

Package: rtemis.draw (via r-universe)

May 27, 2026

Title Interactive Visualization Using ECharts

Version 0.2.1

Description Provides a comprehensive R interface to Apache ECharts (<https://echarts.apache.org>) for creating interactive charts and visualizations. Offers type-checked, validated configuration objects built with S7 classes that mirror the ECharts TypeScript API, with convenience functions for common chart types and full htmlwidgets integration for use in Quarto, Shiny, and IDE viewers.

URL <https://www.rtemis.org>, <https://docs.rtemis.org/draw/r>,
<https://docs.rtemis.org/draw/r-api>

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports cli, htmlwidgets, jsonlite, rtemis.core (>= 0.1.0), S7,
signal, viridisLite

Suggests knitr, mgcv, rmarkdown, rtemis.a3, testthat (>= 3.0.0)

Additional_repositories <https://rtemis-org.r-universe.dev>

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel true

Collate '00_init.R' 'rtemis.draw-package.R' 'rtemis_color_system.R'
'utils.R' 'styles.R' 'label.R' 'axis.R' 'components.R'
'series.R' 'series_sankey.R' 'option.R' 'theme.R' 'draw.R'
'draw_a3.R' 'draw_spectrogram.R' 'export.R' 'zzz.R'

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake make libuv1-dev

Repository <https://rtemis-org.r-universe.dev>

Date/Publication 2026-05-12 18:57:57 UTC

RemoteUrl <https://github.com/rtemis-org/draw>

RemoteRef HEAD

RemoteSha 5f4427e5f3ee4105e487c24756968273b83ffae8

RemoteSubdir r

Contents

rtemis.draw-package	3
AreaStyle	4
Axis	5
AxisLabel	7
AxisLine	8
AxisTick	9
BarSeries	9
BoxplotSeries	11
DataZoom	12
draw	15
draw_a3	16
draw_bar	18
draw_boxplot	19
draw_density	21
draw_heatmap	22
draw_histogram	25
draw_line	26
draw_pie	27
draw_sankey	28
draw_scatter	30
draw_spectrogram	31
drawOutput	34
EChartsOption	34
Grid	36
HeatmapSeries	37
ItemStyle	38
LabelLine	39
LabelOption	40
Legend	41
LineSeries	43
LineStyle	45
MarkArea	46
MarkAreaDataPoint	46
MinorSplitLine	47
MinorTick	48
PieSeries	48
renderDraw	50
rtemis_colors	51
SankeyEdgeItem	51
SankeyLevelOption	52
SankeyNodeItem	52

SankeySeries	53
save_drawing	55
ScatterSeries	56
SplitArea	57
SplitLine	58
TextStyle	58
Theme	61
theme_dark	62
theme_light	63
Title	65
to_json	66
to_list	67
Tooltip	67
VisualMap	69
Index	71

rtemis.draw-package *rtemis.draw: Drawing Utilities*

Description

Comprehensive R interface to Apache ECharts (<https://echarts.apache.org>) for creating interactive charts and visualizations. Offers type-checked, validated configuration objects built with S7 classes that mirror the ECharts TypeScript API, with convenience functions for common chart types and full htmlwidgets integration for use in Quarto, Shiny, and IDE viewers.

Author(s)

Maintainer: E.D. Gennatas <gennatas@gmail.com> ([ORCID](#)) [copyright holder]

Authors:

- E.D. Gennatas <gennatas@gmail.com> ([ORCID](#)) [copyright holder]

See Also

Useful links:

- <https://www.rtemis.org>
- <https://docs.rtemis.org/draw/r>
- <https://docs.rtemis.org/draw/r-api>

AreaStyle

Area Style

Description

Fill styling for areas (polygons, area under curves, etc.).

Usage

```
AreaStyle(  
  color = NULL,  
  opacity = NULL,  
  origin = NULL,  
  shadow_blur = NULL,  
  shadow_color = NULL,  
  shadow_offset_x = NULL,  
  shadow_offset_y = NULL  
)
```

Arguments

color	Optional Character: CSS fill color.
opacity	Optional Numeric [0, 1]: Fill opacity.
origin	Optional Character {"auto", "start", "end"} or Numeric: Area origin.
shadow_blur	Optional Numeric [0, Inf): Shadow blur radius.
shadow_color	Optional Character: Shadow color.
shadow_offset_x	Optional Numeric: Shadow horizontal offset.
shadow_offset_y	Optional Numeric: Shadow vertical offset.

Details

Corresponds to AreaStyleOption in src/util/types.ts (line 1169). ECharts docs: <https://echarts.apache.org/en/option.html#series-line.areaStyle>

*Axis**Axis*

Description

Configuration for a cartesian axis (xAxis or yAxis). Combines fields from AxisBaseOptionCommon, NumericAxisBaseOptionCommon, CategoryAxisBaseOption, ValueAxisBaseOption, LogAxisBaseOption, and TimeAxisBaseOption.

Usage

```
Axis(  
    type = NULL,  
    show = NULL,  
    inverse = NULL,  
    name = NULL,  
    name_location = NULL,  
    name_rotate = NULL,  
    name_gap = NULL,  
    name_text_style = NULL,  
    silent = NULL,  
    trigger_event = NULL,  
    min = NULL,  
    max = NULL,  
    scale = NULL,  
    split_number = NULL,  
    interval = NULL,  
    min_interval = NULL,  
    max_interval = NULL,  
    align_ticks = NULL,  
    log_base = NULL,  
    boundary_gap = NULL,  
    data = NULL,  
    axis_line = NULL,  
    axis_tick = NULL,  
    minor_tick = NULL,  
    axis_label = NULL,  
    split_line = NULL,  
    minor_split_line = NULL,  
    split_area = NULL,  
    position = NULL,  
    grid_index = NULL  
)
```

Arguments

type Optional Character {"value", "category", "time", "log"}: Axis type.

show	Optional Logical: Whether to show the axis.
inverse	Optional Logical: Whether to invert the axis.
name	Optional Character: Axis name.
name_location	Optional Character {"start", "middle", "center", "end"}: Axis name location.
name_rotate	Optional Numeric: Axis name rotation angle in degrees.
name_gap	Optional Numeric [0, Inf): Gap between the axis name and axis line.
name_text_style	Optional TextStyle : Axis name text style.
silent	Optional Logical: Whether the axis is silent.
trigger_event	Optional Logical: Whether the axis triggers events.
min	Optional Numeric or Character {"dataMin", "dataMax"}: Minimum axis value.
max	Optional Numeric or Character {"dataMin", "dataMax"}: Maximum axis value.
scale	Optional Logical: Whether a value axis should avoid forcing zero.
split_number	Optional Numeric [0, Inf): Suggested number of split segments.
interval	Optional Numeric [0, Inf): Tick interval.
min_interval	Optional Numeric [0, Inf): Minimum auto-calculated tick interval.
max_interval	Optional Numeric [0, Inf): Maximum auto-calculated tick interval.
align_ticks	Optional Logical: Whether to align ticks with the first axis.
log_base	Optional Numeric [0, Inf): Logarithm base for log axes.
boundary_gap	Optional Logical, Numeric, or Character: Gap at axis boundaries. Logical (category) or length-2 vector of numbers/strings (numeric axes).
data	Optional Vector: Category data.
axis_line	Optional AxisLine : Axis line configuration.
axis_tick	Optional AxisTick : Axis tick configuration.
minor_tick	Optional MinorTick : Minor tick configuration.
axis_label	Optional AxisLabel : Axis label configuration.
split_line	Optional SplitLine : Split line configuration.
minor_split_line	Optional MinorSplitLine : Minor split line configuration.
split_area	Optional SplitArea : Split area configuration.
position	Optional Character {"left", "right", "top", "bottom"}: Axis position. For yAxis: "left" (default) or "right". For xAxis: "bottom" (default) or "top".
grid_index	Optional Numeric [0, Inf): Index of the grid this axis belongs to. Used when the chart has multiple grids (e.g. heatmap + dendrogram panels).

Details

Corresponds to `AxisBaseOption` in `src/coord/axisCommonTypes.ts` (line 376). ECharts docs: <https://echarts.apache.org/en/option.html#xAxis>

AxisLabel

Axis Label

Description

Configuration for axis labels. Combines label-specific positioning with text styling (flattened in `to_list` like `LabelOption`).

Usage

```
AxisLabel(  
  show = NULL,  
  inside = NULL,  
  rotate = NULL,  
  margin = NULL,  
  formatter = NULL,  
  show_min_label = NULL,  
  show_max_label = NULL,  
  hide_overlap = NULL,  
  interval = NULL,  
  text_style = NULL  
)
```

Arguments

<code>show</code>	Optional Logical: Whether to show axis labels.
<code>inside</code>	Optional Logical: Whether labels are inside the grid.
<code>rotate</code>	Optional Numeric: Label rotation angle in degrees.
<code>margin</code>	Optional Numeric [0 , Inf): Space between the label and axis line in pixels.
<code>formatter</code>	Optional Character or function: Label formatter.
<code>show_min_label</code>	Optional Logical: Whether to show the label at the minimum end.
<code>show_max_label</code>	Optional Logical: Whether to show the label at the maximum end.
<code>hide_overlap</code>	Optional Logical: Whether to hide overlapping labels.
<code>interval</code>	Optional Numeric or Character {"auto"}: Label interval.
<code>text_style</code>	Optional TextStyle : Text appearance.

Details

Corresponds to `AxisLabelBaseOption` in `src/coord/axisCommonTypes.ts` (line 312). ECharts docs: <https://echarts.apache.org/en/option.html#xAxis.axisLabel>

AxisLine

Axis Line

Description

Configuration for the axis line itself.

Usage

```
AxisLine(  
  show = NULL,  
  on_zero = NULL,  
  on_zero_axis_index = NULL,  
  symbol = NULL,  
  symbol_size = NULL,  
  symbol_offset = NULL,  
  line_style = NULL  
)
```

Arguments

show	Optional Logical or Character {"auto"}: Whether to show the axis line.
on_zero	Optional Logical: Whether the axis line is on the zero position of the other axis.
on_zero_axis_index	Optional Numeric [0, Inf): Index of the other axis on whose zero position this axis line is drawn.
symbol	Optional Character: Arrow symbols at both ends.
symbol_size	Optional Numeric: Arrow symbol size.
symbol_offset	Optional Numeric or Character: Arrow symbol offset.
line_style	Optional LineStyle : Line styling.

Details

Corresponds to AxisLineOption in src/coord/axisCommonTypes.ts (line 231). ECharts docs: <https://echarts.apache.org/en/option.html#xAxis.axisLine>

AxisTick	<i>Axis Tick</i>
----------	------------------

Description

Configuration for axis tick marks.

Usage

```
AxisTick(  
  show = NULL,  
  inside = NULL,  
  length = NULL,  
  align_with_label = NULL,  
  line_style = NULL  
)
```

Arguments

show	Optional Logical or Character {"auto"}: Whether to show ticks.
inside	Optional Logical: Whether ticks are inside the grid.
length	Optional Numeric [0, Inf): Tick mark length in pixels.
align_with_label	Optional Logical: Whether to align ticks with labels.
line_style	Optional LineStyle : Line styling.

Details

Corresponds to AxisTickOption in src/coord/axisCommonTypes.ts (line 244). ECharts docs: <https://echarts.apache.org/en/option.html#xAxis.axisTick>

BarSeries	<i>Bar Series</i>
-----------	-------------------

Description

Configuration for a bar chart series.

Usage

```

BarSeries(
  name = NULL,
  data = NULL,
  x_axis_index = NULL,
  y_axis_index = NULL,
  stack = NULL,
  silent = NULL,
  legend_hover_link = NULL,
  color = NULL,
  z_level = NULL,
  z = NULL,
  clip = NULL,
  bar_width = NULL,
  bar_max_width = NULL,
  bar_min_width = NULL,
  bar_min_height = NULL,
  bar_gap = NULL,
  bar_category_gap = NULL,
  round_cap = NULL,
  show_background = NULL,
  item_style = NULL,
  label = NULL
)

```

Arguments

name	Optional Character: Series name for legend display.
data	Optional Vector or list: Data values.
x_axis_index	Optional Numeric [0, Inf): X-axis index.
y_axis_index	Optional Numeric [0, Inf): Y-axis index.
stack	Optional Character: Stack group name.
silent	Optional Logical: Whether the series is silent.
legend_hover_link	Optional Logical: Whether to highlight related legends on hover.
color	Optional Character: Series color override.
z_level	Optional Numeric: Canvas layer index.
z	Optional Numeric: Front-back order within the same layer.
clip	Optional Logical: Whether to clip overflow.
bar_width	Optional Numeric or Character: Bar width.
bar_max_width	Optional Numeric or Character: Maximum bar width.
bar_min_width	Optional Numeric or Character: Minimum bar width.
bar_min_height	Optional Numeric [0, Inf): Minimum bar height.
bar_gap	Optional Numeric or Character: Gap between bars in the same category.

bar_category_gap	Optional Numeric or Character: Gap between categories.
round_cap	Optional Logical: Whether to use a round cap.
show_background	Optional Logical: Whether to show the bar background.
item_style	Optional ItemStyle : Bar styling.
label	Optional LabelOption : Data-label configuration.

