Package: ggmapcn (via r-universe)

January 14, 2025

Title Customizable China Map Visualizations

Version 0.1.2

Description A 'ggplot2' extension for visualizing China's map, offering customizable projections, boundary styles, and buffer zones for thematic maps. Suitable for spatial data analysis and enhancing map visualization with flexible styling options.

License GPL-3

Encoding UTF-8

Depends R (>= 3.5.0), ggplot2 (>= 3.4.0)

Imports sf (>= 1.0.0), dplyr (>= 1.1.0), rlang (>= 1.1.3), ggspatial (>= 1.1.8), terra (>= 1.7), tidyterra (>= 0.6.0), curl (>= 6.0.0)

URL https://rimagination.github.io/ggmapcn/

BugReports https://github.com/Rimagination/ggmapcn/issues

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.2

NeedsCompilation no

Author Liang Ren [aut, cre] (<https://orcid.org/0000-0002-2360-7900>)

Maintainer Liang Ren <rl23@mails.tsinghua.edu.cn>

Repository CRAN

Date/Publication 2025-01-14 10:10:01 UTC

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libicu-dev libjpeg-dev libpng-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Contents

`basemap_dem` *Elevation Map of China Layer for ggplot2*

Description

'basemap_dem' adds a digital elevation model (DEM) raster map of China as a layer to ggplot2. The function ensures the output map remains rectangular, regardless of the chosen projection. It supports displaying the DEM either within China's boundary or in a larger rectangular area around China. Users can provide their own DEM data using the 'data' parameter, or the default built-in DEM data will be used.

Usage

```
basemap_dem(
  data = NULL,
  crs = NULL,
  within_china = FALSE,
  maxcell = 1e+06,
  na.rm = FALSE,
  ...
)
```

Arguments

`data`	Optional. A 'terra' raster object for custom DEM data.
`crs`	Coordinate reference system (CRS) for the projection. Defaults to the CRS of the DEM data. Users can specify other CRS strings (e.g., '"EPSG:4326"' or custom projections).
`within_china`	Logical. If 'TRUE', displays only the DEM within China's boundary. If 'FALSE', displays the DEM for a larger rectangular area around China. Default is 'FALSE'.
`maxcell`	Maximum number of cells for rendering (to improve performance). Defaults to '1e6'.
`na.rm`	Logical. If 'TRUE', removes missing values. Default is 'FALSE'.
`...`	Additional parameters passed to 'geom_spatraster'.

Value

A 'ggplot' object containing the elevation map of China as a layer, which can be further customized or plotted.

See Also

geom_boundary_cn

Examples

```
# Before using the basemap_dem function, make sure the required data files are available.
# The required files are: "gebco_2024_China.tif" and "China_mask.gpkg".
# You can use check_geodata() to download them from GitHub if they are not available locally.

# Check and download the required data files if they are missing
check_geodata(files = c("gebco_2024_China.tif", "China_mask.gpkg"))

# Define the CRS for China (EPSG:4326 is a common global geographic coordinate system)
china_proj <- "+proj=aeqd +lat_0=35 +lon_0=105 +ellps=WGS84 +units=m +no_defs"

# Example 1: Display full rectangular area around China using built-in DEM data
ggplot() +
  basemap_dem(within_china = FALSE) +
  tidyterra::scale_fill_hypso_tint_c(
    palette = "gmt_globe",
    breaks = c(-10000, -5000, 0, 2000, 5000, 8000)
  ) +
  theme_minimal()

# Example 2: Display only China's DEM and boundaries using built-in DEM data
ggplot() +
  basemap_dem(crs = china_proj, within_china = TRUE) +
  geom_boundary_cn(crs = china_proj) +
  tidyterra::scale_fill_hypso_c(
    palette = "dem_print",
    breaks = c(0, 2000, 4000, 6000),
    limits = c(0, 7000)
  ) +
  labs(fill = "Elevation (m)") +
  theme_minimal()
```

---

basemap_vege *Vegetation Map of China Layer for ggplot2*

Description

Adds a vegetation raster map of China to a ggplot2 plot, with color-coded vegetation types.

