Package 'envar'

July 31, 2026

Type Package

Title Download and Process Environmental Variables

Version 0.1.0

Description Provides a unified interface to download, harmonise and extract a wide range of environmental and socio-economic variables from established open data web services (such as 'WorldClim' <https://www.worldclim.org/>, 'CHELSA' <https://chelsa-climate.org/> and 'Bio-ORACLE' <https://www.bio-oracle.org/>, among others) for use in macroecology and biogeography. The package handles spatial subsetting to a study area, reprojection to a common coordinate reference system, and extraction of values at sampling points, so that predictors from heterogeneous sources can be assembled within a single reproducible workflow. Helper functions for collinearity checking and variable exploration are also included.

License GPL-3

Language en-US

Encoding UTF-8

LazyData true

LazyDataCompression xz

Imports terra, dplyr, httr, sf, rnaturalearth, usdm, corrplot, cli, fs, utils, digest, rangeBuilder

RoxygenNote 7.3.2

Config/Needs/website terra, dplyr, sf, pkgdown, usdm, corrplot, rangeBuilder

Suggests knitr, rmarkdown, rnaturalearthdata

VignetteBuilder knitr

URL https://animalbiodiversitylab.github.io/envar/, https://github.com/animalbiodiversitylab/envar

BugReports https://github.com/animalbiodiversitylab/envar/issues

Depends R (>= 3.5.0)

NeedsCompilation no

Author Andrea Simoncini [aut, cre] (ORCID:
<https://orcid.org/0000-0002-9856-6111>),
Michele Bertoncini [aut] (ORCID:
<https://orcid.org/0000-0002-1866-0045>),
Andrea Cerofolini [aut] (ORCID:
<https://orcid.org/0009-0009-4439-3014>),
Andrea Dalpasso [aut] (ORCID: <https://orcid.org/0000-0003-1471-789X>),
Mattia Falaschi [aut] (ORCID: <https://orcid.org/0000-0002-4511-4816>),
Gentile Francesco Ficetola [aut] (ORCID:
<https://orcid.org/0000-0003-3414-5155>),
Elia Lo Parrino [aut] (ORCID: <https://orcid.org/0000-0001-9619-9568>)

Maintainer Andrea Simoncini <simonciniandre@gmail.com>

Repository CRAN

Date/Publication 2026-07-31 14:30:02 UTC

Contents

`accessibility` *Download and process Global Accessibility Indicators*

Description

This function downloads, processes, and extracts variables from the Global Accessibility Indicators dataset. Each variable corresponds to a global raster representing the travelling time (in minutes) to cities or ports of specific sizes.

Usage

```
accessibility(x, vars, ...)
```

Arguments

`x`	The output from ‘par_set()‘ defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use ‘par_set()‘ to define parameters for download.
`vars`	Character vector of one or more variables to download and process.
`...`	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Cities

- 1 - "cities1" ("cities 1", "city 1", "cities >5m", "huge cities", "travel time cities 1")
- 2 - "cities2" ("cities 2", "city 2", "cities >1m", "large cities", "travel time cities 2")
- 3 - "cities3" ("cities 3", "city 3", "medium cities", "travel time cities 3")
- 4 - "cities4" ("cities 4", "city 4", "small cities", "travel time cities 4")
- 5 - "cities5" ("cities 5", "city 5", "travel time cities 5")
- 6 - "cities6" ("cities 6", "city 6", "travel time cities 6")
- 7 - "cities7" ("cities 7", "city 7", "travel time cities 7")
- 8 - "cities8" ("cities 8", "city 8", "towns", "travel time cities 8")
- 9 - "cities9" ("cities 9", "city 9", "small towns", "travel time cities 9")
- 10 - "cities10" ("cities 10", "city 10", "aggregated cities 1", "travel time cities 10")
- 11 - "cities11" ("cities 11", "city 11", "aggregated cities 2", "travel time cities 11")
- 12 - "cities12" ("cities 12", "city 12", "aggregated cities 3", "travel time cities 12")

Ports

- 13 - "ports1" ("ports 1", "port 1", "large ports", "travel time ports 1")
- 14 - "ports2" ("ports 2", "port 2", "medium ports", "travel time ports 2")
- 15 - "ports3" ("ports 3", "port 3", "small ports", "travel time ports 3")
- 16 - "ports4" ("ports 4", "port 4", "very small ports", "travel time ports 4")
- 17 - "ports5" ("ports 5", "port 5", "any port", "all ports", "travel time ports 5")

Citation:
Nelson A, Weiss DJ, van Etten J et al (2019). "A suite of global accessibility indicators." Scientific Data 6, 266. https://doi.org/10.1038/s41597-019-0265-5

Note: Data extent is [-180, 180, -60, 85].

