Package 'greenSD'

August 23, 2026

Type Package

Title Access and Analyze Global GreenSpace Spatial Data

Version 0.2.2

Description Access and analyze multi-band greenspace seasonality data cubes (available for 1,028 major global cities), global Normalized Difference Vegetation Index / land cover data from the European Space Agency WorldCover 10m Dataset, and Sentinel-2-l2a images. Users can download data using bounding boxes, city names, and filter by year or seasonal time window. The package also supports calculating human exposure to greenspace using a population-weighted greenspace exposure model introduced by Chen et al. (2022) <doi:10.1038/s41467-022-32258-4> based on Global Human Settlement Layer population data, and calculating a set of greenspace morphology metrics at patch and landscape levels.

URL https://github.com/billbillbilly/greenSD, https://billbillbilly.github.io/greenSD/

BugReports https://github.com/billbillbilly/greenSD/issues

License GPL-3

Encoding UTF-8

Depends R (>= 4.1)

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Imports terra, maptiles, sf, future, furrr, purrr, nominatimlite, utils, cli, dsmSearch, rstac, landscapemetrics, magick, aws.s3, dplyr, tidyr, stringr, tibble, grDevices, rlang

RoxygenNote 7.3.3

NeedsCompilation no

Author Xiaohao Yang [aut, cre, cph]

Maintainer Xiaohao Yang <xiaohaoy111@gmail.com>

Repository CRAN

Date/Publication 2026-08-23 03:50:02 UTC

Contents

`check_available_urban` *Get all of the urban areas in the Greenspace Seasonality Data Cube*

Description

This function returns all of the urban areas in the Greenspace Seasonality Data Cube dataset.

Usage

```
check_available_urban(test = FALSE)
```

Arguments

`test`	logical. (ignored) Only for testing.

Value

dataframe

Note

You can explore all available urban areas in an interacive map at: https://github.com/billbillbilly/greenSD/blob/main/scripts/city_urban_boundaries.geojson

References

Wu, S., Song, Y., An, J. et al. High-resolution greenspace dynamic data cube from Sentinel-2 satellites over 1028 global major cities. Sci Data 11, 909 (2024). https://doi.org/10.1038/s41597-024-03746-7

Examples

```
check_available_urban(test = TRUE)
```

check_urban_boundary *Get an urban area boundary based on the UID*

Description

This function returns a polygon of a city boundary based on the UID

Usage

```
check_urban_boundary(uid = NULL, plot = TRUE, test = FALSE)
```

Arguments

uid	numeric. Urban area ID. To check the ID of an available urban area, use check_available_urban()
plot	logical. Whether to plot city boundary
test	logical. (ignored) Only for testing.

Value

sf

References

Wu, S., Song, Y., An, J. et al. High-resolution greenspace dynamic data cube from Sentinel-2 satellites over 1028 global major cities. Sci Data 11, 909 (2024). https://doi.org/10.1038/s41597-024-03746-7

Examples

```
check_urban_boundary(test = TRUE)
```

compute_exposure *exposure*

Description

Computes population-weighted greenspace fraction or human exposure to greenspace based on a population-weighted exposure model (Chen et al., 2022), using population data from the Global Human Settlement Layer (GHSL; Pesaresi et al., 2024). See **Details** for the underlying method and assumptions.

Usage

```
compute_exposure(
  r = NULL,
  res = c(10, 10),
  pop_year = 2020,
  radius = 500,
  grid_size = NULL,
  height = FALSE,
  pop_out = FALSE,
  quiet = TRUE
)
```

Arguments

r	A SpatRaster with single/multiple greenspace layer(s), either fractional or binary (where non-green = 0 and green = 1), typically the output from get_gsdc(), get_esa_wc(), or get_tile_green().
res	numeric vector of length 2. The actual spatial resolution (in meters). Default is c(10, 10).
pop_year	numeric. Year of the GHSL dataset to use. Must be one of: 2015, 2020, 2025, or 2030. Default is 2020.
radius	numeric. Buffer radius (in meters) used for local averaging. Default is 500.
grid_size	numeric. Optional. If provided, output is aggregated to grid cells of this size (in meters) and returned as an sf object.
height	logical. Whether to compute greenspace volume for population-weighted greenspace fraction or human exposure to greenspace using Meta's global canopy height map (Tolan et al., 2024). (The default is FALSE)
pop_out	logical. Whether return population layer.
quiet	logical. Whether show progress bars for some process.

