

Package ‘orbis’

August 5, 2026

Title Interactive and High-Resolution Layered Graphics with Built-in World Maps

Version 0.1.0

Description A layered grammar of graphics that compiles plots to a resolution-independent scene description and renders it through two back-ends: a self-contained SVG writer with embedded 'JavaScript' for interactive figures (tooltips, hover highlighting, zoom, pan and legend toggling) and R's own graphics devices for publication-quality output at any resolution. Geographic layers are first class: a simplified world polygon dataset ships with the package and can be drawn with several map projections, including Robinson, Equal Earth and an orthographic globe. The layered grammar follows Wickham (2010) <[doi:10.1198/jcgs.2009.07098](https://doi.org/10.1198/jcgs.2009.07098)>; projections follow Snyder (1987) <[doi:10.3133/pp1395](https://doi.org/10.3133/pp1395)> and, for Equal Earth, Savric, Patterson and Jenny (2019) <[doi:10.1080/13658816.2018.1504949](https://doi.org/10.1080/13658816.2018.1504949)>; line simplification uses Douglas and Peucker (1973) <[doi:10.3138/FM57-6770-U75U-7727](https://doi.org/10.3138/FM57-6770-U75U-7727)>; the default colour scales follow the guidance on perceptually uniform palettes of Crameri, Shephard and Heron (2020) <[doi:10.1038/s41467-020-19160-7](https://doi.org/10.1038/s41467-020-19160-7)>.

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

Depends R (>= 4.1.0)

Imports grDevices, grid, stats, utils, htmltools

Suggests testthat (>= 3.0.0), knitr, rmarkdown, spelling

VignetteBuilder knitr

URL <https://github.com/mqfarooqi1/orbis>,
<https://mqfarooqi1.github.io/orbis/>

BugReports <https://github.com/mqfarooqi1/orbis/issues>

Config/testthat/edition 3

LazyData true

Config/roxygen2/version 8.0.0

NeedsCompilation no
Author Muhammad Farooqi [aut, cre] (ORCID:
 <<https://orcid.org/0000-0003-4918-9791>>)
Maintainer Muhammad Farooqi <mqfarooqi@gmail.com>
Repository CRAN
Date/Publication 2026-08-05 07:40:02 UTC

Contents

<code>+.orb_spec</code>	2
<code>orb</code>	3
<code>orb_area</code>	4
<code>orb_bars</code>	5
<code>orb_coord_cartesian</code>	6
<code>orb_coord_map</code>	6
<code>orb_draw</code>	7
<code>orb_facet</code>	8
<code>orb_geo_points</code>	9
<code>orb_interactive</code>	10
<code>orb_labs</code>	11
<code>orb_line</code>	12
<code>orb_map</code>	13
<code>orb_options</code>	14
<code>orb_palettes</code>	15
<code>orb_points</code>	15
<code>orb_save</code>	16
<code>orb_scale</code>	17
<code>orb_svg</code>	18
<code>orb_text</code>	19
<code>orb_theme</code>	19
<code>orb_worldmap</code>	20
<code>world_map</code>	21
Index	23

<code>+.orb_spec</code>	<i>Add layers, scales, labels and themes to a plot</i>
-------------------------	--

Description

The `+` operator assembles a plot from parts, in the tradition of a layered grammar of graphics.

Usage

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

Arguments

- e1 An orb_spec object.
- e2 A layer, scale, theme, labels or coordinate object to add.

Value

An orb_spec object.

Examples

```
orb(mtcars, x = wt, y = mpg) + orb_points() + orb_theme_dark()
```

orb	<i>Start a plot</i>
-----	---------------------

Description

orb() creates a plot specification: it records the data and the mapping from variables to visual channels. Layers, labels, scales and themes are added with +. Nothing is drawn until the plot is printed or saved, so a specification is cheap to build and modify.

Usage

```
orb(data, ...)
```

Arguments

- data A data frame.
- ... Aesthetic mappings, given as channel = column.

