

# Package ‘highdir’

July 31, 2026

**Title** Unified Data Visualization Framework for Dynamic and Static Graphics

**Version** 0.6.0

**Description** Provides a unified API for creating data visualizations across dynamic and static rendering modes. Visualizations are defined once using a specification object and can be rendered with 'highcharter', 'ggplot2', or other supported packages without modifying user code. The package supports declarative and layered workflows, reusable themes and colour palettes, optional 'JavaScript' enhancements, export tools, and interactive exploration through 'shiny' applications.

**URL** <https://folkehelsestats.github.io/highdir/>,  
<https://github.com/folkehelsestats/highdir>

**BugReports** <https://github.com/folkehelsestats/highdir/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**Depends** R (>= 4.1.0)

**Imports** highcharter (>= 0.9.4), ggplot2 (>= 3.4.0), htmlwidgets (>= 1.6.0), jsonlite (>= 1.8.0), rlang (>= 1.1.0), viridis (>= 0.6.0), shiny (>= 1.7.0), tools, utils, stats, scales (>= 1.4.0), lifecycle (>= 1.0.5)

**Suggests** sf (>= 1.0.0), maps (>= 3.4.0), webshot2, bslib (>= 0.7.0), testthat (>= 3.0.0), withr (>= 2.5.0), data.table, knitr, eulerr, ggVennDiagram, rmarkdown, tibble

**Config/testthat/edition** 3

**VignetteBuilder** knitr

**Config/roxygen2/version** 8.0.0

**NeedsCompilation** no

**Author** Yusman Kamaleri [aut, cre] (ORCID:  
<<https://orcid.org/0000-0001-5014-3665>>)

**Maintainer** Yusman Kamaleri <ybkamaleri@gmail.com>

**Repository** CRAN

**Date/Publication** 2026-07-31 15:20:14 UTC

## Contents

|                                  |    |
|----------------------------------|----|
| +.hd . . . . .                   | 3  |
| alco1 . . . . .                  | 4  |
| alco2 . . . . .                  | 4  |
| geom_args . . . . .              | 5  |
| get_palette . . . . .            | 5  |
| gg_theme . . . . .               | 6  |
| hd . . . . .                     | 7  |
| hd_add_js . . . . .              | 9  |
| hd_app . . . . .                 | 10 |
| hd_geom_arearange . . . . .      | 11 |
| hd_geom_column . . . . .         | 12 |
| hd_geom_line . . . . .           | 13 |
| hd_geom_pie . . . . .            | 15 |
| hd_geom_ranked_bar . . . . .     | 16 |
| hd_geom_scatter . . . . .        | 18 |
| hd_geom_stacked_column . . . . . | 18 |
| hd_geom_venn . . . . .           | 20 |
| hd_make . . . . .                | 22 |
| hd_opts . . . . .                | 25 |
| hd_reset_theme . . . . .         | 27 |
| hd_save . . . . .                | 28 |
| hd_set_theme . . . . .           | 29 |
| hd_spec . . . . .                | 30 |
| hd_spec_venn . . . . .           | 31 |
| hd_theme . . . . .               | 32 |
| hd_venn_df . . . . .             | 33 |
| hd_venn_intersect . . . . .      | 35 |
| hd_venn_set . . . . .            | 36 |
| hd_venn_sets_from_spec . . . . . | 36 |
| list_geoms . . . . .             | 37 |
| list_modes . . . . .             | 37 |
| list_palettes . . . . .          | 38 |
| print.hd . . . . .               | 38 |
| register_backend . . . . .       | 39 |
| register_geom . . . . .          | 39 |
| register_palette . . . . .       | 40 |
| venn_df_to_list . . . . .        | 41 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>43</b> |
|--------------|-----------|

---

|      |                                   |
|------|-----------------------------------|
| +.hd | <i>Add Layers to an hd Object</i> |
|------|-----------------------------------|

---

## Description

Implements the + operator for `hd()` objects, mirroring how `ggplot2` builds plots incrementally. Recognised right-hand sides:

## Usage

```
## S3 method for class 'hd'
e1 + e2
```

## Arguments

`e1`                    An `hd` object (left-hand side).  
`e2`                    An `hd_geom` or `hd_opts` object (right-hand side).

## Details

`hd_geom` **object** Sets the geometry (from any `hd_geom_*`() call). Adding a second geom replaces the first where `highdir` renders one geometry per figure.

`hd_opts` **object** Sets presentation options. Adding a second `hd_opts` replaces the first.

Unknown right-hand sides produce an informative error rather than silently being ignored.

## Value

The updated `hd` object (invisibly, so the chain prints once).

## Examples

```
df <- data.frame(age = c("18-24", "25-34"), pct = c(42, 55))

# Chain layers
p <- hd(df, x = "age", y = "pct") +
  hd_geom_column() +
  hd_opts(title = "Demo")

# Reuse a partial object with different opts
base <- hd(df, x = "age", y = "pct") + hd_geom_column()
base + hd_opts(title = "English title")
base + hd_opts(title = "Norsk tittel")
```

---

|       |                                 |
|-------|---------------------------------|
| alco1 | <i>Alcohol Consumption Data</i> |
|-------|---------------------------------|

---

**Description**

Annual estimates of alcohol consumption, including adjusted mean values, standard errors, and 95% confidence intervals. Includes an alternative unit (adj\_enhet) with corresponding uncertainty estimates.

**Format**

A data frame with 14 rows and 9 variables:

**Source**

Helsedirektoratet. *Tall om alkohol*. <https://www.helsedirektoratet.no/rapporter/tall-om-alkohol>

---

|       |                                      |
|-------|--------------------------------------|
| alco2 | <i>Alcohol Consumption by Gender</i> |
|-------|--------------------------------------|

---

**Description**

Annual estimates of alcohol consumption stratified by kjønn (gender), including adjusted means, standard errors, and 95% intervals. Also includes an alternative unit (adj\_enhet) with uncertainty.

**Format**

A data frame with 28 rows and 10 variables:

**Source**

Helsedirektoratet. *Tall om alkohol*. <https://www.helsedirektoratet.no/rapporter/tall-om-alkohol>

---

|           |                                      |
|-----------|--------------------------------------|
| geom_args | <i>Show Arguments for a Geometry</i> |
|-----------|--------------------------------------|

---

**Description**

Prints the required and optional . . . arguments accepted by a geometry when used with `hd_make()`. This is the primary discoverability tool for geometry-specific arguments that do not appear in `hd_make()`'s signature.

**Usage**

```
geom_args(type = NULL)
```

**Arguments**

|      |                                                                                                                         |
|------|-------------------------------------------------------------------------------------------------------------------------|
| type | Character. Geometry name, e.g. "line", "ranked_bar". If NULL (default), prints a summary for every registered geometry. |
|------|-------------------------------------------------------------------------------------------------------------------------|

**Value**

A data frame of argument metadata, invisibly. The primary purpose is the side-effect of printing.

**Why this exists**

`hd_make()` uses . . . for all geometry-specific arguments so its own signature stays clean regardless of how many geometries are registered. The trade-off is that users cannot see available args from `hd_make()` alone. `geom_args()` solves that: it reads the `required_args` and `optional_args` fields registered for each geometry and presents them in a readable table.

**Examples**

```
geom_args("line")
geom_args("ranked_bar")
geom_args("arearange")
geom_args()          # all registered geometries
```

---

|             |                                 |
|-------------|---------------------------------|
| get_palette | <i>Retrieve a Named Palette</i> |
|-------------|---------------------------------|

---

**Description**

Retrieve a Named Palette

**Usage**

```
get_palette(name)
```

Arguments

name                      Character. Palette name (see [list\\_palettes\(\)](#)).

Value

Character vector of colours, or NULL if not found.

Examples

```
get_palette("hdir")
```

---

|          |                                     |
|----------|-------------------------------------|
| gg_theme | <i>Build a ggplot2 Theme Object</i> |
|----------|-------------------------------------|

---

Description

Constructs a ggplot2 theme by resolving a base theme, then merging colour and font overrides – exactly mirroring what [hd\\_theme\(\)](#) does for the highcharter backend.

Usage

```
gg_theme(theme = NULL, colors = NULL, font = NULL)
```

Arguments

|        |                                                                                                                                                                                                                                                                                  |
|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| theme  | Character name string, ggplot2 theme object, or NULL. NULL reads from <code>getOption("highdir.gg_theme")</code> .                                                                                                                                                               |
| colors | Character vector, palette name string, or NULL. Resolved colours are stored on the returned object and applied by <code>ggplot_engine()</code> via <code>scale_color_manual</code> / <code>scale_fill_manual</code> . NULL reads from <code>getOption("highdir.colors")</code> . |
| font   | Character or NULL. Font family applied to all text elements via <code>theme(text = element_text(family = font))</code> . NULL reads from <code>getOption("highdir.font")</code> .                                                                                                |

Details

Priority for each argument:

- theme: explicit argument > `getOption("highdir.gg_theme")` > "classic"
- colors: explicit argument > `getOption("highdir.colors")` > NULL
- font: explicit argument > `getOption("highdir.font")` > NULL

Called automatically inside `ggplot_engine()`; also useful for applying the package theme to a ggplot built outside highdir.

