Package 'echarts4r'

September 17, 2018

Title Create Interactive Graphs with 'Echarts JavaScript' Version 4

Date 2018-09-12

Version 0.1.1

Description
Easily create interactive charts by leveraging the 'Echarts Javascript' library which includes 33 chart types, themes, 'Shiny' proxies and animations.

License Apache License (>= 2.0)

Encoding UTF-8

LazyData true

Imports htmlwidgets, magrittr, dplyr, purrr, data.tree, crosstalk, base64enc, countrycode, d3r, broom

RoxygenNote 6.1.0

Suggests jsonlite, shiny, quantmod, png, tibble, htmltools

URL http://echarts4r.john-coene.com/

BugReports https://github.com/JohnCoene/echarts4r/issues

NeedsCompilation no

Author John Coene [aut, cre]

Maintainer John Coene <jcoenep@gmail.com>

Repository CRAN

Date/Publication 2018-09-17 16:40:03 UTC

R topics documented:

`echarts4r-shiny` *Shiny bindings for echarts4r*

Description

Output and render functions for using echarts4r within Shiny applications and interactive Rmd documents.

Usage

```
echarts4rOutput(outputId, width = "100%", height = "400px")

renderEcharts4r(expr, env = parent.frame(), quoted = FALSE)

echarts4rProxy(id, session = shiny::getDefaultReactiveDomain())
```

Arguments

`outputId`	output variable to read from.
`width, height`	Must be a valid CSS unit (like `'100%'`, `'400px'`, `'auto'`) or a number, which will be coerced to a string and have `'px'` appended.
`expr`	An expression that generates a echarts4r
`env`	The environment in which to evaluate `expr`.
`quoted`	Is `expr` a quoted expression (with `quote()`)? This is useful if you want to save an expression in a variable.

id	Target chart id.
session	Shiny session.

`e_angle_axis` *Angle axis*

Description

Customise angle axis.

Usage

```
e_angle_axis(e, show = TRUE, ...)
```

Arguments

e	An `echarts4r` object as returned by `e_charts`.
show	Whether to display the axis.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#angleAxis

Examples

```
df <- data.frame(x = 1:100, y = seq(1, 200, by = 2))

df %>%
  e_charts(x) %>%
  e_polar(FALSE) %>%
  e_angle_axis(FALSE) %>%
  e_radius_axis(FALSE) %>%
  e_line(y, coord.system = "polar", smooth = TRUE) %>%
  e_legend(show = FALSE)
```

e_animation *Animation*

Description

Customise animations.

Usage

```
e_animation(e, show = TRUE, threshold = NULL, duration = NULL,
  easing = NULL, delay = NULL, duration.update = NULL,
  easing.update = NULL, delay.update = NULL)
```

Arguments

e	An echarts4r object as returned by e_charts.
show	Set to show animation.
threshold	Whether to set graphic number threshold to animation. Animation will be disabled when graphic number is larger than threshold.
duration	Duration of the first animation.
easing	Easing method used for the first animation.
delay	Delay before updating the first animation.
duration.update	Time for animation to complete.
easing.update	Easing method used for animation.
delay.update	Delay before updating animation.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#animation

Examples

```
mtcars %>%
  e_charts(mpg) %>%
  e_area(drat) %>%
  e_animation(duration = 10000)
```

e_append1_p *Append data*

Description

Append data.

Usage

```
e_append1_p(proxy, series.index = NULL, data, x, y)

e_append1_p_(proxy, series.index = NULL, data, x, y)

e_append2_p(proxy, series.index = NULL, data, x, y, z)

e_append2_p_(proxy, series.index = NULL, data, x, y, z)
```

Arguments

proxy	An echarts4r proxy as returned by echarts4rProxy.
series.index	Index of serie to append to (starts from 0).
data	Data.frame containing data to append.
x, y, z	Columns names to plot.

Details

Currently not all types of series supported incremental rendering when using appendData. Only these types of series support it: e_scatter and e_line of pure echarts, and e_scatter_3d, and e_line_3d of echarts-gl.

Examples

```
## Not run:
  library(shiny)

  ui <- fluidPage(
    actionButton("add", "Add Data to y"),
    echarts4rOutput("plot"),
    h4("Brush"),
    verbatimTextOutput("selected"),
    h4("Legend select change"),
    verbatimTextOutput("legend")
  )

  server <- function(input, output, session){

    data <- data.frame(x = rnorm(10, 5, 3), y = rnorm(10, 50, 12), z = rnorm(10, 50, 5))
```

```
  react <- eventReactive(input$add, {
    set.seed(sample(1:1000, 1))
    data.frame(x = rnorm(10, 5, 2), y = rnorm(10, 50, 10), z = rnorm(10, 1, 5))
  })

  output$plot <- renderEcharts4r({
    data %>%
     e_charts(x) %>%
     e_scatter(y, z) %>%
     e_scatter(z) %>%
     e_brush()
  })

  observeEvent(input$add, {
    echarts4rProxy("plot") %>%
      e_append2_p(0, react(), x, y, z)
  })

  output$selected <- renderPrint({
    input$plot_brush
  })

  output$legend <- renderPrint({
    input$plot_legend_change
  })

}

shinyApp(ui, server)


## End(Not run)
```

---

e_area *Area*

Description

Add area serie.

Usage

```
e_area(e, serie, bind, name = NULL, legend = TRUE, y.index = 0,
  x.index = 0, ...)

e_area_(e, serie, bind = NULL, name = NULL, legend = TRUE,
  y.index = 0, x.index = 0, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Column name of serie to plot.
bind	Binding between datasets, namely for use of e_brush.
name	name of the serie.
legend	Whether to add serie to legend.
y.index	Indexes of x and y axis.
x.index	Indexes of x and y axis.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-line

Examples

```
CO2 %>%
  group_by(Plant) %>%
  e_charts(conc) %>%
  e_area(uptake)
```

---

e_axis *Axis*

Description

Customise axis.

Usage

```
e_axis(e, axis = c("x", "y", "z"), index = 0, ...)

e_x_axis(e, index = 0, ...)

e_y_axis(e, index = 0, ...)

e_z_axis(e, index = 0, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
axis	Axis to customise.
index	Index of axis to customise.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#xAxis, https://ecomfe.github.io/echarts-doc/public/en/option.html#yAxis

Examples

```
USArrests %>%
  e_charts(Assault) %>%
  e_line(Murder, smooth = TRUE) %>%
  e_line(Rape, y.index = 1) %>% # add secondary axis
  e_y_axis(index = 1, show = FALSE) # hide secondary axis

# plot all labels & rotate
USArrests %>%
  head(10) %>%
  dplyr::mutate(State = row.names(.)) %>%
  e_charts(State) %>%
  e_area(Murder) %>%
  e_x_axis(axisLabel = list(interval = 0, rotate = 45)) # rotate
```

---

e_axis_3d *Axis 3D*

Description

Customise 3D axis.

Usage

```
e_axis_3d(e, axis = c("x", "y", "z"), index = 0, ...)

e_x_axis_3d(e, index = 0, ...)

e_y_axis_3d(e, index = 0, ...)

e_z_axis_3d(e, index = 0, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
axis	Axis to customise.
index	Index of axis to customise.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#xAxis3D, https://ecomfe.github.io/echarts-doc/public/en/option.html#yAxis3D, https://ecomfe.github.io/echarts-doc/public/en/option.html#zAxis3D

e_axis_pointer *Axis pointer*

Description

Customise axis pointer.

Usage

```
e_axis_pointer(e, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#axisPointer

e_bar *Bar and Line chart*

Description

Add bar serie.

Usage

```
e_bar(e, serie, bind, name = NULL, legend = TRUE, y.index = 0,
  x.index = 0, ...)

e_bar_(e, serie, bind = NULL, name = NULL, legend = TRUE,
  y.index = 0, x.index = 0, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Column name of serie to plot.
bind	Binding between datasets, namely for use of e_brush.
name	name of the serie.
legend	Whether to add serie to legend.
x.index, y.index	Indexes of x and y axis.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-bar

Examples

```
iris %>%
  group_by(Species) %>%
  e_charts(Sepal.Length) %>%
  e_line(Sepal.Width)
```

`e_bar_3d` *Bar 3D*

Description

Add 3D bars

Usage

```
e_bar_3d(e, y, z, bind, coord.system = "cartesian3D", name = NULL,
  rm.x = TRUE, rm.y = TRUE, ...)

e_bar_3d_(e, y, z, bind = NULL, coord.system = "cartesian3D",
  name = NULL, rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`y, z`	Coordinates.
`bind`	Binding.
`coord.system`	Coordinate system to use, one of `cartesian3D`, `geo3D`, `globe`.
`name`	name of the serie.
`rm.x, rm.y`	Whether to remove x and y axis, defaults to `TRUE`.
`...`	Any other option to pass, check See Also section.