Details

Mappings are given as bare column names, in the style of a formula-free grammar: orb(df, x = wt, y = mpg, colour = cyl). The recognised channels are x, y, colour (or color), fill, size, label, group and tooltip.

Value

An object of class orb_spec.

See Also

[orb_points\(\)](#), [orb_line\(\)](#), [orb_bars\(\)](#), [orb_area\(\)](#), [orb_map\(\)](#), [orb_save\(\)](#), [orb_interactive\(\)](#)

Examples

```
p <- orb(mtcars, x = wt, y = mpg, colour = cyl) + orb_points()
p
```

orb_area	<i>Area layer</i>
----------	-------------------

Description

Fills the region between a line and a baseline.

Usage

```
orb_area(
  ...,
  data = NULL,
  fill = NULL,
  alpha = 0.45,
  baseline = 0,
  smooth = FALSE
)
```

Arguments

...	Aesthetic mappings for this layer only, as <code>channel = column</code> .
data	Optional data frame overriding the plot data.
fill	Fixed fill used when the <code>fill</code> channel is not mapped.
alpha	Opacity between 0 and 1.
baseline	Value the area is filled down to (default 0).
smooth	Draw a smoothed upper boundary. Default FALSE.

Value

An object of class `orb_layer`.

Examples

```
df <- data.frame(t = 1:40, v = abs(cumsum(rnorm(40))))
orb(df, x = t, y = v) + orb_area(alpha = 0.5) + orb_line()
```

orb_bars	<i>Bar layer</i>
----------	------------------

Description

Draws one bar per observation. `x` may be discrete (a factor or character column) or continuous. Bars start at zero.

Usage

```
orb_bars(
  ...,
  data = NULL,
  width = 0.72,
  fill = NULL,
  colour = NULL,
  alpha = 0.95,
  horizontal = FALSE
)
```

Arguments

<code>...</code>	Aesthetic mappings for this layer only, as <code>channel = column</code> .
<code>data</code>	Optional data frame overriding the plot data.
<code>width</code>	Bar width as a fraction of the available slot (0-1).
<code>fill</code>	Fixed fill used when the fill channel is not mapped.
<code>colour</code>	Fixed colour used when the colour channel is not mapped.
<code>alpha</code>	Opacity between 0 and 1.
<code>horizontal</code>	Draw horizontal bars. Default FALSE.

Value

An object of class `orb_layer`.

Examples

```
df <- data.frame(g = c("a", "b", "c"), v = c(3, 7, 5))
orb(df, x = g, y = v, fill = g) + orb_bars()
```

orb_coord_cartesian	<i>Cartesian coordinates</i>
---------------------	------------------------------

Description

The default coordinate system. Use it to return to Cartesian coordinates or to force a fixed aspect ratio.

Usage

```
orb_coord_cartesian(ratio = NULL)
```

Arguments

ratio	Data units per y unit relative to x, or NULL for free.
-------	--

Value

An object of class orb_coord.

Examples

```
orb(mtcars, x = wt, y = mpg) + orb_points() + orb_coord_cartesian()
```

orb_coord_map	<i>Map projections</i>
---------------	------------------------

Description

Sets the coordinate system to a geographic projection. Longitude and latitude are projected before being mapped to the panel, so map layers and point layers stay aligned.

Usage

```
orb_coord_map(projection = "equirectangular", centre = c(0, 0), max_lat = 84)
```

Arguments

projection	Projection name (see Details).
centre	Length-2 numeric vector, c(longitude, latitude), giving the centre of an orthographic view.
max_lat	Latitude at which the Mercator projection is clipped.

Details

Available projections:

`equiarectangular` Longitude and latitude used directly; simple and fast, but distorts area away from the equator.

`mercator` Conformal cylindrical projection; preserves angles, greatly exaggerates high latitudes. Latitudes are clipped at $\pm\text{max_lat}$.

`robinson` A compromise projection widely used for world maps.