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
processed <- par_set(country = "Italy", crs = 3035) %>%
accessibility(vars=c("large cities", "ports1"))
```

Alps *Alpine region*

Description

A spatial dataset (class "sf", "data.frame") containing the borders of the European Alps region, retrieved from the Alpine Convention website (https://www.alpconv.org/fr/page-daccueil/). Used for examples and testing within the envar package.

Usage

```
Alps
```

Format

An sf object :

name European Alps

geometry sfc_MULTIPOLYGON geometry

Source

Alpine Convention

Apollo — *Apollo butterfly occurrences*

Description

A "data.frame" containing the occurrence data of Apollo butterfly (Parnassius apollo), downloaded from the Global Biodiversity Information Facility website (www.gbif.org), on December 10, 2025. Used for examples and testing within the envar package.

Usage

```
Apollo
```

Format

A data.frame object :

X longitude in WGS84

Y latitude in WGS84

Source

GBIF

aridity — *Download and process Global Aridity Index and Potential Evapotranspiration (ET0)*

Description

This function downloads, processes, and extracts variables from the Global Aridity Index and ET0 Database v3. Each variable corresponds to a global raster representing aridity index or potential evapotranspiration values.

Usage

```
aridity(x, vars, ...)
```

Arguments

x	The output from ‘par_set()‘ defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use ‘par_set()‘ to define parameters for download.
vars	Character vector of one or more variables to download and process.
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Annual Variables

- "ai_v3_yr.tif" ("aridity index annual", "ai annual", "aridity annual", "ai year")
- "et0_v3_yr.tif" ("et0 annual", "potential evapotranspiration annual", "et0 year")
- "et0_v3_yr_sd.tif" ("et0 standard deviation", "et0 sd", "et0 variability")

Monthly Aridity Index

- "ai_v3_01.tif" ... "ai_v3_12.tif" ("aridity index january"..."december", "ai jan"..."dec", "ai 01"..."12")

Monthly Potential Evapotranspiration (ET0)

- "et0_v3_01.tif" ... "et0_v3_12.tif" ("et0 january"..."december", "et0 jan"..."dec", "et0 01"..."12")

Citation:
Zomer RJ, Xu J, Trabucco A (2022). "Version 3 of the Global Aridity Index and Potential Evapotranspiration Database." Scientific Data 9, 409. https://doi.org/10.1038/s41597-022-01493-1

Note: Data is downloaded from Figshare (Article ID 7504448).

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
processed <- par_set(country= "Italy", crs=3035) %>%
aridity(vars=c("aridity index annual", "et0 january"))
```

`biooracle` *Download Bio-ORACLE marine data*

Description

This function downloads, processes, and extracts variables from the Bio-ORACLE v3.0 dataset.

Usage

```
biooracle(
  x,
  vars,
  realm = "surface",
  years = "2000-2010",
  ssp = NULL,
  algorithm = "mean",
  ...
)
```

Arguments

x	The output from 'par_set()' defining the area or locations. It must have been created with 'res = 6' (Bio-ORACLE's native 0.05-degree grid).
vars	Character vector of one or more variables or synonyms to download.
realm	Character. One of "surface" (default), "benthic_minimum", "benthic_average", or "benthic_maximum".
years	Character. The time period for the data in "YYYY-YYYY" format. Use "2000-2010" or "2010-2020" for baseline current conditions (default is "2000-2010"). For future projections, specify the decade (e.g., "2040-2050", "2090-2100") and provide the 'ssp' argument.
ssp	numeric or character. Shared Socioeconomic Pathway (119, 126, 245, 370, 460, 585). Required if 'years' is in the future (>= 2020).
algorithm	Character. Statistic to apply (max, mean, min, ltmax, ltmin, range). Default "mean".
...	Additional arguments.