See Also

http://echarts.baidu.com/option-gl.html#series-bar3D

Examples

```
## Not run:
url <- paste0("https://ecomfe.github.io/echarts-examples/",
              "public/data-gl/asset/data/population.json")
data <- jsonlite::fromJSON(url)
data <- as.data.frame(data)
names(data) <- c("lon", "lat", "value")

data %>%
  e_charts(lon) %>%
  e_globe(
    environment = e_stars_texture(),
    base.texture = e_globe_texture()
  ) %>%
  e_bar_3d(lat, value, coord.system = "globe") %>%
  e_visual_map()

data %>%
  e_charts(lon) %>%
  e_geo_3d() %>%
  e_bar_3d(lat, value, coord.system = "geo3D") %>%
  e_visual_map()

v <- LETTERS[1:10]
matrix <- data.frame(
  x = sample(v, 300, replace = TRUE),
  y = sample(v, 300, replace = TRUE),
  z1 = rnorm(300, 10, 1),
  z2 = rnorm(300, 10, 1),
  stringsAsFactors = FALSE
) %>%
  dplyr::group_by(x, y) %>%
  dplyr::summarise(
    z1 = sum(z1),
    z2 = sum(z2)
  ) %>%
  dplyr::ungroup()

trans <- list(opacity = 0.4) # transparency
emphasis <- list(itemStyle = list(color = "#313695"))

matrix %>%
  e_charts(x) %>%
  e_bar_3d(y, z1, stack = "stack", name = "Serie 1", itemStyle = trans, emphasis = emphasis) %>%
  e_bar_3d(y, z2, stack = "stack", name = "Serie 2", itemStyle = trans, emphasis = emphasis) %>%
  e_legend()

## End(Not run)
```

e_boxplot *Boxplot*

Description

Draw boxplot.

Usage

```
e_boxplot(e, serie, name = NULL, outliers = TRUE, ...)

e_boxplot_(e, serie, name = NULL, outliers = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Column name of serie to plot.
name	name of the serie.
outliers	Whether to plot outliers.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-boxplot

Examples

```
df <- data.frame(
  x = c(1:10, 25),
  y = c(1:10, -6)
)

df %>%
  e_charts() %>%
  e_boxplot(y, outliers = TRUE) %>%
  e_boxplot(x, outliers = TRUE)
```

e_brush *Brush*

Description

Add a brush.

Usage

```
e_brush(e, x.index = NULL, y.index = NULL, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
x.index	Indexes of x and y axis.
y.index	Indexes of x and y axis.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#brush

Examples

```
quakes %>%
  e_charts(long) %>%
  e_geo(
    boundingCoords = list(
      c(190, -10),
      c(180, -40)
    )
  ) %>%
  e_scatter(lat, mag, stations, coord.system = "geo", scale = "* 1.5", name = "mag") %>%
  e_data(quakes, depth) %>%
  e_scatter(mag, mag, stations, scale = "* 3", name = "mag & depth") %>%
  e_grid(right = 40, top = 100, width = "30%") %>%
  e_y_axis(type = "value", name = "depth", min = 3.5) %>%
  e_brush() %>%
  e_theme("dark")
```

e_calendar *Calendar*

Description

Calendar

Usage

```
e_calendar(e, range, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
range	Range of calendar format, string or vector.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#calendar

Examples

```
# year
mtcars %>%
  e_charts() %>%
  e_calendar(range = "2017")

# month
mtcars %>%
  e_charts() %>%
  e_calendar(range = "2018-01")

# range
mtcars %>%
  e_charts() %>%
  e_calendar(range = c("2018-01", "2018-07"))
```

e_candle *Candlestick*

Description

Add a candlestick chart.

Usage

```
e_candle(e, opening, closing, low, high, bind, name = NULL,
  legend = TRUE, ...)

e_candle_(e, opening, closing, low, high, bind = NULL, name = NULL,
  legend = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
opening, closing, low, high	Stock prices.
bind	Binding between datasets, namely for use of e_brush.
name	name of the serie.
legend	Whether to add serie to legend.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-candlestick

Examples

```
date <- c("2017-01-01", "2017-01-02", "2017-01-03", "2017-01-04", "2017-03-05",
         "2017-01-06", "2017-01-07")

stock <- data.frame(
  date = date,
  opening = c(200.60, 200.22, 198.43, 199.05, 203.54, 203.40, 208.34),
  closing = c(200.72, 198.85, 199.05, 203.73, 204.08, 208.11, 211.88),
  low = c(197.82, 198.07, 197.90, 198.10, 202.00, 201.50, 207.60),
  high = c(203.32, 200.67, 200.00, 203.95, 204.90, 208.44, 213.17)
)

stock %>%
  e_charts(date) %>%
  e_candle(opening, closing, low, high) %>%
  e_y_axis(min = 190, max = 220)
```

e_charts *Initialise*

Description

Initialise a chart.

Usage

```
e_charts(data, x, width = NULL, height = NULL, elementId = NULL,
  dispose = TRUE, renderer = "canvas", ...)

e_charts_(data, x = NULL, width = NULL, height = NULL,
  elementId = NULL, dispose = TRUE, renderer = "canvas", ...)

e_chart(data, x, width = NULL, height = NULL, elementId = NULL,
  dispose = TRUE, renderer = "canvas", ...)

e_data(e, data, x)
```

Arguments

data	A data.frame.
x	Column name containing x axis.
width, height	Must be a valid CSS unit (like '100%', '400px', 'auto') or a number, which will be coerced to a string and have 'px' appended.
elementId	Id of element.
dispose	Set to TRUE to force redraw of chart, set to FALSE to update.
renderer	Renderer, takes canvas (default) or svg.
...	Any other argument.
e	An object of class echarts4r as returned by e_charts.

Examples

```
mtcars %>%
  e_charts_("qsec") %>%
  e_line(mpg)
```

e_clean *Clean*

Description

Removes base `data.frame`.

Usage

```
e_clean(e)
```

Arguments

e	An `echarts4r` object as returned by `e_charts`.

Details

Removes the core database after all operations are exectuted, lightens up the load on final visualisation.

Examples

```
df <- data.frame(
  x = 1:10,
  y = round(
    runif(10, 1, 100), 2
  )
)

df %>%
  e_charts(x) %>%
  e_line(y) %>%
  e_format_y_axis(suffix = "%") %>%
  e_format_x_axis(prefix = "A") %>%
  e_clean()
```

e_cloud *Wordcloud*

Description

Draw a wordcloud.

Usage

```
e_cloud(e, word, freq, color, rm.x = TRUE, rm.y = TRUE, ...)

e_cloud_(e, word, freq, color = NULL, rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
word, freq	Terms and their frequencies.
color	Word color.
rm.x, rm.y	Whether to remove x and y axis, defaults to TRUE.
...	Any other option to pass, check See Also section.

See Also

https://github.com/ecomfe/echarts-wordcloud

Examples

```
words <- function(n = 5000) {
  a <- do.call(paste0, replicate(5, sample(LETTERS, n, TRUE), FALSE))
  paste0(a, sprintf("%04d", sample(9999, n, TRUE)), sample(LETTERS, n, TRUE))
}

tf <- data.frame(terms = words(100),
  freq = rnorm(100, 55, 10)) %>%
  dplyr::arrange(-freq)

tf %>%
  e_color_range(freq, color) %>%
  e_charts() %>%
  e_cloud(terms, freq, color, shape = "circle", sizeRange = c(3, 15))
```

---

e_color *Color*

Description

Customise chart and background colors.

Usage

```
e_color(e, color = NULL, background = NULL)
```

Arguments

e	An echarts4r object as returned by e_charts.
color	Vector of colors.
background	Background color.

See Also

e_theme, https://ecomfe.github.io/echarts-doc/public/en/option.html#color, https://ecomfe.github.io/echarts-doc/public/en/option.html#backgroundColor

Examples

```
mtcars %>%
  e_charts(drat) %>%
  e_line(mpg) %>%
  e_area(qsec) %>%
  e_color(
    c("red", "blue"),
    "#d3d3d3"
  )
```

---

e_color_range *Color range*

Description

Build manual color range

Usage

```
e_color_range(data, input, output, colors = c("#bf444c", "#d88273",
  "#f6efa6"), ...)

e_color_range_(data, input, output, colors = c("#bf444c", "#d88273",
  "#f6efa6"), ...)
```

Arguments

data	Data.frame in which to find column names.
input, output	Input and output columns.
colors	Colors to pass to colorRampPalette.
...	Any other argument to pass to colorRampPalette.

Examples

```
df <- data.frame(val = 1:10)

e_color_range(df, val, colors)
```

`e_country_names` *Country names*

Description

Convert country names to echarts format.

Usage

```
e_country_names(data, input, output, type = "iso2c", ...)

e_country_names_(data, input, output = NULL, type = "iso2c", ...)
```

Arguments

`data`	Data.frame in which to find column names.
`input, output`	Input and output columns.
`type`	Passed to countrycode `origin` parameter.
`...`	Any other parameter to pass to countrycode.

Details

Taiwan and Hong Kong cannot be plotted.

Examples

```
cns <- data.frame(country = c("US", "BE"))

# replace
e_country_names(cns, country)

# specify output
e_country_names(cns, country, country.name)
```

e_datazoom *Data zoom*

Description

Add data zoom.

Usage

```
e_datazoom(e, x.index = NULL, y.index = NULL, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
x.index	Indexes of x and y axis.
y.index	Indexes of x and y axis.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#dataZoom

Examples

```
USArrests %>%
  e_charts(UrbanPop) %>%
  e_line(Assault) %>%
  e_area(Murder, y.index = 1, x.index = 1) %>%
  e_y_axis(gridIndex = 1) %>%
  e_x_axis(gridIndex = 1) %>%
  e_grid(height = "35%") %>%
  e_grid(height = "35%", top = "50%") %>%
  e_toolbox_feature("dataZoom", title = list(zoom = "zoom", back = "back")) %>%
  e_datazoom(x.index = c(0, 1))
```

e_flip_coords *Flip coordinates*

Description

Flip cartesian 2D coordinates.

