Package 'mapme.biodiversity'

April 30, 2024

Title Efficient Monitoring of Global Biodiversity Portfolios

Version 0.6.0

Description Biodiversity areas, especially primary forest, serve a multitude of functions for local economy, regional functionality of the ecosystems as well as the global health of our planet. Recently, adverse changes in human land use practices and climatic responses to increased greenhouse gas emissions, put these biodiversity areas under a variety of different threats. The present package helps to analyse a number of biodiversity indicators based on freely available geographical datasets. It supports computational efficient routines that allow the analysis of potentially global biodiversity portfolios. The primary use case of the package is to support evidence based reporting of an organization's effort to protect biodiversity areas under threat and to identify regions were intervention is most duly needed.

License GPL (>= 3)

URL https://mapme-initiative.github.io/mapme.biodiversity/index.html, https://github.com/mapme-initiative/mapme.biodiversity/

BugReports https://github.com/mapme-initiative/mapme.biodiversity/issues

Depends R (>= 3.5.0)

SystemRequirements GDAL (>= 3.0.0), PROJ (>= 4.8.0)

Imports curl, dplyr, furrr, httr, magrittr, progressr, purrr, rvest, R.utils, sf, stringr, terra, tibble, tidyr, tidyselect

Suggests exactextractr, future, knitr, landscapemetrics, rmarkdown, rstac, SPEI, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/Needs/website DiagrammeR, ggplot2, lubridate, manipulateWidget, rgl, spatstat, wdpar

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.1

Collate 'register.R' 'calc_active_fire_counts.R'
'calc_active_fire_properties.R' 'calc_biome.R'
'calc_deforestation_drivers.R' 'calc_drought_indicator.R'
'calc_ecoregion.R' 'calc_elevation.R' 'calc_fatalities.R'
'calc_gsw_change.R' 'calc_gsw_occurrence.R'
'calc_gsw_recurrence.R' 'calc_gsw_seasonality.R'
'calc_gsw_transitions.R' 'get_resources.R' 'calc_indicators.R'
'calc_landcover.R' 'calc_mangroves_area.R'
'calc_population_count.R' 'calc_precipitation_chirps.R'
'calc_precipitation_wc.R' 'calc_soilproperties.R'
'calc_temperature_max_wc.R' 'calc_temperature_min_wc.R'
'calc_traveltime.R' 'calc_treecover_area.R'
'calc_treecover_area_and_emissions.R'
'calc_treecoverloss_emissions.R' 'calc_tri.R' 'engines.R'
'get_chirps.R' 'get_esalandcover.R' 'get_fritz_et_al.R'
'get_gfw_emissions.R' 'get_gfw_lossyear.R'
'get_gfw_treecover.R' 'get_gmw.R' 'get_gsw.R'
'get_nasa_firms.R' 'get_nasa_grace.R' 'get_nasa_srtm.R'
'get_nelson_et_al.R' 'get_soilgrids.R' 'get_teow.R'
'get_ucdp_ged.R' 'get_worldclim.R' 'get_worldpop.R'
'mapme.biodiversity-pkg.R' 'portfolio.R' 'utils.R'

NeedsCompilation no

Author Darius A. Görgen [aut, cre],
Om Prakash Bhandari [aut]

Maintainer Darius A. Görgen <darius2402@web.de>

Repository CRAN

Date/Publication 2024-04-30 10:20:02 UTC

R topics documented:

`active_fire_counts` *Calculate active fire counts based on NASA FIRMS polygons*

Description

This function allows to efficiently calculate the number of fire events occurred in the region of interest from the NASA FIRMS active fire polygon datasets. For each polygon, the fire event counts for the desired year is returned.

Usage

```
calc_active_fire_counts()
```

Details

The required resources for this indicator are:

- nasa_firms

Value

A function that returns a tibble with a column for number of fire events per year and instrument.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_nasa_firms(years = 2021, instrument = "VIIRS")) %>%
  calc_indicators(calc_active_fire_counts()) %>%
  tidyr::unnest(active_fire_counts)

aoi

## End(Not run)
```

`active_fire_properties` *Calculate active fire properties based on NASA FIRMS polygons*

Description

This function allows to efficiently extract the properties of fire events occurred in the region of interest from the NASA FIRMS active fire polygon datasets. For each polygon, the fire events properties like fire pixel brightness temperature, and fire radiative power (frp) along with fire hotspots for the desired year is returned. The required resources for this indicator are:

- nasa_firms

Usage

```
calc_active_fire_properties()
```

Value

A function that returns a tibble with a column for the 15 different fire events variables including lon/lat coordinates.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_nasa_firms(years = 2021, instrument = "VIIRS")) %>%
  calc_indicators(calc_active_fire_properties()) %>%
  tidyr::unnest(active_fire_properties)

aoi

## End(Not run)
```

biome *Calculate biomes statistics (TEOW) based on WWF*

Description

This function allows to efficiently retrieve the name of the biomes and compute the corresponding area from Terrestrial Ecoregions of the World (TEOW) - World Wildlife Fund (WWF) for polygons. For each polygon, the name and area of the biomes (in hectare) is returned. The required resources for this indicator are:

- teow

Usage

```
calc_biome()
```

Value

A function that returns a tibble with a column for name of the biomes and corresponding area (in ha).

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_teow()) %>%
  calc_indicators(calc_biome()) %>%
  tidyr::unnest(biome)

aoi

## End(Not run)
```

check_available_years *Helper to check yearly availability*

Description

Use this function to check if a specifed vector of years intersects with the yearly availablity of a resource.

Usage

```
check_available_years(target_years, available_years, indicator)
```

Arguments

target_years	Numeric indicating the target year.
available_years	Numeric indicating the available years.
indicator	A character vector with target resource/indicator name.

check_namespace *Checks if namespace is available*

Description

Use this function if your resource/indicator function requires the namespace of a certain package to be available. An informative error message is printed if that is not the case.

Usage

```
check_namespace(pkg)
```

Arguments

pkg	A character vector of length one indicating a package name for which the namespace is tested

Value

TRUE, invisible, if the namespace is available. An error message otherwise.

chirps *Climate Hazards Group InfraRed Precipitation with Station data (CHIRPS)*

Description

This resource is published by Funk et al. (2015) and represents a quasi-global (50°S-50°S) rainfall estimation at a monthly resolution starting with the year 1981 to the near-present. It has a spatial resolution of 0.05°. The data can be used to retrieve information on the amount of rainfall. Due to the availability of +30 years, anomaly detection and long-term average analysis is also possible. The routine will download the complete archive in order to support long-term average and anomaly calculations with respect to the 1981 - 2010 climate normal period. Thus no additional arguments need to be specified.

Usage

```
get_chirps()
```

Value

A function that returns a character of file paths.

Source

https://data.chc.ucsb.edu/products/CHIRPS-2.0/global_monthly/cogs/

References

Funk, C., Peterson, P., Landsfeld, M. et al. The climate hazards infrared precipitation with stations—a new environmental record for monitoring extremes. Sci Data 2, 150066 (2015). doi:10.1038/sdata.2015.66

deforestation_drivers *Calculate deforestation drivers*

Description

This function extracts areal statistics for the drivers of deforestation based on the data source produced by Fritz et al (2022).

Usage

```
calc_deforestation_drivers()
```

Details

The required resource for this indicator is:

- fritz_et_al

Value

A function that returns a tibble with 3 columns indicating the class of a deforestation driver, the absolute area in ha, and the percentage in relation to the total area of forest loss as indicated by the Fritz et al. (2022) resource.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_fritz_et_al(resolution = 100)) %>%
  calc_indicators(calc_deforestation_drivers()) %>%
  tidyr::unnest(deforestation_drivers)

aoi

## End(Not run)
```

---

download_or_skip *Helper to check and download urls*

Description

Use this function to fetch a number of remote URLs to local files. In case of unreliable source servers, failed downlaods are retried up to the number of times specified with `stubbornness`. A path to an aria2c executable can be specified to use it instead of the default R download function.