Details

Available variables (working synonyms in parentheses and ""):

- 1 - "thetao" (Ocean temperature; [ºC]) ("temperature", "temp", "sea temperature")
- 2 - "so" (Salinity; [-]) ("sal", "salt", "saltiness")
- 3 - "sws" (Sea water velocity; [m.s-1]) ("velocity", "current speed", "speed")
- 4 - "swd" (Sea water direction; [degree]) ("direction", "current direction")
- 5 - "no3" (Nitrate; [mmol . m-3]) ("nitrate")
- 6 - "po4" (Phosphate; [mmol . m-3]) ("phosphate")
- 7 - "si" (Silicate; [mmol . m-3]) ("silicate", "silicon")
- 8 - "o2" (Dissolved molecular oxygen; [mmol . m-3]) ("oxygen", "dissolved oxygen", "o2")
- 9 - "dfe" (Iron; [mmol . m-3]) ("iron", "fe")
- 10 - "phyc" (Primary productivity; [mmol . m-3]) ("productivity", "pp", "primary production")
- 11 - "ph" (pH; [-]) ("acidity")

- 12 - "chl" (Chlorophyll; [mg . m-3]) ("chlorophyll", "chla")
- 13 - "sithick" (Sea ice thickness; [m]) ("ice thickness")
- 14 - "siconc" (Sea ice cover; [Fraction]) ("ice cover", "sea ice")
- 15 - "clt" (Cloud cover; [
- 16 - "mlotst" (Mixed layer depth; [m]) ("mld", "mixed layer")
- 17 - "tas" (Air temperature; [ºC]) ("air temperature", "air temp")
- 18 - "par" (Photosynt. Avail. Radiation; [E.m-2.day-1]) ("light", "radiation")
- 19 - "kdpar" (Diffuse attenuation; [m-1]) ("attenuation", "turbidity")
- 20 - "bathymetry" (Bathymetry; [m]) ("depth", "elevation", "altitude")
- 21 - "slope" (Topographic slope; [-]) ("topographic slope")
- 22 - "aspect" (Topographic aspect; [-]) ("topographic aspect")
- 23 - "tpi" (Topographic position index; [-]) ("topographic position index")
- 24 - "tri" (Terrain ruggedness index; [-]) ("terrain ruggedness index", "ruggedness")

Citation:
Assis J, Fernández Bejarano SJ, Salazar VW, Schepers L, Gouvêa L, Fragkopoulou E, Leclercq F, Vanhoorne B, Tyberghein L, Serrão EA, Verbruggen H, De Clerck O (2024). "Bio-ORACLE v3.0. Pushing marine data layers to the CMIP6 Earth system models of climate change research." Global Ecology and Biogeography, 33, e13813. https://doi.org/10.1111/geb.13813

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Resolution

Bio-ORACLE layers are distributed on a 0.05-degree grid (~5.5 km at the equator). Because `res` is a multiplier of the 30 arc-second base grid, the value that reproduces this grid exactly is `res = 6` ($6 \times 30'' = 0.05°$). You must therefore call `par_set()` with `res = 6`; any other value (including the default) raises an error.

Examples

```
# Example 1: Current conditions (Baseline)
current_env <- par_set(country = "Italy", crs = 3035, res = 6) %>%
  biooracle(vars = c("temperature", "salinity"),
            years = "2000-2010")

# Example 2: Future projections (2050, SSP 585)
future_env <- par_set(country = "Italy", crs = 3035, res = 6) %>%
  biooracle(vars = c("temperature", "salinity"),
            years = "2040-2050",
            ssp = 585)
```

chelsa *Download CHELSA climate data*

Description

This function downloads, processes, and extracts variables from the CHELSA (Climatologies at High Resolution for the Earth's Land Surface Areas) dataset.

Usage

```
chelsa(
  x,
  vars,
  years = NULL,
  months = NULL,
  gcm = NULL,
  rcp = NULL,
  ssp = NULL,
  cruts_years = NULL,
  ...
)
```