Usage

```
e_flip_coords(e)
```

Arguments

e	An echarts4r object as returned by e_charts.

Examples

```
df <- data.frame(
  x = LETTERS[1:5],
  y = runif(5, 1, 5),
  z = runif(5, 3, 10)
)

df %>%
 e_charts(x) %>%
 e_bar(y) %>%
 e_line(z) -> plot

plot # normal
e_flip_coords(plot) # flip
```

---

e_flow_gl *Flow GL*

Description

Flow GL

Usage

```
e_flow_gl(e, y, sx, sy, color, name = NULL, coord.system = NULL,
  rm.x = TRUE, rm.y = TRUE, ...)

e_flow_gl_(e, y, sx, sy, color = NULL, name = NULL,
  coord.system = NULL, rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
y	Vector position on the y axis.
sx, sy	Velocity in respective axis.
color	Vector color.
name	name of the serie.
coord.system	Coordinate system to use.
rm.x, rm.y	Whether to remove x and y axis, only applies if coord.system is not null.
...	Any other option to pass, check See Also section.

See Also

http://echarts.baidu.com/option-gl.html#series-flowGL

Examples

```
# coordinates
vectors <- expand.grid(0:9, 0:9)
names(vectors) <- c("x", "y")
vectors$sx <- rnorm(100)
vectors$sy <- rnorm(100)
vectors$color <- log10(runif(100, 1, 10))

vectors %>%
  e_charts(x) %>%
  e_flow_gl(y, sx, sy, color) %>%
  e_visual_map(
    min = 0, max = 1, # log 10
    dimension = 4, # x = 0, y = 1, sx = 3, sy = 4
    show = FALSE, # hide
    inRange = list(
      color = c('#313695', '#4575b4', '#74add1', '#abd9e9', '#e0f3f8',
                '#ffffbf', '#fee090', '#fdae61', '#f46d43', '#d73027', '#a50026')
    )
  ) %>%
  e_x_axis(
    splitLine = list(show = FALSE)
  ) %>%
  e_y_axis(
    splitLine = list(show = FALSE)
  )

# map
latlong <- seq(-180, 180, by = 5)
wind = expand.grid(lng = latlong, lat = latlong)
wind$slng <- rnorm(nrow(wind), 0, 200)
wind$slat <- rnorm(nrow(wind), 0, 200)
wind$color <- abs(wind$slat) - abs(wind$slng)
rng <- range(wind$color)

trans <- list(opacity = 0.5) # transparency

wind %>%
  e_charts(lng, backgroundColor = '#333') %>%
  e_geo(
    itemStyle = list(
      normal = list(
        areaColor = "#323c48",
        borderColor = "#111"
      )
    )
  ) %>%
  e_flow_gl(lat, slng, slat, color,
```

```
  coord.system = "geo",
  itemStyle = trans,
  particleSize = 2
) %>%
e_visual_map(
  min = rng[1], max = rng[2], # range
  dimension = 4, # lng = 0, lat = 1, slng = 2, slat = 3, color = 4
  show = FALSE, # hide
  inRange = list(
    color = c('#313695', '#4575b4', '#74add1', '#abd9e9', '#e0f3f8',
              '#ffffbf', '#fee090', '#fdae61', '#f46d43', '#d73027', '#a50026')
  )
)
```

`e_format_axis` *Formatters*

Description

Simple formatters as helpers.

Usage

```
e_format_axis(e, axis = "y", suffix = NULL, prefix = NULL, ...)

e_format_x_axis(e, suffix = NULL, prefix = NULL, ...)

e_format_y_axis(e, suffix = NULL, prefix = NULL, ...)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`axis`	Axis to apply formatter to.
`suffix, prefix`	Suffix and prefix of label.
`...`	Any other arguments to pass to `e_axis`.

Examples

```
# Y = %
df <- data.frame(
  x = 1:10,
  y = round(
    runif(10, 1, 100), 2
  )
)

df %>%
```

```
  e_charts(x) %>%
  e_line(y) %>%
  e_format_y_axis(suffix = "%") %>%
  e_format_x_axis(prefix = "A")
```

`e_funnel` *Funnel*

Description

Add a funnel.

Usage

```
e_funnel(e, values, labels, name = NULL, legend = TRUE, rm.x = TRUE,
  rm.y = TRUE, ...)

e_funnel_(e, values, labels, name = NULL, legend = TRUE, rm.x = TRUE,
  rm.y = TRUE, ...)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`values, labels`	Values and labels of funnel.
`name`	name of the serie.
`legend`	Whether to add serie to legend.
`rm.x, rm.y`	Whether to remove x and y axis, defaults to `TRUE`.
`...`	Any other option to pass to `bar` or `line` char types.

Details

No `bind` argument here, with a funnel `bind = labels`.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-funnel

Examples

```
funnel <- data.frame(stage = c("View", "Click", "Purchase"), value = c(80, 30, 20))

funnel %>%
  e_charts() %>%
  e_funnel(value, stage)
```

e_gauge *Gauge*

Description

Plot a gauge.

Usage

```
e_gauge(e, value, name, rm.x = TRUE, rm.y = TRUE, ...)

e_gauge_(e, value, name, rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
value	Value to gauge.
name	Text on gauge.
rm.x, rm.y	Whether to remove x and y axis, defaults to TRUE.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-gauge

Examples

```
e_charts() %>%
  e_gauge(57, "PERCENT")
```

e_geo *Geo*

Description

Initialise geo.

Usage

```
e_geo(e, map = "world", ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
map	Map type.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#geo

Examples

```
flights <- read.csv(
  paste0("https://raw.githubusercontent.com/plotly/datasets/",
         "master/2011_february_aa_flight_paths.csv")
)

flights %>%
  e_charts() %>%
  e_geo() %>%
  e_lines(
    start_lon,
    start_lat,
    end_lon,
    end_lat,
    name = "flights",
    lineStyle = list(normal = list(curveness = 0.3))
   )
```

---

e_geo_3d *Geo 3D*

Description

Initialise geo 3D.

Usage

```
e_geo_3d(e, serie, color, type = "world", rm.x = TRUE, rm.y = TRUE,
  ...)

e_geo_3d_(e, serie = NULL, color = NULL, type = "world",
  rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Column name of serie to plot.
color	Color.
type	Map type.
rm.x, rm.y	Whether to remove x and y axis, defaults to TRUE.
...	Any other option to pass, check See Also section.

See Also

e_country_names, http://echarts.baidu.com/option-gl.html#geo3D

Examples

```
choropleth <- data.frame(
  countries = c("France", "Brazil", "China", "Russia", "Canada", "India", "United States",
                "Argentina", "Australia"),
  height = runif(9, 1, 5),
  color = c("#F7FBFF", "#DEEBF7", "#C6DBEF", "#9ECAE1", "#6BAED6", "#4292C6",
            "#2171B5", "#08519C", "#08306B")
)

choropleth %>%
  e_charts(countries) %>%
  e_geo_3d(height, color)
```

---

e_get_data *Get data*

Description

Get data passed to e_charts.

Usage

```
e_get_data(e)
```

Arguments

e	An echarts4r object as returned by e_charts.

e_globe *Globe*

Description

Add globe.