Usage

```
download_or_skip(
  urls = NULL,
  filenames = NULL,
  verbose = mapme_options()[["verbose"]],
  stubbornness = 6,
  check_existence = TRUE,
  aria_bin = mapme_options()[["aria_bin"]]
)
```

Arguments

urls	A character vector with URLs to be downloaded.
filenames	A character vector with local file paths the same length as urls
verbose	A logical controlling the verbosity.
stubbornness	A numeric indicating the number of retries for failed downloads.
check_existence	A logical indicating if urls are to be checked before trying to download. Defaults to TRUE.
aria_bin	A character vector pointing towards an aria2c executable.

---

drought_indicator *Calculate drought indicator statistics*

Description

This function allows to efficiently calculate the relative wetness in the shallow groundwater section with regard to the the 1948-2012 reference period. The values represent the wetness percentile a given area achieves at a given point in time in regard to the reference period. For each polygon, the desired statistic/s (mean, median or sd) is/are returned.

Usage

```
calc_drought_indicator(engine = "extract", stats = "mean")
```

Arguments

engine	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
stats	Function to be applied to compute statistics for polygons either one or multiple inputs as character "mean", "median" or "sd".

Details

The required resources for this indicator are:

- nasa_grace

Value

A function that returns a tibble with a column for each specified stats and a column with the respective date.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_nasa_grace(years = 2022)) %>%
  calc_indicators(
    calc_drought_indicator(
      engine = "extract",
      stats = c("mean", "median")
    )
  ) %>%
  tidyr::unnest(drought_indicator)

aoi

## End(Not run)
```

`ecoregion` *Calculate terrestrial ecoregions statistics (TEOW) based on WWF*

Description

This function allows to efficiently retrieve the name of the ecoregions and compute the corresponding area from Terrestrial Ecoregions of the World (TEOW) - World Wildlife Fund (WWF) for polygons. For each polygon, the name and area of the ecoregions (in hectare) is returned. The required resources for this indicator are:

- teow

Usage

```
calc_ecoregion()
```

Value

A function that returns a tibble with a column for name of the ecoregions and corresponding area (in ha).

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_teow()) %>%
  calc_indicators(calc_ecoregion()) %>%
  tidyr::unnest(ecoregion)

aoi

## End(Not run)
```

---

elevation *Calculate elevation statistics*

Description

This function allows to calculate elevation statistics for polygons. For each polygon, the desired statistic(s) are returned.

Usage

```
calc_elevation(engine = "extract", stats = "mean")
```

Arguments

engine	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
stats	Function to be applied to compute statistics for polygons either one or multiple inputs as character "mean", "median" or "sd".

Details

The required resources for this indicator are:

- nasa_srtm

Value

A function that returns a tibble with a column for each statistics.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_nasa_srtm()) %>%
  calc_indicators(
    calc_elevation(engine = "extract", stats = c("mean", "median", "sd", "var"))
  ) %>%
  tidyr::unnest(elevation)

aoi

## End(Not run)
```

engine *Function to select processing engines*

Description

check_engine() checks if an extraction engine for zonal vector-raster operations is supported by the backend.

check_stats checks if one or multiple statistics are supported for zonal vector-raster extraction by the backend.

select_engine extracts zonal vector-raster statistics for supported engine and for one or more statistics. Columns are named according to the argument name plus the respective stat. Both portfolio and asset modes are supported.

Usage

```
check_engine(queried_engine)

check_stats(queried_stats)

select_engine(x, raster, stats, engine, name = NULL, mode = "asset")
```

Arguments

queried_engine	A character vector of length one indicating the engine to check for.
queried_stats	A character vector with statistic names to be checked if they are supported by the backend
x	An sf object representing a portfolio.
raster	An terra SpatRaster from which values are to be extracted.
stats	A character vector of statistics to aggregate the raster values with.
engine	A character vector of length one specifying the engine to be used for the extraction.
name	A character vector indicating the name to append to the columns names.
mode	A character vector indicating in which mode to conduct the extraction (e.g. asset-wise or for the whole portfolio at once).

Value

check_engine() returns the character of the queried engine, if supported. Throws an error otherwise.

check_stats returns a character vector of supported statistics. Throws an error if any of the queried statistics is not supported.

select_engine returns a tibble.

esalandcover *ESA Copernicus Global Land Cover layer*

Description

This 100 meter spatial resolution land cover resource is published by Buchhorn et al. (2020) "Copernicus Global Land Cover Layers—Collection 2". The resource represents the actual surface cover of ground available annually for the period 2015 to 2019. The cell values range from 0 to 200, representing total of 23 discrete classifications from ESA.

Usage

```
get_esalandcover(years = 2015:2019)
```

Arguments

years	A numeric vector indicating the years for which to make the resource available.

Value

A function that returns a character of file paths.

Source

https://lcviewer.vito.be/download

References

© European Union, Copernicus Land Monitoring Service (year), European Environment Agency (EEA)", f.ex. in 2018: "© European Union, Copernicus Land Monitoring Service 2018, European Environment Agency (EEA)

fatalities *Calculate number of fatalities of violent conflict from UCDP GED*

Description

The indicator aggregated the number of fatalities within a given asset on a monthly cadence stratified by the type of conflict. The different types of conflicts encoded in the UCDP GED database are:

- state-based conflict
- non-state conflict
- one-sided violence

Usage

```
calc_fatalities(years = 1989:2023, precision_location = 1, precision_time = 1)
```

Arguments

`years`	A numeric vector indicating the years for which to summarize fatalities.
`precision_location`	A numeric indicating precision value for the geolocation up to which events are included. Defaults to 1.
`precision_time`	A numeric indicating the precision value of the temporal coding up to which events are included. Defaults to 1.

Details

The required resources for this indicator are:

- ucdp_ged

You may apply quality filters based on the precision of the geolocation of events and the temporal precision. By default, these are set to only include events with the highest precision scores.

For geo-precision there are levels 1 to 7 with decreasing accuracy:

- value 1: the location information corresponds exactly to the geographical coordinates available
- value 2: the location information refers to a limited area around a specified location
- value 3: the source refers to or can be specified to a larger location at the level of second order administrative divisions (ADM2), such as district or municipality, the GED uses centroid point coordinates for that ADM2.
- value 4: the location information refers to a first order administrative division, such as a province (ADM1), the GED uses the coordinates for the centroid point of ADM1
- value 5: is used in different cases if the source refers to parts of a country which are larger than ADM1, but smaller than the entire country; if two locations are mentioned a representiative point in between is selected; if the location mentioned is an non-independend island; if the location is not very specifically mentioned or in relation to another location
- value 6: the location mentioned refers to an entire country and its centroid is used
- value 7: If the event takes place over water or in international airspace, the geographical coordinates in the dataset either represent the centroid point of a certain water area or estimated coordinates

For temporal precision there are levels 1 to 5 with decreasing precision:

- value 1: if the exact date of an event is known
- value 2: if start and enddates for events are of unspecified character, spanning more than one calendar day though no longer than six days
- value 3: if when start and end dates for events are specified to a certain week, but specific dates are not provided
- value 4: if start and end dates for events are specified to a certain month
- value 5: if start and enddates for events are specified to a certain year, but specific dates are not provided

Value

A function that returns a tibble with a column for the date (year and month), the type of violence an counts of civilian fatalities, unknown fatalities and the total sum of fatalities.

References

Sundberg, Ralph, and Erik Melander, 2013, "Introducing the UCDP Georeferenced Event Dataset", Journal of Peace Research, vol.50, no.4, 523-532

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "burundi.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_ucdp_ged(version = "22.1")) %>%
  calc_indicators(
    calc_fatalities(
      years = 1991:1992,
      precision_location = 1,
      precision_time = 1
    )
  ) %>%
  tidyr::unnest(fatalities)

aoi

## End(Not run)
```

---

fritz_et_al *Drivers of deforestation for tropical forests*

Description

This resource is produced by a neirest-neighbour matching of a crowd-sourced campaign to map dominant driver of forest loss based on visual interpretation of VHR images matched with Global

Forest Loss data by Hansen (2013) version 1.7 The forest loss layer was re sampled to a resolution of 100 and 1.000 meters. Dominant drivers were determined for the period 2008 to 2009.