Built-in name strings and their ggplot2 equivalents:

| Name                | ggplot2 function             |
|---------------------|------------------------------|
| "classic" (default) | <code>theme_classic()</code> |

|                 |                 |
|-----------------|-----------------|
| "minimal"       | theme_minimal() |
| "bw"            | theme_bw()      |
| "light"         | theme_light()   |
| "dark"          | theme_dark()    |
| "void"          | theme_void()    |
| "grey" / "gray" | theme_grey()    |

## Value

An object of class "hd\_gg\_theme" - a list with two fields: \$theme (a ggplot2 theme object with font baked in) and \$colors (a resolved character vector or NULL). ggplot\_engine() unpacks both. The object can also be added directly to a ggplot with + via the +.gg method (only the theme is applied; colors are handled separately by apply\_gg\_colors()).

## Examples

```
gg_theme()                                # session defaults
gg_theme("bw")                            # theme_bw(), session colors/font
gg_theme("classic", colors = c("#025169", "#7C145C"))
gg_theme("minimal", font = "Source Sans Pro")
gg_theme(ggplot2::theme_bw(base_size = 14), font = "mono")
```

---

hd

*Initialise a Composable highdir Figure*


---

## Description

Creates an hd object that accumulates geometry and presentation layers via +, then renders when printed. This mirrors the way ggplot2 builds plots: data mapping is declared first; visual decisions are added incrementally; nothing is rendered until the object hits the console (or an explicit print() / knit call).

## Usage

```
hd(
  data = NULL,
  x = NULL,
  y = NULL,
  group = NULL,
  n = NULL,
  colour = NULL,
  mode = NULL,
  backend = lifecycle::deprecated()
)
```

## Arguments

|         |                                                                                                                                                                                                                                                                              |
|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data    | A data.frame <b>or</b> an <a href="#">hd_spec()</a> object. When an hd_spec is supplied every other mapping argument (x, y, ...) is ignored - the spec carries them already. Default is NULL, which creates an empty data.frame.                                             |
| x       | Character. Column name for the x-axis variable. Ignored when data is an hd_spec.                                                                                                                                                                                             |
| y       | Character. Column name for the y-axis variable. Ignored when data is an hd_spec.                                                                                                                                                                                             |
| group   | Character or NULL. Column used to split data into multiple series. Ignored when data is an hd_spec.                                                                                                                                                                          |
| n       | Character or NULL. Column of raw counts shown in highcharter tooltips alongside the y value. Ignored when data is an hd_spec.                                                                                                                                                |
| colour  | Character or NULL. ggplot2 colour aesthetic column. Defaults to group when NULL and group is set. Ignored when data is an hd_spec.                                                                                                                                           |
| mode    | Character. Rendering mode - "dynamic" (default, interactive) or "static". "dynamic" uses the highcharter backend, "static" uses the ggplot2 backend and others. See <a href="#">list_modes()</a> .                                                                           |
| backend | Character. Rendering engine - "dynamic" (default, interactive) or "static", or any engine added with <a href="#">register_backend()</a> . Falls back to <code>getOption("highdir.backend", "dynamic")</code> . This will be deprecated in favor of mode in a future release. |

## Details

`hd()` accepts either raw data columns **or** an existing [hd\\_spec\(\)](#) object. Passing an `hd_spec` is the recommended bridge for code that already constructs specs separately (e.g. in Shiny or a reporting pipeline).

## Value

An S3 object of class "hd" with slots:

`$spec` An [hd\\_spec\(\)](#) object.

`$geom` NULL until a `+ hd_geom_*()` layer is added.

`$opts` An [hd\\_opts\(\)](#) object (defaults until overridden).

`$mode` Character. The resolved engine name.

## See Also

[hd\\_geom\\_column\(\)](#), [hd\\_geom\\_line\(\)](#), [hd\\_geom\\_arearange\(\)](#), [hd\\_opts\(\)](#), [hd\\_make\(\)](#)

## Examples

```
df <- data.frame(
  age = rep(c("18-24", "25-34", "35-44", "45-54"), each = 2),
  sex = rep(c("Male", "Female"), 4),
  pct = c(42, 38, 55, 61, 48, 52, 60, 57),
```

```

    n = c(120, 115, 200, 210, 180, 175, 160, 155)
  )

  # Composable style
  hd(df, x = "age", y = "pct", group = "sex") +
    hd_geom_column() +
    hd_opts(title = "Health survey", ylim = c(0, 80))

  # Pass an existing hd_spec
  spec <- hd_spec(df, x = "age", y = "pct", group = "sex", n = "n")
  hd(spec) +
    hd_geom_line(smooth = TRUE) +
    hd_opts(title = "Trend")

  # Switch backend per figure
  hd(df, x = "age", y = "pct", mode = "static") +
    hd_geom_column() +
    hd_opts(title = "Static version")

```

hd\_add\_js

*Inject JavaScript into a Highcharts Widget***Description**

Appends custom JavaScript to a highchart object via `chart.events.<where>`. Use this for hand-written callbacks and plugins. For Highcharts built-in modules (accessibility, exporting, etc.) use [highcharter::hc\\_add\\_dependency\(\)](#) instead.

**Usage**

```

hd_add_js(
  hc,
  code = NULL,
  file = NULL,
  plugin = NULL,
  where = c("load", "render")
)

```

**Arguments**

|        |                                                                        |
|--------|------------------------------------------------------------------------|
| hc     | A highchart object.                                                    |
| code   | Character or NULL. Raw JavaScript string.                              |
| file   | Character or NULL. Path to a .js file to read and inject.              |
| plugin | Character or NULL. Name of a plugin bundled in <code>inst/js/</code> . |
| where  | "load" (default) or "render".                                          |

**Details**

Exactly one of code, file, or plugin must be supplied.

**Value**

The highchart object with JS injected.

**Examples**

```
if (interactive()) {
  spec <- hd_spec(mtcars, "wt", "mpg")
  fig <- hd_make(spec, "scatter")
  fig <- hd_add_js(fig, code = "console.log('chart loaded');")
}
```

---

hd\_app

*Launch the highdir Shiny GUI*


---

**Description**

Opens an interactive browser-based application for building figures with highcharter or ggplot2 without writing R code.

**Usage**

```
hd_app(return.app = FALSE)
```

**Arguments**

|            |                                                                                                   |
|------------|---------------------------------------------------------------------------------------------------|
| return.app | Logical. When deploying to server like Shiny.io to return the app object instead of launching it. |
|------------|---------------------------------------------------------------------------------------------------|

**Details**

The UI (inst/app/ui.R), server (inst/app/server.R) and shared setup (inst/app/global.R) live in inst/app/ so the folder can be deployed independently to Shiny Server or shinyapps.io.

**Value**

Launches a Shiny app; does not return a value.

## Features

- Upload datasets in any format supported by **rio** (CSV, XLSX, SPSS, Stata, RDS, ...).
- Choose geometry (column, line, scatter, arearange, pie), backend, axis variables, and group column.
- Set title, subtitle, caption, colour palette, and HC theme.
- Toggle JS hover band per figure.
- Render on demand with the **Draw** button.
- Download as HTML / JSON / PNG (highcharter) or PNG / SVG (ggplot2).
- Copy the equivalent hd\_make() call from the **R code** tab.

## See Also

[hd\\_make\(\)](#), [hd\\_spec\(\)](#), [hd\\_opts\(\)](#), [hd\\_save\(\)](#)

---

|                   |                                                 |
|-------------------|-------------------------------------------------|
| hd_geom_arearange | <i>Add an Arearange (Confidence Band) Layer</i> |
|-------------------|-------------------------------------------------|

---

## Description

Geometry layer for ribbon / confidence-interval charts. Unlike the other `hd_geom_*`() functions, `ymin` and `ymax` are **required** named arguments (they map to column names in `spec$data`) rather than optional ... extras. This makes the contract explicit at the call site instead of burying required information inside ....

## Usage

```
hd_geom_arearange(ymin = NULL, ymax = NULL, ...)
```

## Arguments

|                   |                                                                                                                                                                                                          |
|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>ymin</code> | Character. Column name for the lower bound of the range.                                                                                                                                                 |
| <code>ymax</code> | Character. Column name for the upper bound of the range.                                                                                                                                                 |
| <code>...</code>  | Additional optional arguments forwarded to the geom function (e.g. <code>show_line = FALSE</code> , <code>single_colour = "#025169"</code> ). Run <code>geom_args("arearange")</code> for the full list. |

## Value

An S3 object of class "hd\_geom" for use with `+.hd`.