Details

Corresponds to `BarSeriesOption` in `src/chart/bar/BarSeries.ts` (line 68) and `BaseBarSeriesOption` in `src/chart/bar/BaseBarSeries.ts` (line 38). ECharts docs: <https://echarts.apache.org/en/option.html#series-bar>

BoxplotSeries	<i>Boxplot Series</i>
---------------	-----------------------

Description

Configuration for a boxplot chart series.

Usage

```
BoxplotSeries(
  name = NULL,
  data = NULL,
  x_axis_index = NULL,
  y_axis_index = NULL,
  silent = NULL,
  legend_hover_link = NULL,
  color = NULL,
  z_level = NULL,
  z = NULL,
  layout = NULL,
  box_width = NULL,
  item_style = NULL,
  label = NULL
)
```

Arguments

name	Optional Character: Series name.
data	Optional list: Boxplot data. Each element is <code>c(min, Q1, median, Q3, max)</code> .
x_axis_index	Optional Numeric <code>[0, Inf)</code> : X-axis index.
y_axis_index	Optional Numeric <code>[0, Inf)</code> : Y-axis index.

silent	Optional Logical: Whether the series is silent.
legend_hover_link	Optional Logical: Whether to highlight related legends on hover.
color	Optional Character: Series color override.
z_level	Optional Numeric: Canvas layer index.
z	Optional Numeric: Front-back order within the same layer.
layout	Optional Character {"horizontal", "vertical"}: Layout orientation.
box_width	Optional Numeric or Character: Box-width range as a length-2 vector.
item_style	Optional ItemStyle : Boxplot styling.
label	Optional LabelOption : Data-label configuration.

Details

Corresponds to `BoxplotSeriesOption` in `src/chart/boxplot/BoxplotSeries.ts` (line 59). ECharts docs: <https://echarts.apache.org/en/option.html#series-boxplot>

DataZoom

Data Zoom

Description

Axis-range zoom component. ECharts renders `dataZoom` as an array; common usage pairs a "slider" (visible scrollbar) with an "inside" (mouse-wheel / drag) entry targeting the same axis. Unused styling / mouse-behavior fields stay NULL and are dropped from the serialized output.

Usage

```
DataZoom(
  type = "slider",
  id = NULL,
  disabled = NULL,
  x_axis_index = NULL,
  y_axis_index = NULL,
  radius_axis_index = NULL,
  angle_axis_index = NULL,
  single_axis_index = NULL,
  filter_mode = NULL,
  start = NULL,
  end = NULL,
  start_value = NULL,
  end_value = NULL,
  min_span = NULL,
  max_span = NULL,
  min_value_span = NULL,
  max_value_span = NULL,
```

```

    orient = NULL,
    throttle = NULL,
    range_mode = NULL,
    show = NULL,
    background_color = NULL,
    left = NULL,
    right = NULL,
    top = NULL,
    bottom = NULL,
    width = NULL,
    height = NULL,
    zoom_lock = NULL,
    zoom_on_mouse_wheel = NULL,
    move_on_mouse_move = NULL,
    move_on_mouse_wheel = NULL,
    prevent_default_mouse_move = NULL
)

```

Arguments

type	Character {"slider", "inside"}: Zoom type. "slider" shows a draggable handle; "inside" enables zoom/drag directly on the coordinate system.
id	Optional Character: Component identifier used for updates.
disabled	Optional Logical: Whether the component is disabled.
x_axis_index	Optional Numeric, numeric vector, or Character {"all"}: Index of the x-axis (or axes) this zoom targets.
y_axis_index	Optional Numeric, numeric vector, or Character {"all"}: Index of the y-axis (or axes) this zoom targets.
radius_axis_index	Optional Numeric, numeric vector, or Character {"all"}: Polar radius axis index.
angle_axis_index	Optional Numeric, numeric vector, or Character {"all"}: Polar angle axis index.
single_axis_index	Optional Numeric, numeric vector, or Character {"all"}: Single-axis index.
filter_mode	Optional Character {"filter", "weakFilter", "empty", "none"}: How out-of-window data points are handled.
start	Optional Numeric [0, 100]: Left/start position as a percentage of the full data range.
end	Optional Numeric [0, 100]: Right/end position as a percentage of the full data range.
start_value	Optional Numeric, Character, or Date: Absolute start value (alternative to start).
end_value	Optional Numeric, Character, or Date: Absolute end value (alternative to end).
min_span	Optional Numeric [0, 100]: Minimum window size, as a percentage.
max_span	Optional Numeric [0, 100]: Maximum window size, as a percentage.
min_value_span	Optional Numeric or Character: Minimum window size in data units.

<code>max_value_span</code>	Optional Numeric or Character: Maximum window size in data units.
<code>orient</code>	Optional Character {"horizontal", "vertical"}: Layout orientation. Defaults are inferred from the axis the zoom targets.
<code>throttle</code>	Optional Numeric [0, Inf): Throttle delay in milliseconds for zoom events.
<code>range_mode</code>	Optional length-2 Character vector of {"value", "percent"}: Per-boundary interpretation for start/end.
<code>show</code>	Optional Logical: Whether the slider is visible (slider type only).
<code>background_color</code>	Optional Character: Slider background color.
<code>left</code>	Optional Numeric or Character: Slider left position.
<code>right</code>	Optional Numeric or Character: Slider right position.
<code>top</code>	Optional Numeric or Character: Slider top position.
<code>bottom</code>	Optional Numeric or Character: Slider bottom position.
<code>width</code>	Optional Numeric or Character: Slider width.
<code>height</code>	Optional Numeric or Character: Slider height.
<code>zoom_lock</code>	Optional Logical: Whether to lock the window size (allow move but not zoom).
<code>zoom_on_mouse_wheel</code>	Optional Logical or Character {"shift", "ctrl", "alt"}: Whether/how the mouse wheel zooms (inside type only).
<code>move_on_mouse_move</code>	Optional Logical or Character {"shift", "ctrl", "alt"}: Whether/how mouse drag pans (inside type only).
<code>move_on_mouse_wheel</code>	Optional Logical or Character {"shift", "ctrl", "alt"}: Whether/how the mouse wheel pans (inside type only).
<code>prevent_default_mouse_move</code>	Optional Logical: Whether to call <code>event.preventDefault()</code> on drag (inside type only).

Details

Corresponds to `DataZoomOption` in `src/component/dataZoom/DataZoomModel.ts` (line 38), `SliderDataZoomOption` in `src/component/dataZoom/SliderZoomModel.ts` (line 38), and `InsideDataZoomOption` in `src/component/dataZoom/InsideZoomModel.ts` (line 23). ECharts docs: <https://echarts.apache.org/en/option.html#dataZoom>

draw	<i>Render an ECharts option as an htmlwidget</i>
------	--

Description

Low-level function that takes an [EChartsOption](#) (or a plain list) and renders it as an interactive htmlwidget.

Usage

```
draw(
  option,
  theme = NULL,
  renderer = "canvas",
  width = NULL,
  height = NULL,
  elementId = NULL,
  filename = NULL,
  meta = list()
)
```

Arguments

option	EChartsOption or named list: Option object to render.
theme	Optional Theme , list, or NA: Theme override. NULL enables auto-detection of light/dark mode, or NA for no theme (raw ECharts defaults). When NULL, the widget detects dark mode from VS Code, RStudio, or the browser's <code>prefers-color-scheme</code> and applies theme_light() or theme_dark() accordingly.
renderer	Character {"canvas", "svg"}: Rendering engine.
width	Optional Character or Numeric: Widget width.
height	Optional Character or Numeric: Widget height.
elementId	Optional Character: Explicit element ID.
filename	Optional Character: If provided, the widget is also written to this file via save_drawing() . Extension determines the format (currently only .svg is supported).
meta	Optional named list: Extra fields merged into the widget payload. Used internally (e.g. by draw_heatmap() to pass square-cell layout parameters to the JS binding).

Value

htmlwidget: Widget object.

draw_a3

*Draw an A3 Amino Acid Sequence Visualization***Description**

Renders an A3 object (from `rtemis.a3`) as an interactive ECharts amino-acid sequence diagram. The sequence is wrapped into rows in a meander/serpentine path, with optional site, region, PTM, processing, and variant annotations overlaid as distinct series and collected in a vertical legend.

Usage

```
draw_a3(
  x,
  n_per_row = 21L,
  residue_spacing = 0.3,
  marker_size = 28,
  font_size = 18,
  line_width = 2,
  show_markers = TRUE,
  show_labels = TRUE,
  position_every = 10L,
  region_opacity = 0.35,
  ptm_placement = "radial",
  residue_fill = "#E7E5E4",
  residue_stroke = "#44403C",
  label_color = "#1C1917",
  pos_label_color = "#78716C",
  variant_color = "#FA6E1E",
  disease_variant_color = "#E266AE",
  enable_zoom = TRUE,
  title = NULL,
  grid = NULL,
  theme = NULL,
  width = NULL,
  height = NULL,
  filename = NULL
)
```

Arguments

<code>x</code>	A3 object from <code>rtemis.a3::create_A3()</code> or <code>rtemis.a3::read_A3json()</code> .
<code>n_per_row</code>	Integer [2, Inf): Number of residues per wrapped row.
<code>residue_spacing</code>	Numeric (0, Inf): Horizontal spacing multiplier. Values below 1 compress the layout; 1 gives natural hex spacing.
<code>marker_size</code>	Numeric (0, Inf): Residue circle diameter in pixels.

font_size	Numeric (0, Inf): Base font size in pixels for residue and position labels.
line_width	Numeric (0, Inf): Backbone line width in pixels.
show_markers	Logical: Whether to render the backbone line and residue circles.
show_labels	Logical: Whether to render single-letter residue labels.
position_every	Optional Integer [1, Inf): Show a numeric position label every N residues. NULL disables position labels.
region_opacity	Numeric [0, 1]: Opacity of region band overlays.
ptm_placement	Character {"radial", "innerRadial", "outerRadial"}: Placement of PTM symbols relative to the residue circle. "radial" centres on the circle edge; "innerRadial" places inside; "outerRadial" places outside.
residue_fill	Character: Residue circle fill color.
residue_stroke	Character: Residue circle stroke and backbone line color.
label_color	Character: Default residue label text color.
pos_label_color	Character: Position label text color.
variant_color	Character: Label color for variant residues.
disease_variant_color	Character: Label color for disease-associated variant residues (takes precedence over variant_color).
enable_zoom	Logical: Whether to enable Shift+scroll zoom.
title	Optional Character: Chart title.
grid	Optional Grid : Override any plot-area margin. The defaults (left = 24, top auto, right driven by legend width, bottom = 24) are merged with the properties of the supplied Grid object, so only the fields you set are changed — e.g. Grid(left = 8, top = 8). legend_top is re-derived from the final grid top automatically.
theme	Optional Theme : Theme override. NULL auto-detects light/dark mode; NA uses raw ECharts defaults.
width	Optional Numeric or Character: Widget width.
height	Optional Numeric or Character: Widget height. Auto-computed from layout bounds and marker_size when NULL.
filename	Optional Character: If provided, save the widget to this file via save_drawing() .

Details

Corresponds to `createA3EChartsOption()` in `src/lib/a3/visualization/echarts.ts`.

Value

htmlwidget: Widget object.

Examples

```
if (requireNamespace("rtemis.a3", quietly = TRUE)) {
  a <- rtemis.a3::create_A3(
    "MAEPRQEFVEMEDHAGTYGLGDRK",
    site = list(
      Active_site = rtemis.a3::annotation_position(c(5L, 17L))
    ),
    region = list(
      Domain = rtemis.a3::annotation_range(
        matrix(c(3L, 10L, 15L, 22L), ncol = 2, byrow = TRUE)
      )
    )
  )
  draw_a3(a)
}
```

draw_bar

Draw a Bar Chart

Description

Quick bar chart from x/y data.

Usage

```
draw_bar(
  x,
  y,
  color = NULL,
  stack = FALSE,
  horizontal = FALSE,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  theme = NULL,
  margins = DEFAULT_MARGINS,
  width = NULL,
  height = NULL,
  filename = NULL
)
```

Arguments

x Character: Category labels.

y Numeric or named list: Bar heights.

color	Optional Character: Bar color or colors. For multiple series, colors are applied per series and recycled as needed. For a single series, a single color styles the whole series; multiple colors are recycled across individual bars. color takes precedence over the theme palette.
stack	Logical: Whether to stack bars.
horizontal	Logical: Whether to draw horizontal bars.
xlab	Optional Character: X-axis title (bottom axis).
ylab	Optional Character: Y-axis title (left axis).
title	Optional Character: Chart title.
theme	Optional Theme : Theme override. The palette inside the theme can be overridden per-chart with the color argument.
margins	Optional Named numeric vector or named list: Plot margins in pixels (or percentage strings) for any of "top", "right", "bottom", "left". Unspecified sides keep echarts' default auto-sizing. See draw_line() for details.
width	Optional Character or Numeric: Widget width.
height	Optional Character or Numeric: Widget height.
filename	Optional Character: If provided, save the widget to this file via save_drawing() .

