```
bb_compute_curve(
  bb,
  curve,
  values = bb$values,
  xlen = bb$xlen,
  ylen = bb$ylen,
  xrange = first.non.null(curve$xrange, bb$xrange),
  yrange = first.non.null(curve$yrange, bb$yrange),
  ...
)
```

Arguments

bb	A 'bb' graphic object.
curve	A curve specification.
values	Values used to evaluate the curve equation.
xlen, ylen	Numbers of grid points in each direction.
xrange, yrange	Two-element coordinate ranges.
...	Reserved for additional computation options.

Value

The curve specification with computed geometry.

Examples

```
## Not run:
bb <- bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |>
  bb_curve(id = "diagonal", eq = "y_ = x_")
bb_compute_curve(bb, bb$objs$diagonal)

## End(Not run)
```

bb_compute_obj	<i>Compute one object in a graphic</i>
----------------	--

Description

Compute one object in a graphic

Usage

```
bb_compute_obj(bb, obj, i)
```

Arguments

bb	A 'bb' graphic object.
obj	An object specification from 'bb\$objs'.
i	The object's position in 'bb\$objs'.

Value

The 'bb' object with the selected geometry computed.

Examples

```
## Not run:
bb <- bb_pane(xrange = c(0, 1), yrange = c(0, 1)) |>
  bb_point(0.5, 0.5)
bb_compute_obj(bb, bb$objs[[1]], 1)

## End(Not run)
```

bb_compute_objs	<i>Compute all objects in a graphic</i>
-----------------	---

Description

Compute all objects in a graphic

Usage

```
bb_compute_objs(bb)
```

Arguments

bb A ‘bb’ graphic object.

Value

The ‘bb’ object with computed geometry.

Examples

```
## Not run:
bb <- bb_pane(xrange = c(0, 1), yrange = c(0, 1)) |>
  bb_point(0.5, 0.5)
bb_compute_objs(bb)

## End(Not run)
```

bb_compute_slopecurve	<i>Compute a slope-defined curve’s coordinates</i>
-----------------------	--

Description

Compute a slope-defined curve’s coordinates

Usage

```
bb_compute_slopecurve(bb, obj, ...)
```

Arguments

bb A 'bb' graphic object.
 obj A slope-curve specification.
 ... Reserved for additional computation options.

Value

The curve specification with computed geometry.

Examples

```
## Not run:
bb <- bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |>
  bb_slopecurve(x = 2, y = 2, slope = "-y_ / x_")
bb_compute_slopecurve(bb, bb$objs[[1]])

## End(Not run)
```

bb_curve

Add a curve

Description

Add a curve

Usage

```
bb_curve(
  bb,
  id = random.string(),
  eq,
  latex = NULL,
  label = NULL,
  data = NULL,
  color = NULL,
  lwd = NULL,
  style = nlist(stroke = color, stroke_width = lwd, `stroke-opacity` = alpha, ...),
  var.funs = NULL,
  labpos = NULL,
  labx = NULL,
  laby = NULL,
  tooltip = NULL,
  dy = NULL,
  dx = NULL,
  no.draw = FALSE,
  xrange = bb$xrange,
  yrange = bb$yrange,
```

```

    alpha = NULL,
    ...
)

```

Arguments

bb	A ‘bb’ graphic object.
id	A unique object identifier.
eq	An equation supplied as a character string.
latex	Optional LaTeX curve label.
label	Optional plain-text curve label.
data	Optional data used to evaluate the equation.
color	Stroke color.
lwd	Stroke width.
style	A list of SVG style properties.
var.funs	Optional variable functions used to transform the equation.
labpos	Optional two-element label position.
labx, laby	Optional label coordinates.
tooltip	Optional tooltip text.
dx, dy	Coordinate offsets.
no.draw	If ‘TRUE’, compute the curve without drawing it.
xrange, yrange	Two-element coordinate ranges.
alpha	Stroke opacity.
...	Additional SVG style properties.

Value

The modified ‘bb’ object.

Examples

```

## Not run:
bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |>
  bb_curve(eq = "y_ = 4 / x_", color = "blue")

## End(Not run)

```

bb_data	<i>Attach data to a graphic</i>
---------	---------------------------------

Description

Attach data to a graphic

Usage

```
bb_data(bb, data)
```

Arguments

bb	A 'bb' graphic object.
data	A data frame or similar object.