## Examples

```
# Single series
spec_ar1 <- hd_spec(alco1,
                    x     = "year",
                    y     = "adj_enhet")

opts_ar <- hd_opts(
  title     = "Alcohol consumption with 95% CI",
  subtitle  = "Source: Norwegian Directorate of Health",
  ylim      = c(0, 30),
  ylab      = "Number of units alcohol"
)

hd_make(spec_ar1, "arearange", opts_ar,
        ymin = "lower_enhet", ymax = "upper_enhet")

# Static ggplot2 version
hd_make(spec_ar1, "arearange", opts_ar,
        ymin = "lower_enhet", ymax = "upper_enhet", mode = "static")

#' # Multi-series with group column
# Grouped by kjonn
spec_ar2 <- hd_spec(alco2,
                    x     = "year",
                    y     = "adj_enhet",
                    group = "kjonn")

hd_make(spec_ar2, "arearange", opts_ar,
        ymin = "lower_enhet", ymax = "upper_enhet")
```

---

hd\_geom\_column

*Column Geometry Layer for hd Objects*


---

## Description

hd\_geom\_column() creates a column geometry layer that is added to an [hd\(\)](#) object via +. The layer records the geometry type and any geometry-specific arguments; rendering only happens when the hd object is printed.

## Usage

```
hd_geom_column(...)
```

## Arguments

...                      Geometry-specific arguments forwarded to [hd\\_make\(\)](#).

**Value**

An S3 object of class "hd\_geom" for use with `+.hd`.

**Examples**

```
survey <- data.frame(
  age_group = rep(c("18-24", "25-34", "35-44", "45-54", "55-64"), each = 2),
  kjonn      = rep(c("Male", "Female"), times = 5),
  pct        = c(42, 38, 55, 61, 48, 52, 60, 57, 65, 70),
  n          = c(120, 115, 200, 210, 180, 175, 160, 155, 140, 145)
)

spec_col <- hd_spec(survey,
  x      = "age_group",
  y      = "pct",
  group  = "kjonn",
  n      = "n")

opts_col <- hd_opts(
  title   = "Alcohol use by age group and kjonn",
  subtitle = "Source: Norwegian Directorate of Health",
  ylim    = c(0, 100),
  yint    = 20,
  ylab    = "Percentage (%)"
)

# Interactive (default)
hd_make(spec_col, "column", opts_col)

# Static ggplot2
hd_make(spec_col, "column", opts_col, backend = "static")

# Composable style
p <- hd(survey, x = "age_group", y = "pct", group = "kjonn")
p2 <- p + hd_geom_column()

# More options
p2 + hd_opts(title = "Health survey", ylim = c(0, 100))

# Pass an existing hd_spec
spec <- hd_spec(survey, x = "age_group", y = "pct", group = "kjonn", n = "n")

hd(spec, backend = "static") +
  hd_geom_column() +
  hd_opts(title = "Health survey", ylim = c(0, 80))
```

## Description

hd\_geom\_line() creates a line geometry layer that is added to an `hd()` object via `+`. The layer records the geometry type and any geometry-specific arguments; rendering only happens when the `hd` object is printed.

## Usage

```
hd_geom_line(smooth = TRUE, dot_size = 4, line_symbols = NULL, ...)
```

## Arguments

|                           |                                                                                                                          |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------|
| <code>smooth</code>       | Logical. TRUE = spline curves, FALSE = straight segments. Both backends.                                                 |
| <code>dot_size</code>     | Numeric. Marker radius in pixels. Both backends.                                                                         |
| <code>line_symbols</code> | Character vector. Highcharter only. Per-group marker shapes: "circle", "square", "diamond", "triangle", "triangle-down". |
| <code>...</code>          | Geometry-specific arguments forwarded to <code>hd_make()</code> .                                                        |

## Value

An S3 object of class "hd\_geom" for use with `+.hd`.

## Examples

```
# Single series - no group column
spec_line1 <- hd_spec(alco1,
  x = "year",
  y = "adj_mean"
)

opts_line <- hd_opts(
  title = "Alcohol consumption over time",
  subtitle = "Source: Norwegian Directorate of Health",
  ylim = c(0, 50),
  ylab = "Litres per capita"
)

# Straight segments
hd_make(spec_line1, "line", opts_line, smooth = FALSE)

# Composite example with multiple geoms and custom line symbols
hd(alco2, x = "year", y = "adj_mean", group = "kjonnn", backend = "static") +
  hd_geom_line(smooth = TRUE, dot_size = 3) +
  hd_opts(title = "Alcohol consumption over time by kjonnn",
    subtitle = "Source: Norwegian Directorate of Health")
```

hd\_geom\_pie

*Pie Geometry Layer for hd Objects***Description**

hd\_geom\_pie() creates a pie geometry layer that is added to an [hd\(\)](#) object via +. The layer records the geometry type and any geometry-specific arguments; rendering only happens when the hd object is printed.

**Usage**

```
hd_geom_pie(inner_size = NULL, value_suffix = NULL, ...)
```

**Arguments**

|              |                                                                                                                                                                                                                                                                                                                                      |
|--------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| inner_size   | A string specifying the inner radius of the pie as a percentage of the total radius. For example, "50%" creates a donut chart with a hole in the middle. The default "0%" creates a standard pie chart. This argument is only applicable to the Highcharts backend; it is ignored by ggplot2 since it does not support donut charts. |
| value_suffix | A string to append to the labels on the pie slices. Default is "%".                                                                                                                                                                                                                                                                  |
| ...          | Geometry-specific arguments forwarded to <a href="#">hd_make()</a> .                                                                                                                                                                                                                                                                 |

**Value**

An S3 object of class "hd\_geom" for use with +.hd.

**Examples**

```
# Category share dataset (pie)
drinking_freq <- data.frame(
  category = c("Never", "Rarely", "Monthly", "Weekly", "Daily"),
  pct      = c(18, 25, 30, 20, 7)
)

spec_pie <- hd_spec(drinking_freq,
  x = "category",
  y = "pct"
)

opts_pie <- hd_opts(
  title = "Drinking frequency",
  subtitle = "Source: Norwegian Directorate of Health",
  ylab = "Share (%)"
)

# Donut interactive
hd_make(spec_pie, "pie", opts_pie, inner_size = "50%")
```

```
# Composable API style (ggplot2 ignores inner_size)
hd(drinking_freq, x = "category", y = "pct", backend = "static") +
  hd_geom_pie() +
  hd_opts(
    title = "Drinking frequency",
    subtitle = "Source: Norwegian Directorate of Health"
  )
```

---

hd\_geom\_ranked\_bar      *Ranked bar geometry*

---

## Description

Draws a ranked bar chart with smart label placement and optional comparison highlighting. Bars are sorted by value, and labels are placed inside or outside the bar depending on available space. A single category can be highlighted with a contrasting fill colour, and an optional horizontal line can be added to indicate a target or benchmark value.

## Usage

```
hd_geom_ranked_bar(
  ascending = TRUE,
  vs = NULL,
  aim = NULL,
  char_scale = 0.045,
  min_frac = 0.08,
  ...
)
```

## Arguments

|            |                                                                                                                                                                                                                                                            |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ascending  | Logical. If TRUE (default) bars are sorted in ascending order of y.                                                                                                                                                                                        |
| vs         | Character string (partial match) identifying one category to highlight with a contrasting fill colour. If omitted all bars use the same colour.                                                                                                            |
| aim        | Numeric. Optional horizontal line indicating a target or benchmark value. If NULL (default) no line is drawn.                                                                                                                                              |
| char_scale | Numeric scaling factor that converts label character-count into axis-range units. Controls how generously space is estimated for each character. Defaults to 0.045; increase (e.g. 0.060.06) for larger text sizes, decrease (e.g. 0.03) for smaller ones. |
| min_frac   | Numeric. Minimum fraction of the axis range that a bar must span before its label is considered to fit inside. Acts as a safety floor for very short labels. Defaults to 0.08 (8%).                                                                        |
| ...        | Geometry-specific arguments forwarded to <a href="#">hd_make()</a> .                                                                                                                                                                                       |

**Value**

An S3 object of class "hd\_geom" for use with `+.hd`.

**See Also**

[hd\\_geom\\_column\(\)](#), [hd\\_geom\\_line\(\)](#), [hd\\_geom\\_arearange\(\)](#), [hd\\_opts\(\)](#), [hd\\_make\(\)](#)

**Examples**

```
# Regional health indicator dataset
regions <- data.frame(
  region = c("Oslo", "Viken", "Vestland", "Rogaland",
            "Trondelag", "Innlandet", "Agder",
            "Nordland", "Troms og Finnmark"),
  rate   = c(68.4, 71.2, 87.8, 10.5, 61.3, 6.1, 54.2, 49.8, 42.1),
  n      = c(402, 448, 681, 318, 297, 251, 198, 177, 148)
)

# Declarative API ----
spec_rb <- hd_spec(regions,
  x = "region",
  y = "rate",
  n = "n")

opts_rb <- hd_opts(
  title   = "Health indicator by region",
  subtitle = "Source: Norwegian Directorate of Health",
  ylab    = "Rate per 100 000",
  flip    = TRUE
)

hd_make(spec_rb, "ranked_bar", opts_rb, vs = "Oslo", aim = 63)

# Layered API ----
hd(regions, x = "region", y = "rate", n = "n", mode = "static") +
  hd_geom_ranked_bar(
    ascending = TRUE,
    vs        = "Oslo",
    aim       = 63,
    char_scale = 0.045,
    min_frac  = 0.08) +
  hd_opts(
    title   = "Health indicator by region",
    subtitle = "Source: Norwegian Directorate of Health",
    ylab    = "Rate per 100 000",
    flip    = TRUE
  )
```

---

|                 |                                              |
|-----------------|----------------------------------------------|
| hd_geom_scatter | <i>Scatter Geometry Layer for hd Objects</i> |
|-----------------|----------------------------------------------|

---

### Description

hd\_geom\_scatter() creates a scatter geometry layer that is added to an [hd\(\)](#) object via +. Use [geom\\_args\(\)](#) to discover available arguments per geometry, e.g. geom\_args("scatter") lists dot\_size.