Value

htmlwidget: Widget object.

draw_boxplot	<i>Draw a Boxplot</i>
--------------	-----------------------

Description

Quick boxplot from raw data with optional grouping for multiple traces. Boxplot statistics (min, Q1, median, Q3, max) are computed automatically using [grDevices::boxplot.stats\(\)](#).

Usage

```
draw_boxplot(
  data,
  labels = NULL,
  group = NULL,
  horizontal = FALSE,
  color = NULL,
  fill_alpha = 0.25,
  na.rm = TRUE,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  theme = NULL,
```

```

    margins = DEFAULT_MARGINS,
    width = NULL,
    height = NULL,
    verbosity = 1L,
    filename = NULL
)

```

Arguments

data	Numeric or list: A list of numeric vectors (one per box), or a single numeric vector when group is provided. For ungrouped named lists, <code>names(data)</code> are used as labels when <code>labels</code> is not supplied.
labels	Optional Character: Category labels for each box. Ignored when group is provided (group levels are used instead). When omitted for ungrouped named lists, <code>names(data)</code> are used.
group	Optional Vector: Grouping variable. When provided, data must be a numeric vector, and boxplot statistics are computed per group. Each group gets its own colored series.
horizontal	Logical: Whether to draw horizontal boxplots.
color	Optional Character: Box color or colors. For ungrouped boxplots, a single color used at full opacity for borders and at <code>fill_alpha</code> opacity for the fill. For grouped boxplots, defaults to <code>rtemis_colors</code> ; recycled as needed.
fill_alpha	Numeric [0, 1]: Opacity for the box fill color.
na.rm	Logical: Whether to remove NA values before computing boxplot statistics.
xlab	Optional Character: X-axis title (bottom axis). Overrides the value-axis name inferred from a single-element named list passed as data.
ylab	Optional Character: Y-axis title (left axis). Overrides the value-axis name inferred from a single-element named list passed as data.
title	Optional Character: Chart title.
theme	Optional Theme : Theme override.
margins	Optional Named numeric vector or named list: Plot margins in pixels (or percentage strings) for any of "top", "right", "bottom", "left". Unspecified sides keep echarts' default auto-sizing. See draw_line() for details.
width	Optional Character or Numeric: Widget width.
height	Optional Character or Numeric: Widget height.
verbosity	Integer [0, Inf): Verbosity level for removed-NA messages.
filename	Optional Character: If provided, save the widget to this file via save_drawing() .

Value

htmlwidget: Widget object.

draw_density	<i>Draw a Density Plot</i>
--------------	----------------------------

Description

Kernel density estimation plot from numeric data, with optional grouping for multiple traces. A list input creates one density trace per vector when ungrouped, or one trace per variable/group combination when group is supplied.

Usage

```
draw_density(
  x,
  group = NULL,
  n = 512,
  bw = "nrd0",
  na.rm = TRUE,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  theme = NULL,
  margins = DEFAULT_MARGINS,
  width = NULL,
  height = NULL,
  verbosity = 1L,
  filename = NULL
)
```

Arguments

x	Numeric or list: Values used for density estimation. An ungrouped list creates one density trace per element; with group, each list element is split by group into separate traces.
group	Optional Vector: Grouping variable for multiple density traces.
n	Numeric [1, Inf): Number of equally spaced points for density estimation.
bw	Character or Numeric: Bandwidth passed to <code>stats::density()</code> .
na.rm	Logical: Whether to remove NA values before computing densities.
xlab	Optional Character: X-axis title.
ylab	Optional Character: Y-axis title.
title	Optional Character: Chart title.
theme	Optional Theme : Theme override.
margins	Optional Named numeric vector or named list: Plot margins in pixels (or percentage strings) for any of "top", "right", "bottom", "left". Unspecified sides keep echarts' default auto-sizing. See <code>draw_line()</code> for details.

width	Optional Character or Numeric: Widget width.
height	Optional Character or Numeric: Widget height.
verbosity	Integer [0, Inf): Verbosity level for removed-NA messages.
filename	Optional Character: If provided, save the widget to this file via save_drawing() .

Value

htmlwidget: Widget object.

draw_heatmap	<i>Draw a Heatmap</i>
--------------	-----------------------

Description

Quick heatmap from a numeric matrix. Designed to handle all common use cases including correlation matrices (square cells, diverging color scale), general rectangular heatmaps, and clustered heatmaps.

Usage

```
draw_heatmap(
  x,
  row_names = NULL,
  col_names = NULL,
  triangle = NULL,
  cluster_rows = FALSE,
  cluster_cols = FALSE,
  dist_method = "euclidean",
  hclust_method = "complete",
  show_row_dendro = TRUE,
  show_col_dendro = TRUE,
  dendro_row_width = 60,
  dendro_col_height = 60,
  dendro_color = NULL,
  dendro_uniform = FALSE,
  dendro_row_side = "right",
  dendro_col_side = "top",
  square_cells = NULL,
  color = NULL,
  zlim = NULL,
  show_values = FALSE,
  value_digits = 2L,
  show_colorbar = TRUE,
  colorbar_orient = "vertical",
  title = NULL,
  theme = NULL,
```

```

    margins = NULL,
    width = NULL,
    height = NULL,
    filename = NULL
)

```

Arguments

<code>x</code>	Numeric matrix: Input data. Rows map to y-axis categories and columns to x-axis categories.
<code>row_names</code>	Optional Character: Row labels. Defaults to <code>rownames(x)</code> , or "1", "2", ... when row names are absent.
<code>col_names</code>	Optional Character: Column labels. Defaults to <code>colnames(x)</code> .
<code>triangle</code>	Optional Character {"upper", "lower"}: Mask one triangle of the matrix to NA. "upper" keeps only the upper triangle; "lower" keeps only the lower triangle. The diagonal is always masked, as it is uninformative for symmetric matrices (e.g. always 1 for correlations).
<code>cluster_rows</code>	Logical: Whether to reorder rows via hierarchical clustering.
<code>cluster_cols</code>	Logical: Whether to reorder columns via hierarchical clustering.
<code>dist_method</code>	Character: Distance method passed to <code>stats::dist()</code> . Common values: "euclidean", "manhattan".
<code>hclust_method</code>	Character: Linkage method passed to <code>stats::hclust()</code> . Common values: "complete", "ward.D2", "average".
<code>show_row_dendro</code>	Logical: Whether to render the row dendrogram panel when <code>cluster_rows = TRUE</code> . Set to FALSE to reorder rows but suppress the visual dendrogram.
<code>show_col_dendro</code>	Logical: Whether to render the col dendrogram panel when <code>cluster_cols = TRUE</code> .
<code>dendro_row_width</code>	Optional Numeric [1, Inf): Pixel width of the row dendrogram panel (between row labels and the heatmap grid).
<code>dendro_col_height</code>	Optional Numeric [1, Inf): Pixel height of the col dendrogram panel (between the chart title and the heatmap grid).
<code>dendro_color</code>	Optional Character: Stroke color for dendrogram branch lines. NULL (default) uses a semi-transparent grey ("#99999988"), which works in both light and dark themes.
<code>dendro_uniform</code>	Logical: Whether to render dendrograms with uniform level heights — each merge step occupies equal visual space — rather than heights proportional to actual merge distances. FALSE (default) preserves merge distances in the visual scaling.
<code>dendro_row_side</code>	Character {"right", "left"}: Side of the heatmap on which the row dendrogram is placed. "right" (default) keeps row labels on the left with the dendrogram on the right for a clean, symmetric layout.

dendro_col_side	Character {"top", "bottom"}: Side of the heatmap on which the column dendrogram is placed. "top" (default) places it above the heatmap. "bottom" automatically moves column labels to the top.
square_cells	Optional Logical: Whether to compute widget dimensions so cells are square. NULL (default) enables this automatically for square matrices (e.g. correlation matrices). When TRUE, both width and height are calculated from the number of cells; supply explicit width/height to override.
color	Optional Character: Color palette — a vector of 2 or more colors defining the continuous color scale from zlim[1] to zlim[2]. When NULL (default) a diverging teal–background–orange palette is used when data spans zero (with the background colour pinned exactly at 0, even for asymmetric ranges), otherwise a sequential single-hue palette is used. Two variants (light / dark) are computed automatically and the JS binding selects the correct one based on the active theme.
zlim	Optional Numeric: Length-2 vector c(min, max) for the color scale. Defaults to the observed data range. For correlation matrices, c(-1, 1) is recommended.
show_values	Logical: Whether to print the cell value as a label inside each cell.
value_digits	Integer: Decimal places used in cell labels and the tooltip.
show_colorbar	Logical: Whether to display the continuous color-scale bar.
colorbar_orient	Character {"vertical", "horizontal"}: Orientation of the color bar.
title	Optional Character: Chart title.
theme	Optional Theme : Theme override. NULL auto-detects light/dark mode.
margins	Optional Named numeric vector or named list: Override the auto-computed heatmap plot-area margins in pixels (or percentage strings) for any of "top", "right", "bottom", "left" — e.g. c(left = 200) to widen the left gutter for long row labels. Unspecified sides use the values computed from label lengths, title, colorbar, and dendrogram panels. See draw_line() for the general convention.
width	Optional Character or Numeric: Widget width.
height	Optional Character or Numeric: Widget height.
filename	Optional Character: If provided, save the widget to this file via save_drawing() .

Details

Color encoding is driven by a continuous [VisualMap](#) component. Hierarchical clustering is performed in R via [hclust\(\)](#); rows and columns are reordered accordingly. When `cluster_rows` or `cluster_cols` is TRUE (and the corresponding `show_*_dendro` flag is not FALSE), a dendrogram panel is rendered alongside the heatmap as an ECharts custom series.

Value

htmlwidget: Widget object.

draw_histogram	<i>Draw a Histogram</i>
----------------	-------------------------

Description

Histogram from numeric data, with optional grouping for multiple traces. Bins are computed using [graphics::hist\(\)](#) with consistent break points across groups.

Usage

```
draw_histogram(  
  x,  
  group = NULL,  
  breaks = "Sturges",  
  xlab = NULL,  
  ylab = NULL,  
  title = NULL,  
  theme = NULL,  
  margins = DEFAULT_MARGINS,  
  width = NULL,  
  height = NULL,  
  filename = NULL  
)
```

Arguments

x	Numeric: Values used for histogram binning.
group	Optional Vector: Grouping variable for multiple series.
breaks	Numeric, Character, or Numeric vector: Binning method. A single number (number of bins), a character string naming an algorithm (e.g. "Sturges", "Scott", "FD"), or a numeric vector of break points. Passed to graphics::hist() .
xlab	Optional Character: X-axis title.
ylab	Optional Character: Y-axis title.
title	Optional Character: Chart title.
theme	Optional Theme : Theme override.
margins	Optional Named numeric vector or named list: Plot margins in pixels (or percentage strings) for any of "top", "right", "bottom", "left". Unspecified sides keep echarts' default auto-sizing. See draw_line() for details.
width	Optional Character or Numeric: Widget width.
height	Optional Character or Numeric: Widget height.
filename	Optional Character: If provided, save the widget to this file via save_drawing() .

Value

htmlwidget: Widget object.

draw_line

*Draw a Line Chart***Description**

Quick line chart from x/y data.