Usage

```
get_fritz_et_al(resolution = 100)
```

Arguments

`resolution`	An integer indicating the resolution to download. Defaults to 100.

Details

It indicates 9 different classes:

- commercial agriculture
- commercial oil palm plantations
- managed forests
- mining
- natural disturbances
- pasture
- roads
- wildfire
- other subsistence agriculture
- shifting cultivation

Value

A function that returns a character of file paths.

Source

https://zenodo.org/record/7997885

References

Steffen, F., Carlos, J.C.L., See. L., Schepaschenko D., Hofhansl F., Jung M., Dürauer M., Georgieva I., Danylo O., Lesiv M., McCallum I. (2022) A Continental Assessment of the Drivers of Tropical Deforestation With a Focus on Protected Areas. F.Cos.Sc.(3) doi:10.3389/fcosc.2022.830248

`gfw_emissions` *Forest greenhouse gas emissions*

Description

This resource is part of the publication by Harris et al. (2021) "Global maps of twenty-first century forest carbon fluxes.". It represents "the greenhouse gas emissions arising from stand-replacing forest disturbances that occurred in each modelled year (megagrams CO2 emissions/ha, between 2001 and 2021). Emissions include all relevant ecosystem carbon pools (aboveground biomass, belowground biomass, dead wood, litter, soil) and greenhouse gases (CO2, CH4, N2O)." The area unit that is downloaded here corresponds to the "megagrams of CO2 emissions/pixel" layer, in order to support the calculation of area-wise emissions.

Usage

```
get_gfw_emissions()
```

Details

There are no arguments users need to specify. However, users should note that the spatial extent for this dataset does not totally cover the same extent as the `treecover2000` and `lossyear` resources by Hansen et al. (2013). A missing value (NA) will be inserted for greenhouse gas emissions for areas where no data is available.

Value

A function that returns a character of file paths.

Source

https://data.globalforestwatch.org/datasets/gfw::forest-greenhouse-gas-emissions/about

References

Harris, N.L., Gibbs, D.A., Baccini, A. et al. Global maps of twenty-first century forest carbon fluxes. Nat. Clim. Chang. 11, 234–240 (2021). https://doi.org/10.1038/s41558-020-00976-6

`gfw_lossyear` *Year of forest loss occurrence*

Description

This resource is part of the publication by Hansen et al. (2013) "High-Resolution Global Maps of 21st-Century Forest Cover Change". It represents "Forest loss during the period 2000–2021, defined as a stand-replacement disturbance, or a change from a forest to non-forest state. Encoded as either 0 (no loss) or else a value in the range 1–20, representing loss detected primarily in the year 2001–2021, respectively." Due to changes in the satellites products used in the compilation of the tree loss product, results before the year 2011 and afterwards are not directly comparable until reprocessing has finished. Users should be aware of this limitation, especially when the timeframe of the analysis spans over the two periods delimited by the year 2011.

Usage

```
get_gfw_lossyear(version = "GFC-2022-v1.10")
```

Arguments

`version`	The version of the dataset to download. Defaults to "GFC-2022-v1.10". Check `mapme.biodiversity:::.available_gfw_versions()` to get a list of available versions

Value

A function that returns a character of file paths.

Source

https://data.globalforestwatch.org/documents/tree-cover-loss/explore

References

Hansen, M. C., P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, and J. R. G. Townshend. 2013. “High-Resolution Global Maps of 21st-Century Forest Cover Change.” Science 342 (15 November): 850–53.

gfw_treecover *Treecover for the year 2000*

Description

This resource is part of the publication by Hansen et al. (2013) represents "tree cover in the year 2000, defined as canopy closure for all vegetation taller than 5m in height. Encoded as a percentage per output grid cell, in the range 0–100." Due to changes in the satellites products used in the compilation of the treecover product, results before the year 2011 and afterwards are not directly comparable until reprocessing has finished. Users should be aware of this limitation, especially when the timeframe of the analysis spans over the two periods delimited by the year 2011.

Usage

```
get_gfw_treecover(version = "GFC-2022-v1.10")
```

Arguments

version	The version of the dataset to download. Defaults to "GFC-2022-v1.10". Check mapme.biodiversity:::.available_gfw_versions() to get a list of available versions

Value

A function that returns a character of file paths.

Source

https://data.globalforestwatch.org/documents/tree-cover-2000/explore

References

Hansen, M. C., P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. V. Stehman, S. J. Goetz, T. R. Loveland, A. Kommareddy, A. Egorov, L. Chini, C. O. Justice, and J. R. G. Townshend. 2013. “High-Resolution Global Maps of 21st-Century Forest Cover Change.” Science 342 (15 November): 850–53.

global_surface_water_change *Global Surface Water Change*

Description

The Global Surface Water dataset was developed by the European Commission's Joint Research Centre in the framework of the Copernicus Programme. It maps the location and temporal distribution of water surfaces at the global scale over the past 3.8 decades and provides statistics on their extent and change. It is provisioned as a global tiled raster resource available for all land areas. The reported data represent aggregated observations between 1984 - 2021.

Usage

```
get_global_surface_water_change(version = "v1_4_2021")
```

Arguments

`version`	A character vector indicating the version of the GSW data set to make available.

Details

The change in water occurrence intensity between the two periods is derived from homologous pairs of months (i.e. same months containing valid observations in both periods). The difference in the occurrence of surface water was calculated for each homologous pair of months. The average of all of these differences constitutes the Surface Water Occurrence change intensity. The raster files have integer cell values between `[0, 200]` where 0 represents surface water loss and 200 represents surface water gain.

Value

A function that returns a character of file paths.

Source

https://global-surface-water.appspot.com/

References

Pekel, JF., Cottam, A., Gorelick, N. et al. High-resolution mapping of global surface water and its long-term changes. Nature 540, 418–422 (2016). https://doi.org/10.1038/nature20584

`global_surface_water_occurrence`

Global Surface Water Occurrence

Description

The Global Surface Water dataset was developed by the European Commission's Joint Research Centre in the framework of the Copernicus Programme. It maps the location and temporal distribution of water surfaces at the global scale over the past 3.8 decades and provides statistics on their extent and change. It is provisioned as a global tiled raster resource available for all land areas. The reported data represent aggregated observations between 1984 - 2021.

Usage

```
get_global_surface_water_occurrence(version = "v1_4_2021")
```

Arguments

version	A character vector indicating the version of the GSW data set to make available.

Details

GSW occurrence raw data comes in raster files with integer cell values between `[0, 100]`. This value gives the percentage of the time that a given pixel was classified as water during the entire observation period. So a 0 denotes a pixel that was never classified as water, 100 denotes a pixel with permanent water.

Value

A character of file paths.

Source

https://global-surface-water.appspot.com/

References

Pekel, JF., Cottam, A., Gorelick, N. et al. High-resolution mapping of global surface water and its long-term changes. Nature 540, 418–422 (2016). https://doi.org/10.1038/nature20584

---

global_surface_water_recurrence

Global Surface Water Recurrence

Description

The Global Surface Water dataset was developed by the European Commission's Joint Research Centre in the framework of the Copernicus Programme. It maps the location and temporal distribution of water surfaces at the global scale over the past 3.8 decades and provides statistics on their extent and change. It is provisioned as a global tiled raster resource available for all land areas. The reported data represent aggregated observations between 1984 - 2021.

Usage

```
get_global_surface_water_recurrence(version = "v1_4_2021")
```

Arguments

version	A character vector indicating the version of the GSW data set to make available.

Details

Water Recurrence is a measurement of the degree of variability in the presence of water from year to year. It describes the frequency with which water returned to a particular location from one year to another, and is expressed as a percentage. The raster files have integer cell values between `[0, 100]`, where 100 represents that water reoccurs predictably every year, whereas lower values indicate that water only occurs episodically.