Usage

```
e_globe(e, environment = NULL, base.texture = NULL,
  height.texture = NULL, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
environment	Texture of background.
base.texture	Base texture of globe.
height.texture	Texture of height.
...	Any other option to pass, check See Also section.

See Also

e_country_names, http://echarts.baidu.com/option-gl.html#globe

Examples

```
## Not run:
url <- paste0("https://ecomfe.github.io/echarts-examples/",
              "public/data-gl/asset/data/population.json")
data <- jsonlite::fromJSON(url)
data <- as.data.frame(data)
names(data) <- c("lon", "lat", "value")

data %>%
  e_charts(lon) %>%
  e_globe(
    environment = e_stars_texture(),
    base.texture = e_globe_texture(),
    height.texture = e_globe_texture(),
    displacementScale = 0.04
  ) %>%
  e_bar_3d(lat, value, "globe") %>%
  e_visual_map(show = FALSE)

## End(Not run)
```

e_graph *Graph*

Description

Create a graph.

Usage

```
e_graph(e, layout = "force", name = NULL, rm.x = TRUE, rm.y = TRUE,
  ...)

e_graph_gl(e, layout = "force", name = NULL, rm.x = TRUE,
  rm.y = TRUE, ...)

e_graph_nodes(e, nodes, names, value, size, category, legend = TRUE)

e_graph_edges(e, edges, source, target)
```

Arguments

e	An echarts4 object as returned by e_charts.
layout	Layout, one of force, none or circular.
name	Name of graph.
rm.x, rm.y	Whether to remove the x and y axis, defaults to TRUE.
...	Any other parameter.
nodes	Data.frame of nodes.
names	Names of nodes, unique.
value	values of nodes.
size	Size of nodes.
category	Group of nodes (i.e.: group membership).
legend	Whether to add serie to legend.
edges	Data.frame of edges.
source, target	Column names of source and target.

See Also

Additional arguments, e_modularity

Examples

```
value <- rnorm(10, 10, 2)

nodes <- data.frame(
  name = sample(LETTERS, 10),
  value = value,
  size = value,
  grp = rep(c("grp1", "grp2"), 5),
  stringsAsFactors = FALSE
)

edges <- data.frame(
  source = sample(nodes$name, 20, replace = TRUE),
  target = sample(nodes$name, 20, replace = TRUE),
  stringsAsFactors = FALSE
)

e_charts() %>%
  e_graph() %>%
  e_graph_nodes(nodes, name, value, size, grp) %>%
  e_graph_edges(edges, source, target)

#Use graphGL for larger networks
nodes <- data.frame(
  name = paste0(LETTERS, 1:1000),
  value = rnorm(1000, 10, 2),
  size = rnorm(1000, 10, 2),
  grp = rep(c("grp1", "grp2"), 500),
  stringsAsFactors = FALSE
)

edges <- data.frame(
  source = sample(nodes$name, 2000, replace = TRUE),
  target = sample(nodes$name, 2000, replace = TRUE),
  stringsAsFactors = FALSE
)

e_charts() %>%
  e_graph_gl() %>%
  e_graph_nodes(nodes, name, value, size, grp) %>%
  e_graph_edges(edges, source, target)
```

---

e_grid *Grid*

Description

Customise grid.

Usage

```
e_grid(e, index = NULL, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
index	Index of axis to customise.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#grid

Examples

```
USArrests %>%
  e_charts(UrbanPop) %>%
  e_line(Assault, smooth = TRUE) %>%
  e_area(Murder, y.index = 1, x.index = 1) %>%
  e_y_axis(gridIndex = 1) %>%
  e_x_axis(gridIndex = 1) %>%
  e_grid(height = "40%") %>%
  e_grid(height = "40%", top = "55%")
```

---

e_grid_3d *Grid*

Description

Customise grid.

Usage

```
e_grid_3d(e, index = 0, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
index	Index of axis to customise.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#grid3D

Examples

```
USArrests %>%
  e_charts(UrbanPop) %>%
  e_line(Assault, smooth = TRUE) %>%
  e_area(Murder, y.index = 1, x.index = 1) %>%
  e_y_axis(gridIndex = 1) %>%
  e_x_axis(gridIndex = 1) %>%
  e_grid(height = "40%") %>%
  e_grid(height = "40%", top = "55%")
```

---

e_heatmap *Heatmap*

Description

Draw heatmap by coordinates.

Usage

```
e_heatmap(e, y, z, name = NULL, coord.system = "cartesian2d",
  rm.x = TRUE, rm.y = TRUE, ...)

e_heatmap_(e, y, z = NULL, name = NULL, coord.system = "cartesian2d",
  rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
y, z	Coordinates and values.
name	name of the serie.
coord.system	Coordinate system to plot against, takes cartesian2d, geo or calendar.
rm.x, rm.y	Whether to remove x and y axis, only applies if coord.system is not set to cartesian2d.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-heatmap

Examples

```
v <- LETTERS[1:10]
matrix <- data.frame(
  x = sample(v, 300, replace = TRUE),
  y = sample(v, 300, replace = TRUE),
  z = rnorm(300, 10, 1),
  stringsAsFactors = FALSE
) %>%
  dplyr::group_by(x, y) %>%
  dplyr::summarise(z = sum(z)) %>%
  dplyr::ungroup()

matrix %>%
  e_charts(x) %>%
  e_heatmap(y, z) %>%
  e_visual_map(z)

# calendar
dates <- seq.Date(as.Date("2018-01-01"), as.Date("2018-12-31"), by = "day")
values <- rnorm(length(dates), 20, 6)

year <- data.frame(date = dates, values = values)

year %>%
  e_charts(date) %>%
  e_calendar(range = "2018") %>%
  e_heatmap(values, coord.system = "calendar") %>%
  e_visual_map(max = 30)
```

---

e_highlight_p *Highlight Proxy*

Description

Proxies to highlight and downplay series.

Usage

```
e_highlight_p(proxy, series.index = NULL, series.name = NULL)

e_downplay_p(proxy, series.index = NULL, series.name = NULL)
```

Arguments

proxy	An echarts4r proxy as returned by echarts4rProxy.
series.index	Series index, can be a vector.
series.name	Series Name, can be vector.

Examples

```
## Not run:

 ui <- fluidPage(
   fluidRow(
     column(
       3,
       actionButton("highlightmpg", "Highlight MPG")
     ),
     column(
       3,
       actionButton("highlighthp", "Highlight HP")
     ),
     column(
       3,
       actionButton("downplaympg", "Downplay MPG")
     ),
     column(
       3,
       actionButton("downplayhp", "Downplay HP")
     )
   ),
   echarts4rOutput("plot")
 )

 server <- function(input, output, session){
   output$plot <- renderEcharts4r({
     mtcars %>%
       e_charts(mpg) %>%
       e_line(disp) %>%
       e_line(hp, name = "HP") # explicitly pass name
   })

   # highlight

   observeEvent(input$highlightmpg, {
     echarts4rProxy("plot") %>%
       e_highlight_p(series.index = 0) # using index
   })

   observeEvent(input$highlighthp, {
     echarts4rProxy("plot") %>%
       e_highlight_p(series.name = "HP") # using name
   })

   # downplay

   observeEvent(input$downplaympg, {
     echarts4rProxy("plot") %>%
       e_downplay_p(series.name = "disp")
   })
```

```
  observeEvent(input$downplayhp, {
    echarts4rProxy("plot") %>%
      e_downplay_p(series.index = 1)
  })
}

shinyApp(ui, server)


## End(Not run)
```

---

e_histogram *Histogram & Density*

Description

Add a histogram or density plots.

Usage

```
e_histogram(e, serie, breaks = "Sturges", name = NULL, legend = TRUE,
  bar.width = "99%", x.index = 0, y.index = 0, ...)

e_density(e, serie, breaks = "Sturges", name = NULL, legend = TRUE,
  x.index = 0, y.index = 0, ...)

e_histogram_(e, serie, breaks = "Sturges", name = NULL,
  legend = TRUE, bar.width = "99%", x.index = 0, y.index = 0, ...)

e_density_(e, serie, breaks = "Sturges", name = NULL, legend = TRUE,
  x.index = 0, y.index = 0, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Column name of serie to plot.
breaks	Passed to hist.
name	name of the serie.
legend	Whether to add serie to legend.
bar.width	Width of bars.
x.index	Indexes of x and y axis.
y.index	Indexes of x and y axis.
...	Any other option to pass, check See Also section.

Examples

```
mtcars %>%
  e_charts() %>%
  e_histogram(mpg, name = "histogram") %>%
  e_density(mpg, areaStyle = list(opacity = .4), smooth = TRUE, name = "density", y.index = 1) %>%
  e_tooltip(trigger = "axis")
```

---

e_keras_history *History*

Description

Plot **keras** history in R.

Usage

```
e_keras_history(e)
```

Arguments

e	An echarts4r object as returned by e_charts.

Examples

```
## Not run:
history <- model %>%
 fit(...)

history %>%
 e_charts() %>%
 e_keras_history()

## End(Not run)
```

---

e_leaflet *Leaflet*

Description

Leaflet extension.

Usage

```
e_leaflet(e, roam = TRUE, ...)

e_leaflet_tile(e,
  template = "https://{s}.tile.openstreetmap.fr/hot/{z}/{x}/{y}.png",
  options = NULL, ...)
```

Arguments

e	An `echarts4r` object as returned by `e_charts`.
roam	Whether to allow the user to roam.
...	Any other option to pass, check See Also section.
template	`urlTemplate`, should not be changed.
options	List of options, including `attribution` and `label`.

Note

Will not render in the RStudio, open in browser.

Examples

```
## Not run:
url <- paste0("https://ecomfe.github.io/echarts-examples/",
              "public/data-gl/asset/data/population.json")
data <- jsonlite::fromJSON(url)
data <- as.data.frame(data)
names(data) <- c("lon", "lat", "value")
data$value <- log(data$value)

data %>%
  e_charts(lon) %>%
  e_leaflet() %>%
  e_leaflet_tile() %>%
  e_scatter(lat, size = value, coord.system = "leaflet")

## End(Not run)
```

---

e_legend *Legend*

Description

Customise the legend.

Usage

```
e_legend(e, show = TRUE, type = c("plain", "scroll"), ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
show	Set to FALSE to hide the legend.
type	Type of legend, plain or scroll.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#legend

Examples

```
mtcars %>%
  head() %>%
  dplyr::mutate(model = row.names(.)) %>%
  e_charts(model) %>%
  e_pie(carb) %>%
  e_legend(FALSE)
```

---

e_line *Line*

Description

Add line serie.