Usage

```
draw_line(
  x,
  y,
  names = NULL,
  smooth = FALSE,
  area = FALSE,
  points = TRUE,
  blocks = NULL,
  block_color = NULL,
  block_opacity = 0.2,
  color = NULL,
  line_style = NULL,
  xlim = NULL,
  ylim = NULL,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  theme = NULL,
  zoom = FALSE,
  margins = DEFAULT_MARGINS,
  width = NULL,
  height = NULL,
  filename = NULL
)
```

Arguments

x	Vector: X-axis values.
y	Numeric or named list: Y values.
names	Optional Character: Series names used when y is an unnamed list.
smooth	Logical: Whether to smooth lines.
area	Logical: Whether to show area fill.
points	Logical: Whether to show point markers on each data value. Defaults to TRUE; set to FALSE on long time-series where the symbols add visual noise.
blocks	Optional factor, integer, character, or logical of length length(x): Per-x-value group label used to shade vertical background bands. Contiguous runs of the same level become one band. NA entries produce no band.

block_color	Optional Character vector or list of length k, where k is the number of unique levels in blocks: Fill color for each level. Match by name to factor levels if named, else positional. NA, NULL, or "transparent" entries skip that level.
block_opacity	Numeric [0, 1]: Fill opacity applied to all bands. Defaults to 0.2.
color	Optional Character: Series color palette — a single color string or character vector that overrides the theme palette for this chart. color takes precedence over the theme palette (it sets option.color).
line_style	Optional Character {"solid", "dashed", "dotted"}: Line dash style — one value per series, recycled to match the number of series.
xlim	Optional Numeric [length 2]: X-axis limits c(min, max). Only supported when x is numeric; passing xlim with a non-numeric x errors. Defaults to range(x) (no padding) when x is numeric.
ylim	Optional Numeric [length 2]: Y-axis limits c(min, max). Defaults to the range of all y values across series (no padding).
xlab	Optional Character: X-axis title.
ylab	Optional Character: Y-axis title.
title	Optional Character: Chart title.
theme	Optional Theme : Theme override. The palette inside the theme can be overridden per-chart with the color argument.
zoom	Logical, DataZoom , or list of DataZoom : Enable x-axis zoom. TRUE adds a slider plus mouse-wheel/drag zoom on the x-axis; FALSE (default) disables zoom. Pass DataZoom objects (or a list of them) for full control over zoom behavior and styling.
margins	Optional Named numeric vector or named list: Plot margins in pixels (or percentage strings) for any of "top", "right", "bottom", "left" — e.g. c(left = 80, right = 20) or list(left = 80, right = "10%"). Unspecified sides keep echarts' default auto-sizing (outerBoundsMode = "same"), which shrinks the grid to fit axis labels and names with no dead space.
width	Optional Character or Numeric: Widget width.
height	Optional Character or Numeric: Widget height.
filename	Optional Character: If provided, save the widget to this file via save_drawing() .

Value

htmlwidget: Widget object.

draw_pie

Draw a Pie Chart

Description

Quick pie chart from values and labels.

Usage

```
draw_pie(
  values,
  labels,
  radius = "75%",
  rose_type = NULL,
  color = NULL,
  title = NULL,
  theme = NULL,
  width = NULL,
  height = NULL,
  filename = NULL
)
```

Arguments

values	Numeric: Slice values.
labels	Character: Slice labels.
radius	Numeric or Character: Pie radius.
rose_type	Optional Character {"radius", "area"}: Nightingale chart type.
color	Optional Character: Series color palette — a single color string or character vector that overrides the theme palette for this chart. color takes precedence over the theme palette (it sets option.color).
title	Optional Character: Chart title.
theme	Optional Theme : Theme override. The palette inside the theme can be overridden per-chart with the color argument.
width	Optional Character or Numeric: Widget width.
height	Optional Character or Numeric: Widget height.
filename	Optional Character: If provided, save the widget to this file via save_drawing() .

Value

htmlwidget: Widget object.

draw_sankey

Draw a Sankey Diagram

Description

Sankey diagram from a data frame of directed links. Node names are derived automatically from the unique values in the source and target columns; no separate node list is required.

Usage

```
draw_sankey(
  links,
  orient = "horizontal",
  node_width = NULL,
  node_gap = NULL,
  node_align = NULL,
  title = NULL,
  color = NULL,
  theme = NULL,
  width = NULL,
  height = NULL,
  filename = NULL
)
```

Arguments

links	data.frame: Directed links with columns source (Character), target (Character), and value (Numeric).
orient	Optional Character {"horizontal", "vertical"}: Flow direction.
node_width	Optional Numeric [0, Inf): Node rectangle width in pixels.
node_gap	Optional Numeric [0, Inf): Vertical gap between nodes in pixels.
node_align	Optional Character {"justify", "left", "right"}: Node alignment within each column.
title	Optional Character: Chart title.
color	Optional Character: Node color palette — a single color string or a character vector that overrides the theme palette for this chart.
theme	Optional Theme : Theme override.
width	Optional Character or Numeric: Widget width.
height	Optional Character or Numeric: Widget height.
filename	Optional Character: If provided, save the widget to this file via save_drawing() .

Value

htmlwidget: Widget object.

Examples

```
links <- data.frame(
  source = c("A", "A", "B", "C"),
  target = c("B", "C", "D", "D"),
  value = c(8, 4, 6, 3)
)
draw_sankey(links)
```

`draw_scatter`*Draw a Scatter Plot*

Description

Quick scatter plot from x/y data with optional fitted line and confidence band.

Usage

```
draw_scatter(  
  x,  
  y,  
  size = NULL,  
  group = NULL,  
  fit = NULL,  
  se = TRUE,  
  fit_alpha = 0.25,  
  n_fit = 200,  
  color = NULL,  
  xlim = NULL,  
  ylim = NULL,  
  xlab = NULL,  
  ylab = NULL,  
  title = NULL,  
  theme = NULL,  
  margins = DEFAULT_MARGINS,  
  width = NULL,  
  height = NULL,  
  filename = NULL  
)
```

Arguments

<code>x</code>	Numeric: X values.
<code>y</code>	Numeric: Y values.
<code>size</code>	Optional Numeric: Symbol sizes.
<code>group</code>	Optional Vector: Grouping variable for multiple series.
<code>fit</code>	Optional Character {"glm", "gam"}: Fit method. NULL disables fitting. "gam" for <code>mgcv::gam()</code> . The fitted line and 95% are computed per group when group is provided.
<code>se</code>	Logical: Whether to show the confidence band.
<code>fit_alpha</code>	Numeric [0, 1]: Opacity for the confidence-band fill.
<code>n_fit</code>	Numeric [1, Inf): Number of evaluation points for the fit.

color	Optional Character: Series color palette — a single color string or character vector that overrides the theme palette for this chart. When group is set, colors are assigned per group in order. color takes precedence over the theme palette.
xlim	Optional Numeric [length 2]: X-axis limits c(min, max). Defaults to range(x) padded by 4% of the span on each side.
ylim	Optional Numeric [length 2]: Y-axis limits c(min, max). Defaults to range(y) padded by 4% of the span on each side.
xlab	Optional Character: X-axis title.
ylab	Optional Character: Y-axis title.
title	Optional Character: Chart title.
theme	Optional Theme : Theme override. The palette inside the theme can be overridden per-chart with the color argument.
margins	Optional Named numeric vector or named list: Plot margins in pixels (or percentage strings) for any of "top", "right", "bottom", "left". Unspecified sides keep echarts' default auto-sizing. See draw_line() for details.
width	Optional Character or Numeric: Widget width.
height	Optional Character or Numeric: Widget height.
filename	Optional Character: If provided, save the widget to this file via save_drawing() .

Value

htmlwidget: Widget object.

draw_spectrogram	<i>Draw a Spectrogram</i>
------------------	---------------------------

Description

Renders an interactive time-frequency spectrogram as an ECharts heatmap widget. Accepts either a raw signal vector (STFT is computed internally via `signal::specgram()`) or a pre-computed spectrogram matrix (freq x time).

Usage

```
draw_spectrogram(
  x,
  sample_rate = NULL,
  time = NULL,
  frequency = NULL,
  n_fft = 256L,
  window = "hann",
  overlap = NULL,
  power = TRUE,
  db = TRUE,
```

```

db_range = 80,
freq_scale = "linear",
freq_range = NULL,
freq_unit = "Hz",
time_range = NULL,
time_unit = "s",
palette = "magma",
palette_reverse = FALSE,
n_colors = 256L,
zlim = NULL,
show_colorbar = TRUE,
colorbar_title = NULL,
title = NULL,
xlab = NULL,
ylab = NULL,
theme = NULL,
margins = NULL,
width = NULL,
height = NULL,
filename = NULL
)

```

Arguments

x	Numeric matrix (freq x time) or numeric vector (raw signal). A matrix is used directly; time and frequency vectors supply axis values (defaults to sample indices when absent). A complex matrix (raw STFT output, e.g. from <code>signal::specgram()\$S</code>) is also accepted. A vector triggers STFT computation via <code>signal::specgram()</code> ; <code>sample_rate</code> is required.
sample_rate	Optional Numeric (0, Inf): Sampling frequency in Hz. Required when x is a raw signal vector.
time	Optional Numeric: Time axis values in seconds, length <code>ncol(x)</code> . Only used when x is a matrix.
frequency	Optional Numeric: Frequency axis values in Hz, length <code>nrow(x)</code> . Only used when x is a matrix.
n_fft	Integer [2, Inf): FFT window size in samples. Passed to <code>signal::specgram()</code> as n. Only used when x is a raw signal.
window	Character {"hann", "hamming", "blackman", "bartlett", "rectangular"} or Numeric: Window function name or a pre-built window vector. Passed to <code>signal::specgram()</code> . Only used when x is a raw signal.
overlap	Optional Integer [0, n_fft): Overlap between consecutive frames in samples. Defaults to <code>n_fft / 2</code> . Only used when x is a raw signal.
power	Logical: Treat spectral values as power (TRUE) or amplitude (FALSE). When x is complex, controls whether the STFT magnitude is squared ($ S ^2$) or left as-is ($ S $). When <code>db = TRUE</code> , also determines the dB scaling for real matrices: $10 * \log_{10}()$ for power, $20 * \log_{10}()$ for amplitude.

db	Logical: Convert to dB. For power: $10 * \log_{10}()$; for amplitude: $20 * \log_{10}()$. Set FALSE when passing a pre-computed dB matrix.
db_range	Numeric (0, Inf): Dynamic range to display in dB below the spectral peak. Values below peak - db_range are clipped to the floor.
freq_scale	Character {"linear", "log"}: Frequency axis scale. With "log" the DC component (0 Hz) is automatically dropped.
freq_range	Optional Numeric[2]: Frequency range to display in Hz, e.g. c(20, 8000). Applied after STFT computation.
freq_unit	Character {"Hz", "kHz"}: Unit for the frequency axis.
time_range	Optional Numeric[2]: Time range to display in seconds.
time_unit	Character {"s", "ms"}: Unit for the time axis.
palette	Character: Colour palette. Accepts a viridisLite palette name ("magma" (default), "inferno", "plasma", "viridis", "cividis", "mako", "rocket", "turbo"), "diverging" for the rtemis teal-background-orange scale (suitable for signed data such as EEG/MEG amplitudes), or a character vector of ≥ 2 hex colours for a custom ramp.
palette_reverse	Logical: Reverse the palette direction.
n_colors	Integer [2, Inf): Number of discrete colours in the generated palette.
zlim	Optional Numeric[2]: Colour-scale limits after all transformations (dB clipping, unit conversion). Defaults to the data range.
show_colorbar	Logical: Show the continuous visual-map colorbar.
colorbar_title	Optional Character: Colorbar label. Default: "dB", "Power", or "Amplitude" derived from db and power.
title	Optional Character: Chart title.
xlab	Optional Character: X-axis label. Default: "Time (s)" or "Time (ms)" depending on time_unit.
ylab	Optional Character: Y-axis label. Default: "Frequency (Hz)" or "Frequency (kHz)" depending on freq_unit.
theme	Optional Theme , list, or NA: Theme override passed to draw() .
margins	Optional Named numeric or character vector / list: Plot margins in pixels. Valid names: "top", "right", "bottom", "left".
width	Optional Numeric or Character: Widget width.
height	Optional Numeric or Character: Widget height.
filename	Optional Character: If provided, the widget is saved via save_drawing() .

Details

Corresponds to HeatmapSeriesOption in src/chart/heatmap/HeatmapSeries.ts. ECharts docs: <https://echarts.apache.org/en/option.html#series-heatmap>

Value

htmlwidget

Examples

```
if (requireNamespace("signal", quietly = TRUE)) {
  t_vec <- seq(0, 2, by = 1 / 8000)
  sig <- signal::chirp(t_vec, 200, 2, 2000)
  draw_spectrogram(sig, sample_rate = 8000)
}
```

drawOutput	<i>Shiny output for draw widget</i>
------------	-------------------------------------

Description

Shiny output for draw widget

Usage

```
drawOutput(outputId, width = "100%", height = "400px")
```

Arguments

outputId	Character: Shiny output ID.
width	Character or Numeric: CSS width.
height	Character or Numeric: CSS height.

EChartsOption	<i>ECharts Option</i>
---------------	-----------------------

Description

The top-level configuration object passed to `echarts.setOption()`. Holds all chart components, series, and global settings.