Usage

```
basemap_vege(
  color_table = NULL,
  crs = NULL,
  maxcell = 1e+06,
  use_coltab = TRUE,
  na.rm = FALSE,
  ...
)
```

Arguments

color_table	A data frame containing vegetation types and their corresponding colors. It should have columns "code" (raster values), "type" (vegetation names), and "col" (hex color codes). If NULL, a default color table based on standard vegetation classifications for China is used.
crs	A character string specifying the coordinate reference system for the projection. If NULL, the default projection "+proj=aeqd +lat_0=35 +lon_0=105 +ellps=WGS84 +units=m +no_defs" is applied.
maxcell	An integer indicating the maximum number of cells for rendering to improve performance. Defaults to 1e6.
use_coltab	A logical value indicating whether to use the color table for raster values. Default is TRUE.
na.rm	A logical value indicating whether to remove missing values. Default is FALSE.
...	Additional parameters passed to ‘geom_spatraster‘.

Value

A ggplot2 layer object representing the vegetation map of China.

References

Zhang X, Sun S, Yong S, et al. (2007). *Vegetation map of the People’s Republic of China (1:1000000)*. Geology Publishing House, Beijing.

Examples

```
# Example1: Check and load the vegetation raster map

# Make sure the required raster data is available
check_geodata(files = c("vege_1km_projected.tif"))

# Once the data is checked or downloaded, add the vegetation raster to a ggplot
ggplot() +
  basemap_vege() +
  theme_minimal()

# Example2: Customize color table
```

```
custom_colors <- data.frame(
  code = 0:11,
  type = c(
    "Non-vegetated", "Needleleaf forest", "Needleleaf and broadleaf mixed forest",
    "Broadleaf forest", "Scrub", "Desert", "Steppe", "Grassland",
    "Meadow", "Swamp", "Alpine vegetation", "Cultivated vegetation"
  ),
  col = c(
    "#8D99B3", "#97B555", "#34BF36", "#9ACE30", "#2EC6C9", "#E5CE0E",
    "#5BB1ED", "#6494EF", "#7AB9CB", "#D97A80", "#B87701", "#FEB780"
  )
)
ggplot() +
  basemap_vege(color_table = custom_colors) +
  labs(fill = "Vegetation type group") +
  theme_minimal()
```

---

check_geodata *Check and Download Geospatial Data from GitHub*

Description

Checks if the required geospatial data files are present in the package's 'inst/extdata' directory. If the files are missing, they are downloaded from a specified GitHub repository. The user can specify which files to download, or download all available files by default.

Usage

```
check_geodata(files = NULL, overwrite = FALSE, quiet = FALSE, max_retries = 3)
```

Arguments

files	Character vector. The names of the data files to download. If 'NULL', all available files will be downloaded.
overwrite	Logical. Whether to overwrite existing files in 'inst/extdata/'. Default is 'FALSE'.
quiet	Logical. If 'TRUE', suppresses message outputs when files already exist. Default is 'FALSE'.
max_retries	Integer. The maximum number of retries if a download fails. Default is '3'.

Details

This function uses the 'curl' package to download files, and supports displaying a progress bar during the download process. It is particularly useful for large files or when the default download method ('download.file') is slow or unreliable.

Value

A character vector of file paths to the downloaded or existing files.

Examples

```
# Check for and download all geospatial data from GitHub
file_paths <- check_geodata()

# Check and download specific files
file_paths <- check_geodata(files = c("boundary.rda"))
```

`coord_proj` *Coordinate System with Transformed Limits for Custom Projections*

Description

'coord_proj' is a wrapper around `ggplot2::coord_sf()`. It simplifies specifying map limits ('xlim', 'ylim') in longitude and latitude (WGS84 CRS) and automatically transforms them into the specified CRS for accurate projections.

This function extends the functionality of `coord_sf()` to seamlessly handle user-specified geographic boundaries in any projection, ensuring accurate mapping.