Usage

```
e_line(e, serie, bind, name = NULL, legend = TRUE, y.index = 0,
  x.index = 0, coord.system = "cartesian2d", ...)

e_line_(e, serie, bind = NULL, name = NULL, legend = TRUE,
  y.index = 0, x.index = 0, coord.system = "cartesian2d", ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Column name of serie to plot.
bind	Binding between datasets, namely for use of e_brush.
name	name of the serie.
legend	Whether to add serie to legend.
y.index	Indexes of x and y axis.
x.index	Indexes of x and y axis.
coord.system	Coordinate system to plot against.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-line

Examples

```
iris %>%
  group_by(Species) %>%
  e_charts(Sepal.Length) %>%
  e_line(Sepal.Width) %>%
  e_tooltip(trigger = "axis")
```

---

e_lines *Lines*

Description

Add lines.

Usage

```
e_lines(e, source.lon, source.lat, target.lon, target.lat,
  coord.system = "geo", name = NULL, rm.x = TRUE, rm.y = TRUE, ...)

e_lines_(e, source.lon, source.lat, target.lon, target.lat,
  coord.system = "geo", name = NULL, rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

e	An `echarts4r` object as returned by `e_charts`.
source.lon, source.lat, target.lon, target.lat	coordinates.
coord.system	Coordinate system to use, one of `geo`, or `cartesian2d`.
name	name of the serie.
rm.x, rm.y	Whether to remove x and y axis, defaults to `TRUE`.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-lines

Examples

```
flights <- read.csv(
  paste0("https://raw.githubusercontent.com/plotly/datasets/",
         "master/2011_february_aa_flight_paths.csv")
)

flights %>%
  e_charts() %>%
  e_geo() %>%
  e_lines(
    start_lon,
    start_lat,
    end_lon,
    end_lat,
    name = "flights",
    lineStyle = list(normal = list(curveness = 0.3))
   )
```

---

e_lines_3d *Lines 3D*

Description

Add 3D lines.

Usage

```
e_lines_3d(e, source.lon, source.lat, target.lon, target.lat,
  name = NULL, coord.system = "globe", rm.x = TRUE, rm.y = TRUE,
  ...)

e_line_3d(e, y, z, name = NULL, coord.system = NULL, rm.x = TRUE,
  rm.y = TRUE, ...)

e_lines_3d_(e, source.lon, source.lat, target.lon, target.lat,
  name = NULL, coord.system = "globe", rm.x = TRUE, rm.y = TRUE,
  ...)

e_line_3d_(e, y, z, name = NULL, coord.system = NULL, rm.x = TRUE,
  rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
source.lon, source.lat, target.lon, target.lat	coordinates.
name	name of the serie.

coord.system	Coordinate system to use, such as cartesian3D, or globe.
rm.x, rm.y	Whether to remove x and y axis, defaults to TRUE.
...	Any other option to pass, check See Also section.
y, z	Coordinates of lines.

See Also

http://echarts.baidu.com/option-gl.html#series-lines3D, http://echarts.baidu.com/option-gl.html#series-line3D

Examples

```
# get data
flights <- read.csv(
  paste0("https://raw.githubusercontent.com/plotly/datasets/",
         "master/2011_february_aa_flight_paths.csv")
)

# Lines 3D
# Globe
flights %>%
  e_charts() %>%
  e_globe(
    base.texture = e_map_texture(),
    height.texture = e_map_texture(),
    environment = e_stars_texture(),
    displacementScale = 0.05
  ) %>%
  e_lines_3d(
    start_lon,
    start_lat,
    end_lon,
    end_lat,
    name = "flights",
    effect = list(show = TRUE)
  ) %>%
  e_legend(FALSE)

# Geo 3D
flights %>%
  e_charts() %>%
  e_geo_3d() %>%
  e_lines_3d(
    start_lon,
    start_lat,
    end_lon,
    end_lat,
    coord.system = "geo3D"
  )

# line 3D
```

```
df <- data.frame(
  x = 1:100,
  y = runif(100, 10, 25),
  z = rnorm(100, 100, 50)
)

df %>%
  e_charts(x) %>%
  e_line_3d(y, z) %>%
  e_visual_map() %>%
  e_title("nonsense")
```

`e_liquid` *Liquid fill*

Description

Draw liquid fill.

Usage

```
e_liquid(e, serie, color, rm.x = TRUE, rm.y = TRUE, ...)

e_liquid_(e, serie, color = NULL, rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`serie`	Column name of serie to plot.
`color`	Color to plot.
`rm.x, rm.y`	Whether to remove x and y axis, defaults to `TRUE`.
`...`	Any other option to pass, check See Also section.

See Also

https://github.com/ecomfe/echarts-liquidfill

Examples

```
df <- data.frame(val = c(0.6, 0.5, 0.4))

df %>%
  e_charts() %>%
  e_liquid(val) %>%
  e_theme("dark")
```

e_lm *Smooth*

Description

Aids the eye in seeing patterns in the presence of overplotting.

Usage

```
e_lm(e, formula, name = NULL, legend = TRUE, symbol = "none",
  smooth = TRUE, ...)

e_glm(e, formula, name = NULL, legend = TRUE, symbol = "none",
  smooth = TRUE, ...)

e_loess(e, formula, name = NULL, legend = TRUE, symbol = "none",
  smooth = TRUE, x.index = 0, y.index = 0, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
formula	formula to pass to lm.
name	name of the serie.
legend	Whether to add serie to legend.
symbol	Symbol to use in e_line.
smooth	Whether to smooth the line.
...	Additional arguments to pass to e_line.
x.index	Indexes of x and y axis.
y.index	Indexes of x and y axis.

Examples

```
mtcars %>%
  e_charts(mpg) %>%
  e_scatter(qsec) %>%
  e_lm(qsec ~ mpg, name = "y = ax + b")

mtcars %>%
  e_charts(disp) %>%
  e_scatter(mpg) %>%
  e_loess(mpg ~ disp)

CO2 %>%
  e_charts(conc) %>%
  e_scatter(uptake) %>%
  e_glm(uptake ~ conc, name = "GLM")
```

e_map *Choropleth*

Description

Draw maps.

Usage

```
e_map(e, serie, map = "world", name = NULL, rm.x = TRUE,
  rm.y = TRUE, ...)

e_map_(e, serie, map = "world", name = NULL, rm.x = TRUE,
  rm.y = TRUE, ...)

e_map_3d(e, serie, map = "world", name = NULL, rm.x = TRUE,
  rm.y = TRUE, ...)

e_map_3d_(e, serie, map = "world", name = NULL, rm.x = TRUE,
  rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Values to plot.
map	Map type.
name	name of the serie.
rm.x, rm.y	Whether to remove x and y axis, defaults to TRUE.
...	Any other option to pass, check See Also section.

See Also

e_country_names, https://ecomfe.github.io/echarts-doc/public/en/option.html#series-map, http://echarts.baidu.com/option-gl.html#series-map3D

Examples

```
## Not run:
choropleth <- data.frame(
  countries = c("France", "Brazil", "China", "Russia", "Canada", "India", "United States",
                "Argentina", "Australia"),
  values = round(runif(9, 10, 25))
)

choropleth %>%
  e_charts(countries) %>%
  e_map(values) %>%
```

```
  e_visual_map(min = 10, max = 25)

choropleth %>%
  e_charts(countries) %>%
  e_map_3d(values, shading = "lambert") %>%
  e_visual_map(min = 10, max = 30)

## End(Not run)
```

`e_map_register` *Register map*

Description

Register a http://geojson.org/ map.