`mollweide` Equal-area pseudo-cylindrical projection.

`equalearth` The Equal Earth projection of Savric, Patterson and Jenny (2019): equal-area, like Mollweide, but with continents shaped much closer to the familiar Robinson outline. A good modern default for thematic world maps, because areas are not distorted.

`orthographic` A view of the globe from infinite distance, centred on centre; the hemisphere facing away from the viewer is hidden.

Value

An object of class `orb_coord`, to be added to a plot.

References

Snyder, J. P. (1987) "Map Projections: A Working Manual." US Geological Survey Professional Paper 1395. [doi:10.3133/pp1395](https://doi.org/10.3133/pp1395)

Savric, B., Patterson, T. & Jenny, B. (2019) "The Equal Earth map projection." International Journal of Geographical Information Science 33, 454-465. [doi:10.1080/13658816.2018.1504949](https://doi.org/10.1080/13658816.2018.1504949)

Examples

```
orb_worldmap() + orb_coord_map("robinson")
orb_worldmap() + orb_coord_map("equalearth")
orb_worldmap() + orb_coord_map("orthographic", centre = c(20, 15))
```

orb_draw

Draw a plot on the current graphics device

Description

Renders through R's grid graphics, so the plot appears on whatever device is active. `orb_save()` is usually more convenient.

Usage

```
orb_draw(plot, width = 820, height = 520)
```

Arguments

plot An orb_spec object.
width, height Logical size in pixels.

Value

NULL, invisibly. Called for its side effect.

Examples

```
p <- orb(mtcars, x = wt, y = mpg) + orb_points()
f <- tempfile(fileext = ".png")
grDevices::png(f, width = 820, height = 520)
orb_draw(p)
grDevices::dev.off()
unlink(f)
```

orb_facet	<i>Split a plot into small multiples</i>
-----------	--

Description

Faceting draws one panel per level of a grouping variable, laid out in a grid. Comparing many small panels that share a scale is often clearer than overplotting every group in one panel.

Usage

```
orb_facet(by, ncol = NULL, scales = c("fixed", "free_x", "free_y", "free"))
```

Arguments

by Bare column name to facet by, for example orb_facet(cyl).
ncol Number of columns in the grid. The default chooses a roughly square arrangement.
scales Whether panels share their axes: "fixed" (the default, so panels are directly comparable), "free_x", "free_y" or "free".

Value

An object of class orb_facet, to be added to a plot.

Examples

```
orb(mtcars, x = wt, y = mpg) + orb_points() + orb_facet(cyl)
orb(mtcars, x = wt, y = mpg) + orb_points() +
  orb_facet(cyl, ncol = 3, scales = "free")
```

orb_geo_points	<i>Point layer positioned by longitude and latitude</i>
----------------	---

Description

Places points on a map using the active projection. Use inside a plot that also has an `orb_map()` layer, mapping x to longitude and y to latitude.

Usage

```
orb_geo_points(
  ...,
  data = NULL,
  size = 4,
  colour = "#F76707",
  alpha = 0.9,
  stroke = 0.8
)
```

Arguments

<code>...</code>	Aesthetic mappings for this layer only, as <code>channel = column</code> .
<code>data</code>	Optional data frame overriding the plot data.
<code>size</code>	Point radius in pixels when size is not mapped (default 3.2).
<code>colour</code>	Fixed colour used when the colour channel is not mapped.
<code>alpha</code>	Opacity between 0 and 1.
<code>stroke</code>	Outline width in pixels.

Value

An object of class `orb_layer`.

Examples

```
cities <- data.frame(long = c(2.35, -74, 151.2), lat = c(48.86, 40.7, -33.87),
  pop = c(11, 19, 5))
orb(cities, x = long, y = lat, size = pop) +
  orb_map() + orb_points(colour = "#F76707") + orb_coord_map("robinson")
```

orb_interactive	<i>Open or write an interactive plot</i>
-----------------	--

Description

Builds a self-contained HTML document containing the plot as inline SVG plus a small embedded script. The result works offline, needs no JavaScript library, and supports tooltips, hover highlighting, mouse-wheel zoom, dragging to pan, double-click to reset, and clicking legend keys to show or hide series.