Arguments

x	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
vars	Character vector of one or more variables to download and process.
years	A character or numeric vector of years or year ranges (e.g., "1981-2010", 2015).
months	A numeric vector (1–12) specifying which months to download. If NULL and 'years' are single years, all 12 months are downloaded.
gcm	General Circulation Model(s) for future projections.
rcp	Representative Concentration Pathway, given as the radiative-forcing level (e.g., `2.6`, `4.5`, `6.0`, `8.5`). For CMIP5 projections (year ranges `"2041-2060"`, `"2061-2080"`) it selects the RCP directly. For CMIP6/BIOCLIM+ projections it is combined with `ssp` to build the scenario code (e.g., `ssp = 5` and `rcp = 8.5` request the `ssp585` scenario).
ssp	Shared Socioeconomic Pathway family for CMIP6/BIOCLIM+ data (e.g., `1`, `2`, `3`, `5`). Combined with `rcp` as described above. A complete code such as `"585"` may also be supplied directly (with `rcp = NULL`).
cruts_years	Numeric vector. Years to download from CHELSAcruts (must be 1901–2016).
...	Additional arguments (currently unused).

Details

Available variables

Please note the distinction between "Monthly" time-series data and "Climatologies". Unlike other functions in this package, there is only one code-name for each variable and no working synonyms. The meaning of each variable code-name is provided in parentheses.

Monthly Time-Series (Available from 1979 onwards)

- 1 - "pr" (Precipitation amount; mass per unit area)
- 2 - "tas" (Mean daily air temperature at 2 meters)
- 3 - "tasmax" (Mean daily maximum air temperature at 2 meters)
- 4 - "tasmin" (Mean daily minimum air temperature at 2 meters)
- 5 - "hurs" (Near-surface relative humidity)
- 6 - "clt" (Total cloud cover at surface; considers entire atmospheric column)
- 7 - "sfcWind" (Near-surface wind speed at 10m above ground)
- 8 - "vpd" (Vapor pressure deficit)
- 9 - "rsds" (Surface downwelling shortwave flux in air)
- 10 - "pet_penman" (Potential evapotranspiration; Penman-Monteith equation)
- 11 - "cmi" (Climate Moisture Index)
- 12 - "swb" (Site water balance; cumulative available water)

Climatologies & Derived Indices (1981-2010, 2011-2040, 2041-2070, 2071-2100)

Cloud Cover

- 13 - "clt_mean" (Mean monthly total cloud cover over 1 year)
- 14 - "clt_max" (Maximum monthly total cloud cover)
- 15 - "clt_min" (Minimum monthly total cloud cover)
- 16 - "clt_range" (Annual range of monthly total cloud cover)

Climate Moisture Index

- 17 - "cmi_mean" (Mean monthly climate moisture index)
- 18 - "cmi_max" (Maximum monthly climate moisture index; highest surplus)
- 19 - "cmi_min" (Minimum monthly climate moisture index; highest deficit)
- 20 - "cmi_range" (Annual range of monthly climate moisture index)

Relative Humidity

- 21 - "hurs_mean" (Mean monthly near-surface relative humidity)
- 22 - "hurs_max" (Maximum monthly near-surface relative humidity)
- 23 - "hurs_min" (Minimum monthly near-surface relative humidity)
- 24 - "hurs_range" (Annual range of monthly near-surface relative humidity)

Potential Evapotranspiration

- 25 - "pet_penman_mean" (Mean monthly PET)
- 26 - "pet_penman_max" (Maximum monthly PET)
- 27 - "pet_penman_min" (Minimum monthly PET)
- 28 - "pet_penman_range" (Annual range of monthly PET)

Solar Radiation

- 29 - "rsds_mean" (Mean monthly surface downwelling shortwave flux)
- 30 - "rsds_max" (Maximum monthly surface downwelling shortwave flux)
- 31 - "rsds_min" (Minimum monthly surface downwelling shortwave flux)
- 32 - "rsds_range" (Annual range of monthly surface downwelling shortwave flux)

Wind Speed

- 33 - "sfcWind_mean" (Mean monthly near-surface wind speed)
- 34 - "sfcWind_max" (Maximum monthly near-surface wind speed)
- 35 - "sfcWind_min" (Minimum monthly near-surface wind speed)
- 36 - "sfcWind_range" (Annual range of monthly near-surface wind speed)

Vapor Pressure Deficit

- 37 - "vpd_mean" (Mean monthly vapor pressure deficit)
- 38 - "vpd_max" (Maximum monthly vapor pressure deficit)
- 39 - "vpd_min" (Minimum monthly vapor pressure deficit)
- 40 - "vpd_range" (Annual range of monthly vapor pressure deficit)

Growing Season Characteristics (TREELIM model)