Usage

```
e_map_register(e, name, json)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`name`	Name of map, to use in `e_map`
`json`	http://geojson.org/.

Examples

```
## Not run:
json <- jsonlite::read_json("http://www.echartsjs.com/gallery/data/asset/geo/USA.json")

USArrests %>%
  dplyr::mutate(states = row.names(.)) %>%
  e_charts(states) %>%
  e_map_register("USA", json) %>%
  e_map(Murder, map = "USA") %>%
  e_visual_map(min = 0, max = 18)

## End(Not run)
```

e_map_texture *Textures*

Description

Path to textures.

Usage

```
e_map_texture(convert = TRUE)

e_globe_texture(convert = TRUE)

e_composite_texture(convert = TRUE)

e_globe_dark_texture(convert = TRUE)

e_stars_texture(convert = TRUE)

e_convert_texture(file)
```

Arguments

convert	Converts image to JSON formatted arrays.
file	Path to file.

Details

Due to browser "same origin policy" security restrictions, loading textures from a file system in three.js may lead to a security exception, see https://github.com/mrdoob/three.js/wiki/How-to-run-things-locally. References to file locations work in Shiny apps, but not in standalone examples. The *texture functions facilitates transfer of image texture data from R into textures when convert is set to TRUE.

Functions

- e_map_texture globe texture
- e_stars_texture starts texture
- e_mconvert_texture convert file to JSON formatted array

Examples

```
## Not run:
e_map_texture(FALSE) # returns path to file
e_convert_texture("path/to/file.png") # converts file

## End(Not run)
```

e_mark_point *Mark point*

Description

Mark points and lines.

Usage

```
e_mark_point(e, serie = NULL, data = NULL, ...)

e_mark_line(e, serie = NULL, data = NULL, ...)

e_mark_area(e, serie = NULL, data = NULL, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Serie to mark on passed to grep, defaults to last added.
data	Placement.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-line.markPoint, https://ecomfe.github.io/echarts-doc/public/en/option.html#series-line.markLine

Examples

```
max <- list(
  name = "Max",
  type = "max"
)

min <- list(
  name = "Min",
  type = "min"
)

avg <- list(
  type = "average",
  name = "AVG"
)

USArrests %>%
  e_charts(Murder) %>%
  e_line(Rape) %>%
  e_line(UrbanPop) %>%
```

```
  e_mark_point(data = max) %>%
  e_mark_point(data = min) %>%
  e_mark_line(serie = "Rape", data = avg) %>%
  e_mark_area(serie = "Rape", data = list(
    list(xAxis = "min", yAxis = "min"),
    list(xAxis = "max", yAxis = "max"))
 )
```

`e_modularity` *Modularity*

Description

Graph modularity extension will do community detection and partian a graph's vertices in several subsets. Each subset will be assigned a different color.

Usage

```
e_modularity(e, modularity = TRUE)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`modularity`	Either set to `TRUE`, or a `list`.

Modularity

- `resolution` Resolution
- `sort` Whether to sort to comunities

Note

Does not work in RStudio viewer, open in browser.

See Also

Official documentation

Examples

```
nodes <- data.frame(
  name = paste0(LETTERS, 1:100),
  value = rnorm(100, 10, 2),
  stringsAsFactors = FALSE
)

edges <- data.frame(
  source = sample(nodes$name, 200, replace = TRUE),
```

```
  target = sample(nodes$name, 200, replace = TRUE),
  stringsAsFactors = FALSE
)

e_charts() %>%
  e_graph() %>%
  e_graph_nodes(nodes, name, value) %>%
  e_graph_edges(edges, source, target) %>%
  e_modularity(
    list(
      resolution = 5,
      sort = TRUE
    )
  )
```

`e_parallel` *Parallel*

Description

Draw parallel coordinates.

Usage

```
e_parallel(e, ..., name = NULL, rm.x = TRUE, rm.y = TRUE)

e_parallel_(e, ..., name = NULL, rm.x = TRUE, rm.y = TRUE)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`...`	Any other option to pass, check See Also section.
`name`	name of the serie.
`rm.x, rm.y`	Whether to remove x and y axis, defaults to `TRUE`.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-parallel

Examples

```
df <- data.frame(
  price = rnorm(5, 10),
  amount = rnorm(5, 15),
  letter = LETTERS[1:5]
)
```

```
df %>%
  e_charts() %>%
  e_parallel(price, amount, letter)
```

`e_pictorial` *Pictorial*

Description

Pictorial bar chart is a type of bar chart that custimzed glyph (like images, SVG PathData) can be used instead of rectangular bar.

Usage

```
e_pictorial(e, serie, symbol, bind, name = NULL, legend = TRUE,
  y.index = 0, x.index = 0, ...)

e_pictorial_(e, serie, symbol, bind = NULL, name = NULL,
  legend = TRUE, y.index = 0, x.index = 0, ...)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`serie`	Column name of serie to plot.
`symbol`	Symbol to plot.
`bind`	Binding between datasets, namely for use of `e_brush`.
`name`	name of the serie.
`legend`	Whether to add serie to legend.
`y.index`	Indexes of x and y axis.
`x.index`	Indexes of x and y axis.
`...`	Any other option to pass, check See Also section.

Symbols

- Built-in `circle`, `rect`, `roundRect`, `triangle`, `diamond`, `pin`, `arrow`.
- SVG Path
- Images Path to image, don't forget to precede it with `image://`, see examples.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-pictorialBar

Examples

```
# built-in symbols
y <- rnorm(10, 10, 2)
df <- data.frame(
  x = 1:10,
  y = y,
  z = y - rnorm(10, 5, 1)
)

df %>%
  e_charts(x) %>%
  e_bar(z, barWidth = 10) %>%
  e_pictorial(y, symbol = "rect", symbolRepeat = TRUE, z = -1,
    symbolSize = c(10, 4)) %>%
  e_theme("westeros")

# svg path
path <- "path://M0,10 L10,10 C5.5,10 5.5,5 5,0 C4.5,5 4.5,10 0,10 z"

style <- list(
  normal = list(opacity = 0.5), # normal
  emphasis = list(opacity = 1) # on hover
)

df %>%
  e_charts(x) %>%
  e_pictorial(y, symbol = path, barCategoryGap = "-130%",
    itemStyle = style)

# image
# might not work in RStudio viewer
# open in browser
qomo <- paste0(
  "https://ecomfe.github.io/echarts-examples/public/",
  "data/asset/img/hill-Qomolangma.png"
)

kili <- paste0(
  "https://ecomfe.github.io/echarts-examples/public/",
  "data/asset/img/hill-Kilimanjaro.png"
)

data <- data.frame(
  x = c("Qomolangma", "Kilimanjaro"),
  value = c(8844, 5895),
  symbol = c(paste0("image://", qomo),
    paste0("image://", kili))
)

data %>%
  e_charts(x) %>%
  e_pictorial(value, symbol) %>%
```

```
e_legend(FALSE)
```

---

e_pie *Pie*

Description

Draw pie and donut charts.

Usage

```
e_pie(e, serie, name = NULL, legend = TRUE, rm.x = TRUE,
  rm.y = TRUE, ...)

e_pie_(e, serie, name = NULL, legend = TRUE, rm.x = TRUE,
  rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Column name of serie to plot.
name	name of the serie.
legend	Whether to add serie to legend.
rm.x, rm.y	Whether to remove x and y axis, defaults to TRUE.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-pie

Examples

```
mtcars %>%
  head() %>%
  dplyr::mutate(model = row.names(.)) %>%
  e_charts(model) %>%
  e_pie(carb)
```

e_polar *Polar*

Description

Customise polar coordinates.

Usage

```
e_polar(e, show = TRUE, ...)
```

Arguments

e	An `echarts4r` object as returned by `e_charts`.
show	Whether to display the axis.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#polar

Examples

```
df <- data.frame(x = 1:10, y = seq(1, 20, by = 2))

df %>%
  e_charts(x) %>%
  e_polar() %>%
  e_angle_axis() %>%
  e_radius_axis() %>%
  e_line(y, coord.system = "polar", smooth = TRUE)
```

e_radar *Radar*

Description

Add a radar chart

Usage

```
e_radar(e, serie, max = 100, name = NULL, legend = TRUE,
  rm.x = TRUE, rm.y = TRUE, ...)

e_radar_(e, serie, max = 100, name = NULL, legend = TRUE,
  rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Column name of serie to plot.
max	Maximum value.
name	name of the serie.
legend	Whether to add serie to legend.
rm.x, rm.y	Whether to remove x and y axis, defaults to TRUE.
...	Any other option to pass, check See Also section.

Examples

```
df <- data.frame(
  x = LETTERS[1:5],
  y = runif(5, 1, 5),
  z = runif(5, 3, 7)
)

df %>%
  e_charts(x) %>%
  e_radar(y, max = 7) %>%
  e_radar(z) %>%
  e_tooltip(trigger = "item")
```

---

e_radar_opts *Radar axis*

Description

Radar axis setup and options.

Usage

```
e_radar_opts(e, index = 0, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
index	Index of axis to customise.
...	Any other option to pass, check See Also section.

e_radius_axis *Radius axis*

Description

Customise radius axis.

Usage

```
e_radius_axis(e, show = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
show	Whether to display the axis.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#radiusAxis

Examples

```
df <- data.frame(x = 1:10, y = seq(1, 20, by = 2))

df %>%
  e_charts(x) %>%
  e_polar() %>%
  e_angle_axis() %>%
  e_radius_axis(FALSE) %>%
  e_line(y, coord.system = "polar", smooth = TRUE)
```

e_river *River*

Description

Build a theme river.

Usage

```
e_river(e, serie, name = NULL, legend = TRUE, rm.x = TRUE,
  rm.y = TRUE, ...)

e_river_(e, serie, name = NULL, legend = TRUE, rm.x = TRUE,
  rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Column name of serie to plot.
name	name of the serie.
legend	Whether to add serie to legend.
rm.x, rm.y	Whether to remove x and y axis, defaults to TRUE.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-themeRiver

Examples

```
dates <- seq.Date(Sys.Date() - 30, Sys.Date(), by = "day")

df <- data.frame(
  dates = dates,
  apples = runif(length(dates)),
  bananas = runif(length(dates)),
  pears = runif(length(dates))
)

df %>%
  e_charts(dates) %>%
  e_river(apples) %>%
  e_river(bananas) %>%
  e_river(pears) %>%
  e_tooltip(trigger = "axis")
```

---

e_sankey *Sankey*

Description

Draw a sankey diagram.