Usage

```
EChartsOption(
  title = NULL,
  legend = NULL,
  grid = NULL,
  x_axis = NULL,
  y_axis = NULL,
  tooltip = NULL,
  visual_map = NULL,
  data_zoom = NULL,
  series = NULL,
  color = NULL,
```

```

    background_color = NULL,
    text_style = NULL,
    animation = NULL,
    animation_threshold = NULL,
    animation_duration = NULL,
    animation_easing = NULL,
    animation_delay = NULL,
    dark_mode = NULL,
    use_utc = NULL
  )

```

Arguments

title	Optional Title or list: Title configuration.
legend	Optional Legend or list: Legend configuration.
grid	Optional Grid or list: Grid configuration.
x_axis	Optional Axis or list: X-axis configuration (xAxis).
y_axis	Optional Axis or list: Y-axis configuration (yAxis).
tooltip	Optional Tooltip : Tooltip configuration.
visual_map	Optional VisualMap or list: Visual map configuration (visualMap). Required for heatmaps to map data values to colors.
data_zoom	Optional DataZoom or list of DataZoom : Axis zoom component(s) (dataZoom). Pass a list to combine, e.g., a "slider" with an "inside" zoom.
series	Optional series object or list: Series configuration (LineSeries , BarSeries , ScatterSeries , PieSeries , BoxplotSeries , HeatmapSeries).
color	Optional Character: Color palette.
background_color	Optional Character: Chart background color.
text_style	Optional TextStyle : Global default text style.
animation	Optional Logical: Whether to enable animation.
animation_threshold	Optional Numeric [0, Inf): Data-count threshold that disables animation.
animation_duration	Optional Numeric [0, Inf): Initial animation duration in milliseconds.
animation_easing	Optional Character: Animation easing function name.
animation_delay	Optional Numeric [0, Inf): Initial animation delay in milliseconds.
dark_mode	Optional Logical or Character {"auto"}: Dark-mode setting.
use_utc	Optional Logical: Whether to use UTC for time axes.

Details

Corresponds to EChartsOption in src/export/option.ts (line 258) and ECUntilOption in src/util/types.ts (line 697).

 Grid

*Grid***Description**

Configuration for the cartesian coordinate grid container.

Usage

```
Grid(
  show = NULL,
  left = NULL,
  right = NULL,
  top = NULL,
  bottom = NULL,
  width = NULL,
  height = NULL,
  contain_label = NULL,
  background_color = NULL,
  border_width = NULL,
  border_color = NULL,
  shadow_blur = NULL,
  shadow_color = NULL,
  shadow_offset_x = NULL,
  shadow_offset_y = NULL
)
```

Arguments

show	Optional Logical: Whether to show the grid border.
left	Optional Numeric or Character: Distance from the left side of the container.
right	Optional Numeric or Character: Distance from the right side of the container.
top	Optional Numeric or Character: Distance from the top of the container.
bottom	Optional Numeric or Character: Distance from the bottom of the container.
width	Optional Numeric or Character: Grid width.
height	Optional Numeric or Character: Grid height.
contain_label	Optional Logical: Whether the grid region contains axis labels.
background_color	Optional Character: Grid background color.
border_width	Optional Numeric [0, Inf): Grid border width.
border_color	Optional Character: Grid border color.
shadow_blur	Optional Numeric [0, Inf): Shadow blur radius.
shadow_color	Optional Character: Shadow color.

shadow_offset_x
Optional Numeric: Shadow horizontal offset.

shadow_offset_y
Optional Numeric: Shadow vertical offset.

Details

Corresponds to GridOption in src/coord/cartesian/GridModel.ts (line 37). ECharts docs: <https://echarts.apache.org/en/option.html#grid>

HeatmapSeries	<i>Heatmap Series</i>
---------------	-----------------------

Description

Configuration for a heatmap chart series on Cartesian coordinates. Each data point is a triple [col_index, row_index, value] and color encoding is driven by a [VisualMap](#) component.

Usage

```
HeatmapSeries(
  name = NULL,
  data = NULL,
  x_axis_index = NULL,
  y_axis_index = NULL,
  silent = NULL,
  legend_hover_link = NULL,
  z_level = NULL,
  z = NULL,
  label = NULL,
  item_style = NULL,
  emphasis = NULL
)
```

Arguments

name	Optional Character: Series name.
data	Optional list: Heatmap data. Each element is a length-3 list list(col_index, row_index, value).
x_axis_index	Optional Numeric [0, Inf): X-axis index.
y_axis_index	Optional Numeric [0, Inf): Y-axis index.
silent	Optional Logical: Whether the series is silent (no events).
legend_hover_link	Optional Logical: Whether to link hover with the legend.
z_level	Optional Numeric: Canvas layer index.

z	Optional Numeric: Front-back order within the same layer.
label	Optional LabelOption : Cell label configuration (e.g. show the value).
item_style	Optional ItemStyle : Cell styling (e.g. border color/width).
emphasis	Optional list: Emphasis (hover) styling. A plain list passed through to ECharts, e.g. <code>list(itemStyle = list(shadowBlur = 10))</code> .

Details

Corresponds to `HeatmapSeriesOption` in `src/chart/heatmap/HeatmapSeries.ts`. ECharts docs: <https://echarts.apache.org/en/option.html#series-heatmap>

ItemStyle	<i>Item Style</i>
-----------	-------------------

Description

Styling for graphical items (bar segments, scatter points, pie slices, etc.). Includes both fill and stroke options.

Usage

```
ItemStyle(
  color = NULL,
  opacity = NULL,
  border_color = NULL,
  border_width = NULL,
  border_type = NULL,
  border_cap = NULL,
  border_join = NULL,
  border_dash_offset = NULL,
  border_miter_limit = NULL,
  border_radius = NULL,
  decal = NULL,
  shadow_blur = NULL,
  shadow_color = NULL,
  shadow_offset_x = NULL,
  shadow_offset_y = NULL
)
```

Arguments

color	Optional Character: CSS fill color.
opacity	Optional Numeric [0, 1]: Fill opacity.
border_color	Optional Character: Border color.
border_width	Optional Numeric [0, Inf): Border width in pixels.

<code>border_type</code>	Optional Character {"solid", "dashed", "dotted"}: Border line type.
<code>border_cap</code>	Optional Character {"butt", "round", "square"}: Border line cap.
<code>border_join</code>	Optional Character {"bevel", "round", "miter"}: Border line join.
<code>border_dash_offset</code>	Optional Numeric: Border dash offset.
<code>border_miter_limit</code>	Optional Numeric [0, Inf): Border miter limit.
<code>border_radius</code>	Optional Numeric, Character, or Numeric vector: Corner border radius.
<code>decal</code>	Optional Any: Decal pattern for accessibility.
<code>shadow_blur</code>	Optional Numeric [0, Inf): Shadow blur radius.
<code>shadow_color</code>	Optional Character: Shadow color.
<code>shadow_offset_x</code>	Optional Numeric: Shadow horizontal offset.
<code>shadow_offset_y</code>	Optional Numeric: Shadow vertical offset.

Details

Corresponds to `ItemStyleOption` in `src/util/types.ts` (line 1142). ECharts docs: <https://echarts.apache.org/en/option.html#series-bar.itemStyle>

LabelLine

Label Line

Description

Connecting line between a label and its chart element (e.g. pie label lines).

Usage

```
LabelLine(
  show = NULL,
  show_above = NULL,
  length = NULL,
  length2 = NULL,
  smooth = NULL,
  min_turn_angle = NULL,
  line_style = NULL
)
```

Arguments

show	Optional Logical: Whether to show the label line.
show_above	Optional Logical: Whether the line renders above other elements.
length	Optional Numeric [0, Inf): Length of the first segment.
length2	Optional Numeric [0, Inf): Length of the second segment.
smooth	Optional Numeric [0, 1] or Logical: Line smoothness.
min_turn_angle	Optional Numeric: Minimum angle for turning the line.
line_style	Optional LineStyle : Line styling.

Details

Corresponds to LabelLineOption in src/util/types.ts (line 1379). ECharts docs: <https://echarts.apache.org/en/option.html#series-pie.labelLine>

 LabelOption

Label Option

Description

Text label configuration for chart elements. Combines label-specific positioning (show, position, distance, rotate) with text styling.

Usage

```
LabelOption(
  show = NULL,
  position = NULL,
  distance = NULL,
  rotate = NULL,
  offset = NULL,
  formatter = NULL,
  silent = NULL,
  precision = NULL,
  value_animation = NULL,
  min_margin = NULL,
  text_style = NULL
)
```

Arguments

show	Optional Logical: Whether to show the label.
------	--

position	Optional Character {"top", "left", "right", "bottom", "inside", "insideLeft", "insideRight", "insideTop", "insideBottom", "insideTopLeft", "insideBottomLeft", "insideTopRight", "insideBottomRight", "outside"} or Numeric: Label position. "bottom", "inside", "insideLeft", "insideRight", "insideTop", "insideBottom", "insideTopLeft", "insideBottomLeft", "insideTopRight", "insideBottomRight", "outside".
distance	Optional Numeric: Distance from the element.
rotate	Optional Numeric: Rotation angle in degrees.
offset	Optional Numeric: Offset c(x, y) in pixels.
formatter	Optional Character or function: Label text formatter.
silent	Optional Logical: Whether the label is silent.
precision	Optional Numeric or Character {"auto"}: Numeric precision.
value_animation	Optional Logical: Whether to animate value changes.
min_margin	Optional Numeric [0, Inf): Minimum margin between labels.
text_style	Optional TextStyle : Text appearance.

Details

Corresponds to `SeriesLabelOption / LabelOption` in `src/util/types.ts` (line 1283, 1348).
 ECharts docs: <https://echarts.apache.org/en/option.html#series-bar.label>

Legend

Legend

Description

Legend configuration for selecting/filtering series.

Usage

```
Legend(
  show = NULL,
  orient = NULL,
  align = NULL,
  left = NULL,
  right = NULL,
  top = NULL,
  bottom = NULL,
  width = NULL,
  height = NULL,
  padding = NULL,
  item_gap = NULL,
  item_width = NULL,
```

```

    item_height = NULL,
    icon = NULL,
    selected_mode = NULL,
    selected = NULL,
    formatter = NULL,
    inactive_color = NULL,
    inactive_border_color = NULL,
    text_style = NULL,
    item_style = NULL,
    background_color = NULL,
    border_color = NULL,
    border_width = NULL,
    border_radius = NULL,
    trigger_event = NULL,
    data = NULL
)

```

Arguments

show	Optional Logical: Whether to show the legend.
orient	Optional Character {"horizontal", "vertical"}: Layout orientation.
align	Optional Character {"auto", "left", "right"}: Marker and text alignment.
left	Optional Numeric or Character: Left position.
right	Optional Numeric or Character: Right position.
top	Optional Numeric or Character: Top position.
bottom	Optional Numeric or Character: Bottom position.
width	Optional Numeric or Character: Legend box width.
height	Optional Numeric or Character: Legend box height.
padding	Optional Numeric: Padding inside the legend box.
item_gap	Optional Numeric [0, Inf): Gap between legend items.
item_width	Optional Numeric [0, Inf): Legend symbol width.
item_height	Optional Numeric [0, Inf): Legend symbol height.
icon	Optional Character: Default legend icon.
selected_mode	Optional Logical or Character {"single", "multiple"}: Selection mode.
selected	Optional Named logical vector: Pre-selected items.
formatter	Optional Character or function: Label formatter.
inactive_color	Optional Character: Color for unselected items.
inactive_border_color	Optional Character: Border color for unselected items.
text_style	Optional TextStyle : Legend label text style.
item_style	Optional ItemStyle : Legend icon style.
background_color	Optional Character: Background color.

border_color	Optional Character: Border color.
border_width	Optional Numeric [0 , Inf): Border width.
border_radius	Optional Numeric: Border radius.
trigger_event	Optional Logical: Whether the legend emits events.
data	Optional Character vector or list: Legend data items.

Details

Corresponds to LegendOption in `src/component/legend/LegendModel.ts` (line 162). ECharts docs: <https://echarts.apache.org/en/option.html#legend>

LineSeries

Line Series

Description

Configuration for a line chart series.

Usage

```
LineSeries(  
  name = NULL,  
  data = NULL,  
  x_axis_index = NULL,  
  y_axis_index = NULL,  
  stack = NULL,  
  silent = NULL,  
  legend_hover_link = NULL,  
  color = NULL,  
  z_level = NULL,  
  z = NULL,  
  smooth = NULL,  
  step = NULL,  
  connect_nulls = NULL,  
  clip = NULL,  
  show_symbol = NULL,  
  symbol = NULL,  
  symbol_size = NULL,  
  line_style = NULL,  
  area_style = NULL,  
  item_style = NULL,  
  label = NULL,  
  mark_area = NULL  
)
```

Arguments

name	Optional Character: Series name for legend display.
data	Optional Numeric or list: Data values.
x_axis_index	Optional Numeric [0, Inf): X-axis index.
y_axis_index	Optional Numeric [0, Inf): Y-axis index.
stack	Optional Character: Stack group name.
silent	Optional Logical: Whether the series is silent.
legend_hover_link	Optional Logical: Whether to highlight related legends on hover.
color	Optional Character: Series color override.
z_level	Optional Numeric: Canvas layer index.
z	Optional Numeric: Front-back order within the same layer.
smooth	Optional Logical or Numeric [0, 1]: Line smoothing.
step	Optional Logical or Character {"start", "end", "middle"}: Step line mode.
connect_nulls	Optional Logical: Whether to connect across NULL points.
clip	Optional Logical: Whether to clip overflow.
show_symbol	Optional Logical: Whether to show data-point symbols.
symbol	Optional Character {"circle", "rect", "roundRect", "triangle", "diamond", "pin", "arrow", "none"}: Symbol type. "diamond", "pin", "arrow", "none", or custom SVG path.
symbol_size	Optional Numeric: Symbol size in pixels.
line_style	Optional LineStyle : Line styling.
area_style	Optional AreaStyle : Area fill styling.
item_style	Optional ItemStyle : Data-point marker styling.
label	Optional LabelOption : Data-label configuration.
mark_area	Optional MarkArea : Shaded bands (e.g. vertical highlight regions across x-ranges).