Value

The modified 'bb' object.

Examples

```
## Not run:  
bb_pane() |> bb_data(data.frame(x = 1:3, y = 3:1))  
  
## End(Not run)
```

bb_hide_object	<i>Hide a graphic object</i>
----------------	------------------------------

Description

Hide a graphic object

Usage

```
bb_hide_object(bb, id)
```

Arguments

bb	A 'bb' graphic object.
id	Identifier of the object to hide.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb <- bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |>
  bb_point(2, 3, id = "p")
bb_hide_object(bb, "p")

## End(Not run)
```

bb_hline

*Add a horizontal line***Description**

Add a horizontal line

Usage

```
bb_hline(
  bb,
  y,
  alpha = NULL,
  color = NULL,
  class = "segment",
  linetype = "solid",
  lwd = NULL,
  style = list(stroke = color, `stroke-opacity` = alpha, `stroke-width` = lwd, ...),
  ...,
  id = paste0("hline_", random.string())
)
```

Arguments

bb	A ‘bb’ graphic object.
y	Vertical coordinate.
alpha	Stroke opacity.
color	Stroke color.
class	SVG class name.
linetype	Line type.
lwd	Stroke width.
style	A list of SVG style properties.
...	Additional SVG style properties.
id	A unique object identifier.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |> bb_hline(y = 2)

## End(Not run)
```

bb_isoquant	<i>Add an isoquant through a point</i>
-------------	--

Description

Add an isoquant through a point

Usage

```
bb_isoquant(
  bb,
  Q,
  x,
  y,
  ...,
  id = paste0("isoquant_", random.string()),
  xvar = bb$xvar,
  yvar = bb$yvar
)
```

Arguments

bb	A 'bb' graphic object.
Q	A production function or expression accepted by 'isoquant.slope()'.
x, y	Coordinates of a point on the isoquant.
...	Additional arguments passed to [bb_slopecurve()].
id	A unique object identifier.
xvar, yvar	Names of the horizontal and vertical variables.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |>
  bb_isoquant(Q = "x_ * y_", x = 2, y = 2)

## End(Not run)
```

bb_margins	<i>Specify graphic margins</i>
------------	--------------------------------

Description

Specify graphic margins

Usage

```
bb_margins(bb, bottom = NULL, left = NULL, top = NULL, right = NULL, ...)
```

Arguments

bb	A 'bb' graphic object.
bottom, left, top, right	Margins in pixels. 'NULL' preserves the current value.
...	Reserved for additional margin settings.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |>
  bb_margins(left = 60, bottom = 50)

## End(Not run)
```

bb_object	<i>Add an object specification</i>
-----------	------------------------------------

Description

Add an object specification

Usage

```
bb_object(
  bb,
  obj = NULL,
  ...,
  id = first.non.null(obj[["id"]], random.string())
)
```

Arguments

bb	A 'bb' graphic object.
obj	Optional list describing the object.
...	Named properties added to 'obj'.
id	A unique object identifier.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb <- bb_pane(xrange = c(0, 5), yrange = c(0, 5))
bb_object(bb, list(type = "point", x = 2, y = 3), id = "p")

## End(Not run)
```

bb_pane

Create a blackboard-style graphic

Description

Creates a plotting pane or modifies an existing 'bb' graphic object.

Usage

```
bbsvg(
  bb = NULL,
  id = NULL,
  data = NULL,
  xvar = xy[1],
  yvar = xy[2],
  xy = c("x_", "y_"),
  xrange = NULL,
  yrange = NULL,
  show.ticks = FALSE,
  arrow.axis = NULL,
  xlen = 201,
  ylen = 201,
  org.width = width,
  org.height = height,
  margins = NULL,
  show = ".all",
  hide = NULL,
  init.data = FALSE,
```