### Usage

```
hd_geom_scatter(dot_size = 4, ...)
```

### Arguments

|          |                                                                                                                                    |
|----------|------------------------------------------------------------------------------------------------------------------------------------|
| dot_size | Numeric. Size of the points in the scatter plot. Default is 4.                                                                     |
| ...      | Geometry-specific arguments forwarded to <a href="#">hd_make()</a> .<br>@return An S3 object of class "hd_geom" for use with +.hd. |

### Value

An S3 object of class "hd\_geom" for use with +.hd.

### Examples

```
# Basic scatter plot - layered API
hd(mtcars, x = "wt", y = "mpg", backend = "static") +
  hd_geom_scatter() +
  hd_opts(title = "Scatter Plot of mtcars")

# Basic scatter plot - declarative API
car <- hd_spec(mtcars, x = "wt", y = "mpg")
opt <- hd_opts(title = "Scatter Plot of mtcars")
hd_make(car, type = "scatter")
```

---

|                        |                                      |
|------------------------|--------------------------------------|
| hd_geom_stacked_column | <i>Stacked Column Geometry Layer</i> |
|------------------------|--------------------------------------|

---

### Description

Create a stacked column geometry layer for hd objects. Each stack is a facet (sub-panel) containing one or more series. The stack argument specifies the column in the data that defines the stacks if exists. The group aesthetic in hd\_spec() defines the series within each stack. The stacking argument controls how the stacks are rendered: "normal" (default) stacks values on top of each other, while "percent" stacks values as percentages of the total stack height.

**Usage**

```
hd_geom_stacked_column(facet = NULL, stacking = c("normal", "percent"), ...)
```

**Arguments**

|          |                                                                                                                                                                                                                                                                                                                                             |
|----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| facet    | Character. Column name for the facet variable. Each unique value in this column creates a separate facet (stack) containing all series with that faceted value.                                                                                                                                                                             |
| stacking | Character. Stacking mode for the column geometry. One of "normal" (default) or "percent". For ggplot2, is equivalent to position = "fill", else see Highcharts documentation for details: <a href="https://api.highcharts.com/highcharts/plotOptions.column.stacking">https://api.highcharts.com/highcharts/plotOptions.column.stacking</a> |
| ...      | Additional optional arguments forwarded to the geom function (e.g. show_line = FALSE, single_colour = "#025169"). Run geom_args("arearange") for the full list.                                                                                                                                                                             |

**Value**

An S3 object of class "hd\_geom" for use with +.hd.

**Examples**

```
# Example data: medal counts for four countries across three medal types
olympics <- data.frame(
  Country = rep(c("Norway", "Germany", "United States", "Canada"), each = 3),
  Continent = rep(c("Europe", "Europe", "North America", "North America"), each = 3),
  Medal = rep(c("Gold", "Silver", "Bronze"), times = 4),
  Count = c(148, 133, 124, 102, 98, 65, 113, 122, 95, 77, 72, 80)
)

# Define Specification and Options
spec_st <- hd_spec(olympics,
  x = "Medal",
  y = "Count",
  group = "Country"
)

opts_st <- hd_opts(
  title = "Olympic Games all-time medal table, grouped by continent",
  subtitle = "Source: Olympics",
  ylab = "Count medals"
)

hd_make(spec_st, "stacked_column", opts_st)

# Interactive - stacks are separated by continent
hd_make(spec_st, "stacked_column", opts_st, facet = "Continent")

# Static ggplot2 - stacks are separated by continent
hd(spec_st, mode = "static") +
  hd_geom_stacked_column(facet = "Continent") +
  hd_opts(title = "Olympic Games all-time medal table, grouped by continent", ylab = "Count medals")
```

hd\_geom\_venn

*Venn / Euler Diagram Layer for hd Objects***Description**

Creates a Venn or Euler diagram layer that is added to an [hd\(\)](#) object via `+`. The diagram is described by a list of set entries supplied via the `sets` argument.

**Usage**

```
hd_geom_venn(
  sets,
  series_name = "Venn Diagram",
  label_font_size = "14px",
  value_suffix = "",
  use_names = FALSE,
  show_legend = FALSE,
  legend_offset_y = 0.6,
  legend_height = 0.85,
  ...
)
```

**Arguments**

|                              |                                                                                                                                                  |
|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>sets</code>            | A list of set entries. See <b>Data format</b> above. Use <a href="#">hd_venn_set()</a> and <a href="#">hd_venn_intersect()</a> to build entries. |
| <code>series_name</code>     | Character. Series label shown in the Highcharts chart. Default "Venn Diagram". Highcharter only.                                                 |
| <code>label_font_size</code> | Character. CSS font-size for set labels. Default "14px". Highcharter only.                                                                       |
| <code>value_suffix</code>    | Character. Appended to each region's value label. Default "" (no suffix). static mode only.                                                      |
| <code>use_names</code>       | Logical. If TRUE, display human-readable name values instead of set ids in labels and legend. Default FALSE. static mode only.                   |
| <code>show_legend</code>     | Logical. If TRUE, render a legend beneath the diagram. Default FALSE. static mode only.                                                          |
| <code>legend_offset_y</code> | Numeric (0–1). Vertical shift of the venn grob. Increase to move diagram upward and create more space for legend.                                |
| <code>legend_height</code>   | Numeric (0–1). Relative height of the venn grob. Decrease to allocate more space for legend.                                                     |
| <code>...</code>             | Additional arguments forwarded to <a href="#">hd_make()</a> .                                                                                    |

**Value**

An S3 object of class "hd\_geom" for use with `+.hd`.

## Venn vs Euler

Supply only the intersections that actually exist in your data. Highcharts will separate circles that share no intersection entry, producing an Euler diagram automatically. No separate geom is needed. Note: mathematically impossible layouts (e.g. three pairs all intersect but the triple intersection is zero) cannot be rendered and will error. If chart fails to display as intended, it might be due to the intersection is larger than the single sets that contain them.

## Data format

sets is a list of named lists. Each element describes either a single set or an intersection between sets:

```
list(
  # Single sets - must have name and value
  list(sets = list("A"), name = "Animals", value = 5),
  list(sets = list("B"), name = "Four legs", value = 4),
  list(sets = list("C"), name = "Mineral", value = 2),
  # Intersections - sets contains two or more names, value is overlap size
  list(sets = list("A", "B"), value = 3)
)
```

Use `hd_venn_set()` and `hd_venn_intersect()` to build entries without typing the nested list structure by hand.

## Extending the set list

Because sets is a plain R list, it can be built and extended programmatically with `c()` or `append()`:

```
base <- list(
  hd_venn_set("A", "Animals", 5),
  hd_venn_set("B", "Four legs", 4)
)
extended <- c(base, list(hd_venn_intersect(c("A", "B"), 3)))
hd(mode = "dynamic") +
  hd_geom_venn(sets = extended)
```

## Mode differences

**dynamic** Uses Highcharts' native "venn" series type. All features (labels, tooltips, interactivity) work out of the box. `series_name` and `label_font_size` are highcharter-only options.

**static** Requires the **eulerr** package (preferred) or **ggVennDiagram** in Suggests. Install with `install.packages("eulerr")`.

## See Also

`hd_venn_df()`, `hd_venn_set()`, `hd_venn_intersect()`, `venn_df_to_list()`

## Examples

```
# Build the set list with helper constructors
my_sets <- list(
  hd_venn_set("A", "Animals", value = 5),
  hd_venn_set("B", "Four legs", value = 4),
  hd_venn_set("C", "Mineral", value = 2),
  hd_venn_intersect(c("A", "B"), value = 3)
)

# Dynamic (interactive)
hd(mode = "dynamic") +
  hd_geom_venn(sets = my_sets) +
  hd_opts(title = "Animals and Minerals")

# Static - requires eulerr or ggVennDiagram
hd(mode = "static") +
  hd_geom_venn(sets = my_sets) +
  hd_opts(title = "Animals and Minerals")

# Extend a base list programmatically
base_sets <- list(
  hd_venn_set("A", "Oslo", value = 120),
  hd_venn_set("B", "Bergen", value = 95)
)

extended <- c(base_sets, list(
  hd_venn_intersect(c("A", "B"), "Both", value = 40)
))

hd(mode = "dynamic") +
  hd_geom_venn(sets = extended) +
  hd_opts(title = "City overlap")

# Convert a data.frame to a venn set list with hd_venn_df() function
id = c("A", "B", "A,B")
name = c("Oslo", "Bergen", "Both")
value = c(80, 30, 10)
n = c(120, 95, 40)
type = c(rep("set", 2), "intersect")

df <- data.frame(type, id, name, value, n)

df2 <- hd_venn_df(df, output = "spec")
hd_make(df2, "venn", hd_opts(title = "City overlap"), mode = "dynamic")
```

## Description

Renders a [hd\\_spec](#) and [hd\\_opts](#) pair using the selected mode and geometry. This is the central function of the package - everything else feeds into or flows out of `hd_make()`.

