Package: ridges (via r-universe)

June 1, 2026

Title Download Elevation Data For Anywhere In The World And Visualize It

Version 0.0.0.9000

Description The ridges package allows downloading topographical elevation data and visualizing it using various plots. Elevation data can be visualized as ridgeline plots and topographical heat maps.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2

Imports sf, elevatr, shiny, leaflet, leaflet.extras, osmdata, ggplot2, terra, ggridges, svglite

Depends R (>= 3.5)

LazyData true

URL http://simson.io/ridges/, https://github.com/jansim/ridges

Suggests testthat (>= 3.0.0), raster, vdiffr, withr

Config/testthat/edition 3

BugReports https://github.com/jansim/ridges/issues

Config/pak/sysreqs libabsl-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev libgdal-dev gdal-bin libgeos-dev make libharfbuzz-dev libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev zlib1g-dev

Repository https://jansim.r-universe.dev

Date/Publication 2025-02-20 15:50:08 UTC

RemoteUrl https://github.com/jansim/ridges

RemoteRef HEAD

RemoteSha 97864e0fa283a9400ec5891f96adf855a2ae50ed

Contents

`bb_wilder_kaiser` *Bounding box of Wilder Kaiser*

Description

A dataset containing the bounding box coordinates of the Wilder Kaiser mountain range.

Usage

```
bb_wilder_kaiser
```

Format

A numeric vector with 4 elements:

xmin Minimum longitude

ymin Minimum latitude

xmax Maximum longitude

ymax Maximum latitude

Source

OpenStreetMap data

calculate_ridgelines *Calculate ridgeline data from elevation*

Description

Takes elevation data and calculates the data needed for plotting ridgelines by sampling the elevation at regular y-intervals. You will only need this function if you want to create your own custom plots from the ground up.

Usage

```
calculate_ridgelines(elevation = NULL, n_lines = 30)
```

Arguments

elevation	A raster object containing elevation data, or NULL to use last elevation
n_lines	Number of ridgelines to calculate

Value

A data frame containing the ridgeline data with columns x, y, elevation, and group

See Also

plot_ridgelines

draw_bb *Draw a bounding box on an interactive map in a web browser*

Description

Opens an interactive map interface in your web browser where you can draw a rectangular bounding box to select an area of interest. Use the rectangle tool in the top-right corner to draw the box, then click 'Submit' to return the selection.

Usage

```
draw_bb(start_place_name = NULL, terrain = TRUE, provider = NULL)
```

Arguments

start_place_name	Optional address or place name to initially center the map on. This helps you quickly navigate to your area of interest.
terrain	Logical, whether to use OpenTopoMap tiles (default: TRUE)
provider	Optional custom tile provider. If specified, overrides the terrain parameter. See leaflet::providers for available providers.

Value

A bounding box object (sf::bbox) that can be used with get_elevation()

Examples

```
## Not run:
# Open map centered on default location
bb <- draw_bb()

# Open map centered on a specific location
bb_innsbruck <- draw_bb("Innsbruck, Austria")

# Use the returned bounding box to get elevation data
elevation <- get_elevation(bb_innsbruck)

## End(Not run)
#
```

---

ele_wilder_kaiser *Elevation data of Wilder Kaiser*

Description

A dataset containing elevation data for the Wilder Kaiser mountain range.

Usage

```
ele_wilder_kaiser
```

Format

A raster object containing elevation data

Source

SRTM elevation data

`export_ridgeline_svgs` *Export individual SVG files for each ridgeline*

Description

Creates separate SVG files for each ridgeline slice, maintaining the same scaling and proportions as plot_ridgelines().

Usage

```
export_ridgeline_svgs(
  elevation = NULL,
  output_dir = "export",
  n_lines = 30,
  scale_factor = 10,
  line_color = "black",
  fill_color = NULL,
  linewidth = 0.5,
  width = 800,
  height = 200,
  closed = TRUE
)
```

Arguments

`elevation`	A raster object containing elevation data, as returned by get_elevation()
`output_dir`	Directory where SVG files will be saved (default: "export")
`n_lines`	Number of ridgelines to generate (default: 30)
`scale_factor`	Scaling factor for the height of ridgelines (default: 10)
`line_color`	Color of the ridgelines (default: "black")
`fill_color`	Color to fill closed shapes (default: same as line_color). Only used when closed = TRUE
`linewidth`	Width of the ridgelines (default: 0.5)
`width`	SVG width in pixels (default: 800)
`height`	SVG height in pixels (default: 200)
`closed`	Logical; if TRUE, creates closed shapes by adding a baseline (default: FALSE)

Value

Invisibly returns the paths to the created SVG files

get_elevation *Get elevation data for a bounding box*

Description

Retrieves elevation data for a specified geographic area using the elevatr package. The data is returned as a raster object that can be used with the plotting functions in this package.