```

    dataenv = parent.frame(),
    css = bb_svg_css(),
    values = if (!is.null(data)) as.list(data[data.row, , drop = FALSE]) else list(),
    data.row = 1,
    enclos = parent.frame(),
    scale = 1,
    width = 420,
    height = 300,
    ...
)

```

Arguments

bb	An optional existing ‘bb’ object to modify.
id	An optional graphic identifier.
data	Optional data frame used to evaluate graphic expressions.
xvar, yvar	Names of the horizontal and vertical variables.
xy	A two-element character vector used as the default variable names.
xrange, yrange	Two-element coordinate ranges.
show.ticks	Whether axes show ticks by default.
arrow.axis	Whether axes use arrow heads by default.
xlen, ylen	Numbers of grid points used for curve computation.
org.width, org.height	Unscaled graphic dimensions in pixels.
margins	Optional plot margins in pixels.
show, hide	Object selectors controlling visibility.
init.data	Whether to initialize data-dependent state.
dataenv	Environment in which data expressions are evaluated.
css	CSS included in the SVG.
values	Named values used to evaluate expressions.
data.row	Row of ‘data’ used to initialize ‘values’.
enclos	Enclosing environment for expression evaluation.
scale	Scale factor applied to the original dimensions.
width, height	Output dimensions in pixels.
...	Additional graphic defaults.

Value

A ‘bb_pane’ object.

Examples

```

## Not run:
bb <- bb_pane(xrange = c(0, 10), yrange = c(0, 5))

## End(Not run)

```

bb_period	<i>Mark a period on a series plot</i>
-----------	---------------------------------------

Description

Mark a period on a series plot

Usage

```
bb_period(  
  bb,  
  from,  
  to = NULL,  
  label = NULL,  
  shade = "#555555",  
  alpha = 0.3,  
  lwd = 1,  
  linetype = "dashed",  
  tooltip = label,  
  area.tooltip = tooltip,  
  latex = NULL,  
  font_size = 11  
)
```

Arguments

bb	A 'bb' graphic object.
from	Start coordinate.
to	Optional end coordinate. If supplied, the period is shaded.
label	Plain-text period label.
shade	Shading color.
alpha	Shading opacity.
lwd	Boundary-line width.
linetype	Boundary-line type.
tooltip	Boundary tooltip text.
area.tooltip	Shaded-area tooltip text.
latex	Optional LaTeX label.
font_size	Label font size.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(2000, 2010), yrange = c(0, 5)) |>
  bb_period(2003, 2005, "Event")

## End(Not run)
```

bb_point	<i>Add a point</i>
----------	--------------------

Description

Add a point

Usage

```
bb_point(
  bb,
  x,
  y,
  r = 4,
  alpha = NULL,
  color = fill,
  fill = NULL,
  class = "point",
  style = list(stroke = color, `fill-color` = fill, `stroke-opacity` = alpha,
    `fill-opacity` = alpha, ...),
  ...,
  id = paste0("point_", random.string())
)
```

Arguments

bb	A ‘bb’ graphic object.
x, y	Point coordinates.
r	Point radius in pixels.
alpha	Point opacity.
color	Stroke color.
fill	Fill color.
class	SVG class name.
style	A list of SVG style properties.
...	Additional SVG style properties.
id	A unique object identifier.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |>
  bb_point(x = 2, y = 3)

## End(Not run)
```

bb_segment

Add a line segment

Description

Add a line segment

Usage

```
bb_segment(
  bb,
  x1 = x,
  x2 = x1,
  y1 = y,
  y2 = y1,
  x,
  y,
  alpha = NULL,
  color = NULL,
  class = "segment",
  linetype = "solid",
  lwd = NULL,
  dasharray = linetype.to.dasharray(linetype),
  style = list(stroke = color, `stroke-opacity` = alpha, `stroke-width` = lwd, ...),
  ...,
  tooltip = NULL,
  id = paste0("segment_", random.string())
)
```

Arguments

bb	A 'bb' graphic object.
x1, y1	Coordinates of the first endpoint.
x2, y2	Coordinates of the second endpoint.
x, y	Convenience coordinates used by the endpoint defaults.

alpha	Stroke opacity.
color	Stroke color.
class	SVG class name.
linetype	Line type.
lwd	Stroke width.
dasharray	SVG stroke-dasharray value.
style	A list of SVG style properties.
...	Additional SVG style properties.
tooltip	Optional tooltip text.
id	A unique object identifier.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_segment(x1 = 1, y1 = 1, x2 = 3, y2 = 3)

## End(Not run)
```