Usage

```
coord_proj(
  crs = NULL,
  xlim = NULL,
  ylim = NULL,
  expand = TRUE,
  default_crs = "EPSG:4326",
  ...
)
```

Arguments

`crs`	A character string specifying the coordinate reference system (CRS) for the projection (e.g., '"EPSG:4326"' or custom projections like '"+proj=merc"').
`xlim`	Longitude range (in degrees) to display, as a numeric vector of length 2.
`ylim`	Latitude range (in degrees) to display, as a numeric vector of length 2.
`expand`	Logical, whether to expand the plot limits. Default is 'TRUE'.
`default_crs`	A character string specifying the CRS of the input 'xlim' and 'ylim'. Default is '"EPSG:4326"'.
`...`	Additional arguments passed to `ggplot2::coord_sf()`.

Value

A ggplot2 `coord_sf` object with the transformed limits.

See Also

ggplot2::coord_sf, geom_world

Examples

```
# World map with default projection and limits
ggplot() +
  geom_world() +
  coord_proj(
    crs = "+proj=longlat +datum=WGS84",
    xlim = c(-180, 180),
    ylim = c(-90, 90),
    expand=FALSE
  ) +
  theme_minimal()

# Focused view with Azimuthal Equidistant projection
china_proj <- "+proj=aeqd +lat_0=35 +lon_0=105 +ellps=WGS84 +units=m +no_defs"
ggplot() +
  geom_world(fill = "lightblue") +
  coord_proj(
    crs = china_proj,
    xlim = c(60, 140),
    ylim = c(-10, 50)
  ) +
  theme_minimal()

# Display a small map of the South China Sea Islands with a custom projection
ggplot() +
  geom_boundary_cn() +
  theme_bw() +
  coord_proj(
    crs = china_proj,
    expand = FALSE,
    xlim = c(105, 123),
    ylim = c(2, 23)
  )
```

---

geom_boundary_cn *Plot Boundaries of China*

Description

Draws various types of boundaries for China. Each boundary type can be customized in terms of color, line width, and line type. This function also allows optional addition of a compass and a scale bar.

Usage

```
geom_boundary_cn(
  crs = "+proj=aeqd +lat_0=35 +lon_0=105 +ellps=WGS84 +units=m +no_defs",
  compass = FALSE,
  scale = FALSE,
  mainland_color = "black",
  mainland_size = 0.5,
  mainland_linetype = "solid",
  coastline_color = "blue",
  coastline_size = 0.3,
  coastline_linetype = "solid",
  ten_segment_line_color = "black",
  ten_segment_line_size = 0.5,
  ten_segment_line_linetype = "solid",
  SAR_boundary_color = "grey",
  SAR_boundary_size = 0.5,
  SAR_boundary_linetype = "dashed",
  undefined_boundary_color = "black",
  undefined_boundary_size = 0.5,
  undefined_boundary_linetype = "longdash",
  province_color = "transparent",
  province_size = 0.3,
  province_linetype = "solid",
  ...
)
```

Arguments

crs	Character. Coordinate reference system (CRS) for the projection. Defaults to ‘"+proj=aeqd +lat_0=35 +lon_0=105 +ellps=WGS84 +units=m +no_defs"‘. Users can specify other CRS strings to customize the projection (e.g., ‘"+proj=merc"‘ for Mercator).
compass	Logical. Whether to display a compass (north arrow). Default is ‘FALSE‘. If set to ‘TRUE‘, a default compass (north arrow) with ‘ggspatial::north_arrow_fancy_orienteering()‘ will be added to the top-left corner. To customize the compass, use ‘ggspatial::annotation_north_arrow()‘ directly.
scale	Logical. Whether to display a scale bar. Default is ‘FALSE‘. If set to ‘TRUE‘, a default scale bar with ‘ggspatial::annotation_scale()‘ will be added to the bottom-left corner. To customize the scale bar, use ‘ggspatial::annotation_scale()‘ directly.
mainland_color	Character. Color for the mainland boundary. Default is ‘"black"‘.
mainland_size	Numeric. Line width for the mainland boundary. Default is ‘0.5‘.
mainland_linetype	Character. Line type for the mainland boundary. Default is ‘"solid"‘.
coastline_color	Character. Color for the coastline. Default is ‘"blue"‘.