Details

This function implements the population-weighted greenspace exposure (PWGE) model:

1. Start with a population raster. Each pixel \(i \) has a population value \(P_i \).
2. Create a circular buffer of radius \(d \) around each pixel center.
3. For each buffer, calculate greenspace fraction:

$$G_i^d = \frac{\text{Area of greenspace within buffer}}{\text{Total buffer area}}$$

4. Repeat for all \(i = 1, 2, ..., N \) grid cells.
5. Compute overall exposure:

$$GE^d = \frac{\sum_i P_i \cdot G_i^d}{\sum_i P_i}$$

Value

SpatRaster or sf. A `SpatRaster` (if `grid_size` is `NULL`) with layers `pwgf_*`, or an `sf` object with columns `pwge_*` representing population-weighted greenspace exposure values aggregated to each grid polygon.

References

Chen, B., Wu, S., Song, Y. et al. Contrasting inequality in human exposure to greenspace between cities of Global North and Global South. Nat Commun 13, 4636 (2022). https://doi.org/10.1038/s41467-022-32258-4

Pesaresi, M., Schiavina, M., Politis, P., Freire, S., Krasnodębska, K., Uhl, J. H., ... Kemper, T. (2024). Advances on the Global Human Settlement Layer by joint assessment of Earth Observation and population survey data. International Journal of Digital Earth, 17(1). https://doi.org/10.1080/17538947.2024.2390454

Tolan, J., Yang, H. I., Nosarzewski, B., Couairon, G., Vo, H. V., Brandt, J., ... & Couprie, C. (2024). Very high resolution canopy height maps from RGB imagery using self-supervised vision transformer and convolutional decoder trained on aerial lidar. Remote Sensing of Environment, 300, 113888.

Examples

```
sample_data <- terra::rast(system.file("extdata", "detroit_gs.tif", package = "greenSD"))
pwgf <- compute_exposure(
  # r = sample_data,
  pop_year = 2020,
  radius = 1500
)
```

`compute_morphology` *compute_morphology*

Description

Compute greenspace morphology metrics at patch (Nowosad & Stepinski, 2019) or landscape level (see details), including average size (AREA_MN), fragmentation (PD), connectedness (COHESION), aggregation (AI), and complexity of the shape (SHAPE_AM), related to public health (Wang et al., 2024)

Usage

```
compute_morphology(r = NULL, directions = 4, grid_size = NULL, quiet = TRUE)
```

Arguments

`r`	SpatRaster. A single-band binary greenspace raster, where 0 or NA represents non-green areas and 1 represents green areas.
`directions`	numeric. The number of directions in which patches should be connected: 4 (default) or 8.

grid_size	numeric or sf polygons. (Optional) If specified, morphology metrics at grid level will be computed based on the size (in meters) of given grid cells or input (sf) polygons.
quiet	logical. Whether show progress bars for some process.

Details

To get information of metrics, please use `landscapemetrics::list_lsm()`.

Value

A SpatVector object contains indivisual patches with metrics at patch level, when `grid_size = NULL`.

A SpatVector object contains landscape-level value of metrics, when `grid_size` is not `NULL`.

References

Nowosad J., TF Stepinski. 2019. Information theory as a consistent framework for quantification and classification of landscape patterns. https://doi.org/10.1007/s10980-019-00830-x

Wang, H., & Tassinary, L. G. (2024). Association between greenspace morphology and prevalence of non-communicable diseases mediated by air pollution and physical activity. Landscape and Urban Planning, 242, 104934.

Examples

```
green <- get_tile_green(
                         # bbox = c(-83.087174,42.333373,-83.042542,42.358748),
                         provider = "esri",
                         zoom = 16)
# p <- terra::ifel(green$green == 0, NA, 1)
m <- compute_morphology(
                        #r = p
                        directions = 8)
```

`get_band_index_by_time` *Get band index based on time period*

Description

Converts a date string in `"MM-DD"` format to the corresponding band index for the Greenspace Seasonality Data Cube, which contains 36 bands representing 10-day intervals over a year.

Usage

```
get_band_index_by_time(time, year)
```

Arguments

time	Character vector of length 2. (optional) Start and end dates in "MM-DD" format (e.g., c("03-20", "10-15")). Used to subset the 10-day interval data cube by time.
year	numeric. (required) The year of interest.