bb_series	<i>Add a data series</i>
-----------	--------------------------

Description

Add a data series

Usage

```
bb_series(
  bb,
  x = data[[xvar]],
  y = data[[yvar]],
  data = bb$data,
  xvar = 1,
  yvar = 2,
  name = if (is.character(yvar)) yvar else id,
  alpha = NULL,
  color = NULL,
  class = "series",
  linetype = "solid",
  lwd = NULL,
```

```

    plot_type = "l",
    line.style = list(stroke = color, `stroke-opacity` = line.alpha, `stroke-width` = lwd,
        ...),
    point.style = list(fill = color, `fill-opacity` = point.alpha),
    dasharray = linetype.to.dasharray(linetype),
    ...,
    id = paste0("series_", random.string()),
    level = 10,
    draw.line = TRUE,
    draw.points = FALSE,
    r = 3,
    line.alpha = alpha,
    point.alpha = alpha
)

```

Arguments

bb	A 'bb' graphic object.
x, y	Series coordinates.
data	Data frame containing the series.
xvar, yvar	Column names or positions used to obtain 'x' and 'y'.
name	Series name used in tooltips.
alpha	Overall opacity.
color	Series color.
class	SVG class name.
linetype	Line type.
lwd	Line width.
plot_type	Plot type descriptor.
line.style, point.style	Lists of SVG style properties.
dasharray	SVG stroke-dasharray value.
...	Additional SVG style properties.
id	A unique object identifier.
level	Drawing level.
draw.line, draw.points	Whether to draw lines and points.
r	Point radius.
line.alpha, point.alpha	Line and point opacity.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane() |>
  bb_series(x = 1:5, y = c(1, 3, 2, 4, 5), color = "blue")

## End(Not run)
```

bb_series_tooltip_bars
Add tooltip bars for data series

Description

Add tooltip bars for data series

Usage

```
bb_series_tooltip_bars(
  bb,
  xname = "t",
  color = "yellow",
  lwd = 11,
  style = list(stroke = color, `stroke-width` = lwd),
  id = paste0("series_tooltip_bars", random.string()),
  level = 11,
  round.digits = 2,
  signif.digits = 5,
  tooltip.fun = NULL,
  tooltip.data = NULL,
  ...
)
```

Arguments

<code>bb</code>	A ‘bb’ graphic object.
<code>xname</code>	Label used for the horizontal coordinate in tooltips.
<code>color</code>	Bar color.
<code>lwd</code>	Bar width.
<code>style</code>	A list of SVG style properties.
<code>id</code>	A unique object identifier.
<code>level</code>	Drawing level.
<code>round.digits, signif.digits</code>	Numeric formatting controls.
<code>tooltip.fun</code>	Optional function that creates tooltip text.
<code>tooltip.data</code>	Optional data passed to ‘tooltip.fun’.
<code>...</code>	Additional object properties.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane() |>
  bb_series(x = 1:5, y = 1:5) |>
  bb_series_tooltip_bars()

## End(Not run)
```

bb_set_data

Set the current data row and values

Description

This helper updates the ‘bb’ object available in the calling evaluation context.

Usage

```
bb_set_data(
  values = if (!is.null(data)) as.list(data[data.row, , drop = FALSE]) else list(),
  data = bb$data,
  data.row = first.non.null(bb$data.row, 1)
)
```

Arguments

values	Named values used to evaluate graphic expressions.
data	Data frame associated with the graphic.
data.row	Row of ‘data’ used as the current observation.

Value

The updated ‘bb’ object.

Examples

```
## Not run:
bb <- bb_pane(data = data.frame(x = 1:2, y = 3:4))
bb_set_data(data = bb$data, data.row = 2)

## End(Not run)
```

bb_slopecurve	<i>Add a curve defined by its slope</i>
---------------	---

Description

Numerically traces a curve through a supplied point using a slope expression.