## Usage

```
hd_make(
  spec,
  type = "column",
  opts = NULL,
  mode = NULL,
  backend = lifecycle::deprecated(),
  use_js = TRUE,
  module = FALSE,
  ...
)
```

## Arguments

|                      |                                                                                                                                                                                                                                                                                                                             |
|----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>spec</code>    | A <a href="#">hd_spec</a> object from <a href="#">hd_spec()</a> .                                                                                                                                                                                                                                                           |
| <code>type</code>    | Character. Geometry name - one of <a href="#">list_geoms()</a> : "column", "line", "scatter", "arearange", "pie", or any custom geometry added with <a href="#">register_geom()</a> .                                                                                                                                       |
| <code>opts</code>    | A <a href="#">hd_opts</a> object or NULL (uses all defaults). Controls title, subtitle, caption, ylim, yint, flip, per-figure colours, and highcharter theme.                                                                                                                                                               |
| <code>mode</code>    | Character. Rendering mode - "dynamic" (default, interactive) or "static". "dynamic" uses the highcharter backend, "static" uses the ggplot2 backend and others. See <a href="#">list_modes()</a> .                                                                                                                          |
| <code>backend</code> | Character. Rendering engine - "dynamic" (default, interactive) or "static", or any engine added with <a href="#">register_backend()</a> . Falls back to <code>getOption("highdir.backend", "dynamic")</code> . This will be deprecated in favor of mode in a future release.                                                |
| <code>use_js</code>  | Logical. When TRUE (default) injects a hover-band <code>htmlwidgets::JS()</code> callback via <code>point.events.mouseOver/Out</code> . Tooltips, accessibility module, and all other Highcharts declarative features are <b>always</b> present. Set FALSE for clean, no-custom-JS widgets. Ignored by the ggplot2 backend. |
| <code>module</code>  | Use available modules.js from CDN <a href="https://api.highcharts.com/highcharts/">https://api.highcharts.com/highcharts/</a>                                                                                                                                                                                               |
| <code>...</code>     | Extra arguments forwarded to the geometry function. Required arguments (e.g. <code>ymin</code> , <code>ymax</code> for "arearange") <b>must</b> be supplied here.                                                                                                                                                           |

## Value

A highchart widget (highcharter backend) or ggplot object (ggplot2 backend), invisibly wrapped so knitr/Shiny render it automatically.

## Workflow

```
spec <- hd_spec(df, x = "age", y = "pct", group = "sex", n = "n")
opts <- hd_opts(title = "Health survey", ylim = c(0, 80))
```

```

hd_make(spec, "column", opts)           # highcharter (default)
hd_make(spec, "column", opts, mode = "static") # static ggplot2
hd_make(spec, "line", opts, smooth = TRUE)   # smooth spline
hd_make(spec, "pie", opts)                  # pie / donut

```

## See Also

[hd\\_spec\(\)](#), [hd\\_opts\(\)](#), [hd\\_save\(\)](#), [hd\\_set\\_theme\(\)](#), [list\\_geoms\(\)](#), [list\\_modes\(\)](#), [hd\\_app\(\)](#)

## Examples

```

df <- data.frame(
  age = rep(c("18-24", "25-34", "35-44", "45-54"), each = 2),
  sex = rep(c("Male", "Female"), 4),
  pct = c(42, 38, 55, 61, 48, 52, 60, 57),
  n    = c(120, 115, 200, 210, 180, 175, 160, 155)
)

spec <- hd_spec(df, x = "age", y = "pct", group = "sex", n = "n")

opts <- hd_opts(title = "Health survey results",
  subtitle = "Source: FHI 2024",
  ylim = c(0, 80))

# -- Interactive charts (highcharter) -----
hd_make(spec, "column", opts)
hd_make(spec, "line", opts, smooth = TRUE)
hd_make(spec, "line", opts, smooth = FALSE, dot_size = 6)
hd_make(spec, "scatter")

# Pie chart - group is ignored; x = label, y = value
pie_df <- data.frame(category = c("A", "B", "C", "D"),
  value = c(35, 25, 20, 20))
pie_spec <- hd_spec(pie_df, x = "category", y = "value")
pie_opts <- hd_opts(title = "Share by category")
hd_make(pie_spec, "pie", pie_opts)
hd_make(pie_spec, "pie", pie_opts, inner_size = "50%") # donut

# Arearange - requires ymin + ymax in ...
df2 <- cbind(df, lo = df$pct - 5, hi = df$pct + 5)
spec2 <- hd_spec(df2, "age", "pct", group = "sex")
hd_make(spec2, "arearange", opts, ymin = "lo", ymax = "hi")

# -- Disable JS hover band -----
hd_make(spec, "column", opts, use_js = FALSE)

# -- Static ggplot2 versions -----
hd_make(spec, "column", opts, mode = "static")
hd_make(spec, "line", opts, mode = "static")
hd_make(spec, "scatter", opts, mode = "static")

```

```

hd_make(pie_spec, "pie", pie_opts, mode = "static")

# -- Reuse spec with different presentation -----
opts_no <- hd_opts(title = "Helseundersøkelse", subtitle = "Alle aldre")
hd_make(spec, "column", opts_no)

# -- Save outputs -----
## Not run:
hd_save(hd_make(spec, "column", opts), "column.html")
hd_save(hd_make(spec, "column", opts, mode = "static"), "column.png")

## End(Not run)

```

---

hd\_opts

---

*Create Figure Presentation Options*


---

## Description

Defines the **visual presentation** of a figure independently from the data mapping. Pass the result as the `opts` argument of `hd_make()`, or omit it to accept all defaults.

## Usage

```

hd_opts(
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  description = NULL,
  xlab = " ",
  ylab = " ",
  ylim = NULL,
  yint = 10,
  ysuffix = NULL,
  xtick_labels = NULL,
  decimals = NULL,
  flip = FALSE,
  colors = NULL,
  hc_theme = NULL,
  gg_theme = NULL
)

```

## Arguments

|                       |                                                                            |
|-----------------------|----------------------------------------------------------------------------|
| <code>title</code>    | Character or NULL. Chart title.                                            |
| <code>subtitle</code> | Character or NULL. Subtitle. Highcharter default: "Kilde: Navn av kilder". |

|              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| caption      | Character or NULL. Caption text displayed below the figure (highcharter only). This is a <b>visible</b> footnote, distinct from description which is read only by assistive technology.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| description  | <p>Character or NULL. Invisible accessibility description of the figure intended for screen readers and other assistive technology.</p> <p><b>highcharter</b> Passed to <code>hc_accessibility(description = ...)</code>. Requires the accessibility module, which <code>highdir</code> loads automatically. Screen readers announce this text when the user focuses the chart.</p> <p><b>ggplot2</b> Set as the alt label via <code>labs(alt = ...)</code>, available since <code>ggplot2</code> 3.3.0. Rendered as an HTML alt attribute when the plot is saved to SVG or included in an R Markdown / Quarto document with <code>fig.alt</code> support.</p> <p>Write a concise one- or two-sentence summary of what the figure shows, including the key trend or comparison, so the information is equally accessible to users who cannot see the chart. Example: "Bar chart showing alcohol use by age group. Use is highest in the 45–54 age group at 65% and lowest in 18–24 at 42%."</p> |
| xlab         | <p>Character or NULL. X-axis label.</p> <p>" " <b>(default)</b> Use the x column name from <code>hd_spec()</code>.</p> <p>NULL Hide the x-axis label completely.</p> <p><b>any string</b> Use that string as the label.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| ylab         | Character or NULL. Y-axis label. Same rules as <code>xlab</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| ylim         | Numeric vector of length 2 or NULL. Fixed y-axis limits, e.g. <code>c(0, 100)</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| yint         | Positive numeric. Y-axis tick interval. Default 10.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| ysuffix      | Character or NULL. String appended to every y-axis tick label, e.g. "%" or " km". NULL shows no suffix.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| xtick_labels | Character or NULL. Column name supplying custom x-axis tick labels when the plotted x-values are numeric but the displayed labels should come from another column. Highcharter only. Note: Highcharts indexes categories from 0, not 1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| decimals     | Integer or NULL. Number of decimal places to round the y-values to before rendering. Applied to the data via <code>check_decimals()</code> in <code>hd_make()</code> . NULL leaves values unchanged.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| flip         | Logical. Invert axes (horizontal bars / inverted chart). Default FALSE.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| colors       | Character vector, palette name string, or NULL. Per-figure colour override; takes precedence over <code>hd_set_theme()</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| hc_theme     | Character or NULL. Per-figure highcharter theme name; takes precedence over <code>hd_set_theme()</code> . See <code>hd_theme()</code> for valid names.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| gg_theme     | Character name string, ggplot2 theme object, or NULL. Per-figure ggplot2 theme; takes precedence over <code>hd_set_theme()</code> . Name strings: "classic" (default), "minimal", "bw", "light", "dark", "void", "grey". Or pass a ggplot2::theme_*() object directly, e.g. <code>ggplot2::theme_bw(base_size = 14)</code> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

**Details**

Because opts are separate from `hd_spec()`, the same data mapping can be rendered with multiple styles without repetition:

```
spec    <- hd_spec(df, "age", "pct", group = "sex")
opts_en <- hd_opts(title = "Health survey",  subtitle = "All ages")
opts_no <- hd_opts(title = "Helseundersøkelse", subtitle = "Alle aldre")

hd_make(spec, "column", opts_en)
hd_make(spec, "column", opts_no)
```

**Value**

An S3 object of class "hd\_opts".