Usage

```
e_sankey(e, source, target, value, layout = "none", rm.x = TRUE,
  rm.y = TRUE, ...)

e_sankey_(e, source, target, value, layout = "none", rm.x = TRUE,
  rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
source, target	Source and target columns.
value	Value change from source to target.
layout	Layout of sankey.
rm.x, rm.y	Whether to remove the x and y axis, defaults to TRUE.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-sankey

Examples

```
sankey <- data.frame(
  source = c("a", "b", "c", "d", "c"),
  target = c("b", "c", "d", "e", "e"),
  value = ceiling(rnorm(5, 10, 1)),
  stringsAsFactors = FALSE
)

sankey %>%
  e_charts() %>%
  e_sankey(source, target, value)
```

---

e_scatter *Scatter*

Description

Add scatter serie.

Usage

```
e_scatter(e, serie, size, bind, scale = "* 1", name = NULL,
  coord.system = "cartesian2d", legend = TRUE, y.index = 0,
  x.index = 0, rm.x = TRUE, rm.y = TRUE, ...)

e_effect_scatter(e, serie, size, bind, scale = "* 1", name = NULL,
  coord.system = "cartesian2d", legend = TRUE, y.index = 0,
  x.index = 0, rm.x = TRUE, rm.y = TRUE, ...)

e_scatter_(e, serie, size = NULL, bind = NULL, scale = "* 1",
  name = NULL, coord.system = "cartesian2d", legend = TRUE,
  y.index = 0, x.index = 0, rm.x = TRUE, rm.y = TRUE, ...)
```

```
e_effect_scatter_(e, serie, size = NULL, bind = NULL, scale = "* 1",
  name = NULL, coord.system = "cartesian2d", legend = TRUE,
  y.index = 0, x.index = 0, rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Column name of serie to plot.
size	Column name containing size of points.
bind	Binding between datasets, namely for use of e_brush.
scale	Scale for size, defaults to * 1 which multiplies the size by 1 (equivalent to no multiplier).
name	name of the serie.
coord.system	Coordinate system to plot against.
legend	Whether to add serie to legend.
y.index	Indexes of x and y axis.
x.index	Indexes of x and y axis.
rm.x, rm.y	Whether to remove x and y axis, only applies if coord.system is not set to cartesian2d.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-scatter, https://ecomfe.github.io/echarts-doc/public/en/option.html#series-effectScatter

Examples

```
USArrests %>%
  e_charts(Assault) %>%
  e_scatter(Murder, Rape) %>%
  e_effect_scatter(Rape, Murder, y.index = 1) %>%
  e_grid(index = c(0, 1)) %>%
  e_tooltip()

quakes %>%
  e_charts(long) %>%
  e_geo(
    roam = TRUE,
    boundingCoords = list(
      c(185, - 10),
      c(165, -40)
    )
  ) %>%
  e_scatter(lat, mag, coord.system = "geo") %>%
  e_visual_map(min = 4, max = 6.5)
```

```
# Calendar
year <- seq.Date(as.Date("2017-01-01"), as.Date("2017-12-31"), by = "day")
values <- rnorm(length(year), 20, 6)

year <- data.frame(year = year, values = values)

year %>%
  e_charts(year) %>%
  e_calendar(range = "2018") %>%
  e_scatter(values, coord.system = "calendar") %>%
  e_visual_map(max = 30)
```

`e_scatter_3d` *Scatter 3D*

Description

Add 3D scatter.

Usage

```
e_scatter_3d(e, y, z, color, size, bind, coord.system = "cartesian3D",
  name = NULL, rm.x = TRUE, rm.y = TRUE, legend = FALSE, ...)

e_scatter_3d_(e, y, z, color = NULL, size = NULL, bind = NULL,
  coord.system = "cartesian3D", name = NULL, rm.x = TRUE,
  rm.y = TRUE, legend = FALSE, ...)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`y, z`	Coordinates.
`color, size`	Color and Size of bubbles.
`bind`	Binding.
`coord.system`	Coordinate system to use, one of `geo3D`, `globe`, or `cartesian3D`.
`name`	name of the serie.
`rm.x, rm.y`	Whether to remove x and y axis, defaults to `TRUE`.
`legend`	Whether to add serie to legend.
`...`	Any other option to pass, check See Also section.

See Also

http://echarts.baidu.com/option-gl.html#series-scatter3D

Examples

```
v <- LETTERS[1:10]
matrix <- data.frame(
  x = sample(v, 300, replace = TRUE),
  y = sample(v, 300, replace = TRUE),
  z = rnorm(300, 10, 1),
  color = rnorm(300, 10, 1),
  size = rnorm(300, 10, 1),
  stringsAsFactors = FALSE
) %>%
  dplyr::group_by(x, y) %>%
  dplyr::summarise(
    z = sum(z),
    color = sum(color),
    size = sum(size)
  ) %>%
  dplyr::ungroup()

matrix %>%
  e_charts(x) %>%
  e_scatter_3d(y, z, size, color) %>%
  e_visual_map(
    min = 1, max = 100,
    inRange = list(symbolSize = c(1, 30)), # scale size
    dimension = 3 # third dimension 0 = x, y = 1, z = 2, size = 3
  ) %>%
  e_visual_map(
    min = 1, max = 100,
    inRange = list(color = c('#bf444c', '#d88273', '#f6efa6')), # scale colors
    dimension = 4, # third dimension 0 = x, y = 1, z = 2, size = 3, color = 4
    bottom = 300 # padding to avoid visual maps overlap
  )

airports <- read.csv(
  paste0("https://raw.githubusercontent.com/plotly/datasets/",
         "master/2011_february_us_airport_traffic.csv")
)

airports %>%
  e_charts(long) %>%
  e_globe(
    environment = e_stars_texture(),
    base.texture = e_globe_texture(),
    globeOuterRadius = 100
  ) %>%
  e_scatter_3d(lat, cnt, coord.system = "globe", blendMode = 'lighter') %>%
  e_visual_map(inRange = list(symbolSize = c(1, 10)))
```

`e_scatter_gl` *Scatter GL*

Description

Draw scatter GL.

Usage

```
e_scatter_gl(e, y, z, name = NULL, coord.system = "geo", rm.x = TRUE,
  rm.y = TRUE, ...)

e_scatter_gl_(e, y, z, name = NULL, coord.system = "geo",
  rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
y, z	Column names containing y and z data.
name	name of the serie.
coord.system	Coordinate system to plot against.
rm.x, rm.y	Whether to remove x and y axis, defaults to TRUE.
...	Any other option to pass, check See Also section.

See Also

http://echarts.baidu.com/option-gl.html#series-scatterGL

Examples

```
quakes %>%
  e_charts(long) %>%
  e_geo(
    roam = TRUE,
    boundingCoords = list(
      c(185, - 10),
      c(165, -40)
     )
  ) %>%
  e_scatter_gl(lat, depth)
```

---

e_showtip_p *Tooltip Proxy*

Description

Proxies to show or hide tooltip.

Usage

```
e_showtip_p(proxy, position = c(10, 10), series.index = NULL,
  name = NULL)

e_hidetip_p(proxy)
```

Arguments

proxy	An echarts4r proxy as returned by echarts4rProxy.
position	Tooltip position.
series.index	Index of serie.
name	Name of serie.

Examples

```
## Not run:
   library(shiny)

   ui <- fluidPage(
     fluidRow(
       actionButton("show", "Show tooltip"),
       actionButton("hide", "Hide tooltip")
     ),
     fluidRow(
       echarts4rOutput("plot"),
       h3("clicked Data"),
       verbatimTextOutput("clickedData"),
       h3("clicked Serie"),
       verbatimTextOutput("clickedSerie"),
       h3("clicked Row"),
       verbatimTextOutput("clickedRow")
     )
   )

   server <- function(input, output, session){
     output$plot <- renderEcharts4r({
       mtcars %>%
         e_charts(mpg) %>%
         e_line(disp, bind = carb, name = "displacement") %>%
         e_line(hp) %>%
         e_x_axis(min = 10) %>%
         e_theme("westeros")
     })

     observeEvent(input$show, {
       echarts4rProxy("plot") %>%
         e_showtip_p(0)
     })

     observeEvent(input$hide, {
```

```
    echarts4rProxy("plot") %>%
      e_hidetip_p()
  })

  output$clickedData <- renderPrint({
    input$plot_clicked_data
  })

  output$clickedSerie <- renderPrint({
    input$plot_clicked_serie
  })

  output$clickedRow <- renderPrint({
    input$plot_clicked_row
  })

}

shinyApp(ui, server)

## End(Not run)
```

---

e_show_loading *Loading*

Description

Show or hide loading.