Details

The Greenspace Data Cube is organized into 36 bands per year, each representing a 10-day interval. This function calculates which of those bands a given date falls into by converting the MM-DD string into the day-of-year (DOY) and dividing by 10 (rounded up).

Value

Interger. a band index.

Examples

```
get_band_index_by_time(c("03-20", "10-15"), year = 2020)
```

---

get_esa_wc *Download Land Cover or NDVI Data from ESA WorldCover or Sentinel-2 10m Land Use/Land Cover Time Series*

Description

Download 11-class land cover or 3-band NDVI data from the ESA WorldCover 10m Annual Dataset (NDVI p90, NDVI p50, NDVI p10), or 9-class annual land use/land cover (LULC) maps from the Sentinel-2 10m Land Use/Land Cover Time Series produced by Impact Observatory, accessed via Microsoft Planetary Computer. Users can define an area of interest using a bounding box or place name.

Usage

```
get_esa_wc(
  bbox = NULL,
  place = NULL,
  datatype = "landcover",
  year = 2021,
  mask = TRUE,
  quiet = TRUE
)
```

Arguments

bbox	sf, sfc, or a numeric vector (xmin, ymin, xmax, ymax) defining the area of interest. Optional if place is provided.
place	character or vector. (optional) A single line address, e.g. ("1600 Pennsylvania Ave NW, Washington") or a vector of addresses (c("Madrid", "Barcelona")).
datatype	character. One of "landcover", "ndvi", or "lulc". "landcover" and "ndvi" retrieve ESA WorldCover data (years 2020–2021). "lulc" retrieves the Sentinel-2 10m LULC Time Series (years 2017–2024).
year	numeric. The year of interest. For "landcover" and "ndvi": 2020 or 2021 (default 2021). For "lulc": an integer from 2017 to 2024.
mask	logical (optional). Default is TRUE. If TRUE, masks the raster data using the given bbox or place.
quiet	logical. Whether show progress bars for some process.

Details

The "lulc" datatype retrieves the Impact Observatory Sentinel-2 10m Land Use/Land Cover Time Series from the public AWS open data bucket (s3://io-10m-annual-lulc, no authentication required). New annual maps are released each January, so coverage currently extends to 2024. The 9 classes are: 1 = Water, 2 = Trees, 4 = Flooded vegetation, 5 = Crops, 7 = Built area, 8 = Bare ground, 9 = Snow/ice, 10 = Clouds, 11 = Rangeland.

Value

A SpatRaster object. For "landcover": 11-class ESA WorldCover land cover map. For "ndvi": NDVI yearly percentile composite (NDVI p90, NDVI p50, NDVI p10). For "lulc": 9-class Sentinel-2 annual land use/land cover classification (layer named "LULC").

References

Zanaga, D., Van De Kerchove, R., De Keersmaecker, W., Souverijns, N., Brockmann, C., Quast, R., Wevers, J., Grosu, A., Paccini, A., Vergnaud, S., Cartus, O., Santoro, M., Fritz, S., Georgieva, I., Lesiv, M., Carter, S., Herold, M., Li, L., Tsendbazar, N.-E., ... Arino, O. (2021). ESA WorldCover 10 m 2020 v100 (Version v100). Zenodo. https://doi.org/10.5281/zenodo.5571936

Zanaga, D., Van De Kerchove, R., Daems, D., De Keersmaecker, W., Brockmann, C., Kirches, G., Wevers, J., Cartus, O., Santoro, M., Fritz, S., Lesiv, M., Herold, M., Tsendbazar, N.-E., Xu, P., Ramoino, F., & Arino, O. (2022). ESA WorldCover 10 m 2021 v200 (Version v200). Zenodo. https://doi.org/10.5281/zenodo.7254221

Karra, K., Kontgis, C., Statman-Weil, Z., Mazzariello, J. C., Mathis, M., & Brumby, S. P. (2021). Global land use / land cover with Sentinel-2 and deep learning. IGARSS 2021. https://doi.org/10.1109/IGARSS47720.2021.9

Examples

```
# ESA WorldCover land cover
result <- get_esa_wc(
  place = 'New York',
  datatype = 'landcover',
```

```
  year = 2021
)

# Sentinel-2 10m LULC Time Series
result <- get_esa_wc(
  place = 'New York',
  datatype = 'lulc',
  year = 2022
)
```

---

get_gsdc *Download Greenspace Seasonality Data Cube*

Description

download Greenspace Seasonality Data Cube for an urban area. Retrieves high-resolution greenspace seasonality data from the Sentinel-2-based global dataset developed by Wu et al. (2024). Users can define a city of interest using a bounding box, place name, coordinates, or unique city ID (UID).