`coastline_size` Numeric. Line width for the coastline. Default is '0.3'.

`coastline_linetype`
: Character. Line type for the coastline. Default is '"solid"'.

`ten_segment_line_color`
: Character. Color for the ten-segment line. Default is '"black"'.

`ten_segment_line_size`
: Numeric. Line width for the ten-segment line. Default is '0.5'.

`ten_segment_line_linetype`
: Character. Line type for the ten-segment line. Default is '"solid"'.

`SAR_boundary_color`
: Character. Color for the SAR boundary. Default is '"grey"'.

`SAR_boundary_size`
: Numeric. Line width for the SAR boundary. Default is '0.5'.

`SAR_boundary_linetype`
: Character. Line type for the SAR boundary. Default is '"dashed"'.

`undefined_boundary_color`
: Character. Color for the undefined boundary. Default is '"black"'.

`undefined_boundary_size`
: Numeric. Line width for the undefined boundary. Default is '0.5'.

`undefined_boundary_linetype`
: Character. Line type for the undefined boundary. Default is '"longdash"'.

`province_color` Character. Color for the provincial boundaries. Default is '"transparent"'.

`province_size` Numeric. Line width for the provincial boundaries. Default is '0.3'.

`province_linetype`
: Character. Line type for the provincial boundaries. Default is '"solid"'.

`...` Additional parameters passed to 'geom_sf'.

Value

A list of ggplot2 layers representing China's multi-segment boundaries with the specified styles. Optionally includes a compass (north arrow) and a scale bar, depending on the 'compass' and 'scale' arguments.

Examples

```
# Plot China's boundaries with default settings
ggplot() +
  geom_boundary_cn() +
  theme_minimal()

# Plot China's boundaries with a compass and scale bar
ggplot() +
  geom_boundary_cn(compass = TRUE, scale = TRUE) +
  theme_minimal()

# For customized compass or scale bar, use ggspatial directly:
ggplot() +
```

```
  geom_boundary_cn() +
  ggspatial::annotation_north_arrow(
    location = "br", style = ggspatial::north_arrow_minimal()
  ) +
  ggspatial::annotation_scale(
    location = "tr", width_hint = 0.3
  ) +
  theme_minimal()
```

`geom_buffer_cn` *Plot Buffered Layers for China's Boundary*

Description

Creates a ggplot2 layer for displaying buffered areas around China's boundaries, including both the mainland boundary and the ten-segment line. Buffers with user-defined distances are generated around each boundary, providing flexibility in projection and appearance.

Usage

```
geom_buffer_cn(
  mainland_dist = 20000,
  ten_line_dist = NULL,
  crs = "+proj=aeqd +lat_0=35 +lon_0=105 +ellps=WGS84 +units=m +no_defs",
  color = NA,
  fill = "#D2D5EB",
  ...
)
```

Arguments

`mainland_dist`	Numeric. The buffer distance (in meters) for the mainland boundary.
`ten_line_dist`	Numeric. The buffer distance (in meters) for each segment of the ten-segment line. If not specified, it defaults to the same value as 'mainland_dist'.
`crs`	Character. The coordinate reference system (CRS) for the projection. Defaults to "+proj=aeqd +lat_0=35 +lon_0=105 +ellps=WGS84 +units=m +no_defs". Users can specify other CRS strings (e.g., "+proj=merc" for Mercator).
`color`	Character. The border color for the buffer area. Default is 'NA' (transparent).
`fill`	Character. The fill color for the buffer area. Default is '"#D2D5EB"'.
`...`	Additional parameters passed to 'geom_sf'.

Value

A ggplot2 layer displaying buffered areas around China's boundaries, with customizable buffer distances for the mainland boundary and the ten-segment line, using the specified projection.