- 41 - "gsl" (Growing season length; days)
- 42 - "gsp" (Accumulated precipitation during growing season)
- 43 - "gst" (Mean temperature of the growing season)
- 44 - "fgd" (First day of the growing season; Julian day)
- 45 - "lgd" (Last day of the growing season; Julian day)

Growing Degree Days (GDD)

- 46 - "gdd0" (Heat sum of all days > 0°C accumulated over 1 year)
- 47 - "gdd5" (Heat sum of all days > 5°C accumulated over 1 year)
- 48 - "gdd10" (Heat sum of all days > 10°C accumulated over 1 year)
- 49 - "ngd0" (Number of days with tas > 0°C)
- 50 - "ngd5" (Number of days with tas > 5°C)
- 51 - "ngd10" (Number of days with tas > 10°C)
- 52 - "gdgfgd0" (First growing degree day > 0°C; Julian day)
- 53 - "gdgfgd5" (First growing degree day > 5°C; Julian day)
- 54 - "gdgfgd10" (First growing degree day > 10°C; Julian day)

- 55 - "gddlgd0" (Last growing degree day > 0°C; Julian day)
- 56 - "gddlgd5" (Last growing degree day > 5°C; Julian day)
- 57 - "gddlgd10" (Last growing degree day > 10°C; Julian day)

Snow and Frost

- 58 - "scd" (Snow cover days; count)
- 59 - "swe" (Snow water equivalent; accumulated amount of liquid water if snow melted)
- 60 - "fcf" (Frost change frequency; events where tmin/tmax cross 0°C)

Biological Productivity

- 61 - "npp" (Net primary productivity; g C m^-2 yr^-1)

Climate Classifications

- 62 - "kg0" (Köppen-Geiger climate category)
- 63 - "kg1" (Köppen-Geiger without As/Aw differentiation)
- 64 - "kg2" (Köppen-Geiger after Peel et al. 2007)
- 65 - "kg3" (Wissmann 1939 classification)
- 66 - "kg4" (Thornthwaite 1931 classification)
- 67 - "kg5" (Troll-Pfaffen classification)

Citation:
Standard bioclimatic variables: Karger D, Conrad O, Böhner J et al (2017). "Climatologies at high resolution for the earth's land surface areas." Scientific Data 4, 170122. https://doi.org/10.1038/sdata.2017.122

BIOCLIM+ dataset: Brun P, Zimmermann NE, Hari C, Pellissier L, Karger DN (2022). "Global climate-related predictors at kilometer resolution for the past and future." Earth System Science Data 14, 5573-5603. https://doi.org/10.5194/essd-14-5573-2022

Note: Users should verify the terms of use for CHELSA data provided at https://chelsa-climate.org/

Value

If 'par_set()' contained a raster/polygon/points with buffer: a 'SpatRaster' stack of processed variables. If 'par_set()' contained spatial points or data.frame of points without buffer: a 'data.frame' of x, y, and extracted values.

Examples

```
# climatic values for one specific year/month
processed <- par_set(zooregion = "Madagascan") %>%
chelsa(vars=c("tas"), years = 2018, months = 1)

# climatic values for a long period (real "climate"), if months are not specified
# all the months are downloaded (12 layers per variable)
processed <- par_set(country = "Iceland") %>%
chelsa(vars=c("pr", "tas"), years = "1981-2010", months = 1)
```

```
# bioclimatic variables are available only over these extended periods, and not
# for the single years
processed <- par_set(country = "Iceland") %>%
chelsa(vars=c("bio1"), years = "1981-2010", months = 1)

# to download a specified set of variables, leave only "bio" and then the
# package will ask which variables to download (all 19 or a selection) in the console
processed <- par_set(country = "Iceland") %>%
chelsa(vars=c("bio"), years = "1981-2010", months = 1)

# climatic values for the future (SSP, RCP and GCM must be specified)
processed <- par_set(country = "Italy", crs = 3035) %>%
chelsa(vars=c("pr", "tas"), years = "2041-2070", months = 1,
 ssp = 5, rcp = 8.5, gcm = "GFDL-ESM4")
```

`clear_cache` *Clear the envar download cache*

Description

Removes all files stored in the persistent download cache used when ‘cache = TRUE‘ in [par_set()]. This is useful to free disk space or to force a fresh download of every variable.

Usage

```
clear_cache()
```

Value

Invisibly, the path of the cache directory that was cleared.