Usage

```
get_elevation(bb = NULL, z = 9)
```

Arguments

bb	A bounding box object (sf::bbox) or NULL to use the last drawn bounding box
z	Zoom level for elevation data (1-14, higher means more detail). Default is 9, which provides a good balance between detail and download size.

Value

A raster object containing elevation data

Examples

```
# Using the included Wilder Kaiser bounding box
elevation <- get_elevation(bb_wilder_kaiser)
plot_elevation()

# Using a higher zoom level for more detail
detailed_elevation <- get_elevation(bb_wilder_kaiser, z = 12)
plot_elevation()
```

plot_contours *Plot elevation data as contour lines*

Description

Creates a visualization of elevation data using contour lines. This function provides a way to visualize terrain elevation changes through isolines.

Usage

```
plot_contours(
  elevation = NULL,
  interval = 100,
  line_color = "black",
  linewidth = 0.25,
  color_by_elevation = "none",
  low_color = "darkblue",
  high_color = "darkred",
  fill_alpha = 1
)
```

Arguments

elevation	A raster object containing elevation data, as returned by get_elevation()
interval	Elevation interval between contour lines in meters (default: 100)
line_color	Color of the contour lines (default: "black")
linewidth	Width of the contour lines (default: 0.25)
color_by_elevation	How to color elevation: "none", "lines", "fill", or "both" (default: "none")
low_color	Color for lowest elevations when coloring by elevation (default: "darkblue")
high_color	Color for highest elevations when coloring by elevation (default: "darkred")
fill_alpha	Alpha transparency for fill color (default: 1)

Value

A ggplot object that can be further customized using ggplot2 functions

Examples

```
# Create a basic contour plot
plot_contours(ele_wilder_kaiser)

# Customize contour interval and appearance
plot_contours(ele_wilder_kaiser,
  interval = 50,
  line_color = "darkred",
  linewidth = 0.1
)

# Color contours by elevation
plot_contours(ele_wilder_kaiser, color_by_elevation = "lines")

# Color fill by elevation
plot_contours(ele_wilder_kaiser, color_by_elevation = "fill")

# Color both lines and fill by elevation
plot_contours(ele_wilder_kaiser, color_by_elevation = "both")
```

plot_elevation *Plot elevation data as a heatmap*

Description

Creates a heatmap visualization of elevation data. This function provides a quick way to visualize the terrain in your area of interest.

Usage

```
plot_elevation(elevation = NULL, low_color = "darkblue", high_color = "white")
```

Arguments

elevation	A raster object containing elevation data, as returned by get_elevation()
low_color	Color for lowest elevations (default: "darkblue")
high_color	Color for highest elevations (default: "white")

Value

A ggplot object that can be further customized using ggplot2 functions

Examples

```
# Create a basic elevation plot
plot_elevation(ele_wilder_kaiser)

# Customize colors
plot_elevation(ele_wilder_kaiser, low_color = "darkgreen", high_color = "yellow")
```

plot_ridgelines *Plot elevation data as ridgelines*

Description

Creates topographical elevation ridgeline plots, inspired by the iconic Joy Division album cover.

Usage

```
plot_ridgelines(
  elevation = NULL,
  n_lines = 30,
  scale_factor = 10,
  line_color = "white",
  fill_color = "#FFFFFF1A",
  linewidth = 0.5,
  background_color = "black"
)
```

Arguments

elevation	A raster object containing elevation data, as returned by get_elevation()
n_lines	Number of ridgelines to draw (default: 30). More lines create a denser visualization but may increase plotting time.
scale_factor	Scaling factor for the height of ridgelines (default: 10). Higher values make the elevation differences more pronounced.
line_color	Color of the ridgelines (default: "white")
fill_color	Fill color below the lines (default: "#FFFFFF1A", semi-transparent white)
linewidth	Width of the ridgelines (default: 0.5)
background_color	Background color of the plot (default: "black")

Value

A ggplot object that can be further customized using ggplot2 functions

Examples

```
# Default style (white on black)
plot_ridgelines(ele_wilder_kaiser)

# No fill, just lines
plot_ridgelines(
  ele_wilder_kaiser,
  fill_color = NA,
  scale_factor = 12
)

# Classic black on white style
plot_ridgelines(
  ele_wilder_kaiser,
  line_color = "#000000",
  fill_color = "white",
  background_color = "white",
  scale_factor = 8
)
```

```
# Get creative!
plot_ridgelines(
  ele_wilder_kaiser,
  n_lines = 35,
  line_color = "#FF4081",
  fill_color = "#FF408133",
  background_color = "#1A237E"
)
```

Index