Usage

```
bb_slopecurve(
  bb,
  x,
  y,
  slope,
  color = NULL,
  lwd = NULL,
  alpha = NULL,
  style = nlist(stroke = color, stroke_width = lwd, `stroke-opacity` = alpha, ...),
  x.move = 0,
  y.move = 0,
  xrange = bb$xrange,
  yrange = bb$yrange,
  var.funs = NULL,
  tooltip = NULL,
  ...,
  data = NULL,
  id = paste0("slopecurve_", random.string()),
  xvar = bb$xvar,
  yvar = bb$yvar
)
```

Arguments

bb	A 'bb' graphic object.
x, y	Coordinates of a point on the curve.
slope	A slope expression or character string.
color	Stroke color.
lwd	Stroke width.
alpha	Stroke opacity.
style	A list of SVG style properties.
x.move, y.move	Offsets applied to the computed coordinates.
xrange, yrange	Two-element coordinate ranges.
var.funs	Optional variable functions used in computations.
tooltip	Optional tooltip text.

...	Additional SVG style properties.
data	Optional data used to evaluate expressions.
id	A unique object identifier.
xvar, yvar	Names of the horizontal and vertical variables.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |>
  bb_slopecurve(x = 2, y = 2, slope = "-y_ / x_")

## End(Not run)
```

bb_svg_css

Default CSS for blackboard-style SVG graphics

Description

Default CSS for blackboard-style SVG graphics

Usage

```
bb_svg_css()
```

Value

A character string containing CSS rules.

Examples

```
## Not run:
css <- bb_svg_css()

## End(Not run)
```

bb_tangent

*Add a tangent line***Description**

Add a tangent line

Usage

```
bb_tangent(
  bb,
  x = NULL,
  y = NULL,
  slope = NULL,
  width = NULL,
  alpha = NULL,
  color = NULL,
  class = "segment",
  linetype = "solid",
  to = NULL,
  ...,
  id = paste0("tangent_", random.string())
)
```

Arguments

bb	A 'bb' graphic object.
x, y	Coordinates of the tangency point.
slope	The tangent slope.
width	Total line width, or a two-element vector of widths on either side of the tangency point.
alpha	Stroke opacity.
color	Stroke color.
class	SVG class name.
linetype	Line type.
to	Optional identifier of a curve from which missing tangent quantities are computed.
...	Additional arguments passed to [bb_segment()].
id	A unique object identifier.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_tangent(x = 2, y = 2, slope = 1)

## End(Not run)
```

bb_text

*Add text to a graphic***Description**

Add text to a graphic

Usage

```
bb_text(
  bb,
  label = NULL,
  latex = NULL,
  x = NULL,
  y = NULL,
  xrel = NULL,
  yrel = NULL,
  align = "center",
  x.offset = 0,
  y.offset = NULL,
  boxed = FALSE,
  font_size = 14,
  color = NULL,
  style = list(`font-size` = font_size, fill = color),
  valign = c("center", "bottom", "top")[1],
  vertical = FALSE,
  ...,
  id = random.string(),
  fill.background = FALSE,
  background.alpha = 0.8,
  background.color = "#ffffff"
)
```

Arguments

bb	A ‘bb’ graphic object.
label	Plain-text label.
latex	LaTeX label. If supplied, it takes precedence over ‘label’.
x, y	Absolute label coordinates.

xrel, yrel	Relative label coordinates between zero and one.
align	Horizontal alignment.
x.offset, y.offset	Pixel offsets from the label coordinates.
boxed	Whether to use boxed-label styling.
font_size	Font size in pixels.
color	Text color.
style	A list of SVG style properties.
valign	Vertical alignment.
vertical	Whether to rotate the text vertically.
...	Additional label properties.
id	A unique object identifier.
fill.background	Whether to draw a background behind plain text.
background.alpha	Background opacity.
background.color	Background color.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_text("Equilibrium", x = 2, y = 2)

## End(Not run)
```

bb_to_pdf

Export a graphic to PDF

Description

Export a graphic to PDF

Usage

```
bb_to_pdf(bb, file, ...)
```

Arguments

bb	A 'bb' graphic object.
file	Output file path.
...	Additional arguments passed through the export pipeline.

Value

The result returned by the SVG conversion function, invisibly where applicable.

Examples

```
## Not run:
bb <- bb_pane(xrange = c(0, 1), yrange = c(0, 1))
bb_to_pdf(bb, "figure.pdf")

## End(Not run)
```

bb_to_png	<i>Export a graphic to PNG</i>
-----------	--------------------------------

Description

Export a graphic to PNG

Usage