`climatezones` *Download and process Köppen-Geiger climate classification maps*

Description

This function downloads, processes, and extracts variables from the High-resolution (1 km) Köppen-Geiger maps dataset. Each variable corresponds to a global GeoTIFF representing climate classification zones based on historical data or future CMIP6 projections.

Usage

```
climatezones(x, vars = "zones", years = "1991-2020", ssp = NULL, ...)
```

Arguments

x	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
vars	Character vector. Defaults to "zones". Accepted aliases include: "koppengeiger", "climate", "climatezones", "koppen".
years	Character vector of time periods. Defaults to "1991-2020". Accepts formats with underscores or hyphens (e.g., "1901-1930" or "1901_1930").
ssp	Numeric or character vector of Shared Socioeconomic Pathways. Required for future projections (e.g., 126, 585).
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

- 1 - "zones" ("koppengeiger", "climate", "climatezones", "koppen", "koppen geiger")

Time Periods (years argument):

Historical

- "1901-1930"
- "1931-1960"
- "1961-1990"
- "1991-2020" (default)

Future

- "2041-2070"
- "2071-2099"

SSP Scenarios (ssp argument, required for future periods):

- 119 (SSP1-1.9)
- 126 (SSP1-2.6)
- 245 (SSP2-4.5)
- 370 (SSP3-7.0)
- 434 (SSP4-3.4)
- 460 (SSP4-6.0)
- 585 (SSP5-8.5)

Citation:

Beck HE, McVicar TR, Vergopolan N, Berg A, Lutsko NJ, Dufour A, Zeng Z, Jiang X, van Dijk AIJM, Miralles DG (2023). "High-resolution (1 km) Köppen-Geiger maps for 1901-2099 based on constrained CMIP6 projections." Scientific Data 10, 724. https://doi.org/10.1038/s41597-023-02549-6

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
processed <- par_set(country= "Italy", crs=3035) %>%
climatezones(vars="zones", years="1991-2020")
```

---

cloudcover *Download and process EarthEnv Global Cloud Cover layers*

Description

This function downloads, processes, and extracts variables from the EarthEnv Global Cloud Cover dataset. Each variable corresponds to a global Cloud-Optimized GeoTIFF (COG) representing cloud cover dynamics.

Usage

```
cloudcover(x, vars, ...)
```

Arguments

x	The output from ‘par_set()‘ defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use ‘par_set()‘ to define parameters for download.
vars	Character vector of one or more variables to download and process.
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Metrics

- 1 - "MODCF_CloudForestPrediction" ("cloud forest prediction", "cloud forest", "cfp")
- 2 - "MODCF_interannualSD" ("inter-annual variability", "interannual sd", "interannual variability")
- 3 - "MODCF_intraannualSD" ("intra-annual variability", "intraannual sd", "intraannual variability")
- 4 - "MODCF_meanannual" ("mean annual", "annual mean", "annual")
- 5 - "MODCF_spatialSD_1deg" ("spatial variability", "spatial sd", "spatial sd 1deg")

Seasonality

- 6 - "MODCF_seasonality_concentration" ("seasonality concentration", "concentration")
- 7 - "MODCF_seasonality_rgb" ("seasonality rgb", "rgb")
- 8 - "MODCF_seasonality_theta" ("seasonality theta", "theta")
- 9 - "MODCF_seasonality_visct" ("seasonality single band", "seasonality visct", "seasonality color")

Monthly Means

- 10 - "MODCF_monthlymean_01" ("january mean", "january", "jan")
- 11 - "MODCF_monthlymean_02" ("february mean", "february", "feb")
- 12 - "MODCF_monthlymean_03" ("march mean", "march", "mar")
- 13 - "MODCF_monthlymean_04" ("april mean", "april", "apr")
- 14 - "MODCF_monthlymean_05" ("may mean", "may")
- 15 - "MODCF_monthlymean_06" ("june mean", "june", "jun")
- 16 - "MODCF_monthlymean_07" ("july mean", "july", "jul")
- 17 - "MODCF_monthlymean_08" ("august mean", "august", "aug")
- 18 - "MODCF_monthlymean_09" ("september mean", "september", "sep")
- 19 - "MODCF_monthlymean_10" ("october mean", "october", "oct")
- 20 - "MODCF_monthlymean_11" ("november mean", "november", "nov")
- 21 - "MODCF_monthlymean_12" ("december mean", "december", "dec")

Citation:
Wilson AM, Jetz W (2016). "Remotely sensed high-resolution global cloud dynamics for predicting ecosystem and biodiversity distributions." PLoS Biol 14(3): e1002415. https://doi.org/10.1371/journal.pbio.1002415

Note: Please cite original sources of primary datasets where appropriate.

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
processed <- par_set(country= "Italy", crs=3035) %>%
cloudcover(vars=c("mean annual", "january"))
```