Details

Corresponds to `LineStyleOption` in `src/chart/line/LineSeries.ts` (line 74). ECharts docs: <https://echarts.apache.org/en/option.html#series-line>

LineStyle

Line Style

Description

Stroke styling for lines in series, axes, and components.

Usage

```
LineStyle(  
  color = NULL,  
  width = NULL,  
  opacity = NULL,  
  type = NULL,  
  cap = NULL,  
  join = NULL,  
  dash_offset = NULL,  
  miter_limit = NULL,  
  shadow_blur = NULL,  
  shadow_color = NULL,  
  shadow_offset_x = NULL,  
  shadow_offset_y = NULL  
)
```

Arguments

color	Optional Character: CSS line color.
width	Optional Numeric [0, Inf): Line width in pixels.
opacity	Optional Numeric [0, 1]: Line opacity.
type	Optional Character {"solid", "dashed", "dotted"}: Line type.
cap	Optional Character {"butt", "round", "square"}: Line cap.
join	Optional Character {"bevel", "round", "miter"}: Line join.
dash_offset	Optional Numeric: Dash offset for dashed lines.
miter_limit	Optional Numeric [0, Inf): Miter limit for miter joins.
shadow_blur	Optional Numeric [0, Inf): Shadow blur radius.
shadow_color	Optional Character: Shadow color.
shadow_offset_x	Optional Numeric: Shadow horizontal offset.
shadow_offset_y	Optional Numeric: Shadow vertical offset.

Details

Corresponds to `LineStyleOption` in `src/util/types.ts` (line 1154). ECharts docs: <https://echarts.apache.org/en/option.html#series-line.lineStyle>

MarkArea	<i>Mark Area</i>
----------	------------------

Description

Container for markArea bands attached to a series. data is a list where each element is a length-2 list of [MarkAreaDataPoint](#) (or plain named list) defining the two opposite corners of one band.

Usage

```
MarkArea(
  data = NULL,
  silent = NULL,
  precision = NULL,
  item_style = NULL,
  label = NULL,
  z2 = NULL
)
```

Arguments

data	Optional list: List of length-2 lists of MarkAreaDataPoint / plain named list. Each inner pair defines one band.
silent	Optional Logical: Whether hover / tooltip are disabled.
precision	Optional Integer [0, Inf): Decimal precision used by statistical methods ("min", "max", etc.).
item_style	Optional ItemStyle : Default fill/border styling for all bands. Per-band overrides go on the first MarkAreaDataPoint of a pair.
label	Optional LabelOption : Default label configuration.
z2	Optional Numeric: Front-back order among markers.

Details

Corresponds to MarkAreaOption in src/component/marker/MarkAreaModel.ts (line 60). ECharts docs: <https://echarts.apache.org/en/option.html#series-line.markArea>

MarkAreaDataPoint	<i>Mark Area Data Point</i>
-------------------	-----------------------------

Description

One endpoint of a markArea band. Two of these — wrapped in a length-2 list — define the opposite corners of a shaded rectangle. Supplying only x_axis on both endpoints produces a full-height vertical band (useful for highlighting x-ranges of a time series); supplying only y_axis produces a full-width horizontal band.

Usage

```
MarkAreaDataPoint(
  x_axis = NULL,
  y_axis = NULL,
  name = NULL,
  value = NULL,
  item_style = NULL,
  label = NULL
)
```

Arguments

x_axis	Optional Numeric or Character: Position on the x-axis.
y_axis	Optional Numeric or Character: Position on the y-axis.
name	Optional Character: Label text for the band.
value	Optional Numeric or Character: Fallback label value when name is not set.
item_style	Optional ItemStyle : Fill and border styling. Styling set on the first endpoint of a pair applies to the whole band.
label	Optional LabelOption : Label configuration.

Details

Corresponds to MarkArea2DDataItemDimOption in src/component/marker/MarkAreaModel.ts (line 49). ECharts docs: <https://echarts.apache.org/en/option.html#series-line.markArea.data>

MinorSplitLine	<i>Minor Split Line</i>
----------------	-------------------------

Description

Configuration for minor split lines (between main split lines).

Usage

```
MinorSplitLine(show = NULL, line_style = NULL)
```

Arguments

show	Optional Logical: Whether to show minor split lines.
line_style	Optional LineStyle : Line styling.

Details

Corresponds to MinorSplitLineOption in src/coord/axisCommonTypes.ts (line 364). ECharts docs: <https://echarts.apache.org/en/option.html#xAxis.minorSplitLine>

MinorTick

Minor Tick

Description

Configuration for minor axis tick marks (between main ticks).

Usage

```
MinorTick(show = NULL, split_number = NULL, length = NULL, line_style = NULL)
```

Arguments

show	Optional Logical: Whether to show minor ticks.
split_number	Optional Numeric [0, Inf): Number of minor ticks between two main ticks.
length	Optional Numeric [0, Inf): Minor tick length in pixels.
line_style	Optional LineStyle : Line styling.

Details

Corresponds to `MinorTickOption` in `src/coord/axisCommonTypes.ts` (line 346). ECharts docs: <https://echarts.apache.org/en/option.html#xAxis.minorTick>

PieSeries

Pie Series

Description

Configuration for a pie chart series.

Usage

```
PieSeries(  
  name = NULL,  
  data = NULL,  
  silent = NULL,  
  legend_hover_link = NULL,  
  color = NULL,  
  z_level = NULL,  
  z = NULL,  
  center = NULL,  
  radius = NULL,  
  rose_type = NULL,  
  clockwise = NULL,  
  start_angle = NULL,
```

```

    end_angle = NULL,
    pad_angle = NULL,
    min_angle = NULL,
    min_show_label_angle = NULL,
    selected_offset = NULL,
    avoid_label_overlap = NULL,
    percent_precision = NULL,
    still_show_zero_sum = NULL,
    animation_type = NULL,
    item_style = NULL,
    label = NULL,
    label_line = NULL,
    left = NULL,
    right = NULL,
    top = NULL,
    bottom = NULL
)

```

Arguments

name	Optional Character: Series name.
data	Optional Numeric or list: Pie data. A list of named lists with value and name fields, or a numeric vector.
silent	Optional Logical: Whether the series is silent.
legend_hover_link	Optional Logical: Whether to highlight related legends on hover.
color	Optional Character: Series color override.
z_level	Optional Numeric: Canvas layer index.
z	Optional Numeric: Front-back order within the same layer.
center	Optional Numeric or Character: Pie center as a length-2 vector.
radius	Optional Numeric or Character: Pie radius.
rose_type	Optional Character {"radius", "area"}: Nightingale chart type.
clockwise	Optional Logical: Whether to lay out sectors clockwise.
start_angle	Optional Numeric: Starting angle in degrees.
end_angle	Optional Numeric or Character {"auto"}: Ending angle in degrees.
pad_angle	Optional Numeric [0, Inf): Padding angle between sectors in degrees.
min_angle	Optional Numeric [0, Inf): Minimum sector angle.
min_show_label_angle	Optional Numeric [0, Inf): Minimum angle required to show labels.
selected_offset	Optional Numeric [0, Inf): Offset of a selected sector.
avoid_label_overlap	Optional Logical: Whether to avoid label overlap automatically.

percent_precision	Optional Numeric [0 , Inf): Decimal precision for percentages.
still_show_zero_sum	Optional Logical: Whether to show the pie when all values are zero.
animation_type	Optional Character {"expansion", "scale"}: First-render animation type.
item_style	Optional ItemStyle : Pie-sector styling.
label	Optional LabelOption : Data-label configuration.
label_line	Optional LabelLine : Label connector line configuration.
left	Optional Numeric or Character: Left position.
right	Optional Numeric or Character: Right position.
top	Optional Numeric or Character: Top position.
bottom	Optional Numeric or Character: Bottom position.

Details

Corresponds to `PieSeriesOption` in `src/chart/pie/PieSeries.ts` (line 105). ECharts docs: <https://echarts.apache.org/en/option.html#series-pie>

renderDraw	<i>Shiny render function for draw widget</i>
------------	--

Description

Shiny render function for draw widget

Usage

```
renderDraw(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

expr	Expression: Expression that returns a draw widget.
env	Environment: Evaluation environment.
quoted	Logical: Whether expr is quoted.

rtemis_colors	<i>rtemis Color Palette</i>
---------------	-----------------------------

Description

rtemis Color Palette

Usage

rtemis_colors

Author(s)

EDG

SankeyEdgeItem	<i>Sankey Edge Item</i>
----------------	-------------------------

Description

One directed link (edge) in a Sankey diagram.

Usage

```
SankeyEdgeItem(
  source = NULL,
  target = NULL,
  value = NULL,
  line_style = NULL,
  edge_label = NULL,
  focus_node_adjacency = NULL
)
```

Arguments

source	Optional Character: Source node name.
target	Optional Character: Target node name.
value	Optional Numeric: Flow value (determines link width).
line_style	Optional LineStyle or list: Link line styling. Accepts a LineStyle object or a plain named list; a list also allows the Sankey-specific curveness field (e.g. <code>list(curveness = 0.5, opacity = 0.3)</code>).
edge_label	Optional LabelOption : Label shown on the link itself.
focus_node_adjacency	Optional Logical or Character {"inEdges", "outEdges", "allEdges"}: Which adjacent elements to highlight on hover.

Details

Corresponds to SankeyEdgeItemOption in src/chart/sankey/SankeySeries.ts (line 88). ECharts docs: <https://echarts.apache.org/en/option.html#series-sankey.links>

SankeyLevelOption *Sankey Level Option*

Description

Style overrides applied to all nodes at a given column depth. depth = 0 targets the leftmost column.

Usage

```
SankeyLevelOption(
  depth = integer(0),
  label = NULL,
  item_style = NULL,
  line_style = NULL
)
```

Arguments

depth	Numeric [0, Inf): Column depth to target (0 = leftmost). Required.
label	Optional LabelOption : Node label configuration at this depth.
item_style	Optional ItemStyle : Node fill/border styling at this depth.
line_style	Optional LineStyle or list: Link styling for edges leaving this depth. Accepts a LineStyle object or a plain list (allows curviness).

Details

Corresponds to SankeyLevelOption in src/chart/sankey/SankeySeries.ts (line 96). ECharts docs: <https://echarts.apache.org/en/option.html#series-sankey.levels>

SankeyNodeItem *Sankey Node Item*

Description

One node in a Sankey diagram. Node names must match the source / target strings used in the corresponding [SankeyEdgeItem](#) objects.

Usage

```
SankeyNodeItem(
  name = NULL,
  value = NULL,
  depth = NULL,
  draggable = NULL,
  local_x = NULL,
  local_y = NULL,
  label = NULL,
  item_style = NULL,
  focus_node_adjacency = NULL
)
```

Arguments

name	Optional Character: Node name (must match source/target strings).
value	Optional Numeric: Node value shown in tooltip.
depth	Optional Numeric [0 , Inf): Fixed column depth (0 = leftmost).
draggable	Optional Logical: Whether the node can be dragged.
local_x	Optional Numeric: Horizontal position inside the layout rect.
local_y	Optional Numeric: Vertical position inside the layout rect.
label	Optional LabelOption : Node label configuration.
item_style	Optional ItemStyle : Node fill/border styling.
focus_node_adjacency	Optional Logical or Character {"inEdges", "outEdges", "allEdges"}: Which adjacent elements to highlight on hover.

Details

Corresponds to SankeyNodeItemOption in src/chart/sankey/SankeySeries.ts (line 73). ECharts docs: <https://echarts.apache.org/en/option.html#series-sankey.data>

 SankeySeries

Sankey Series

Description

Configuration for a Sankey diagram series. Nodes and directed links are represented as two separate lists. Node names in data must match the source and target strings in links.

Usage