Examples

```
# Plot buffers with specified distances for mainland and ten-segment line
ggplot() +
  geom_buffer_cn(
    mainland_dist = 10000,
    ten_line_dist = 5000
  ) +
  theme_minimal()
```

`geom_loc` *Visualize Spatial Point Data*

Description

'geom_loc' is a wrapper around `ggplot2::geom_sf()` designed for visualizing spatial point data. It supports both `sf` objects and tabular data frames with longitude and latitude columns, automatically transforming them into the specified coordinate reference system (CRS).

Usage

```
geom_loc(
  data,
  lon = NULL,
  lat = NULL,
  crs = "+proj=aeqd +lat_0=35 +lon_0=105 +ellps=WGS84 +units=m +no_defs",
  mapping = ggplot2::aes(),
  ...
)
```

Arguments

`data`	A data frame, tibble, or `sf` object containing spatial point data.
`lon`	A character string. The name of the longitude column in `data` (required if `data` is tabular).
`lat`	A character string. The name of the latitude column in `data` (required if `data` is tabular).
`crs`	A character string. The target coordinate reference system (CRS) for the data. Defaults to `"+proj=aeqd +lat_0=35 +lon_0=105 +ellps=WGS84 +units=m +no_defs"`.
`mapping`	Aesthetic mappings created by `ggplot2::aes()`, such as `color` or `size`.
`...`	Additional parameters passed to `ggplot2::geom_sf()`, such as `size`, `alpha`, or `color`.

Details

This function simplifies the process of visualizing spatial data in ggplot2 by automatically handling CRS transformations and providing an interface for both `sf` and tabular data. If the input is a tabular data frame, it will be converted to an `sf` object using the specified longitude and latitude columns.

See `ggplot2::geom_sf()` for details on additional parameters and aesthetics.

Value

A ggplot2 layer for visualizing spatial point data, either from an ‘sf‘ object or a tabular data frame with longitude and latitude columns, after transforming the data to the specified coordinate reference system (CRS).

See Also

`geom_boundary_cn`

Examples

```
# Generate a random dataset with latitude and longitude
set.seed(123)
data_sim <- data.frame(
  Longitude = runif(100, 80, 120),
  Latitude = runif(100, 28, 40),
  Category = sample(c("Type A", "Type B", "Type C"), 100, replace = TRUE)
)

# Visualize the data with China's boundaries
ggplot() +
  geom_boundary_cn() +
  geom_loc(
    data = data_sim, lon = "Longitude", lat = "Latitude",
    mapping = aes(color = Category), size = 1, alpha = 0.7
  ) +
  theme_minimal()
```

---

geom_mapcn *Plot China Map with Customizable Options*

Description

‘geom_mapcn‘ provides a flexible interface for visualizing China’s administrative boundaries. Users can select administrative levels (province, city, or county), apply custom projections, and filter specific regions.

Usage

```
geom_mapcn(
  data = NULL,
  admin_level = "province",
  crs = "+proj=aeqd +lat_0=35 +lon_0=105 +ellps=WGS84 +units=m +no_defs",
  color = "black",
  fill = "white",
  linewidth = 0.5,
  filter_attribute = NULL,
  filter = NULL,
  ...
)
```

Arguments

data	An 'sf' object containing China's map data. If 'NULL', the function loads the package's default map. Users can select provincial, municipal, or county-level maps using the 'admin_level' parameter.
admin_level	A character string specifying the administrative level of the map. Options are '"province"' (default), '"city"', or '"county"'. The corresponding '.rda' files ('China_sheng.rda', 'China_shi.rda', 'China_xian.rda') must be located in the package's 'extdata' folder.
crs	A string specifying the Coordinate Reference System (CRS). Defaults to '"+proj=aeqd +lat_0=35 +lon_0=105 +ellps=WGS84 +units=m +no_defs"'. Users can specify other CRS strings (e.g., '"EPSG:4326"').
color	Border color. Default is '"black"'.
fill	Fill color. Default is '"white"'.
linewidth	Line width for borders. Default is '0.5'. Note: 'linewidth' is used to control the border width in 'ggplot2' version 3.3.0 or higher. Use 'size' for earlier versions.
filter_attribute	Column name for filtering regions (e.g., '"name_en"').
filter	A character vector of values to filter specific regions (e.g., 'c("Beijing", "Shanghai")').
...	Additional parameters passed to 'geom_sf'.