Usage

```
get_gsdc(
  bbox = NULL,
  place = NULL,
  location = NULL,
  UID = NULL,
  year = NULL,
  time = NULL,
  mask = TRUE,
  quiet = TRUE
)
```

Arguments

bbox	sf, sfc, or a numeric vector (xmin, ymin, xmax, ymax) defining the area of interest. Optional if place, location, or UID is provided.
place	character or vector. (optional) A single line address, e.g. ("1600 Pennsylvania Ave NW, Washington") or a vector of addresses (c("Madrid", "Barcelona")). This can be ignored if location is specified.
location	vector or sf point. A point of interest. Ignored if UID is specified.
UID	numeric. Urban area ID. To check the ID of an available urban area, use check_available_urban()
year	numeric. (required) The year of interest.
time	Character vector of length 2 or character. (optional) Start and end dates in "MM-DD" format (e.g., c("03-20", "10-15") or "07-10"). Used to subset the 10-day interval data cube by time.

mask	logical (optional). Default is TRUE. If TRUE, masks the raster data using the given bbox or place if it is specified.
quiet	logical. Whether show progress bars for some process.

Details

The Greenspace Data Cube is organized into 36 bands per year, each representing a 10-day interval.

Value

A SpatRaster object containing the greenspace seasonality data.

Note

Use check_available_urban() and check_urban_boundary() to see supported cities and their boundaries.

References

Wu, S., Song, Y., An, J. et al. High-resolution greenspace dynamic data cube from Sentinel-2 satellites over 1028 global major cities. Sci Data 11, 909 (2024). https://doi.org/10.1038/s41597-024-03746-7

Examples

```
result <- get_gsdc(UID = 0,
                   # year = 2022
                  )
```

---

get_s2a_ndvi *Retrieve Sentinel-2-l2a images to compute NDVI*

Description

download Sentinel-2-l2a imagery data and compute NDVI. Users can define an area of interest using a bounding box or place name.

Usage

```
get_s2a_ndvi(
  bbox = NULL,
  place = NULL,
  datetime = c(),
  cloud_cover = 10,
  vege_perc = 0,
  select = "latest",
  method = "first",
```

```
  mask = TRUE,
  output_bands = NULL,
  quiet = TRUE
)
```

Arguments

bbox	sf, sfc, or a numeric vector (xmin, ymin, xmax, ymax) defining the area of interest. Optional if place is provided.
place	character or vector. (optional) A single line address, e.g. ("1600 Pennsylvania Ave NW, Washington") or a vector of addresses (c("Madrid", "Barcelona")).
datetime	numeric vector of 2. The time of interest such as c("2020-08-01", "2020-09-01").
cloud_cover	numeric. Threshold for the percentage of cloud coverage. Desfault is 10.
vege_perc	numeric. Threshold for the percentage of vegetation coverage. Desfault is 0.
select	character. one of "latest", "earliest", "all". The default is "latest".
method	character. A method for mosaicing layers: one of "mean", "median", "min", "max", "modal", "sum", "first", "last". The default is "first".
mask	logical (optional). Default is TRUE. If TRUE, masks the raster data using the given bbox or place.
output_bands	vector. A list of band names (c('B04', 'B08')). The default is NULL. If output_bands is specified, NDVI will not be computed and only the specified bands will be returned. All available bands can be found here
quiet	logical. Whether show progress bars for some process.

Value

A SpatRaster object containing (multiple) NDVI layer(s) (for different period of time) select = "latest" or select = "first" (or if mask = TRUE and select = "all")

A List of NDVI rasters if mask = FALSE and select = "all".

Examples

```
result <- get_s2a_ndvi(
  place = 'New York',
  datetime = c("2020-08-01", "2020-09-01")
)
```

`get_tile_green` *Classify greenspace based on map tile images*

Description

Generate high-resolution greenspace segmentation using WorldImagery map tiles provided by esri and Sentinel-2 cloudless mosaic tiles provided by EOX.