```
bb_to_png(bb, file, ...)
```

Arguments

bb	A 'bb' graphic object.
file	Output file path.
...	Additional arguments passed through the export pipeline.

Value

The result returned by the SVG conversion function, invisibly where applicable.

Examples

```
## Not run:
bb <- bb_pane(xrange = c(0, 1), yrange = c(0, 1))
bb_to_png(bb, "figure.png")

## End(Not run)
```

bb_to_ps	<i>Export a graphic to PostScript</i>
----------	---------------------------------------

Description

Export a graphic to PostScript

Usage

```
bb_to_ps(bb, file, ...)
```

Arguments

bb	A ‘bb’ graphic object.
file	Output file path.
...	Additional arguments passed through the export pipeline.

Value

The result returned by the SVG conversion function, invisibly where applicable.

Examples

```
## Not run:
bb <- bb_pane(xrange = c(0, 1), yrange = c(0, 1))
bb_to_ps(bb, "figure.ps")

## End(Not run)
```

bb_to_svg	<i>Render a graphic as SVG</i>
-----------	--------------------------------

Description

Render a graphic as SVG

Usage

```
bb_to_svg(
  bb,
  file = outfile,
  id = first.non.null(bb$id, random.string()),
  css = bb$css,
  width = first.non.null(bb$width, bb$org.width, 480),
  height = first.non.null(bb$height, bb$org.height, 320),
  return.svg.object = FALSE,
```

```

    latexsvg = isTRUE(bb$use.latex),
    outfile = NULL,
    ...
)

```

Arguments

bb	A ‘bb’ graphic object.
file	Optional output file path.
id	SVG element identifier.
css	CSS included in the SVG.
width, height	Output dimensions in pixels.
return.svg.object	If ‘TRUE’, return the intermediate SVG object.
latexsvg	Whether to convert LaTeX labels with ‘latexsvg’.
outfile	Deprecated alternative output-file argument.
...	Reserved for additional rendering options.

Value

An SVG character string, or the intermediate SVG object when ‘return.svg.object’ is ‘TRUE’.

Examples

```

## Not run:
bb <- bb_pane(xrange = c(0, 1), yrange = c(0, 1)) |>
  bb_point(0.5, 0.5)
bb_to_svg(bb)

## End(Not run)

```

bb_var

Define computed graphic variables

Description

Define computed graphic variables

Usage

```
bb_var(bb, ..., id = paste0("var_", random.string()))
```

Arguments

bb	A ‘bb’ graphic object.
...	Named expressions defining variables.
id	A unique object identifier.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |> bb_var(a = 2)

## End(Not run)
```

bb_vline	<i>Add a vertical line</i>
----------	----------------------------

Description

Add a vertical line

Usage

```
bb_vline(
  bb,
  x,
  alpha = NULL,
  color = NULL,
  class = "segment",
  linetype = "solid",
  lwd = NULL,
  style = list(stroke = color, `stroke-opacity` = alpha, `stroke-width` = lwd, ...),
  ...,
  id = paste0("hline_", random.string())
)
```

Arguments

bb	A ‘bb’ graphic object.
x	Horizontal coordinate.
alpha	Stroke opacity.
color	Stroke color.
class	SVG class name.
linetype	Line type.
lwd	Stroke width.
style	A list of SVG style properties.
...	Additional SVG style properties.
id	A unique object identifier.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |> bb_vline(x = 2)

## End(Not run)
```

bb_xaxis	<i>Specify the x-axis</i>
----------	---------------------------

Description

Specify the x-axis

Usage

```
bb_xaxis(
  bb,
  label = latex,
  latex = NULL,
  labelpos = c("bottom", "right", "center")[1],
  show.ticks = first.non.null(defaults$show.ticks, TRUE),
  arrow.axis = first.non.null(defaults$arrow.axis, !isTRUE(show.ticks)),
  defaults = bb$defaults,
  y.offset = NULL,
  x.offset = NULL,
  y = "bottom",
  align = NULL,
  num.ticks = 5,
  ticks = NULL
)
```

Arguments

bb	A ‘bb’ graphic object.
label	Plain-text axis label.
latex	Optional LaTeX axis label.
labelpos	Label position: “bottom”, “right”, or “center”.
show.ticks	Whether to draw ticks and tick labels.
arrow.axis	Whether to draw an arrow-headed axis.
defaults	Graphic defaults used to resolve axis settings.

y.offset, x.offset	Label offsets in pixels.
y	Axis position: "bottom", "top", or a numeric coordinate.
align	Label alignment.
num.ticks	Desired number of automatically generated ticks.
ticks	Optional explicit tick coordinates.

Value

The modified 'bb' object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |>
  bb_xaxis(label = "Quantity")

## End(Not run)
```