Value

A character of file paths.

Source

https://global-surface-water.appspot.com/

References

Pekel, JF., Cottam, A., Gorelick, N. et al. High-resolution mapping of global surface water and its long-term changes. Nature 540, 418–422 (2016). https://doi.org/10.1038/nature20584

`global_surface_water_seasonality`

Global Surface Water Seasonality

Description

The Global Surface Water dataset was developed by the European Commission's Joint Research Centre in the framework of the Copernicus Programme. It maps the location and temporal distribution of water surfaces at the global scale over the past 3.8 decades and provides statistics on their extent and change. It is provisioned as a global tiled raster resource available for all land areas. The reported data represent aggregated observations between 1984 - 2021.

Usage

```
get_global_surface_water_seasonality(version = "v1_4_2021")
```

Arguments

`version`	A character vector indicating the version of the GSW data set to make available.

Details

GSW seasonality describes the intra-annual distribution of surface water for each pixel. The raster files have integer cell values between `[0, 12]`, indicating how many months per year the pixel was classified as water.

Value

A character of file paths.

Source

https://global-surface-water.appspot.com/

References

Pekel, JF., Cottam, A., Gorelick, N. et al. High-resolution mapping of global surface water and its long-term changes. Nature 540, 418–422 (2016). https://doi.org/10.1038/nature20584

---

global_surface_water_transitions

Global Surface Water Transitions

Description

The Global Surface Water dataset was developed by the European Commission's Joint Research Centre in the framework of the Copernicus Programme. It maps the location and temporal distribution of water surfaces at the global scale over the past 3.8 decades and provides statistics on their extent and change. It is provisioned as a global tiled raster resource available for all land areas. The reported data represent aggregated observations between 1984 - 2021.

Usage

```
get_global_surface_water_transitions(version = "v1_4_2021")
```

Arguments

version	A character vector indicating the version of the GSW data set to make available.

Details

GSW transition data contains information about the type of surface water change for each pixel. The raster files have integer cell values between `[0, 10]` that code for different transition classes:

Value	Transition Class
1	Permanent
2	New Permanent
3	Lost Permanent
4	Seasonal
5	New Seasonal
6	Lost Seasonal
7	Seasonal to Permanent
8	Permanent to Seasonal
9	Ephemeral Permanent
10	Ephemeral Seasonal

Value

A character of file paths.

Source

https://global-surface-water.appspot.com/

References

Pekel, JF., Cottam, A., Gorelick, N. et al. High-resolution mapping of global surface water and its long-term changes. Nature 540, 418–422 (2016). https://doi.org/10.1038/nature20584

---

gmw *Global Mangrove Extent Polygon*

Description

This resource is part of the publication by Bunting et al. (2018) "The Global Mangrove Watch—A New 2010 Global Baseline of Mangrove Extent". The polygons represent the mangrove, which is tropical coastal vegetation and considered the most significant part of the marine ecosystem. This resource is available for the period 1996- 2020 from Global Mangrove Watch (GMW), providing geospatial information about global mangrove extent.

Usage

```
get_gmw(years = c(1996, 2007:2010, 2015:2020))
```

Arguments

years	A numeric vector of the years for which to make GMW available.

Value

A function that returns a character of file paths.

Source

https://data.unep-wcmc.org/datasets/45

References

Bunting P., Rosenqvist A., Lucas R., Rebelo L-M., Hilarides L., Thomas N., Hardy A., Itoh T., Shimada M. and Finlayson C.M. (2018). The Global Mangrove Watch – a New 2010 Global Baseline of Mangrove Extent. Remote Sensing 10(10): 1669. doi:10.3390/rs10101669.

gsw_change *Calculate Global Surface Water (GSW) Change*

Description

The change in water occurrence intensity between the two periods is derived from homologous pairs of months (i.e. same months containing valid observations in both periods). The difference in the occurrence of surface water was calculated for each homologous pair of months. The average of all of these differences constitutes the Surface Water Occurrence change intensity. The raster files have integer cell values between `[0, 200]` where 0 represents surface water loss and 200 represents surface water gain.

Usage

```
calc_gsw_change(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract". Default: "extract".
`stats`	Aggregation function with which the data are combined. Default: "mean".

Details

The pixel values are aggregated using method provided via the `stats` parameter using the specified `engine`.

The required resources for this indicator are:

- global_surface_water_change

Value

A function that returns a tibble with a column for the aggregated GSW change indicator.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)
```

```
aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_global_surface_water_change()) %>%
  calc_indicators(
    calc_gsw_change(engine = "extract", stats = "mean")
  ) %>%
  tidyr::unnest(gsw_change)

aoi

## End(Not run)
```

`gsw_occurrence` *Calculate Global Surface Water (GSW) Occurrence*

Description

GSW occurrence raw data comes in raster files with integer cell values between `[0, 100]`. This value gives the percentage of the time that a given pixel was classified as water during the entire observation period. So a 0 denotes a pixel that was never classified as water, 100 denotes a pixel with permanent water.

Usage

```
calc_gsw_occurrence(engine = "extract", min_occurrence = NULL)
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract". Default: "extract".
`min_occurrence`	Threshold to define which pixels count towards the GSW occurrence area `[0, 100]`.

Details

The raw data values are aggregated based on a provided threshold parameter `min_occurrence`, the function returns the area covered by values greater or equal than this threshold.

The required resources for this indicator are:

- global_surface_water_occurrence

Value

A function that returns a tibble with a column for the aggregated GSW occurrence indicator.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_global_surface_water_occurrence()) %>%
  calc_indicators(
    calc_gsw_occurrence(engine = "extract", stats = "mean")
  ) %>%
  tidyr::unnest(gsw_occurence)

aoi

## End(Not run)
```

`gsw_recurrence` *Calculate Global Surface Water (GSW) Recurrence*

Description

Water Recurrence is a measurement of the degree of variability in the presence of water from year to year. It describes the frequency with which water returned to a particular location from one year to another, and is expressed as a percentage. The raster files have integer cell values between `[0, 100]`, where 100 represents that water reoccurs predictably every year, whereas lower values indicate that water only occurs episodically.

Usage

```
calc_gsw_recurrence(engine = "extract", min_recurrence = NULL)
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract". Default: "extract".
`min_recurrence`	Threshold to define which pixels count towards the GSW recurrence area `[0, 100]`.

Details

The raw data values are aggregated based on a provided threshold parameter `min_recurrence`, the function returns the area covered by values greater or equal than this threshold.

The required resources for this indicator are:

- global_surface_water_recurrence

Value

A function that returns a tibble with a column for the aggregated GSW recurrence indicator.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_global_surface_water_recurrence()) %>%
  calc_indicators(
    calc_gsw_recurrence(engine = "extract", min_recurrence = 10)
  ) %>%
  tidyr::unnest(gsw_recurrence)

aoi

## End(Not run)
```

`gsw_seasonality` *Calculate Global Surface Water (GSW) Seasonality*

Description

GSW seasonality describes the intra-annual distribution of surface water for each pixel. The raster files have integer cell values between `[0, 12]`, indicating how many months per year the pixel was classified as water.

Usage

```
calc_gsw_seasonality()
```

Details

The pixel values are aggregated using method provided via the `stats` parameter.