Usage

```
orb_interactive(
  plot,
  file = NULL,
  width = 820,
  height = 520,
  title = "orbis plot"
)
```

Arguments

plot	An orb_spec object.
file	Optional output path. If NULL, an object suitable for the RStudio viewer or an R Markdown document is returned instead.
width, height	Size in pixels.
title	Title of the HTML document.

Value

If file is given, the path (invisibly); otherwise an htmltools tag object.

Examples

```
p <- orb(mtcars, x = wt, y = mpg, colour = cyl) + orb_points()
f <- tempfile(fileext = ".html")
orb_interactive(p, file = f)
file.exists(f)
unlink(f)
```

orb_labs	<i>Axis, legend and plot labels</i>
----------	-------------------------------------

Description

Axis, legend and plot labels

Usage

```
orb_labs(  
  title = NULL,  
  subtitle = NULL,  
  x = NULL,  
  y = NULL,  
  colour = NULL,  
  fill = NULL,  
  size = NULL,  
  caption = NULL  
)
```

Arguments

title	Plot title.
subtitle	Plot subtitle.
x, y	Axis titles.
colour, fill, size	Legend titles for those channels.
caption	Caption shown under the plot.

Value

An object of class orb_labs, to be added to a plot.

Examples

```
orb(mtcars, x = wt, y = mpg) + orb_points() +  
  orb_labs(title = "Fuel use", x = "Weight", y = "Miles per gallon")
```

orb_line

*Line layer***Description**

Connects observations in the order they appear, optionally split by a group or colour mapping.

Usage

```
orb_line(
  ...,
  data = NULL,
  width = 1.8,
  colour = NULL,
  alpha = 1,
  smooth = FALSE
)
```

Arguments

...	Aesthetic mappings for this layer only, as <code>channel = column</code> .
data	Optional data frame overriding the plot data.
width	Line width in pixels.
colour	Fixed colour used when the colour channel is not mapped.
alpha	Opacity between 0 and 1.
smooth	Draw a smoothed curve through the points instead of straight segments (a Catmull-Rom style spline). Default FALSE.

Value

An object of class `orb_layer`.

Examples

```
df <- data.frame(t = 1:50, v = cumsum(rnorm(50)))
orb(df, x = t, y = v) + orb_line(width = 2, smooth = TRUE)
```

orb_map	<i>World map layer</i>
---------	------------------------

Description

Draws country polygons from the built-in [world_map](#) dataset, or from a data frame of your own with long, lat and group columns. The projection is set with [orb_coord_map\(\)](#); if none is given, an equirectangular projection is used.

Usage

```
orb_map(
  data = NULL,
  values = NULL,
  region_col = "region",
  value_col = NULL,
  fill = "#DEE2E6",
  colour = "#FFFFFF",
  width = 0.4,
  alpha = 1,
  palette = "viridis",
  na_fill = "#E9ECEF",
  ocean = NULL
)
```

Arguments

data	Optional polygon data frame with columns long, lat, group and (optionally) region. Defaults to world_map .
values	Optional data frame giving one value per region, used to fill the polygons.
region_col, value_col	Column names in values holding the region name and the numeric value.
fill	Fixed fill colour used when values is not supplied.
colour	Border colour.
width	Border width in pixels.
alpha	Opacity between 0 and 1.
palette	Continuous palette used for the choropleth; see orb_palettes() .
na_fill	Fill colour for regions with no value.
ocean	Optional colour for the sea drawn behind the land: a disc for the orthographic globe, the panel background otherwise.

Details

To build a choropleth, supply values: a data frame with a region column and a value column, matched to map regions by name.

Value

An object of class orb_layer.