```

SankeySeries(
  name = NULL,
  data = NULL,
  links = NULL,
  levels = NULL,
  orient = NULL,
  node_width = NULL,
  node_gap = NULL,
  node_align = NULL,
  draggable = NULL,
  layout_iterations = NULL,
  sort = NULL,
  roam = NULL,
  label = NULL,
  edge_label = NULL,
  line_style = NULL,
  item_style = NULL,
  color = NULL,
  left = NULL,
  right = NULL,
  top = NULL,
  bottom = NULL,
  silent = NULL,
  z_level = NULL,
  z = NULL
)

```

Arguments

name	Optional Character: Series name for legend and tooltip.
data	Optional list: Nodes as a list of SankeyNodeItem objects or plain named lists, each with at least a name field.
links	Optional list: Edges as a list of SankeyEdgeItem objects or plain named lists with source, target, and value fields.
levels	Optional list: List of SankeyLevelOption for per-depth styling.
orient	Optional Character {"horizontal", "vertical"}: Flow direction.
node_width	Optional Numeric [0, Inf): Width of each node rectangle in pixels.
node_gap	Optional Numeric [0, Inf): Vertical gap between nodes in pixels.
node_align	Optional Character {"justify", "left", "right"}: Node alignment within columns.
draggable	Optional Logical: Whether nodes can be dragged interactively.
layout_iterations	Optional Numeric [0, Inf): Number of layout iterations.
sort	Optional Character {"desc"} or NULL: Column-internal node ordering. NULL preserves the original data order.

roam	Optional Logical or Character {"move", "scale"}: Whether to enable pan/zoom.
label	Optional LabelOption : Node label configuration.
edge_label	Optional LabelOption : Edge label configuration.
line_style	Optional LineStyle or list: Default link line styling. Accepts a LineStyle object or a plain list; a plain list also allows the Sankey-specific curveness field (e.g. <code>list(curveness = 0.5)</code>).
item_style	Optional ItemStyle : Default node styling.
color	Optional Character: Color palette for nodes as a character vector.
left	Optional Numeric or Character: Series left offset.
right	Optional Numeric or Character: Series right offset.
top	Optional Numeric or Character: Series top offset.
bottom	Optional Numeric or Character: Series bottom offset.
silent	Optional Logical: Whether to disable mouse/touch events.
z_level	Optional Numeric: Canvas layer index.
z	Optional Numeric: Front-back order within the same canvas layer.

Details

Corresponds to `SankeySeriesOption` in `src/chart/sankey/SankeySeries.ts` (line 100). ECharts docs: <https://echarts.apache.org/en/option.html#series-sankey>

save_drawing	<i>Save a Draw Widget to a File</i>
--------------	-------------------------------------

Description

Exports a widget created by `draw()` (or any of the `draw_*` functions) to a static file. Currently supports `.svg` via Node.js server-side rendering. Requires a node binary on `PATH`.

Usage

```
save_drawing(widget, filename, width = 800, height = 600)
```

Arguments

widget	htmlwidget: A widget returned by <code>draw()</code> or a <code>draw_*</code> function.
filename	Character: Output file path. Extension determines the format.
width	Numeric: Image width in pixels.
height	Numeric: Image height in pixels.

Details

Interactive features (tooltip formatters, event handlers) are stripped before rendering — static images cannot express them.

Value

The filename, invisibly.

ScatterSeries

Scatter Series

Description

Configuration for a scatter (point) chart series.

Usage

```
ScatterSeries(
  name = NULL,
  data = NULL,
  x_axis_index = NULL,
  y_axis_index = NULL,
  silent = NULL,
  legend_hover_link = NULL,
  color = NULL,
  z_level = NULL,
  z = NULL,
  clip = NULL,
  symbol = NULL,
  symbol_size = NULL,
  large = NULL,
  large_threshold = NULL,
  item_style = NULL,
  label = NULL
)
```

Arguments

name	Optional Character: Series name for legend display.
data	Optional Vector or list: Data values.
x_axis_index	Optional Numeric [0, Inf): X-axis index.
y_axis_index	Optional Numeric [0, Inf): Y-axis index.
silent	Optional Logical: Whether the series is silent.
legend_hover_link	Optional Logical: Whether to highlight related legends on hover.
color	Optional Character: Series color override.
z_level	Optional Numeric: Canvas layer index.
z	Optional Numeric: Front-back order within the same layer.
clip	Optional Logical: Whether to clip overflow.

symbol	Optional Character: Symbol type.
symbol_size	Optional Numeric or function: Symbol size.
large	Optional Logical: Whether to enable large-data optimization.
large_threshold	Optional Numeric [0, Inf): Threshold for large mode.
item_style	Optional ItemStyle : Data-point marker styling.
label	Optional LabelOption : Data-label configuration.

Details

Corresponds to ScatterSeriesOption in `src/chart/scatter/ScatterSeries.ts` (line 64). ECharts docs: <https://echarts.apache.org/en/option.html#series-scatter>

SplitArea

Split Area

Description

Configuration for alternating colored bands between split lines.

Usage

```
SplitArea(show = NULL, interval = NULL, area_style = NULL)
```

Arguments

show	Optional Logical: Whether to show the split area.
interval	Optional Numeric or Character {"auto"}: Split-area interval.
area_style	Optional AreaStyle : Area styling.

Details

Corresponds to SplitAreaOption in `src/coord/axisCommonTypes.ts` (line 369). ECharts docs: <https://echarts.apache.org/en/option.html#xAxis.splitArea>

SplitLine	<i>Split Line</i>
-----------	-------------------

Description

Configuration for grid split lines perpendicular to the axis.

Usage

```
SplitLine(  
  show = NULL,  
  interval = NULL,  
  show_min_line = NULL,  
  show_max_line = NULL,  
  line_style = NULL  
)
```

Arguments

show	Optional Logical: Whether to show split lines.
interval	Optional Numeric or Character {"auto"}: Split-line interval.
show_min_line	Optional Logical: Whether to show the split line at the minimum end.
show_max_line	Optional Logical: Whether to show the split line at the maximum end.
line_style	Optional LineStyle : Line styling.

Details

Corresponds to `SplitLineOption` in `src/coord/axisCommonTypes.ts` (line 353). ECharts docs: <https://echarts.apache.org/en/option.html#xAxis.splitLine>

TextStyle	<i>Text Style</i>
-----------	-------------------

Description

Text styling used in labels, titles, tooltips, and other text elements.

Usage

```
TextStyle(  
  color = NULL,  
  font_style = NULL,  
  font_weight = NULL,  
  font_family = NULL,  
  font_size = NULL,  
  align = NULL,  
  vertical_align = NULL,  
  opacity = NULL,  
  line_height = NULL,  
  background_color = NULL,  
  border_color = NULL,  
  border_width = NULL,  
  border_type = NULL,  
  border_dash_offset = NULL,  
  border_radius = NULL,  
  padding = NULL,  
  width = NULL,  
  height = NULL,  
  text_border_color = NULL,  
  text_border_width = NULL,  
  text_border_type = NULL,  
  text_border_dash_offset = NULL,  
  text_shadow_blur = NULL,  
  text_shadow_color = NULL,  
  text_shadow_offset_x = NULL,  
  text_shadow_offset_y = NULL,  
  overflow = NULL,  
  ellipsis = NULL,  
  rich = NULL,  
  shadow_blur = NULL,  
  shadow_color = NULL,  
  shadow_offset_x = NULL,  
  shadow_offset_y = NULL  
)
```

Arguments

color	Optional Character: Text color.
font_style	Optional Character {"normal", "italic", "oblique"}: Font style.
font_weight	Optional Character {"normal", "bold", "bolder", "lighter"} or Numeric [100, 900]: Font weight.
font_family	Optional Character: Font family name.
font_size	Optional Numeric or Character: Font size.
align	Optional Character {"left", "center", "right"}: Horizontal alignment.
vertical_align	Optional Character {"top", "middle", "bottom"}: Vertical alignment.

opacity	Optional Numeric [0, 1]: Text opacity.
line_height	Optional Numeric [0, Inf): Line height in pixels.
background_color	Optional Character: Background color behind text.
border_color	Optional Character: Text box border color.
border_width	Optional Numeric [0, Inf): Text box border width.
border_type	Optional Character {"solid", "dashed", "dotted"}: Text box border type.
border_dash_offset	Optional Numeric: Text box border dash offset.
border_radius	Optional Numeric or Numeric vector: Text box corner radius.
padding	Optional Numeric or Numeric vector: Text box padding.
width	Optional Numeric or Character: Text box width.
height	Optional Numeric [0, Inf): Text box height.
text_border_color	Optional Character: Stroke color around the text.
text_border_width	Optional Numeric [0, Inf): Stroke width around the text.
text_border_type	Optional Character {"solid", "dashed", "dotted"}: Stroke type around the text.
text_border_dash_offset	Optional Numeric: Stroke dash offset around the text.
text_shadow_blur	Optional Numeric [0, Inf): Text shadow blur radius.
text_shadow_color	Optional Character: Text shadow color.
text_shadow_offset_x	Optional Numeric: Text shadow horizontal offset.
text_shadow_offset_y	Optional Numeric: Text shadow vertical offset.
overflow	Optional Character {"truncate", "break", "breakAll", "none"}: Text overflow behavior.
ellipsis	Optional Character: Ellipsis string used when truncating.
rich	Optional Named list: Rich text styles.
shadow_blur	Optional Numeric [0, Inf): Box shadow blur radius.
shadow_color	Optional Character: Box shadow color.
shadow_offset_x	Optional Numeric: Box shadow horizontal offset.
shadow_offset_y	Optional Numeric: Box shadow vertical offset.

Details

Corresponds to `TextCommonOption` in `src/util/types.ts` (line 1220). ECharts docs: <https://echarts.apache.org/en/option.html#title.textStyle>

Theme	<i>Theme</i>
-------	--------------

Description

A chart theme that controls default colors, text styles, and component appearance. Corresponds to the object passed as the second argument to `echarts.init(dom, theme)` or registered via `echarts.registerTheme()`.

Usage

```
Theme(  
  color = NULL,  
  background_color = NULL,  
  text_style = NULL,  
  title = NULL,  
  legend = NULL,  
  tooltip = NULL,  
  line = NULL,  
  bar = NULL,  
  pie = NULL,  
  scatter = NULL,  
  category_axis = NULL,  
  value_axis = NULL,  
  log_axis = NULL,  
  time_axis = NULL  
)
```

Arguments

<code>color</code>	Optional Character: Color palette.
<code>background_color</code>	Optional Character: Chart background color.
<code>text_style</code>	Optional TextStyle : Global default text style.
<code>title</code>	Optional Named list: Title overrides.
<code>legend</code>	Optional Named list: Legend overrides.
<code>tooltip</code>	Optional Named list: Tooltip overrides.
<code>line</code>	Optional Named list: Line-series defaults.
<code>bar</code>	Optional Named list: Bar-series defaults.
<code>pie</code>	Optional Named list: Pie-series defaults.
<code>scatter</code>	Optional Named list: Scatter-series defaults.
<code>category_axis</code>	Optional Named list: Category-axis defaults.
<code>value_axis</code>	Optional Named list: Value-axis defaults.
<code>log_axis</code>	Optional Named list: Log-axis defaults.
<code>time_axis</code>	Optional Named list: Time-axis defaults.

Details

TS source: `src/model/globalDefault.ts, src/theme/dark.ts`.

theme_dark

Dark Theme

Description

Returns a dark theme. All parameters have sensible defaults; most users only need `base_font_size`. Title font size defaults to 1.2x the base; all other text defaults to 1x.

Usage

```
theme_dark(
  base_font_size = 12,
  title_font_size = NULL,
  subtitle_font_size = NULL,
  axis_label_font_size = NULL,
  legend_font_size = NULL,
  tooltip_font_size = NULL,
  font_family = "system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif",
  color = rtemis_colors,
  bg_color = "#181818",
  fg_color = "rgba(255, 255, 255, 0.7)",
  title_color = "rgba(255, 255, 255, 0.9)",
  subtitle_color = "rgba(255, 255, 255, 0.5)",
  legend_color = NULL,
  axis_color = "rgba(255, 255, 255, 0.3)",
  grid_color = "rgba(255, 255, 255, 0.1)",
  tooltip_bg = "rgba(20, 20, 20, 0.9)",
  tooltip_border_color = "rgba(255, 255, 255, 0.1)",
  tooltip_color = NULL
)
```

Arguments

`base_font_size` Numeric [0, Inf): Base font size in pixels for all text.

`title_font_size`
Optional Numeric [0, Inf): Title font size.

`subtitle_font_size`
Optional Numeric [0, Inf): Subtitle font size.

`axis_label_font_size`
Optional Numeric [0, Inf): Axis-label font size.

`legend_font_size`
Optional Numeric [0, Inf): Legend font size.

tooltip_font_size	Optional Numeric [0, Inf): Tooltip font size.
font_family	Character: Font family string.
color	Character: Color palette.
bg_color	Optional Character: Chart background color.
fg_color	Optional Character: Global text color.
title_color	Optional Character: Title text color.
subtitle_color	Optional Character: Subtitle text color.
legend_color	Optional Character: Legend text color.
axis_color	Optional Character: Axis line color.
grid_color	Optional Character: Grid or split-line color.
tooltip_bg	Optional Character: Tooltip background color.
tooltip_border_color	Optional Character: Tooltip border color.
tooltip_color	Optional Character: Tooltip text color.

Value

[Theme](#): Theme object.

theme_light	<i>Light Theme</i>
-------------	--------------------

Description

Returns a light theme. All parameters have sensible defaults; most users only need `base_font_size`. Title font size defaults to 1.2x the base; all other text defaults to 1x.

Usage