Usage

```
get_tile_green(
  bbox = NULL,
  place = NULL,
  zoom = 17,
  provider = "esri",
  year = NULL,
  bright_filter = 0.85,
  gr_ratio_filter = 0.05,
  gb_ratio_filter = 0.05,
  clean = TRUE,
  quiet = TRUE
)
```

Arguments

`bbox`	`sf`, `sfc`, or a numeric vector (xmin, ymin, xmax, ymax) defining the area of interest. Optional if `place` is provided.
`place`	character or vector. (optional) A single line address, e.g. ("1600 Pennsylvania Ave NW, Washington") or a vector of addresses (c("Madrid", "Barcelona")).
`zoom`	numeric. Zoom level of map tile. The default is 17.
`provider`	character. One of "esri" and "eox".
`year`	integer. The desired year for Sentinel-2 cloudless mosaic tiles. (This is required when `provider = "eox"`)
`bright_filter`	numeric. To remove very bright pixels (concrete, roofs, glare) Default is 0.85. With lower `bright_filter`, more bright pixels will be excluded.
`gr_ratio_filter`	numeric. (range from 0 to 1) To exclude pixels where green is only microscopically bigger than red. Default is 0.05. With higher `gr_ratio_filter`, more pixels will be excluded.
`gb_ratio_filter`	numeric. (range from 0 to 1) To exclude pixels where green is only microscopically bigger than blue. Default is 0.05. With higher `gb_ratio_filter`, more pixels will be excluded.
`clean`	logical. Whether to remove the blight and less green pixels based on `bright_filter`, `gr_ratio_filter`, and `gb_ratio_filter`.
`quiet`	logical. Whether show progress bars for some process.

Value

A list of two rasters including: greenspace segmentation (where 1 is green and 0 is non-green) and original map tiles

Note

The data derived from Esri WorldImagery may need to include appropriate Esri copyright notice.

Examples

```
g <- get_tile_green(
 # bbox = c(-83.087174,42.333373,-83.042542,42.358748),
 zoom = 15
)
```

`ndvi_to_sem` *ndvi_to_sem*

Description

Convert ndvi raster data into semantic vegetation areas

Usage

```
ndvi_to_sem(r = NULL, threshold = c(0.2, 0.5), quiet = FALSE)
```

Arguments

`r`	A SpatRaster with single greenspace layer, typically the output from `get_esa_wc()`, or `get_s2a_ndvi()`.
`threshold`	numeric vector of two. Thresholds, defaulting to `c(0.2, 0.5)`, for classify two types of vegetation areas according to Hashim et al. (2019): (1) Non-vegetation (Development and bare land): NDVI values generally below `0.2`. (2) Low vegetation (Shrub and grassland): NDVI values generally between `0.2` and `0.5`. (2) High vegetation (Temperate and Tropical urban forest): NDVI values generally between `0.5` and `1.0`.
`quiet`	logical. Whether show progress bars for some process.

Value

SpatRaster. A raster, where `0` represents non-green area, `1` represents shrub and grassland, and `2` represents trees.

References

Hashim, H., Abd Latif, Z., & Adnan, N. A. (2019). Urban vegetation classification with NDVI threshold value method with very high resolution (VHR) Pleiades imagery. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 42, 237-240.

Examples

```
sample_data <- terra::rast(system.file("extdata", "detroit_gs.tif", package = "greenSD"))
seg <- ndvi_to_sem(sample_data$`25_NDVI`, threshold = c(0.2, 0.6))
```

`sample_values` *Sample greenspace-realted data from Greenspace Seasonality Data Cube, ESA WorldCover 10m Annual Composites Dataset, or Sentinel-2-l2a images.*

Description

Samples values by locatoins from the Greenspace Seasonality Data Cube developed by Wu et al. (2024), ESA WorldCover 10m Annual Composites Dataset by Zanaga et al. (2021), or Sentinel-2-l2a images.