**See Also**

[hd\\_spec\(\)](#), [hd\\_make\(\)](#), [hd\\_set\\_theme\(\)](#)

**Examples**

```
opts <- hd_opts(
  title      = "Health survey results",
  subtitle   = "Source: FHI 2024",
  caption    = "Tall om helse",
  description = paste(
    "Grouped bar chart showing alcohol use by age group and sex.",
    "Use is highest in the 45-54 age group at 65% for women."
  ),
  ylim       = c(0, 100),
  yint       = 20,
  colors     = c("#025169", "#7C145C")
)
opts
```

---

hd\_reset\_theme

*Reset Theme*


---

**Description**

Reset package-wide style defaults

**Usage**

```
hd_reset_theme()
```

**See Also**

[hd\\_set\\_theme\(\)](#) to set custom defaults

---

|         |                              |
|---------|------------------------------|
| hd_save | <i>Save a Figure to Disk</i> |
|---------|------------------------------|

---

## Description

Exports a highchart or ggplot figure produced by `hd_make()` to a file. The output format is inferred from the file extension unless `type` is supplied explicitly.

## Usage

```
hd_save(
  fig,
  file,
  type = "auto",
  width = 8,
  height = 5,
  dpi = 300,
  selfcontained = TRUE,
  ...
)
```

## Arguments

|                            |                                                                              |
|----------------------------|------------------------------------------------------------------------------|
| <code>fig</code>           | A highchart or ggplot object (output of <code>hd_make()</code> ).            |
| <code>file</code>          | Character. Output path including extension.                                  |
| <code>type</code>          | "auto" (infer from extension) or an explicit format string. Default: "auto". |
| <code>width</code>         | Numeric. Width in inches (ggplot2). Default: 8.                              |
| <code>height</code>        | Numeric. Height in inches (ggplot2). Default: 5.                             |
| <code>dpi</code>           | Numeric. Raster resolution for ggplot2. Default: 300.                        |
| <code>selfcontained</code> | Logical. Embed all JS/CSS in the HTML file. Default: TRUE.                   |
| <code>...</code>           | Passed to <code>ggplot2::ggsave()</code> for ggplot2 figures.                |

## Details

| Backend     | Supported formats                           |
|-------------|---------------------------------------------|
| highcharter | html, json                                  |
| ggplot2     | png, svg, pdf, jpeg, jpg,<br>tiff, bmp, eps |

To export a highcharter figure as an image, either save as html and screenshot in a browser, or re-render with `backend = "static"` in `hd_make()` and save as png.

## Value

file, invisibly.

hd\_set\_theme

*Set Package-Wide Style Defaults***Description**

Configures the default theme, colour palette, font, and optional JavaScript plugins for all figures produced with [hd\\_make\(\)](#) in the current R session. Call once at the top of a script or in .Rprofile.

**Usage**

```
hd_set_theme(
  hc_theme = NULL,
  gg_theme = NULL,
  colors = NULL,
  font = NULL,
  js_plugins = NULL
)
```

**Arguments**

|            |                                                                                                                                                                                                                                                                                                      |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| hc_theme   | Character or NULL. Built-in highcharter theme name: one of "default", "smp1", "economist", "darkunica", "gridlight", "bloom", "flat", "flatdark", "ggplot2".                                                                                                                                         |
| gg_theme   | Character, ggplot2 theme object, or NULL. Controls the ggplot2 backend appearance. Built-in name strings: "minimal" (default), "classic", "bw", "light", "dark", "void", "grey". Alternatively pass any ggplot2::theme_*() object directly for full control, e.g. ggplot2::theme_bw(base_size = 14). |
| colors     | Character vector, palette name, or NULL. Applied to all figures in the session. See <a href="#">register_palette()</a> .                                                                                                                                                                             |
| font       | Character or NULL. Font family name, e.g. "Source Sans Pro".                                                                                                                                                                                                                                         |
| js_plugins | Character vector or NULL. Names of bundled JS plugins (files in inst/js/) injected into every highcharter figure. Use character(0) to clear all plugins.                                                                                                                                             |

**Details**

Per-figure overrides are provided via [hd\\_opts\(\)](#), which always take precedence over these session defaults.

**Value**

The previous option values invisibly; pass to [options\(\)](#) to restore.

**See Also**

[hd\\_opts\(\)](#) for per-figure overrides

## Examples

```
hd_set_theme(hc_theme = "economist", gg_theme = "classic",
             colors = c("#025169", "#7C145C", "#C68803"))
# Reset
hd_set_theme(hc_theme = "default", gg_theme = "minimal", colors = "hdir")
hd_reset_theme()
```

---

hd\_spec

---

Create a Figure Data Specification

---

## Description

Defines the **data mapping** for a figure - which columns map to x, y, group, and count - independently of any visual presentation choices. Pass the result to [hd\\_make\(\)](#) together with an optional [hd\\_opts\(\)](#) object.

## Usage

```
hd_spec(data = NULL, x, y, group = NULL, n = NULL, colour = NULL)
```

## Arguments

|        |                                                                                                                                                                                                           |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| data   | A data.frame containing all referenced columns. Default is NULL, which creates an empty data.frame. The original data source is recorded in the "src_data" attribute for printing and debugging purposes. |
| x      | Character. Column name for the x-axis variable.                                                                                                                                                           |
| y      | Character. Column name for the y-axis variable (typically a percentage or count).                                                                                                                         |
| group  | Character or NULL. Column used to split data into multiple series.                                                                                                                                        |
| n      | Character or NULL. Column of raw counts shown in highcharter tooltips alongside the y value. Ignored by ggplot2.                                                                                          |
| colour | Character or NULL. ggplot2 colour aesthetic column. Defaults to group when NULL and group is set.                                                                                                         |

## Value

An S3 object of class "hd\_spec".

## See Also

[hd\\_opts\(\)](#), [hd\\_make\(\)](#)

**Examples**

```
df <- data.frame(
  age = rep(c("18-24", "25-34", "35-44", "45-54"), each = 2),
  sex = rep(c("Male", "Female"), 4),
  pct = c(42, 38, 55, 61, 48, 52, 60, 57),
  n = c(120, 115, 200, 210, 180, 175, 160, 155)
)

spec <- hd_spec(df, x = "age", y = "pct", group = "sex", n = "n")
spec
```

hd\_spec\_venn

*Create a Venn / Euler Figure Specification***Description**

Declarative-API equivalent of [hd\\_spec\(\)](#) for venn and euler diagrams. Wraps a pre-built set list inside an `hd_spec` object so it can be passed to [hd\\_make\(\)](#) using the same `spec / opts / hd_make()` pattern as every other `highdir` geometry.

**Usage**

```
hd_spec_venn(sets)
```

**Arguments**

|                   |                                                                                                                                                                                                                              |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>sets</code> | A non-empty list of set entries built with <a href="#">hd_venn_set()</a> and <a href="#">hd_venn_intersect()</a> , or raw named lists following the format: <code>list(sets = list("A"), name = "Label", value = 5)</code> . |
|-------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Value**

An S3 object of class `c("hd_spec_venn", "hd_spec")`.

**Usage with `hd_make()`**

```
sets <- list(
  hd_venn_set("A", "Oslo", 120),
  hd_venn_set("B", "Bergen", 95),
  hd_venn_intersect(c("A", "B"), 40)
)

spec_v <- hd_spec_venn(sets)
opts_v <- hd_opts(title = "City overlap")

# Declarative API - identical pattern to every other geom
hd_make(spec_v, "venn", opts_v) # highcharter
hd_make(spec_v, "venn", opts_v, backend = "static") # ggplot2
```

### Compared to the composable API

Both approaches produce identical output. Use whichever fits your workflow:

```
# Declarative (hd_spec / hd_opts / hd_make)
hd_make(hd_spec_venn(sets), "venn", hd_opts(title = "City overlap"))

# Composable (hd / + / hd_geom_venn)
hd(backend = "dynamic") +
  hd_geom_venn(sets = sets) +
  hd_opts(title = "City overlap")
```

### See Also

[hd\\_venn\\_set\(\)](#), [hd\\_venn\\_intersect\(\)](#), [hd\\_venn\\_sets\\_from\\_spec\(\)](#), [hd\\_make\(\)](#), [hd\\_opts\(\)](#)

### Examples

```
sets <- list(
  hd_venn_set("A", "Oslo", value = 120),
  hd_venn_set("B", "Bergen", value = 95),
  hd_venn_intersect(c("A", "B"), value = 40)
)
spec_v <- hd_spec_venn(sets)
opts_v <- hd_opts(title = "City overlap")

hd_make(spec_v, "venn", opts_v)
hd_make(spec_v, "venn", opts_v, backend = "static")
hd_make(hd_spec_venn(sets), "venn", hd_opts(title = "City overlap"))
```

---

hd\_theme

*Build a Highcharts Theme Object*


---

### Description

Constructs a highcharter theme by merging a named base theme with colour and font overrides from the current [hd\\_set\\_theme\(\)](#) session defaults and any per-figure opts.