See Also

[orb_coord_map\(\)](#), [world_map](#)

Examples

```
orb_worldmap()

vals <- data.frame(region = c("Brazil", "India", "France"), v = c(3, 9, 5))
orb(world_map) + orb_map(values = vals, region_col = "region",
                          value_col = "v") + orb_coord_map("robinson")
```

orb_options	<i>Plot-level options</i>
-------------	---------------------------

Description

Plot-level options

Usage

```
orb_options(legend = NULL, grid = NULL, interactive = NULL)
```

Arguments

- legend Should a legend be drawn when a colour, fill or size mapping is present? Default TRUE.
- grid One of "both", "x", "y" or "none".
- interactive Should the SVG output embed interactive behaviour (tooltips, hover highlighting, zoom, pan and legend toggling)? Default TRUE. Raster output is unaffected.

Value

An object of class orb_opts, to be added to a plot.

Examples

```
orb(mtcars, x = wt, y = mpg) + orb_points() + orb_options(grid = "y")
```

orb_palettes	<i>Colour palettes available in orbis</i>
--------------	---

Description

Lists the palette names accepted by `orb_scale_colour()` and `orb_scale_fill()`.

Usage

```
orb_palettes()
```

Value

A list with two character vectors: continuous (sequential and diverging ramps) and discrete (qualitative palettes).

Examples

```
orb_palettes()
```

orb_points	<i>Point layer</i>
------------	--------------------

Description

Draws one mark per observation: a scatter plot, or a bubble chart when a size mapping is present.

Usage

```
orb_points(  
  ...,  
  data = NULL,  
  size = 3.2,  
  colour = NULL,  
  fill = NULL,  
  alpha = 0.9,  
  stroke = 0.8  
)
```

Arguments

<code>...</code>	Aesthetic mappings for this layer only, as <code>channel = column</code> .
<code>data</code>	Optional data frame overriding the plot data.
<code>size</code>	Point radius in pixels when size is not mapped (default 3.2).
<code>colour</code>	Fixed colour used when the colour channel is not mapped.
<code>fill</code>	Fixed fill used when the fill channel is not mapped.
<code>alpha</code>	Opacity between 0 and 1.
<code>stroke</code>	Outline width in pixels.

Value

An object of class `orb_layer`.

Examples

```
orb(mtcars, x = wt, y = mpg, colour = cyl) + orb_points(alpha = 0.85)
```

<code>orb_save</code>	<i>Save a plot at any resolution</i>
-----------------------	--------------------------------------

Description

Writes a plot to disk. The format is taken from the file extension:

Usage

```
orb_save(plot, file, width = 820, height = 520, dpi = 300, interactive = TRUE)
```

Arguments

<code>plot</code>	An <code>orb_spec</code> object.
<code>file</code>	Output path. The extension selects the format.
<code>width, height</code>	Logical size in pixels (the drawing is laid out at this size, then rendered at dpi).
<code>dpi</code>	Resolution for raster formats. 96 is screen resolution; 300 is a common print requirement; 600 gives very high resolution output.
<code>interactive</code>	For <code>.html</code> and <code>.svg</code> , embed interactive behaviour.

Details

- `.svg` Vector output from the built-in SVG writer. Being a vector format it is resolution independent - it stays sharp at any size.
- `.html` A self-contained interactive page (see [orb_interactive\(\)](#)).
- `.pdf` Vector output through R's PDF device.
- `.png`, `.tiff`, `.jpeg` Raster output at the requested dpi; use 300 or more for print.

Value

The file path, invisibly.

See Also

[orb_svg\(\)](#), [orb_interactive\(\)](#), [orb_draw\(\)](#)

Examples

```
p <- orb(mtcars, x = wt, y = mpg, colour = cyl) + orb_points()
f <- tempfile(fileext = ".png")
orb_save(p, f, dpi = 300)
file.exists(f)
unlink(f)
```

orb_scale	<i>Scales</i>
-----------	---------------

Description

Scales control how data values are converted to visual values: which colours a variable spans, how large points become, and the limits and transformation of the axes.