```
theme_light(
    base_font_size = 12,
    title_font_size = NULL,
    subtitle_font_size = NULL,
    axis_label_font_size = NULL,
    legend_font_size = NULL,
    tooltip_font_size = NULL,
    font_family = "system-ui, -apple-system, BlinkMacSystemFont, 'Segoe UI', sans-serif",
    color = rtemis_colors,
    bg_color = "#ffffff",
    fg_color = "rgba(0, 0, 0, 0.7)",
    title_color = NULL,
    subtitle_color = NULL,
    legend_color = NULL,
```

```

    axis_color = NULL,
    grid_color = NULL,
    tooltip_bg = NULL,
    tooltip_border_color = NULL,
    tooltip_color = NULL
)

```

Arguments

`base_font_size` Numeric [0, Inf): Base font size in pixels for all text.

`title_font_size`
Optional Numeric [0, Inf): Title font size.

`subtitle_font_size`
Optional Numeric [0, Inf): Subtitle font size.

`axis_label_font_size`
Optional Numeric [0, Inf): Axis-label font size.

`legend_font_size`
Optional Numeric [0, Inf): Legend font size.

`tooltip_font_size`
Optional Numeric [0, Inf): Tooltip font size.

`font_family` Character: Font family string.

`color` Character: Color palette.

`bg_color` Optional Character: Chart background color.

`fg_color` Optional Character: Global text color.

`title_color` Optional Character: Title text color.

`subtitle_color` Optional Character: Subtitle text color.

`legend_color` Optional Character: Legend text color.

`axis_color` Optional Character: Axis line color.

`grid_color` Optional Character: Grid or split-line color.

`tooltip_bg` Optional Character: Tooltip background color.

`tooltip_border_color`
Optional Character: Tooltip border color.

`tooltip_color` Optional Character: Tooltip text color.

Value

[Theme](#): Theme object.

 Title

*Title***Description**

Chart title configuration with main text and optional subtext.

Usage

```
Title(
  show = NULL,
  text = NULL,
  link = NULL,
  target = NULL,
  subtext = NULL,
  sublink = NULL,
  subtarget = NULL,
  text_align = NULL,
  text_vertical_align = NULL,
  padding = NULL,
  item_gap = NULL,
  text_style = NULL,
  subtext_style = NULL,
  trigger_event = NULL,
  background_color = NULL,
  border_color = NULL,
  border_width = NULL,
  border_radius = NULL,
  left = NULL,
  right = NULL,
  top = NULL,
  bottom = NULL
)
```

Arguments

show	Optional Logical: Whether to show the title.
text	Optional Character: Main title text. Supports <code>\n</code> for newlines.
link	Optional Character: Hyperlink for the main title.
target	Optional Character {"self", "blank"}: Link target.
subtext	Optional Character: Subtitle text.
sublink	Optional Character: Hyperlink for the subtitle.
subtarget	Optional Character {"self", "blank"}: Subtitle link target.
text_align	Optional Character {"auto", "left", "center", "right"}: Horizontal alignment.

text_vertical_align	Optional Character {"auto", "top", "middle", "bottom"}: Vertical alignment.
padding	Optional Numeric: Padding inside the title box.
item_gap	Optional Numeric [0, Inf): Gap between title and subtitle in pixels.
text_style	Optional TextStyle : Main title text style.
subtext_style	Optional TextStyle : Subtitle text style.
trigger_event	Optional Logical: Whether the title emits events.
background_color	Optional Character: Title-box background color.
border_color	Optional Character: Border color.
border_width	Optional Numeric [0, Inf): Border width.
border_radius	Optional Numeric: Border radius.
left	Optional Numeric or Character: Left position.
right	Optional Numeric or Character: Right position.
top	Optional Numeric or Character: Top position.
bottom	Optional Numeric or Character: Bottom position.

Details

Corresponds to TitleOption in src/component/title/install.ts (line 48). ECharts docs: <https://echarts.apache.org/en/option.html#title>

to_json

Convert EChartsOption to JSON

Description

Convert EChartsOption to JSON

Usage

```
to_json(x, pretty = FALSE, auto_unbox = TRUE, ...)
```

Arguments

x	EChartsOption : Option object to serialize.
pretty	Logical: Whether to pretty-print the JSON.
auto_unbox	Logical: Whether to auto-unbox single-element vectors.
...	Dots: Additional arguments passed to jsonlite::toJSON() .

Value

Character: JSON string.

to_list	<i>Convert to echarts-compatible list</i>
---------	---

Description

Generic function that converts an S7 object to a plain named list matching the ECharts JSON option structure.

Usage

```
to_list(x, ...)
```

Arguments

x	S7 object: Object to serialize.
...	Dots: Unused.

Value

List: Named list.

Tooltip	<i>Tooltip</i>
---------	----------------

Description

Tooltip configuration for showing data on hover/click.

Usage

```
Tooltip(  
  show = NULL,  
  trigger = NULL,  
  trigger_on = NULL,  
  show_content = NULL,  
  always_show_content = NULL,  
  formatter = NULL,  
  value_formatter = NULL,  
  position = NULL,  
  confine = NULL,  
  enterable = NULL,  
  show_delay = NULL,  
  hide_delay = NULL,  
  transition_duration = NULL,  
  background_color = NULL,  
)
```

```

border_color = NULL,
border_width = NULL,
border_radius = NULL,
padding = NULL,
text_style = NULL,
extra_css_text = NULL,
order = NULL,
class_name = NULL
)

```

Arguments

show	Optional Logical: Whether to show the tooltip.
trigger	Optional Character {"item", "axis", "none"}: Trigger type.
trigger_on	Optional Character {"mousemove", "click", "none", "mousemove click"}: Trigger event. "mousemove click".
show_content	Optional Logical: Whether to show tooltip content.
always_show_content	Optional Logical: Whether to keep the tooltip visible.
formatter	Optional Character or function: Tooltip content formatter.
value_formatter	Optional function: Formatter for the value portion.
position	Optional Character or Numeric: Tooltip position.
confine	Optional Logical: Whether to confine the tooltip within the chart container.
enterable	Optional Logical: Whether the mouse can enter the tooltip.
show_delay	Optional Numeric [0, Inf): Delay before showing, in milliseconds.
hide_delay	Optional Numeric [0, Inf): Delay before hiding, in milliseconds.
transition_duration	Optional Numeric [0, Inf): Transition duration in seconds.
background_color	Optional Character: Background color.
border_color	Optional Character: Border color.
border_width	Optional Numeric [0, Inf): Border width.
border_radius	Optional Numeric [0, Inf): Border radius.
padding	Optional Numeric: Padding inside the tooltip.
text_style	Optional TextStyle : Tooltip text style.
extra_css_text	Optional Character: Extra CSS for HTML render mode.
order	Optional Character {"seriesAsc", "seriesDesc", "valueAsc", "valueDesc"}: Tooltip content order.
class_name	Optional Character: CSS class name for the tooltip element.

Details

Corresponds to `TooltipOption` in `src/component/tooltip/TooltipModel.ts` (line 36) and `CommonTooltipOption` in `src/util/types.ts` (line 1527). ECharts docs: <https://echarts.apache.org/en/option.html#tooltip>

 VisualMap

Visual Map

Description

Continuous visual map component that maps data values to visual properties (typically color). Required for heatmaps.

Usage

```
VisualMap(
  type = "continuous",
  min = NULL,
  max = NULL,
  precision = NULL,
  calculable = NULL,
  show = NULL,
  orient = NULL,
  in_range = NULL,
  out_of_range = NULL,
  left = NULL,
  right = NULL,
  top = NULL,
  bottom = NULL
)
```

Arguments

<code>type</code>	Character {"continuous", "piecewise"}: Visual map type.
<code>min</code>	Optional Numeric: Minimum value of the mapped data range.
<code>max</code>	Optional Numeric: Maximum value of the mapped data range.
<code>precision</code>	Optional Integer [0, Inf): Decimal places shown in colorbar labels and the draggable-handle tooltip. ECharts defaults to 0, which rounds labels to whole numbers — set this to match <code>value_digits</code> for accurate display.
<code>calculable</code>	Optional Logical: Whether to show drag handles for interactive range selection.
<code>show</code>	Optional Logical: Whether to display the visual map legend.
<code>orient</code>	Optional Character {"vertical", "horizontal"}: Orientation of the color bar.
<code>in_range</code>	Optional list: Visual encoding for in-range values. Typically <code>list(color = c("blue", "white", "red"))</code> .

out_of_range	Optional list: Visual encoding for out-of-range values.
left	Optional Numeric or Character: Distance from the left of the container.
right	Optional Numeric or Character: Distance from the right of the container.
top	Optional Numeric or Character: Distance from the top of the container.
bottom	Optional Numeric or Character: Distance from the bottom of the container.

Details

Corresponds to `ContinuousVisualMapOption` in `src/component/visualMap/ContinuousModel.ts`.
ECharts docs: <https://echarts.apache.org/en/option.html#visualMap-continuous>

Index

AreaStyle, [4](#), [44](#), [57](#)
Axis, [5](#), [35](#)
AxisLabel, [6](#), [7](#)
AxisLine, [6](#), [8](#)
AxisTick, [6](#), [9](#)

BarSeries, [9](#), [35](#)
BoxplotSeries, [11](#), [35](#)

DataZoom, [12](#), [27](#), [35](#)
draw, [15](#)
draw(), [33](#), [55](#)
draw_a3, [16](#)
draw_bar, [18](#)
draw_boxplot, [19](#)
draw_density, [21](#)
draw_heatmap, [22](#)
draw_heatmap(), [15](#)
draw_histogram, [25](#)
draw_line, [26](#)
draw_line(), [19–21](#), [24](#), [25](#), [31](#)
draw_pie, [27](#)
draw_sankey, [28](#)
draw_scatter, [30](#)
draw_spectrogram, [31](#)
drawOutput, [34](#)

EChartsOption, [15](#), [34](#), [66](#)

graphics::hist(), [25](#)
grDevices::boxplot.stats(), [19](#)
Grid, [17](#), [35](#), [36](#)

hclust(), [24](#)
HeatmapSeries, [35](#), [37](#)

ItemStyle, [11](#), [12](#), [38](#), [38](#), [42](#), [44](#), [46](#), [47](#), [50](#),
[52](#), [53](#), [55](#), [57](#)

jsonlite::toJSON(), [66](#)

LabelLine, [39](#), [50](#)
LabelOption, [11](#), [12](#), [38](#), [40](#), [44](#), [46](#), [47](#),
[50–53](#), [55](#), [57](#)
Legend, [35](#), [41](#)
LineSeries, [35](#), [43](#)
LineStyle, [8](#), [9](#), [40](#), [44](#), [45](#), [47](#), [48](#), [51](#), [52](#), [55](#),
[58](#)

MarkArea, [44](#), [46](#)
MarkAreaDataPoint, [46](#), [46](#)
mgcv::gam(), [30](#)
MinorSplitLine, [6](#), [47](#)
MinorTick, [6](#), [48](#)

PieSeries, [35](#), [48](#)

renderDraw, [50](#)
rtemis.a3::create_A3(), [16](#)
rtemis.a3::read_A3json(), [16](#)
rtemis.draw (rtemis.draw-package), [3](#)
rtemis.draw-package, [3](#)
rtemis_colors, [51](#)

SankeyEdgeItem, [51](#), [52](#), [54](#)
SankeyLevelOption, [52](#), [54](#)
SankeyNodeItem, [52](#), [54](#)
SankeySeries, [53](#)
save_drawing, [55](#)
save_drawing(), [15](#), [17](#), [19](#), [20](#), [22](#), [24](#), [25](#),
[27–29](#), [31](#), [33](#)
ScatterSeries, [35](#), [56](#)
signal::specgram(), [31](#), [32](#)
SplitArea, [6](#), [57](#)
SplitLine, [6](#), [58](#)
stats::density(), [21](#)
stats::dist(), [23](#)
stats::hclust(), [23](#)

TextStyle, [6](#), [7](#), [35](#), [41](#), [42](#), [58](#), [61](#), [66](#), [68](#)
Theme, [15](#), [17](#), [19–21](#), [24](#), [25](#), [27–29](#), [31](#), [33](#),
[61](#), [63](#), [64](#)

theme_dark, [62](#)
theme_dark(), [15](#)
theme_light, [63](#)
theme_light(), [15](#)
Title, [35](#), [65](#)
to_json, [66](#)
to_list, [67](#)
Tooltip, [35](#), [67](#)

VisualMap, [24](#), [35](#), [37](#), [69](#)