### Usage

```
hd_theme(name = NULL, colors = NULL, ...)
```

### Arguments

|        |                                                                                                                   |
|--------|-------------------------------------------------------------------------------------------------------------------|
| name   | Character or NULL. Theme name; NULL reads from <code>getOption("highdir.hc_theme")</code> .                       |
| colors | Character vector or NULL. Colour override for this call.                                                          |
| ...    | Named arguments forwarded to <a href="#">highcharter::hc_theme()</a> as extra overrides on top of the base theme. |

**Details**

Called automatically inside the highcharter engine; useful when you want to apply a theme to a highchart built outside highdir.

**Value**

A highcharter theme object (hc\_theme).

**Examples**

```
if(interactive()) {
  t <- hd_theme("darkunica")
  highcharter::highchart() |> highcharter::hc_add_theme(t)
}
```

hd\_venn\_df

*Create a Venn Diagram Directly from a Data Frame***Description**

A single-call convenience wrapper that accepts the same tidy data frame as [venn\\_df\\_to\\_list\(\)](#) and returns either an hd\_spec\_venn object (for the declarative API) or an hd\_geom layer (for the composable + API), depending on the output argument.

**Usage**

```
hd_venn_df(
  df,
  output = c("spec", "geom"),
  series_name = "Venn Diagram",
  label_font_size = "14px",
  value_suffix = "",
  use_names = FALSE,
  show_legend = FALSE
)
```

**Arguments**

|                 |                                                                                                                         |
|-----------------|-------------------------------------------------------------------------------------------------------------------------|
| df              | A data frame with columns id, name, value, type. See <a href="#">venn_df_to_list()</a> for the full format description. |
| output          | Character. "spec" (default) returns an hd_spec_venn; "geom" returns an hd_geom layer for use with +.                    |
| series_name     | Character. Passed to <a href="#">hd_geom_venn()</a> when output = "geom". Default "Venn Diagram".                       |
| label_font_size | Character. Passed to <a href="#">hd_geom_venn()</a> when output = "geom". Default "14px".                               |

|              |                                                                                                                       |
|--------------|-----------------------------------------------------------------------------------------------------------------------|
| value_suffix | Character. Passed to <a href="#">hd_geom_venn()</a> . Suffix appended to value labels (e.g. "%"). Default "".         |
| use_names    | Logical. Passed to <a href="#">hd_geom_venn()</a> . Use human-readable names instead of ids in labels. Default FALSE. |
| show_legend  | Logical. Passed to <a href="#">hd_geom_venn()</a> . Show a legend beneath the diagram. Default FALSE.                 |

## Details

This means you never need to call [venn\\_df\\_to\\_list\(\)](#), [hd\\_spec\\_venn\(\)](#), or [hd\\_geom\\_venn\(\)](#) directly when starting from a data frame.

## Value

An `hd_spec_venn` object when `output = "spec"`, or an `hd_geom` object when `output = "geom"`.

## Which output to use

`output = "spec" (default)` Returns an `hd_spec_venn` object. Pass to [hd\\_make\(\)](#) just like any other spec. Best for reporting pipelines and scripts that separate data from presentation.

`output = "geom"` Returns an `hd_geom` layer. Add to an [hd\(\)](#) object with `+`. Best for interactive exploration and inline charts.

## Declarative API (output = "spec")

```
df <- data.frame(
  type = c("set", "set", "intersect"),
  id   = c("A", "B", "A,B"),
  name = c("Oslo", "Bergen", "Both"),
  value = c(120, 95, 40)
)

spec_v <- hd_venn_df(df)
hd_make(spec_v, "venn", hd_opts(title = "City overlap"))
```

## Composable API (output = "geom")

```
hd(mode = "dynamic") +
  hd_venn_df(df, output = "geom") +
  hd_opts(title = "City overlap")
```

## See Also

[venn\\_df\\_to\\_list\(\)](#), [hd\\_spec\\_venn\(\)](#), [hd\\_geom\\_venn\(\)](#), [hd\\_make\(\)](#), [hd\(\)](#)

**Examples**

```
df <- data.frame(
  type = c("set", "set", "intersect"),
  id   = c("A", "B", "A,B"),
  name = c("Oslo", "Bergen", "Both"),
  value = c(120, 95, 40)
)

# Declarative API
spec_v <- hd_venn_df(df)
hd_make(spec_v, "venn", hd_opts(title = "City overlap"))

# Composable API
hd(mode = "dynamic") +
  hd_venn_df(df, output = "geom") +
  hd_opts(title = "City overlap")

# static mode
hd(mode = "static") +
  hd_venn_df(df, output = "geom") +
  hd_opts(title = "City overlap")
```

---

|                   |                                                     |
|-------------------|-----------------------------------------------------|
| hd_venn_intersect | <i>Build an Intersection Entry for hd_geom_venn</i> |
|-------------------|-----------------------------------------------------|

---

**Description**

Creates one entry describing the overlap region between two or more sets. The ids must match the id values used in the corresponding `hd_venn_set()` calls.

**Usage**

```
hd_venn_intersect(ids, value, name = NULL)
```

**Arguments**

|       |                                                                                                                               |
|-------|-------------------------------------------------------------------------------------------------------------------------------|
| ids   | Character vector of length $\geq 2$ identifying the overlapping sets. E.g. <code>c("A", "B")</code> for the A-B intersection. |
| value | Numeric. Area / size of the intersection region.                                                                              |
| name  | Character or NULL. Optional label for the intersection region. Usually left NULL.                                             |

**Value**

A named list suitable for inclusion in the sets argument of `hd_geom_venn()`.

**Examples**

```
hd_venn_intersect(c("A", "B"), value = 2)
#> list(sets = list("A", "B"), value = 2)
```

---

|             |                                                  |
|-------------|--------------------------------------------------|
| hd_venn_set | <i>Build a Single-Set Entry for hd_geom_venn</i> |
|-------------|--------------------------------------------------|

---

**Description**

Creates one set entry describing a single, non-overlapping set region. Pass the result directly into the sets list of [hd\\_geom\\_venn\(\)](#).

**Usage**

```
hd_venn_set(id, name, value)
```

**Arguments**

|       |                                                                                           |
|-------|-------------------------------------------------------------------------------------------|
| id    | Character. Short identifier used to reference this set in intersection entries. E.g. "A". |
| name  | Character. Human-readable label shown inside the circle.                                  |
| value | Numeric. Area / size of this set region (excluding overlaps).                             |

**Value**

A named list suitable for inclusion in the sets argument of [hd\\_geom\\_venn\(\)](#).

**Examples**

```
hd_venn_set("A", "Animals", 5)
#> list(sets = list("A"), name = "Animals", value = 5)
```

---

|                        |                                                         |
|------------------------|---------------------------------------------------------|
| hd_venn_sets_from_spec | <i>Extract the Set List from an hd_spec_venn Object</i> |
|------------------------|---------------------------------------------------------|

---

**Description**

Convenience extractor to retrieve the set list stored inside a spec created by [hd\\_spec\\_venn\(\)](#). Useful when you want to inspect or extend the set list before passing to [hd\\_make\(\)](#).

**Usage**

```
hd_venn_sets_from_spec(spec)
```

Arguments

spec                    An hd\_spec\_venn object.

Value

The set list (same object that was passed to [hd\\_spec\\_venn\(\)](#)).

Examples

```
sets <- list(hd_venn_set("A", "Oslo", 120))
spec_v <- hd_spec_venn(sets)
hd_venn_sets_from_spec(spec_v) # returns the original sets list
```

---

|            |                                   |
|------------|-----------------------------------|
| list_geoms | <i>List Registered Geometries</i> |
|------------|-----------------------------------|

---

Description

List Registered Geometries

Usage

```
list_geoms()
```

Value

Character vector of registered geometry names.

---

|            |                                 |
|------------|---------------------------------|
| list_modes | <i>List Registered Backends</i> |
|------------|---------------------------------|

---

Description

List Registered Backends

Usage

```
list_modes()
```

Value

Character vector of registered backend names.

---

|               |                                 |
|---------------|---------------------------------|
| list_palettes | <i>List Registered Palettes</i> |
|---------------|---------------------------------|

---

**Description**

List Registered Palettes

**Usage**

```
list_palettes()
```

**Value**

Character vector of registered palette names.

**Examples**

```
list_palettes()
```

---

|          |                            |
|----------|----------------------------|
| print.hd | <i>Render an hd Object</i> |
|----------|----------------------------|

---

**Description**

Printing an `hd()` object triggers rendering. The geometry type and any geometry-specific parameters are extracted from the stored `hd_geom` layer; presentation options from the `hd_opts` layer; the backend from the `$backend` slot. All of these are forwarded to `hd_make()`, which performs the actual rendering via the registered engine.