The required resources for this indicator are:

- global_surface_water_seasonality

Value

A function that returns a tibble with one column `months` and one column `area`, representing the area covered by each class in ha.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_global_surface_water_seasonality()) %>%
  calc_indicators(calc_gsw_seasonality()) %>%
  tidyr::unnest(gsw_seasonality)

aoi

## End(Not run)
```

---

gsw_transitions *Calculate Global Surface Water (GSW) Transitions*

Description

GSW transition data contains information about the type of surface water change for each pixel. The raster files have integer cell values between `[0, 10]` that code for different transition classes:

Usage

```
calc_gsw_transitions()
```

Details

Value	Transition Class
1	Permanent
2	New Permanent
3	Lost Permanent
4	Seasonal
5	New Seasonal
6	Lost Seasonal
7	Seasonal to Permanent
8	Permanent to Seasonal
9	Ephemeral Permanent
10	Ephemeral Seasonal

To aggregate, we sum up the area of each transition class for a given region.

The required resources for this indicator are:

- global_surface_water_transitions

Value

A function that returns a tibble with a column for name of the transition classes and corresponding area (in ha).

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_global_surface_water_transitions()) %>%
  calc_indicators(calc_gsw_transitions()) %>%
```

```
  tidyr::unnest(gsw_transitions)

aoi

## End(Not run)
```

`indicators` *Register or list indicators in mapme.biodiversity*

Description

`register_indicator()` is used to register a new indicator function with base information to the package's internal environment used to inform users about available indicators. Note, registering a custom indicator will only have effect for the current R session.

`available_indicators()` returns a tibble of registered indicators with basic information such as the required resources.

Usage

```
register_indicator(name = NULL, description = NULL, resources = NULL)

available_indicators(indicators = NULL)
```

Arguments

`name`	A character vector indicating the name of the indicator.
`description`	A character vector with a basic description
`resources`	A character vector of the required resources that need to be available to calculate the indicator. The names must correspond with already registered resources.
`indicators`	If `NULL` returns a list of all registered indicators (default). Otherwise only the ones specified.

Value

`register_indicator()` is called for the side-effect of registering an indicator

`available_resources()` returns a tibble listing available indicators.

Examples

```
## Not run:
register_indicator(
  name = "treecover_area",
  description = "Area of forest cover by year",
  resources = c(
    "gfw_treecover",
    "gfw_lossyear"
  )
```

```
)

## End(Not run)
available_indicators()
```

`landcover` *Calculate area of different landcover classes*

Description

The land cover data shows us how much of the region is covered by forests, rivers, wetlands, barren land, or urban infrastructure thus allowing the observation of land cover dynamics over a period of time. This function allows to efficiently calculate area of different landcover classes for polygons. For each polygon, the area of the classes in hectare(ha) is returned.

Usage

```
calc_landcover()
```

Details

The required resources for this indicator are:

- esalandcover

Value

A function that returns tibble with a column for area (in ha) and the percentage covered per land-cover class.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_esalandcover(years = 2016:2017)) %>%
```

```
  calc_indicators(calc_landcover()) %>%
  tidyr::unnest(landcover)

aoi

## End(Not run)
```

`make_global_grid` *Helper to create a grid of regular resolution and CRS*

Description

Use this function to create a regular grid in a custom CRS. This is used e.g. to create the tile grid for Global Forest Watch in order to retrieve the intersecting tiles with a given portfolio.

Usage

```
make_global_grid(
  xmin = -180,
  xmax = 170,
  dx = 10,
  ymin = -50,
  ymax = 80,
  dy = 10,
  proj = NULL
)
```

Arguments

`xmin`	minimum longitude value (E/W)
`xmax`	maximum longitude value (E/W)
`dx`	difference in longitude value per grid
`ymin`	minimum latitude value (S/N)
`ymax`	maximum latitude value (E/W)
`dy`	difference in latitude value per grid
`proj`	projection system

Value

An sf object with a defined grid.

mangroves_area *Calculate mangrove extent based on Global Mangrove Watch (GMW)*

Description

This function allows to efficiently calculate area of mangrove from Global Mangrove Watch - World Conservation Monitoring Centre (WCMC) for polygons. For each polygon, the area of the mangrove (in hectare) for desired year is returned.

Usage

```
calc_mangroves_area()
```

Details

The required resources for this indicator are:

- gmw

Value

A function that returns a tibble with a column for area of mangrove (in ha) and corresponding year.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "shell_beach_protected_area_41057_B.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_gmw(years = c(1996, 2016))) %>%
  calc_indicators(calc_mangroves_area()) %>%
  tidyr::unnest(mangroves_area)

aoi

## End(Not run)
```

mapme *Portfolio methods for mapme.biodiversity*

Description

`mapme_options()` sets default options for mapme.biodiversity to control the behavior of downstream functions. Mainly, the output path as well as the temporal directory for intermediate files can be set. Additionally, the verbosity can be set. The testing options should not be set by users, as it controls the behavior of the package during automated test pipelines. Might be extended by other options in the future.

`get_resources()` data sets required for the calculation of indicators can be made available. The function supports the specification of several resource functions. To determine the output path, temporary directory and verbosity, the output of `mapme_options()` is used.

`calc_indicators()` calculates specific biodiversity indicators. A requirement is that the resources that are mandatory inputs for the requested indicators are available locally. Multiple indicators and their respective additional arguments can be supplied.

Usage

```
mapme_options(..., outdir, verbose, aria_bin, testing)

get_resources(x, ...)

calc_indicators(x, ...)
```

Arguments

`...`	One or more functions for resources/indicators
`outdir`	A length one character indicating the output path.
`verbose`	A logical, indicating if informative messages should be printed.
`aria_bin`	A character vector to an aria2c executable for parallel downloads.
`testing`	A logical. Not to be set by users. Controls the behavior during automated test pipelines.
`x`	An `sf` object with features of type `"POLYGON"`

Value

`mapme_options()` returns a list of options if no arguments are specified. Otherwise sets matching arguments to new values in the package's internal environment.

`get_resources()` is called for its side effect of making resources available in the package environment. Returns `x`, invisibly.

`calc_indicators()` returns `x`, invisibly, with an additional nested list column per requested indicator.

Examples

```
library(mapme.biodiversity)
mapme_options()
```

`nasa_firms` *Active Fire Polygon*

Description

This resource is published by Fire Information for Resource Management System (FIRMS) from NASA as Near Real Time (NRT) active fire data. The data is collected from Moderate Resolution Imaging Spectroradiometer (MODIS) and the Visible Infrared Imaging Radiometer Suite (VIIRS). The resource represents the fire hotspot with lat/lon coordinates along with information on fire pixel brightness temperature, and fire radiative power (frp). The data from MODIS is available from 2000 to 2021 and that from VIIRS is only available for 2012-2021 year range.

Usage

```
get_nasa_firms(years = 2012:2021, instrument = "VIIRS")
```

Arguments

`years`	A numeric vector indicating the years for which to make the resource available.
`instrument`	A character vector specifying the data collection instrument.

Details

The data from the following instruments are available:

- "MODIS"
- "VIIRS"

Value

A function that returns a character of file paths.

Source

https://firms.modaps.eosdis.nasa.gov/download/

References

NRT VIIRS 375 m Active Fire product VNP14IMGT distributed from NASA FIRMS. Available on-line https://earthdata.nasa.gov/firms. doi:10.5067/FIRMS/VIIRS/VNP14IMGT_NRT.002.

nasa_grace *NASA GRACE-based Drought Indicator layer*

Description

The resource is published by NASA GRACE Tellus. This data set reflects on potential drought conditions in the shallow groundwater section relative to a reference period spanning from 1948 to 2012. It is available as a global raster with a weekly temporal resolution starting with the year 2003. The value indicates the wetness percentile of a given pixel with regard to the reference period.

Usage

```
get_nasa_grace(years = 2003:2022)
```

Arguments

years	A numeric vector indicating the years for which to make the resource available.

Value

A function that returns a character of file paths.

nasa_srtm *NASADEM HGT v001*

Description

This resource is processed by the Land Processes Distributed Active Archive Center (LP DAAC) and made available at the Microsoft Planetery Computer. NASADEM are distributed in 1 degree latitude by 1 degree longitude tiles and consist of all land between 60° N and 56° S latitude. This accounts for about 80% of Earth's total landmass.

Usage

```
get_nasa_srtm()
```

Value

A function that returns a character of file paths.