bb_xmarker

Add a vertical marker

Description

Add a vertical marker

Usage

```
bb_xmarker(
  bb,
  x = NULL,
  y2 = y,
  y = NULL,
  y1 = bb$y.min,
  ...,
  linetype = "dashed",
  label = x,
  latex = NULL,
  align = "center",
  y.offset = -20,
  x.offset = 0,
  id = random.string()
)
```

Arguments

bb	A ‘bb’ graphic object.
x	Horizontal marker coordinate.
y2, y1	End coordinates of the marker line.
y	Optional shorthand for ‘y2’.
...	Additional arguments passed to the tick and segment.
linetype	Marker-line type.
label	Plain-text marker label.
latex	Optional LaTeX marker label.
align	Label alignment.
y.offset, x.offset	Label offsets in pixels.
id	A base identifier for the marker objects.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |> bb_xmarker(x = 2)

## End(Not run)
```

bb_xtick	<i>Add a custom x-axis tick</i>
----------	---------------------------------

Description

Add a custom x-axis tick

Usage

```
bb_xtick(
  bb,
  x = NULL,
  ...,
  label = x,
  latex = NULL,
  align = "center",
  y.offset = -20,
  y = NULL,
  id = random.string(),
  add.tick.line = TRUE
)
```

Arguments

bb	A ‘bb’ graphic object.
x	Horizontal tick coordinate.
...	Additional arguments passed to [bb_text()].
label	Plain-text tick label.
latex	LaTeX tick label.
align	Horizontal label alignment.
y.offset	Vertical pixel offset.
y	Vertical coordinate of the label.
id	A unique object identifier.
add.tick.line	Whether to add a tick mark to the axis.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_xtick(x = 2, label = "two")

## End(Not run)
```

bb_yaxis

Specify the y-axis

Description

Specify the y-axis

Usage

```
bb_yaxis(
  bb,
  label = latex,
  latex = NULL,
  labelpos = c("left", "top", "center")[1],
  show.ticks = first.non.null(defaults$show.ticks, TRUE),
  arrow.axis = first.non.null(defaults$arrow.axis, !isTRUE(show.ticks)),
  defaults = bb$defaults,
  y.offset = NULL,
  x.offset = NULL,
  align = NULL,
```

```

    x = "left",
    ticks = NULL,
    num.ticks = 5,
    show.grid = FALSE,
    grid.ticks = ticks,
    grid.color = "#888888",
    tick.labels = NULL,
    show.line = TRUE,
    ...
)

```

Arguments

bb	A ‘bb’ graphic object.
label	Plain-text axis label.
latex	Optional LaTeX axis label.
labelpos	Label position: “left”, “top”, or “center”.
show.ticks	Whether to draw ticks and tick labels.
arrow.axis	Whether to draw an arrow-headed axis.
defaults	Graphic defaults used to resolve axis settings.
y.offset, x.offset	Label offsets in pixels.
align	Label alignment.
x	Axis position: “left”, “right”, or a numeric coordinate.
ticks	Optional explicit tick coordinates.
num.ticks	Desired number of automatically generated ticks.
show.grid	Whether to draw horizontal grid lines.
grid.ticks	Coordinates of grid lines.
grid.color	Grid-line color.
tick.labels	Optional custom tick labels.
show.line	Whether to draw the main axis line.
...	Additional axis settings.

Value

The modified ‘bb’ object.