Usage

```
orb_scale_colour(palette = "viridis", limits = NULL, reverse = FALSE)

orb_scale_fill(palette = "viridis", limits = NULL, reverse = FALSE)

orb_scale_size(range = c(2, 14), limits = NULL)

orb_scale_x(limits = NULL, trans = "identity", breaks = NULL, expand = 0.04)

orb_scale_y(limits = NULL, trans = "identity", breaks = NULL, expand = 0.04)
```

Arguments

palette	Palette name (see orb_palettes()) or a character vector of colours to interpolate.
limits	Numeric length-2 vector giving the data range to map, or NULL to use the observed range.
reverse	Reverse the direction of the palette.
range	Length-2 numeric vector giving the smallest and largest radius in pixels, for orb_scale_size() .
trans	Axis transformation: "identity" (default), "log10" or "sqrt".
breaks	Numeric vector of axis breaks, or NULL for automatic.
expand	Fraction of the data range added as padding at each end.

Value

An object of class `orb_scale`, to be added to a plot.

Examples

```
p <- orb(mtcars, x = wt, y = mpg, colour = hp) + orb_points()
p + orb_scale_colour("magma", reverse = TRUE)
p + orb_scale_y(limits = c(10, 35))
```

orb_svg	<i>Render a plot as SVG</i>
---------	-----------------------------

Description

Produces a standalone SVG image of a plot. SVG is a vector format, so the result is resolution independent: it stays sharp at any size or zoom level, which is what makes it suitable both for the web and for print.

Usage

```
orb_svg(plot, width = 820, height = 520, interactive = TRUE, id = NULL)
```

Arguments

plot	An <code>orb_spec</code> object.
width, height	Size in pixels.
interactive	Embed JavaScript for tooltips, hover highlighting, zoom, pan and legend toggling. Set FALSE for a static image, for example when embedding in a document that forbids scripts.
id	Element id used by the embedded script; generated if NULL.

Value

A character string containing SVG (and, if requested, a wrapping `div` with the script).

See Also

[orb_save\(\)](#), [orb_interactive\(\)](#)

Examples

```
p <- orb(mtcars, x = wt, y = mpg) + orb_points()
s <- orb_svg(p, interactive = FALSE)
substr(s, 1, 40)
```

orb_text	<i>Text layer</i>
----------	-------------------

Description

Draws a text label at each observation.

Usage

```
orb_text(..., data = NULL, size = 10, colour = NULL, dy = 0, alpha = 1)
```

Arguments

...	Aesthetic mappings for this layer only, as <code>channel = column</code> .
data	Optional data frame overriding the plot data.
size	Font size in points.
colour	Fixed colour used when the colour channel is not mapped.
dy	Vertical offset in pixels (negative moves up).
alpha	Opacity between 0 and 1.

Value

An object of class `orb_layer`.

Examples

```
df <- data.frame(x = 1:3, y = c(2, 4, 3), n = c("a", "b", "c"))
orb(df, x = x, y = y, label = n) + orb_points() + orb_text(dy = -10)
```

orb_theme	<i>Visual themes</i>
-----------	----------------------

Description

Themes control the non-data appearance of a plot: colours of the background, grid, axes and text, plus base font size. Add a theme to a plot with `+`.

Usage

```
orb_theme_light(base_size = 12, ...)

orb_theme_dark(base_size = 12, ...)

orb_theme_minimal(base_size = 12, ...)

orb_theme_ink(base_size = 12, ...)
```

Arguments

`base_size` Base font size in points (default 12).

`...` Named theme elements to override, for example `panel = "#111111"` or `grid = "#333333"`. Recognised elements are `bg`, `panel`, `grid`, `axis`, `text`, `title`, `accent`, `base_size` and `font`.

Value

An object of class `orb_theme`, to be added to a plot.