Source

https://planetarycomputer.microsoft.com/dataset/nasadem

References

NASA JPL (2020). NASADEM Merged DEM Global 1 arc second V001. NASA EOSDIS Land Processes DAAC. Accessed 2023-07-01 from https://doi.org/10.5067/MEaSUREs/NASADEM/NASADEM_HGT.001

`nelson_et_al` *Accessibility to Cities layer*

Description

This resource is published by Weiss et al. (2018) "A global map of travel time to cities to assess inequalities in accessibility in 2015" on journal nature. Accessibility is the ease with which larger cities can be reached from a certain location. This resource represents the travel time to major cities in the year 2015. Encoded as minutes, representing the time needed to reach that particular cell from nearby city of target population range. The following ranges to nearby cities are available:

- "5k_10k"
- "10k_20k"
- "20k_50k"
- "50k_100k"
- "100k_200k"
- "200k_500k"
- "500k_1mio"
- "1mio_5mio"
- "50k_50mio"
- "5k_110mio"
- "20k_110mio"
- "5mio_50mio"

Usage

```
get_nelson_et_al(ranges = "20k_50k")
```

Arguments

`ranges`	A character vector indicating one or more ranges to download.

Value

A function that returns a character of file paths.

Source

https://figshare.com/articles/dataset/Travel_time_to_cities_and_ports_in_the_year_2015/7638134/3

References

Weiss, D. J., Nelson, A., Gibson, H. S., Temperley, W., Peedell, S., Lieber, A., ... & Gething, P. W. (2018). A global map of travel time to cities to assess inequalities in accessibility in 2015. Nature, 553(7688), 333-336.

population_count *Calculate population count statistics*

Description

WorldPop, which was initiated in 2013, offers easy access to spatial demographic datasets, claiming to use peer-reviewed and fully transparent methods to create global mosaics for the years 2000 to 2020. This function allows to efficiently calculate population count statistics (e.g. total number of population) for polygons. For each polygon, the desired statistic/s (min, max, sum, mean, median, sd or var) is/are returned.

Usage

```
calc_population_count(engine = "extract", stats = "sum")
```

Arguments

engine	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
stats	Function to be applied to compute statistics for polygons either one or multiple inputs as character "min", "max", "sum", "mean", "median" "sd" or "var".

Details

The required resources for this indicator are:

- worldpop

Value

A function that returns tibble with a column for population count statistics.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
```

```
) %>%
  read_sf() %>%
  get_resources(get_worldpop(years = 2000:2010)) %>%
  calc_indicators(
    calc_population_count(engine = "extract", stats = c("sum", "median"))
  ) %>%
  tidyr::unnest(population_count)

aoi

## End(Not run)
```

`portfolio` *Writing a portfolio to disk*

Description

`write_portfolio()` writes a processed biodiversity portfolio to disk. In order to ensure interoperability with other geospatial software the only supported format is the GeoPackage. The metadata of a portfolio together with the geometry will be written to a table called `'metadata'`. All calculated indicators, which are expected to be present as nested list columns, will be written to their own respective tables. In order to allow re-joining the metadata with the indicators, it is expected that a column called `'assetid'` which uniquely identifies all assets is present. Usually, users do not have to take care of this since the usual `mapme.biodiversity` workflow will ensure that this columns is present. Additional arguments to `st_write()` can be supplied.

`read_portfolio()` is used to read a portfolio object that was previously written to disk via `write_portfolio()` back into R as an `sf` object. It should be directed against a GeoPackage which was the output of `write_portfolio()`, otherwise the function is very likely to fail. All available indicators will be read back into R as nested list columns reflecting the output once `calc_indicators()` has been called.

Usage

```
write_portfolio(x, dsn, overwrite = FALSE, ...)

read_portfolio(src, ...)
```

Arguments

`x`	A portfolio object processed with `mapme.biodiversity`
`dsn`	A file path for the output file. Should end with `'.gpkg'`
`overwrite`	A logical indicating if the output file should be overwritten if it exists
`...`	Additional arguments supplied to `st_read()`
`src`	A character vector pointing to a GeoPackage that has been previously written to disk via `write_portfolio()`

Value

`write_portfolio()` returns `x`, invisibly.

`read_portfolio()` reutnrs an `sf` object object with nested list columns for every indicator table found in the GeoPackage source file.

`precipitation_chirps` *Calculate precipitation statistics based on CHIRPS*

Description

This functions allows to calculate precipitation statistics based on the CHIRPS rainfall estimates. Corresponding to the time-frame of the analysis of the portfolio, monthly precipitation statistics are calculated. These include the total rainfall amount, rainfall anomaly against the 1981-2010 climate normal, and the Standardized Precipitation Index (SPI) which is available for scales between 1 and 48 months. Th function needs the `SPEI` package to be installed.

Usage

```
calc_precipitation_chirps(
  years = 1981:2020,
  engine = "extract",
  scales_spi = 3,
  spi_prev_years = 8
)
```

Arguments

`years`	A numeric vector indicating the years for which to calculate precipitation statistics.
`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`scales_spi`	Integers specifying time-scales for SPI
`spi_prev_years`	Integer specifying how many previous years to include in order to fit the SPI. Defaults to 8.

Details

The required resources for this indicator are:

- chirps

Value

A function that returns a tibble with a column for years, months, absolute rainfall (in mm), rainfall anomaly (in mm) and one or more columns per selected time-scale for SPI (dimensionless).

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_chirps()) %>%
  calc_indicators(
    calc_precipitation_chirps(
      years = 2010,
      engine = "extract",
      scales_spi = 3,
      spi_prev_years = 8
    )
  ) %>%
  tidyr::unnest(precipitation_chirps)

aoi

## End(Not run)
```

`precipitation_wc` *Calculate precipitation statistics*

Description

This function allows to efficiently calculate precipitation statistics from Worldclim for polygons. For each polygon, the desired statistic/s (min, max, sum, mean, median, sd or var) is/are returned.

Usage

```
calc_precipitation_wc(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.

stats — Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resources for this indicator are:

- precipitation layer from worldclim_precipitation

Value

A function that returns a tibble with a column for precipitation statistics (in mm).

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_worldclim_precipitation(years = 2018)) %>%
  calc_indicators(
    calc_precipitation_wc(
      engine = "extract",
      stats = c("mean", "median")
    )
  ) %>%
  tidyr::unnest(precipitation_wc)

aoi

## End(Not run)
```

resources *Register or list resources in mapme.biodiversity*

Description

`register_resource()` is used to register a new resource function with base information to the package's internal environment used to inform users about available resources. Note, registering a custom resource will only have effect for the current R session.

`available_resources()` returns a tibble of registered resources with basic information such as the source and the licence.

Usage

```
register_resource(
  name = NULL,
  description = NULL,
  licence = NULL,
  source = NULL,
  type = NULL
)

available_resources(resources = NULL)
```

Arguments

`name`	A character vector indicating the name of the resource.
`description`	A character vector with a basic description
`licence`	A character vector indicating the licence of the resource. In case it is a custom licence, put a link to the licence text.
`source`	Optional, preferably a URL where the data is found.
`type`	A character vector indicating the type of the resource. Either 'vector' or 'raster'.
`resources`	If `NULL` returns a list of all resources (default). Otherwise only the ones specified.

Value

`register_resource()` is called for the side-effect of registering a resource.

`available_resources()` returns a tibble listing available resources.

Examples

```
## Not run:
register_resource(
  name = "gfw_treecover",
  description = "Global Forest Watch - Percentage of canopy closure in 2000",
  licence = "CC-BY 4.0",
  source = "https://data.globalforestwatch.org/documents/tree-cover-2000/explore",
```

```
  type = "raster"
)

## End(Not run)
available_resources()
```

`soilgrids` *SoilGrids data layers*

Description

SoilGrids is a project combining global observation data with machine learning to map the spatial distribution of soil properties across the globe. It is produced at a spatial resolution of 250 meters and each parameters is mapped at different depths. In order to be able to assess prediction uncertainty, besides the mean and median prediction, the 0.05 and 0.95 percentile predictions are available. The following parameters are available:

bdod Bulk density of the fine earth fraction (kg/dm3)

cec Cation Exchange Capacity of the soil (cmol(c)/kg)

cfvo Volumetric fraction of coarse fragments > 2 mm (cm3/100cm3 (volPerc))

clay Proportion of clay particles < 0.002 mm in the fine earth fraction (g/100g)

nitrogen Total nitrogen (g/kg)

phh2o Soil pH (pH)

sand Proportion of sand particles > 0.05 mm in the fine earth fraction (g/100g)

silt Proportion of silt particles >= 0.002 mm and <= 0.05 mm in the fine earth fraction (g/100g)

soc Soil organic carbon content in the fine earth fraction (g/kg)

ocd Organic carbon density (kg/m3)

ocs Organic carbon stocks (kg/m²)

Usage

```
get_soilgrids(layers, depths, stats)
```

Arguments

`layers`	A character vector indicating the layers to download from soilgrids
`depths`	A character vector indicating the depths to download
`stats`	A character vector indicating the statistics to download.