Value

A 'ggplot2' layer (of class 'ggproto') for visualizing China's administrative boundaries. This layer can be added to a 'ggplot' object to generate maps with customizable projections, border colors, fill colors, and more.

Examples

```
# Plot provincial map (default)
ggplot() +
  geom_mapcn() +
  theme_minimal()
```

```
# Filter specific provinces
ggplot() +
 geom_mapcn(filter_attribute = "name_en", filter = c("Beijing", "Shanghai"), fill = "red") +
  theme_minimal()

# Use a Mercator projection
ggplot() +
  geom_mapcn(crs = "+proj=merc", linewidth = 0.7) +
  theme_minimal()
```

---

geom_world *Plot World Map with Customizable Options*

Description

A wrapper around [ggplot2::geom_sf()] for visualizing world maps with customizable options. This function allows for custom projections, filtering specific countries or regions, and detailed aesthetic customizations for borders and fills.

Usage

```
geom_world(
  data = NULL,
  crs = "+proj=longlat +datum=WGS84",
  color = "black",
  fill = "white",
  linewidth = 0.5,
  filter_attribute = "SOC",
  filter = NULL,
  ...
)
```

Arguments

data	An ‘sf‘ object containing world map data. If ‘NULL‘, the default world map data from the package will be loaded from a ‘.rda‘ file.
crs	A character string specifying the target coordinate reference system (CRS) for the map projection. Defaults to ‘"+proj=longlat +datum=WGS84"‘.
color	A character string specifying the border color for administrative boundaries. Default is ‘"black"‘.
fill	A character string specifying the fill color for administrative areas. Default is ‘"white"‘.
linewidth	A numeric value specifying the line width for administrative boundaries. Default is ‘0.5‘.

`filter_attribute`
: A character string specifying the column name used for filtering countries or regions. Default is '"SOC"', which refers to the ISO 3166-1 alpha-3 country code in the default dataset.

`filter`
: A character vector specifying the values to filter specific countries or regions. Default is 'NULL'.

`...`
: Additional parameters passed to [ggplot2::geom_sf()], such as 'size', 'alpha', or 'lty'.

Details

This function simplifies the process of creating world maps by combining the functionality of 'geom_sf' with user-friendly options for projections, filtering, and custom styling. Key features include: - **Custom projections**: Easily apply any CRS to the map. - **Filtering by attributes**: Quickly focus on specific countries or regions. - **Flexible aesthetics**: Customize fill, borders, transparency, and other visual properties.

Value

A 'ggplot2' layer for world map visualization.

See Also

[ggplot2::geom_sf()], [sf::st_transform()], [sf::st_read()]

Examples

```
# Plot the default world map
ggplot() +
  geom_world() +
  theme_minimal()

# Using Robinson projection with central meridian at 0°
ggplot() +
  geom_world(crs = "+proj=robin +lon_0=0 +x_0=0 +y_0=0 +datum=WGS84 +units=m") +
  theme_minimal()

# Filter specific countries (e.g., China and its neighbors)
china_neighbors <- c("CHN", "AFG", "BTN", "MMR", "LAO", "NPL", "PRK", "KOR",
                     "KAZ", "KGZ", "MNG", "IND", "BGD", "TJK", "PAK", "LKA", "VNM")
ggplot() +
  geom_world(filter = china_neighbors) +
  theme_minimal()

# Background map + Highlight specific region
ggplot() +
  geom_world(fill = "gray80", color = "gray50", alpha = 0.5) +
  geom_world(filter = c("CHN"), fill = "red", color = "black", linewidth = 1.5) +
  theme_minimal()

# Customize styles with transparency and bold borders
```

```
ggplot() +
  geom_world(fill = "lightblue", color = "darkblue", linewidth = 1, alpha = 0.8) +
  theme_void()
```

Index