Examples

```
p <- orb(mtcars, x = wt, y = mpg) + orb_points()
p + orb_theme_dark()
p + orb_theme_light(base_size = 14)
```

orb_worldmap	<i>A ready-made world map</i>
--------------	-------------------------------

Description

A convenience wrapper that returns a complete world map plot, optionally shaded by a value per region.

Usage

```
orb_worldmap(
  values = NULL,
  region_col = "region",
  value_col = NULL,
  projection = "equirectangular",
  palette = "viridis",
  ocean = NULL,
  title = NULL
)
```

Arguments

`values`, `region_col`, `value_col` Passed to [orb_map\(\)](#) to build a choropleth.

`projection` Projection name; see [orb_coord_map\(\)](#).

`palette` Continuous palette name; see [orb_palettes\(\)](#).

`ocean` Optional sea colour; see [orb_map\(\)](#).

`title` Optional plot title.

Value

An orb_spec object.

Examples

```
orb_worldmap(projection = "orthographic")
```

world_map	<i>World country polygons</i>
-----------	-------------------------------

Description

A simplified set of world country outlines, suitable for thematic world maps at typical figure sizes. Polygon rings were simplified with the Douglas-Peucker algorithm to keep the dataset small while preserving the recognisable shape of coastlines and borders.

Usage

```
world_map
```

Format

A data frame with four columns:

long Longitude in degrees, from -180 to 180.

lat Latitude in degrees, from -90 to 90.

group Integer identifying the polygon ring a point belongs to. Points must be drawn ring by ring; a country made of several land masses has several rings.

region Country or territory name.

Details

Coordinates are unprojected longitude and latitude in degrees; use `orb_coord_map()` to choose a projection.

Source

Derived from the world database of the **maps** package, whose boundaries come from Natural Earth (<https://www.naturalearthdata.com>), released into the public domain.

References

Douglas, D. H. & Peucker, T. K. (1973) "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature." Cartographica 10, 112-122. doi:10.3138/FM576770U75U7727

Examples

```
str(world_map)
length(unique(world_map$region))
orb_worldmap()
```

Index

- * **datasets**
 - world_map, [21](#)
- + .orb_spec, [2](#)
- orb, [3](#)
- orb_area, [4](#)
- orb_area(), [3](#)
- orb_bars, [5](#)
- orb_bars(), [3](#)
- orb_coord_cartesian, [6](#)
- orb_coord_map, [6](#)
- orb_coord_map(), [13](#), [14](#), [20](#), [21](#)
- orb_draw, [7](#)
- orb_draw(), [17](#)
- orb_facet, [8](#)
- orb_geo_points, [9](#)
- orb_interactive, [10](#)
- orb_interactive(), [3](#), [16–18](#)
- orb_labs, [11](#)
- orb_line, [12](#)
- orb_line(), [3](#)
- orb_map, [13](#)
- orb_map(), [3](#), [9](#), [20](#)
- orb_options, [14](#)
- orb_palettes, [15](#)
- orb_palettes(), [13](#), [17](#), [20](#)
- orb_points, [15](#)
- orb_points(), [3](#)
- orb_save, [16](#)
- orb_save(), [3](#), [7](#), [18](#)
- orb_scale, [17](#)
- orb_scale_colour (orb_scale), [17](#)
- orb_scale_colour(), [15](#)
- orb_scale_fill (orb_scale), [17](#)
- orb_scale_fill(), [15](#)
- orb_scale_size (orb_scale), [17](#)
- orb_scale_size(), [17](#)
- orb_scale_x (orb_scale), [17](#)
- orb_scale_y (orb_scale), [17](#)
- orb_svg, [18](#)
- orb_svg(), [17](#)
- orb_text, [19](#)
- orb_theme, [19](#)
- orb_theme_dark (orb_theme), [19](#)
- orb_theme_ink (orb_theme), [19](#)
- orb_theme_light (orb_theme), [19](#)
- orb_theme_minimal (orb_theme), [19](#)
- orb_worldmap, [20](#)
- world_map, [13](#), [14](#), [21](#)