Details

Except for `ocs`, which is only available for a depth of `"0-30cm"`, all other parameters are available at the following depths:

- "0-5cm"
- "5-15cm"
- "15-30cm"
- "30-60cm"
- "60-100cm"
- "100-200cm"

Each parameter and depth is available for the following statistics:

- "Q0.05"
- "Q0.50"
- "mean"
- "Q0.95"

Value

A function that returns a character of file paths.

Source

https://www.isric.org/explore/soilgrids

References

Hengl T, Mendes de Jesus J, Heuvelink GBM, Ruiperez Gonzalez M, Kilibarda M, et al. (2017) SoilGrids250m: Global gridded soil information based on machine learning. PLOS ONE 12(2): e0169748. doi:10.1371/journal.pone.0169748

`soilproperties` *Calculate Zonal Soil Properties*

Description

This indicator allows the extraction of zonal statistics for resource layers previously downloaded from SoilGrids, thus in total supporting the calculation of zonal statistics for 10 different soil properties at 6 different depths for a total of 4 different model outputs (stat). Zonal statistics will be calculated for all SoilGrid layers that have been previously made available vie `get_resources()`.

Usage

```
calc_soilproperties(engine = "extract", stats = "mean")
```

Arguments

engine	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
stats	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resource for this indicator is:

- soilgrids

Value

A tibble with a column for the SoilGrid layer, the depth and the model output statistic as well as additional columns for all zonal statistics specified via `stats_soil`

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(
    get_soilgrids(
      layers = c("clay", "silt"),
      depths = c("0-5cm", "5-15cm"),
      stats = "mean"
    )
  ) %>%
  calc_indicators(
    calc_soilproperties(engine = "extract", stats = c("mean", "median"))
  ) %>%
  tidyr::unnest(soilproperties)

aoi

## End(Not run)
```

`temperature_max_wc` *Calculate maximum temperature statistics*

Description

This function allows to efficiently calculate maximum temperature statistics from Worldclim for polygons. For each polygon, the desired statistic/s (min, max, sum, mean, median, sd or var) is/are returned.

Usage

```
calc_temperature_max_wc(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resources for this indicator are:

- maximum temperature layer from worldclim_max_temperature

Value

A tibble with a column for maximum temperature statistics (in °C)

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
```

```
  read_sf() %>%
  get_resources(get_worldclim_max_temperature(years = 2018)) %>%
  calc_indicators(
    calc_temperature_max_wc(
      engine = "extract",
      stats = c("mean", "median")
    )
  ) %>%
  tidyr::unnest(temperature_max_wc)

aoi

## End(Not run)
```

`temperature_min_wc` *Calculate minimum temperature statistics based on WorldClim*

Description

This function allows to efficiently calculate minimum temperature statistics from Worldclim for polygons. For each polygon, the desired statistic/s (min, max, sum, mean, median, sd or var) is/are returned.

Usage

```
calc_temperature_min_wc(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resources for this indicator are:

- minimum temperature layer from worldclim_min_temperature

Value

A function that returns a tibble with a column for minimum temperature statistics (in °C).

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_worldclim_min_temperature(years = 2018)) %>%
  calc_indicators(
    calc_temperature_min_wc(
      engine = "extract",
      stats = c("mean", "median")
    )
  ) %>%
  tidyr::unnest(temperature_min_wc)

aoi

## End(Not run)
```

---

teow *Terrestrial Ecoregions of the World (TEOW) Polygon*

Description

This resource is part of the publication by Olson et al. (2004) "Terrestrial Ecosystems of the World (TEOW) from WWF-US (Olson)". It depicts 867 terrestrial ecoregions around the world classified into 14 different terrestrial biomes such as forests, grasslands, or deserts. The polygons represent the ecoregions, defined as relatively large units of land or inland water sharing a large majority of biodiversity. The datasets is made available from World Wildlife Fund (WWF) for the year 2001.

Usage

```
get_teow()
```

Value

A function that returns a character of file paths.

References

Olson, D. M., Dinerstein, E., Wikramanayake, E. D., Burgess, N. D., Powell, G. V. N., Underwood, E. C., D'Amico, J. A., Itoua, I., Strand, H. E., Morrison, J. C., Loucks, C. J., Allnutt, T. F., Ricketts, T. H., Kura, Y., Lamoreux, J. F., Wettengel, W. W., Hedao, P., Kassem, K. R. 2001. Terrestrial ecoregions of the world: a new map of life on Earth. Bioscience 51(11):933-938. doi:10.1641/00063568(2001)051[0933:TEOTWA]2.0.CO;2

`traveltime` *Calculate accessibility statistics*

Description

Accessibility is the ease with which larger cities can be reached from a certain location. This function allows to efficiently calculate accessibility statistics (i.e. travel time to nearby major cities) for polygons. For each polygon, the desired statistic/s (mean, median or sd) is/are returned.

Usage

```
calc_traveltime(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resources for this indicator are:

- nelson_et_al

Value

A function that returns a tibble with a column for accessibility statistics (in minutes).

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)
```

```
mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_nelson_et_al(
    ranges = c("5k_10k", "100k_200k", "500k_1mio", "1mio_5mio")
  )) %>%
  calc_indicators(
    calc_traveltime(engine = "extract", stats = c("min", "max"))
  ) %>%
  tidyr::unnest(traveltime)

aoi

## End(Not run)
```

`treecoverloss_emissions` *Calculate emission statistics*

Description

This functions allows to efficiently calculate emission statistics for areas of interest. For each year in the analysis timeframe, the forest losses from Hansen et al. (2013) are overlayed with the respective emission layer from Harris et al. (2021) and area-wise emission statistics are calculated for each year.

Usage

```
calc_treecoverloss_emissions(years = 2000:2020, min_size = 10, min_cover = 35)
```

Arguments

`years`	A numeric vector with the years for which to calculate emissions caused by treecover loss.
`min_size`	The minimum size of a forest patch in ha.
`min_cover`	The minimum threshold of stand density for a pixel to be considered forest in the year 2000.

Details

The required resources for this indicator are:

- gfw_treecover

- gfw_lossyear
- gfw_emissions

Value

A function that returns a tibble with a column for years and emissions (in Mg).

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(
    get_gfw_treecover(version = "GFC-2020-v1.8"),
    get_gfw_lossyear(version = "GFC-2020-v1.8"),
    get_gfw_emissions()
  ) %>%
  calc_indicators(
    calc_treecoverloss_emissions(years = 2016:2017, min_size = 1, min_cover = 30)
  ) %>%
  tidyr::unnest(treecoverloss_emissions)

aoi

## End(Not run)
```

---

treecover_area *Calculate treecover statistics*

Description

This functions allows to efficiently calculate treecover statistics for polygons. For each year in the analysis timeframe, the forest losses in preceding and the current years are subtracted from the treecover in the year 2000 and actual treecover figures within the polygon are returned.

Usage

```
calc_treecover_area(years = 2000:2020, min_size = 10, min_cover = 35)
```

Arguments

years	A numeric vector with the years for which to calculate treecover area.
min_size	The minimum size of a forest patch to be considered as forest in ha.
min_cover	The minimum cover percentage per pixel to be considered as forest.