Examples

```

## Not run:
bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |>
  bb_yaxis(label = "Price")

## End(Not run)

```

bb_ymarker	<i>Add a horizontal marker</i>
------------	--------------------------------

Description

Add a horizontal marker

Usage

```
bb_ymarker(
  bb,
  y = NULL,
  x2 = x,
  x = NULL,
  ...,
  linetype = "dashed",
  label = y,
  latex = NULL,
  align = "right",
  id = random.string()
)
```

Arguments

bb	A ‘bb’ graphic object.
y	Vertical marker coordinate.
x2	Ending horizontal coordinate.
x	Optional shorthand for ‘x2’.
...	Additional arguments passed to the tick and segment.
linetype	Marker-line type.
label	Plain-text marker label.
latex	Optional LaTeX marker label.
align	Label alignment.
id	A base identifier for the marker objects.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 5), yrange = c(0, 5)) |> bb_ymarker(y = 2)

## End(Not run)
```

bb_ytick	<i>Add a custom y-axis tick</i>
----------	---------------------------------

Description

Add a custom y-axis tick

Usage

```
bb_ytick(
  bb,
  y = NULL,
  ...,
  label = y,
  latex = NULL,
  align = "right",
  x.offset = ifelse(add.tick.line, -15, -5),
  x = NULL,
  id = random.string(),
  add.tick.line = TRUE
)
```

Arguments

bb	A ‘bb’ graphic object.
y	Vertical tick coordinate.
...	Additional arguments passed to [bb_text()].
label	Plain-text tick label.
latex	LaTeX tick label.
align	Horizontal label alignment.
x.offset	Horizontal pixel offset.
x	Horizontal coordinate of the label.
id	A unique object identifier.
add.tick.line	Whether to add a tick mark to the axis.

Value

The modified ‘bb’ object.

Examples

```
## Not run:
bb_pane(xrange = c(0, 4), yrange = c(0, 4)) |>
  bb_ytick(y = 2, label = "two")

## End(Not run)
```

```
geom.curves.intersections
```

Find the intersections of two curves, which are characterized by their xy values

Description

Find the intersections of two curves, which are characterized by their xy values

Usage

```
geom.curves.intersections(geom1, geom2, grid.length = 201)
```

```
geom.to.geom.pos
```

Find relative position (above, below, left, right) of a geom to a geom

Description

If the geoms intersect non of the attributes holds true, i.e. the geom new must be stricly above old

Usage

```
geom.to.geom.pos(new, old, check = c("above", "below", "left", "right"))
```

```
point.to.geom.pos
```

Find relative position (above, below, left, right) of a point to a geom

Description

Find relative position (above, below, left, right) of a point to a geom

Usage

```
point.to.geom.pos(
  xy,
  geom,
  check = c("above", "below", "left", "right", "on"),
  tol.on = 0.01
)
```

`two.point.lines.intersections`

intersection of two geoms that are characterized by two points each

Description

Formula based on Wikipedia entry <http://en.wikipedia.org/wiki/Line>

Usage

`two.point.lines.intersections(x1, x2, x3, x4, y1, y2, y3, y4)`

Index

* package

EconCurves-package, 3

<pkg>, 3

bb_area, 4

bb_area_above_curve, 5

bb_area_below_curve, 6

bb_area_beside_curve, 7

bb_area_left_of_curve, 8

bb_area_rect, 9

bb_area_right_of_curve, 10

bb_arrow, 11

bb_compute_curve, 12

bb_compute_obj, 13

bb_compute_objs, 14

bb_compute_slopecurve, 14

bb_curve, 15

bb_data, 17

bb_hide_object, 17

bb_hline, 18

bb_isoquant, 19

bb_margins, 20

bb_object, 20

bb_pane, 21

bb_period, 23

bb_point, 24

bb_segment, 25

bb_series, 26

bb_series_tooltipBars, 28

bb_set_data, 29

bb_slopecurve, 30

bb_svg_css, 31

bb_tangent, 32

bb_text, 33

bb_to_pdf, 34

bb_to_png, 35

bb_to_ps, 36

bb_to_svg, 36

bb_var, 37

bb_vline, 38

bb_xaxis, 39

bb_xmarker, 40

bb_xtick, 41

bb_yaxis, 42

bb_ymarker, 44

bb_ytick, 45

bbsvg (bb_pane), 21

EconCurves (EconCurves-package), 3

EconCurves-package, 3

geom.curves.intersections, 46

geom.to.geom.pos, 46

point.to.geom.pos, 46

two.point.lines.intersections, 47