Usage

```
e_show_loading(e, hide.overlay = TRUE, text = "loading",
  color = "#c23531", text.color = "#000",
  mask.color = "rgba(255, 255, 255, 0.8)", zlevel = 0)

e_hide_loading(e)
```

Arguments

e	An echarts4r object as returned by e_charts.
hide.overlay	Hides the white overaly that appears in shiny when a plot is recalculating.
text	Text to display.
color	Color of spinner.
text.color	Color of text.
mask.color	Color of mask.
zlevel	Z level.

Details

This only applies to Shiny.

Examples

```
## Not run:

# no redraw
# no loading
library(shiny)
ui <- fluidPage(
  fluidRow(
    column(12, actionButton("update", "Update"))
  ),
  fluidRow(
    column(12, echarts4rOutput("plot"))
  )
)

server <- function(input, output){
  data <- eventReactive(input$update, {
    data.frame(
      x = 1:10,
      y = rnorm(10)
    )
  })

  output$plot <- renderEcharts4r({
    data() %>%
      e_charts(x) %>%
      e_bar(y)
  })
}

shinyApp(ui, server)

# add loading
server <- function(input, output){
  data <- eventReactive(input$update, {
    Sys.sleep(1) # sleep one second to show loading
    data.frame(
      x = 1:10,
      y = rnorm(10)
    )
  })

  output$plot <- renderEcharts4r({
    data() %>%
      e_charts(x) %>%
      e_bar(y) %>%
      e_show_loading()
  })
```

```
}

shinyApp(ui, server)

## End(Not run)
```

`e_step` *Step*

Description

Add step serie.

Usage

```
e_step(e, serie, bind, step = c("start", "middle", "end"),
  fill = FALSE, name = NULL, legend = TRUE, y.index = 0,
  x.index = 0, ...)

e_step_(e, serie, bind = NULL, step = c("start", "middle", "end"),
  fill = FALSE, name = NULL, legend = TRUE, y.index = 0,
  x.index = 0, ...)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`serie`	Column name of serie to plot.
`bind`	Binding between datasets, namely for use of `e_brush`.
`step`	Step type, one of `start`, `middle` or end.
`fill`	Set to fill as area.
`name`	name of the serie.
`legend`	Whether to add serie to legend.
`y.index`	Indexes of x and y axis.
`x.index`	Indexes of x and y axis.
`...`	Any other option to pass, check See Also section.

See Also

Additional arguments

Examples

```
USArrests %>%
  dplyr::mutate(State = row.names(.)) %>%
  e_charts(State) %>%
  e_step(Murder, name = "Start", step = "start", fill = TRUE) %>%
  e_step(Rape, name = "Middle", step = "middle") %>%
  e_step(Assault, name = "End", step = "end") %>%
  e_tooltip(trigger = "axis")
```

---

e_sunburst *Sunburst*

Description

Build a sunburst.

Usage

```
e_sunburst(e, parent, child, value, itemStyle, rm.x = TRUE,
  rm.y = TRUE, ...)

e_sunburst_(e, parent, child, value, itemStyle = NULL, rm.x = TRUE,
  rm.y = TRUE, ...)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`parent, child`	Edges.
`value`	Name of column containing values.
`itemStyle`	Name of column containing styles to pass to `child`, expects a `data.frame` or a `list`.
`rm.x, rm.y`	Whether to remove x and y axis, defaults to `TRUE`.
`...`	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-sunburst

Examples

```
df <- data.frame(
  parent = c("earth", "earth", "earth", "mars", "mars"),
  child = c("forest", "ocean", "iceberg", "elon", "curiosity"),
  value = ceiling(rnorm(5, 10, 2))
)
```

```
df %>%
  e_charts() %>%
  e_sunburst(parent, child, value) %>%
  e_theme("westeros")
```

---

e_theme *Themes*

Description

Add a theme.

Usage

```
e_theme(e, theme)

e_theme_custom(e, theme)
```

Arguments

e	An echarts4r object as returned by e_charts.
theme	Theme, see below.

Themes

- dark
- vintage
- westeros
- essos
- wonderland
- walden
- chalk
- infographic
- macarons
- roma
- shine
- purple-passion
- halloween

See Also

create your own theme.

Examples

```
mtcars %>%
  e_charts(mpg) %>%
  e_line(disp) %>%
  e_area(hp) %>%
  e_x_axis(min = 10) -> p

p %>% e_theme("chalk")
p %>% e_theme_custom('{"color":["#ff715e","#ffaf51"]}')
```

`e_title` *Title*

Description

Add title.

Usage

```
e_title(e, text, subtext = NULL, link = NULL, sublink = NULL, ...)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`text, subtext`	Title and Subtitle.
`link, sublink`	Title and Subtitle link.
`...`	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#title

Examples

```
quakes %>%
  dplyr::mutate(mag = exp(mag) / 60) %>%
  e_charts(stations) %>%
  e_scatter(depth, mag) %>%
  e_visual_map(min = 3, max = 7) %>%
  e_title("Quakes", "Stations and Magnitude")
```

e_toolbox_feature *Toolbox*

Description

Add toolbox interface.

Usage

```
e_toolbox_feature(e, feature, ...)

e_toolbox(e, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
feature	Feature to add, defaults to all.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#toolbox

Examples

```
USArrests %>%
  e_charts(UrbanPop) %>%
  e_line(Assault) %>%
  e_area(Murder, y.index = 1, x.index = 1) %>%
  e_datazoom(x.index = 0)
```

e_tooltip *Tooltip*

Description

Customise tooltip

Usage

```
e_tooltip(e, trigger = c("item", "axis"), ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
trigger	What triggers the tooltip, one of item or item.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#tooltip

Examples

```
USArrests %>%
  e_charts(Assault) %>%
  e_bar(Murder) %>%
  e_tooltip()
```

---

e_tree *Tree*

Description

Build a tree.

Usage

```
e_tree(e, parent, child, rm.x = TRUE, rm.y = TRUE, ...)

e_tree_(e, parent, child, rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
parent, child	Edges.
rm.x, rm.y	Whether to remove x and y axis, defaults to TRUE.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-tree

Examples

```
df <- data.frame(
  parent = c("earth","earth","forest","forest","ocean","ocean","ocean","ocean"),
  child = c("ocean","forest","tree","sasquatch","fish","seaweed","mantis shrimp","sea monster")
)

df %>%
  e_charts() %>%
  e_tree(parent, child)
```

`e_treemap` *Treemap*

Description

Build a treemap.

Usage

```
e_treemap(e, parent, child, value, rm.x = TRUE, rm.y = TRUE, ...)

e_treemap_(e, parent, child, value, rm.x = TRUE, rm.y = TRUE, ...)
```

Arguments

`e`	An `echarts4r` object as returned by `e_charts`.
`parent, child`	Edges.
`value`	Value of edges.
`rm.x, rm.y`	Whether to remove x and y axis, defaults to `TRUE`.
`...`	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#series-treemap

Examples

```
df <- data.frame(
  parent = c("earth", "earth", "earth", "mars", "mars"),
  child = c("forest", "ocean", "iceberg", "elon", "curiosity"),
  value = ceiling(rnorm(5, 10, 2))
)

df %>%
  e_charts() %>%
  e_treemap(parent, child, value)
```

e_utc *Use UTC*

Description

Use UTC

Usage

```
e_utc(e)
```

Arguments

e	An echarts4r object as returned by e_charts.

e_visual_map *Visual Map*

Description

Visual Map

Usage

```
e_visual_map(e, serie, calculable = TRUE, type = c("continuous",
  "piecewise"), ...)

e_visual_map_(e, serie = NULL, calculable = TRUE,
  type = c("continuous", "piecewise"), ...)
```

Arguments

e	An echarts4r object as returned by e_charts.
serie	Column name of serie to scale against.
calculable	Whether show handles, which can be dragged to adjust "selected range".
type	One of continuous or piecewise.
...	Any other option to pass, check See Also section.

See Also

https://ecomfe.github.io/echarts-doc/public/en/option.html#visualMap

Examples

```
v <- LETTERS[1:10]
matrix <- data.frame(
  x = sample(v, 300, replace = TRUE),
  y = sample(v, 300, replace = TRUE),
  z = rnorm(300, 10, 1),
  color = rnorm(300, 10, 1),
  size = rnorm(300, 10, 1),
  stringsAsFactors = FALSE
) %>%
  dplyr::group_by(x, y) %>%
  dplyr::summarise(
    z = sum(z),
    color = sum(color),
    size = sum(size)
  ) %>%
  dplyr::ungroup()

matrix %>%
  e_charts(x) %>%
  e_scatter_3d(y, z, color, size) %>%
  e_visual_map(
    z, # scale to z
    inRange = list(symbolSize = c(1, 30)), # scale size
    dimension = 3 # third dimension 0 = x, y = 1, z = 2, size = 3
  ) %>%
  e_visual_map(
    z, # scale to z
    inRange = list(color = c('#bf444c', '#d88273', '#f6efa6')), # scale colors
    dimension = 4, # third dimension 0 = x, y = 1, z = 2, size = 3, color = 4
    bottom = 300 # padding to avoid visual maps overlap
  )
```

Index