Details

The required resources for this indicator are:

- gfw_treecover
- gfw_lossyear

Value

A function that returns a tibble with a column for years and treecover (in ha).

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(
    get_gfw_treecover(version = "GFC-2022-v1.10"),
    get_gfw_lossyear(version = "GFC-2022-v1.10")
  ) %>%
 calc_indicators(calc_treecover_area(years = 2016:2017, min_size = 1, min_cover = 30)) %>%
  tidyr::unnest(treecover_area)

aoi

## End(Not run)
```

`treecover_area_and_emissions` *Calculate treeloss statistics*

Description

This functions allows to efficiently calculate the treecover and emissions indicators in a single function call together. Since most of the pre-processing operations for treecover and emissions are the same, it is more efficient to calculate them in one run if users are actually interested in both statistics. Otherwise users are advised to use the respective single indicator functions.

Usage

```
calc_treecover_area_and_emissions(
  years = 2000:2020,
  min_size = 10,
  min_cover = 35
)
```

Arguments

`years`	A numeric vector with the years for which to calculate treecover area and emissions.
`min_size`	The minimum size of a forest patch in ha.
`min_cover`	The minimum threshold of stand density for a pixel to be considered forest in the year 2000.

Details

The required resources for this indicator are:

- gfw_treecover
- gfw_lossyear
- gfw_emissions

Value

A function that returns tibble with a column for years, treecover (in ha), and emissions (in Mg CO2).

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
```

```
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(
    get_gfw_treecover(version = "GFC-2020-v1.8"),
    get_gfw_lossyear(version = "GFC-2020-v1.8"),
    get_gfw_emissions()
  ) %>%
  calc_indicators(
    calc_treecover_area_and_emissions(years = 2016:2017, min_size = 1, min_cover = 30)
  ) %>%
  tidyr::unnest(treecover_area_and_emissions)

aoi

## End(Not run)
```

`tri` *Calculate Terrain Ruggedness Index (TRI) statistics*

Description

Terrain Ruggedness Index is a measurement developed by Riley, et al. (1999). The elevation difference between the centre pixel and its eight immediate pixels are squared and then averaged and its square root is taken to get the TRI value. This function allows to calculate terrain ruggedness index (tri) statistics for polygons. For each polygon, the desired statistic(s) are returned.

Usage

```
calc_tri(engine = "extract", stats = "mean")
```

Arguments

`engine`	The preferred processing functions from either one of "zonal", "extract" or "exactextract" as character.
`stats`	Function to be applied to compute statistics for polygons either single or multiple inputs as character. Supported statistics are: "mean", "median", "sd", "min", "max", "sum" "var".

Details

The required resources for this indicator are:

- nasa_srtm

Value

A function that returns a tibble with a column for terrain ruggedness index statistics (in meters). The range of index values and corresponding meaning: (1) 0 - 80 m :- level surface (2) 81-116 m :- nearly level surface (3) 117-161 m :- slightly rugged surface (4) 162-239 m :- intermediately rugged surface (5) 240-497 m :- moderately rugged surface (6) 498-958 m :- highly rugged surface (7) 959-4367 m:- extremely rugged surface

References

Riley, S. J., DeGloria, S. D., & Elliot, R. (1999). Index that quantifies topographic heterogeneity. intermountain Journal of sciences, 5(1-4), 23-27.

Examples

```
## Not run:
library(sf)
library(mapme.biodiversity)

outdir <- file.path(tempdir(), "mapme-data")
dir.create(outdir, showWarnings = FALSE)

mapme_options(
  outdir = outdir,
  verbose = FALSE
)

aoi <- system.file("extdata", "sierra_de_neiba_478140_2.gpkg",
  package = "mapme.biodiversity"
) %>%
  read_sf() %>%
  get_resources(get_nasa_srtm()) %>%
  calc_indicators(
    calc_tri(stats = c("mean", "median", "sd", "var"), engine = "extract")
  ) %>%
  tidyr::unnest(tri)

aoi

## End(Not run)
```

ucdp_ged *UCDP Georeferenced Event Dataset (UCDP GED)*

Description

This resource distributed by the Uppsala Conflict Data Program (UCDP) constitutes its most disaggregated dataset on individual events of organized violence. It encodes the different actors involved, is spatially disaggregated down to village levels anc currently covers the time period of 1989 to 2021. Older versions of the data set can be downloaded, but users are recommended to download the latest data set.

Usage

```
get_ucdp_ged(version = "latest")
```

Arguments

version	A character vector specifying the version to download. Defaults to "latest".

Details

The following versions are available

- 5.0
- 17.1
- 17.2
- 18.1
- 19.1
- 20.1
- 21.1
- 22.1
- latest

Value

A function that returns a character of file paths.

Source

https://ucdp.uu.se/downloads/

References

Davies, Shawn, Therese Pettersson & Magnus Öberg (2022). Organized violence 1989-2021 and drone warfare. Journal of Peace Research 59(4). doi:10.1177/00223433221108428

unzip_and_remove *Helper to unzip and remove zip/gzip files*

Description

Use this function to unzip a zip/gzip file and remove the original archive, if required.

Usage

```
unzip_and_remove(zip = NULL, dir = tempdir(), remove = TRUE)
```

Arguments

zip	zip file to unzip
dir	A directory where the extracted files are written to.
remove	if TRUE, removes the zip else keeps it

worldclim_max_temperature *Downloads WorldClim Maximum Temperature layer*

Description

This resource is published by Fick et al. (2017) "WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas" and represents multiple climatic variables from which we will be requiring minimum temperature, maximum temperature, and mean precipitation layers. The layers are available to download for the period 2000 - 2018 on monthly basis from WorldClim.

Usage

```
get_worldclim_max_temperature(years = 2000:2018)
```

Arguments

years	A numeric vector indicating for which years to make the resource available.

Details

This resource represents the maximum temperature, layers available to download for the period 2000 - 2018 on monthly basis from WorldClim. Encoded as (°C), representing the maximum temperature per output grid cell.

Value

A character of file paths.

Source

https://www.worldclim.org/data/index.html

`worldclim_min_temperature`
Downloads WorldClim Minimum Temperature layer

Description

This resource is published by Fick et al. (2017) "WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas" and represents multiple climatic variables from which we will be requiring minimum temperature, maximum temperature, and mean precipitation layers. The layers are available to download for the period 2000 - 2018 on monthly basis from WorldClim.

Usage

```
get_worldclim_min_temperature(years = 2000:2018)
```

Arguments

`years`	A numeric vector indicating for which years to make the resource available.

Details

This resource represents the minimum temperature, layers available to download for the period 2000 - 2018 on monthly basis from WorldClim. Encoded as (°C), representing the minimum temperature per output grid cell.

Value

A function that returns a character of file paths.

Source

https://www.worldclim.org/data/index.html

`worldclim_precipitation` *Downloads WorldClim Mean Precipitation layer*

Description

This resource is published by Fick et al. (2017) "WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas" and represents multiple climatic variables from which we will be requiring minimum temperature, maximum temperature, and mean precipitation layers. The layers are available to download for the period 2000 - 2018 on monthly basis from WorldClim.

Usage

```
get_worldclim_precipitation(years = 2000:2018)
```

Arguments

`years`	A numeric vector indicating for which years to make the resource available.

Details

This resource represents the average precipitation, layers available to download for the period 2000 - 2018 on monthly basis from WorldClim. Encoded as (mm), representing the mean precipitation per output grid cell.

Value

A character of file paths.

Source

https://www.worldclim.org/data/index.html

`worldpop` *Population Count layer for year 2000-2020*

Description

This resource is published by open spatial demographic data and research organization called WorldPop. This resource represents the population count, 1 km spatial resolution layers available to download from the year 2000 to 2020. The dataset is called as WorldPop Unconstrained Global Mosaics. The encoded cell value represents the total number of people in that particular grid cell.

Usage

```
get_worldpop(years = 2000)
```

Arguments

years	A numeric vector indicating the years for which to make the resource available.

Value

A function that returns a character of file paths.

Source

https://www.worldpop.org/

Index