**Usage**

```
## S3 method for class 'hd'
print(x, ...)
```

**Arguments**

|     |                                      |
|-----|--------------------------------------|
| x   | An hd object.                        |
| ... | Ignored; present for S3 consistency. |

**Details**

You rarely need to call `print.hd()` directly since R calls it automatically when the object appears at the top level, in knitr/Quarto chunks, or in Shiny `renderHighchart()` / `renderPlot()` blocks.

**Value**

The rendered output (a highchart widget or a ggplot object), invisibly.

---

|                  |                                                                        |
|------------------|------------------------------------------------------------------------|
| register_backend | <i>Register a Rendering Backend and used when loding in zzz.R file</i> |
|------------------|------------------------------------------------------------------------|

---

**Description**

Register a Rendering Backend and used when loding in zzz.R file

**Usage**

```
register_backend(name, engine)
```

**Arguments**

|        |                                                       |
|--------|-------------------------------------------------------|
| name   | Character. Unique backend identifier (e.g. "static"). |
| engine | Function.                                             |

**Value**

name, invisibly.

---

|               |                            |
|---------------|----------------------------|
| register_geom | <i>Register a Geometry</i> |
|---------------|----------------------------|

---

**Description**

Adds a geometry to the registry. The optional\_args field is the key mechanism for user discoverability: it is what geom\_args() prints, what the Shiny app reads to build dynamic UI, and what automated documentation can harvest without parsing source code.

**Usage**

```
register_geom(  
  name,  
  ggplot_fun = NULL,  
  highcharter_fun = NULL,  
  required_args = list(),  
  optional_args = list(),  
  skip_base_fig = FALSE  
)
```

**Arguments**

|                 |                                                                                                                                                                                                                                                                 |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name            | Character. Unique geometry identifier.                                                                                                                                                                                                                          |
| ggplot_fun      | Function or NULL.                                                                                                                                                                                                                                               |
| highcharter_fun | Function or NULL.                                                                                                                                                                                                                                               |
| required_args   | Named list of list(default, desc). Args that <b>MUST</b> be supplied via ... in hd_make(). Validation fails if any are missing.                                                                                                                                 |
| optional_args   | Named list of list(default, desc). Args that <b>MAY</b> be supplied and have a sensible default when omitted. These are purely informational from the registry's perspective - the geom function applies the defaults itself via geom_params\$key %  % default. |
| skip_base_fig   | Logical. Bypasses base_fig() in both engines.                                                                                                                                                                                                                   |

**Details**

Structure of optional\_args: A named list where each element is itself a list with two fields: default - the value used when the arg is not supplied (may be NULL) desc - a short human-readable description (shown by geom\_args())

Example:

```
optional_args = list(
  smooth = list(default = TRUE, desc = "Spline smoothing (logical)"),
  dot_size = list(default = 4, desc = "Marker radius in px (numeric)")
)
```

**Value**

name, invisibly.

---

|                  |                                        |
|------------------|----------------------------------------|
| register_palette | <i>Register a Named Colour Palette</i> |
|------------------|----------------------------------------|

---

**Description**

Adds a named palette to the highdir palette registry so it can be referenced by name wherever colours are accepted (e.g. hd\_opts(colors = "my\_palette")). This function is evaluated when loading a file in zzz.R file.

**Usage**

```
register_palette(name, colors)
```

**Arguments**

|        |                                                       |
|--------|-------------------------------------------------------|
| name   | Character. Unique palette identifier.                 |
| colors | Non-empty character vector of CSS/hex colour strings. |

## Details

Built-in palettes registered at package load time:

| Name    | Description                               |
|---------|-------------------------------------------|
| "hdir"  | Helsedirektoratet 10-colour brand palette |
| "hdir2" | 2-colour teal / purple pair               |

## Value

name, invisibly.

## Examples

```
register_palette("blues", c("#084594", "#2171b5", "#4292c6", "#6baed6"))
get_palette("blues")
```

---

|                 |                                                |
|-----------------|------------------------------------------------|
| venn_df_to_list | <i>Convert a Data Frame to a Venn Set List</i> |
|-----------------|------------------------------------------------|

---

## Description

Converts a tidy data frame with columns `id`, `name`, `value`, and `type` into the nested list format required by [hd\\_geom\\_venn\(\)](#) and [hd\\_spec\\_venn\(\)](#). This is the recommended way to build venn data from a spreadsheet, CSV, or database query without calling [hd\\_venn\\_set\(\)](#) and [hd\\_venn\\_intersect\(\)](#) row by row.

## Usage

```
venn_df_to_list(df)
```

## Arguments

|                 |                                                                                                                                                                                                      |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>df</code> | A data frame with columns <code>id</code> , <code>name</code> , <code>value</code> , <code>type</code> . Can be a plain <code>data.frame</code> , <code>data.table</code> , or <code>tibble</code> . |
|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Value

A list of set entries suitable for [hd\\_geom\\_venn\(\)](#), [hd\\_spec\\_venn\(\)](#), and [hd\\_venn\\_df\(\)](#).

### Data frame format

The input data frame must have these four columns:

- id Character. For type = "set": a single identifier such as "A". For type = "intersect": a comma-separated list of the set ids that overlap, such as "A,B" or "A,B,C". Spaces around commas are trimmed automatically.
- name Character. Human-readable label shown in the diagram. For intersections this is optional - supply NA or "" to omit it.
- value Numeric. Area or size of this region.
- type Character. Either "set" (a single circle) or "intersect" (an overlap region between two or more circles).

### Relationship to `hd_venn_set` / `hd_venn_intersect`

Each row is converted by the corresponding constructor:

- type = "set" calls `hd_venn_set()`
- type = "intersect" calls `hd_venn_intersect()` where `ids` is the comma-split vector from `id`.

The result is identical to building the list by hand with those functions.

### See Also

[hd\\_venn\\_df\(\)](#), [hd\\_spec\\_venn\(\)](#), [hd\\_geom\\_venn\(\)](#)

### Examples

```
venn_df <- data.frame(
  type = c("set", "set", "intersect"),
  id    = c("A", "B", "A,B"),
  name  = c("Oslo", "Bergen", "Both"),
  value = c(120, 95, 40)
)

sets <- venn_df_to_list(venn_df)
# Equivalent to:
# list(
#   hd_venn_set("A", "Oslo", 120),
#   hd_venn_set("B", "Bergen", 95),
#   hd_venn_intersect(c("A", "B"), 40, name = "Both")
# )
```

# Index

- \* **Geoms**
  - hd\_geom\_ranked\_bar, 16
- \* **datasets**
  - alco1, 4
  - alco2, 4
- +.hd, 3
- alco1, 4
- alco2, 4
- geom\_args, 5
- geom\_args(), 18
- get\_palette, 5
- gg\_theme, 6
- ggplot2::ggsave(), 28
- hd, 7
- hd(), 3, 12, 14, 15, 18, 20, 34, 38
- hd\_add\_js, 9
- hd\_app, 10
- hd\_app(), 24
- hd\_geom\_arearange, 11
- hd\_geom\_arearange(), 8, 17
- hd\_geom\_column, 12
- hd\_geom\_column(), 8, 17
- hd\_geom\_line, 13
- hd\_geom\_line(), 8, 17
- hd\_geom\_pie, 15
- hd\_geom\_ranked\_bar, 16
- hd\_geom\_scatter, 18
- hd\_geom\_stacked\_column, 18
- hd\_geom\_venn, 20
- hd\_geom\_venn(), 33–36, 41, 42
- hd\_make, 22
- hd\_make(), 5, 8, 11, 12, 14–18, 20, 25–32, 34, 36, 38
- hd\_opts, 23, 25
- hd\_opts(), 8, 11, 17, 24, 29, 30, 32
- hd\_reset\_theme, 27
- hd\_save, 28
- hd\_save(), 11, 24
- hd\_set\_theme, 29
- hd\_set\_theme(), 24, 26, 27, 32
- hd\_spec, 23, 30
- hd\_spec(), 8, 11, 23, 24, 26, 27, 31
- hd\_spec\_venn, 31
- hd\_spec\_venn(), 34, 36, 37, 41, 42
- hd\_theme, 32
- hd\_theme(), 6, 26
- hd\_venn\_df, 33
- hd\_venn\_df(), 21, 41, 42
- hd\_venn\_intersect, 35
- hd\_venn\_intersect(), 20, 21, 31, 32, 41, 42
- hd\_venn\_set, 36
- hd\_venn\_set(), 20, 21, 31, 32, 35, 41, 42
- hd\_venn\_sets\_from\_spec, 36
- hd\_venn\_sets\_from\_spec(), 32
- highcharter::hc\_add\_dependency(), 9
- highcharter::hc\_theme(), 32
- list\_geoms, 37
- list\_geoms(), 23, 24
- list\_modes, 37
- list\_modes(), 8, 23, 24
- list\_palettes, 38
- list\_palettes(), 6
- print.hd, 38
- register\_backend, 39
- register\_backend(), 8, 23
- register\_geom, 39
- register\_geom(), 23
- register\_palette, 40
- register\_palette(), 29
- venn\_df\_to\_list, 41
- venn\_df\_to\_list(), 21, 33, 34