Usage

```
sample_values(
  samples = NULL,
  time = NULL,
  year = NULL,
  source = "gsdc",
  output_bands = NULL,
  cloud_cover = 10,
  vege_perc = 0,
  select = "latest",
  method = "first",
  quiet = TRUE
)
```

Arguments

`samples`	A list, matrix, `data.frame`, or `sf` object of point locations. Can be a list of length-2 numeric vectors (`list(c(lon, lat))`), a 2-column matrix or data.frame, or an `sf` object with POINT geometry in any CRS.
`time`	numeric or vector. The time of interest. See Detail.
`year`	numeric or vector. Deprecated alias for `time`.
`source`	character. The data source for extracting greenspace values: `gsdc` for Greenspace Seasonality Data Cube (also see `get_gsdc()`]), `esa_ndvi`or `esa_landcover` for ESA WorldCover 10m Annual Dataset (also see `get_esa_wc()`]), and `s2a_ndvi` or `s2a_bands` for Sentinel-2-l2a image data (also see `get_s2a_ndvi()`]). The default is `gsdc`.
`output_bands`	vector. A list of band names (`c('B04', 'B08')`). The default is `NULL`. (Only required, when `source = "s2a_bands"`) All available bands can be found here

cloud_cover	numeric. The percentage of cloud coverage for retrieving Sentinel-2-l2a images. (Only required, when source = "s2a_ndvi" or source = "s2a_bands")
vege_perc	numeric. The percentage of cloud coverage for retrieving Sentinel-2-l2a images. (Only required, when source = "s2a_ndvi" or source = "s2a_bands")
select	character. one of "latest", "earliest", "all". The default is "latest".
method	character. A method for mosaicing layers: one of "mean", "median", "min", "max", "modal", "sum", "first", "last". The default is "first".
quiet	logical. Whether show progress bars for some process.

Details

time: For the greenspace seasonality data cube, only years from 2019 to 2022 are availabe. For ESA WorldCover 10m Annual Composites Dataset, only 2020 and 2021 are available.

Value

A data.frame containing greenspace values extracted at each point across all bands. Each row corresponds to a sample location; columns represent band values.

Note

For sampling data from Greenspace Seasonality Data Cube samples must be located within the same boundary of an available city in the data cube. Use check_available_urban() and check_urban_boundary() to see supported cities and their boundaries.

References

Wu, S., Song, Y., An, J. et al. High-resolution greenspace dynamic data cube from Sentinel-2 satellites over 1028 global major cities. Sci Data 11, 909 (2024). https://doi.org/10.1038/s41597-024-03746-7

Zanaga, D., Van De Kerchove, R., De Keersmaecker, W., Souverijns, N., Brockmann, C., Quast, R., Wevers, J., Grosu, A., Paccini, A., Vergnaud, S., Cartus, O., Santoro, M., Fritz, S., Georgieva, I., Lesiv, M., Carter, S., Herold, M., Li, L., Tsendbazar, N.-E., ... Arino, O. (2021). ESA WorldCover 10 m 2020 v100 (Version v100). Zenodo. https://doi.org/10.5281/zenodo.5571936

Zanaga, D., Van De Kerchove, R., Daems, D., De Keersmaecker, W., Brockmann, C., Kirches, G., Wevers, J., Cartus, O., Santoro, M., Fritz, S., Lesiv, M., Herold, M., Tsendbazar, N.-E., Xu, P., Ramoino, F., & Arino, O. (2022). ESA WorldCover 10 m 2021 v200 (Version v200). Zenodo. https://doi.org/10.5281/zenodo.7254221

Examples

```
## Not run:
# see supported urban areas and their boundaries
check_available_urban()
boundary <- check_urban_boundary(uid = 11, plot = FALSE)

# sample locations within the boundary
samples <- sf::st_sample(boundary, size = 20)
```

```
# extract values
gs_samples <- sample_values(samples, time = 2022)

## End(Not run)
```

`to_gif` *Convert A Multi-layer Raster to GIF*

Description

Export a multi-layer raster (`SpatRaster`) or vector layer (`sf`) with multiple numeric value columns to an animated GIF.

Usage

```
to_gif(
  r,
  fps = 5,
  width = 600,
  height = 600,
  axes = TRUE,
  title_prefix = NULL,
  border = FALSE
)
```

Arguments

`r`	SpatRaster or sf. A SpatRaster with multiple layers or an sf object with multiple numeric value columns.
`fps`	numeric. Frames per second (default 5).
`width`	numeric. Width of output GIF in pixels.
`height`	numeric. Height of output GIF in pixels.
`axes`	logical. Draw axes?
`title_prefix`	character or character vector.
`border`	character. Color of polygon border(s); using NA hides them. Only optional when `r` is an sf object.

Value

An animated magick image object (GIF).

Examples

```
sample_data <- terra::rast(system.file("extdata", "detroit_gs.tif", package = "greenSD"))
gif <- to_gif(sample_data)
```

Index