corr_check *Check Variable Correlation and Multi-collinearity*

Description

'corr_check()' is an optional function for the 'envar' package workflow. It allows you to check for variable correlation and multi-collinearity among the variables over the study area. It can be used before or after 'extr_check()'.

Usage

```
corr_check(x, pearson = NULL, vif = NULL)
```

Arguments

x	A 'SpatRaster', 'data.frame', or a list containing 'data' or 'extracted_df' (e.g., output from 'extr_check()').
pearson	Numeric or 'NULL'. Threshold for the absolute Pearson correlation coefficient above which variables are flagged with a warning. By default ('NULL') a default threshold of '0.7' is used; supply a value (e.g. '0.6') to be warned about variable pairs whose absolute correlation exceeds it instead.
vif	Numeric or 'NULL'. Threshold for the Variance Inflation Factor above which variables are flagged with a warning. By default ('NULL') a default threshold of '3' is used; supply a value (e.g. '5') to be warned about variables whose VIF exceeds it instead.

Details

Citation:
Wei T, Simko V (2021). "R package 'corrplot': Visualization of a Correlation Matrix." GitHub. https://github.com/taiyun/corrplot

Naimi B, Hamm NA, Groen TA, Skidmore AK, Toxopeus AG (2014). "Where is positional uncertainty a problem for species distribution modelling?" Ecography 37, 191-203. https://doi.org/10.1111/j.1600-0587.2013.00205.x

Regardless of whether 'pearson'/'vif' thresholds are set, the function writes two files: the correlation plot ('Corr_plot.png') and a table of VIF values ('VIF_table.csv'). In an interactive R session it asks, at the console, for the directory in which to store them every time it is called. In non-interactive sessions (e.g. scripts or 'R CMD check') a temporary directory is used and no prompt is shown. Their paths are returned as 'plot_path' and 'vif_path'.

Value

A 'list' object containing:

- 'data': The input environmental data used.
- 'correlation_matrix': Pearson correlation matrix.

- ‘vif‘: Variance Inflation Factor data frame.
- ‘summary‘: Character vector highlighting high correlation or VIF, based on the supplied thresholds (or the defaults of ‘0.7‘ and ‘3‘).
- ‘plot_path‘: Path to the saved correlation plot.
- ‘vif_path‘: Path to the saved VIF table.
- Any additional elements from input list (e.g., ‘extrapolation‘ from ‘extr_check()‘).

Examples

```
# corr_check() runs offline on the small example raster bundled with the
# package (a real WorldClim extract for Switzerland), so it needs no download:
switzerland <- terra::rast(
  system.file("extdata", "switzerland.tif", package = "envar")
)
cc <- corr_check(switzerland)
cc$summary

# It also accepts a plain data.frame of predictor values:
set.seed(1)
example_df <- data.frame(bio1 = rnorm(40), bio12 = rnorm(40))
example_df$bio11 <- example_df$bio1 + rnorm(40, sd = 0.2)  # correlated with bio1
corr_check(example_df)$summary


# The typical workflow chains corr_check() after downloading variables.
# Example 1: Basic usage after environmental variable extraction
processed_bilayer_corr_check <- par_set(country = "Italy", crs=3035, buffer = 10) %>%
  melc(vars=c("ice")) %>%
  chelsa(vars=c("pr"), months= 12, year=2015) %>%
  corr_check()

# Example 2: Chain with extr_check() (corr_check before extr_check)
result <- par_set(country = "Italy") %>%
  chelsa(vars = c("bio1", "bio12")) %>%
  corr_check() %>%
  extr_check(calib_points = my_points)

# Example 3: Chain with extr_check() (extr_check before corr_check)
result <- par_set(country = "Italy") %>%
  chelsa(vars = c("bio1", "bio12")) %>%
  extr_check(calib_points = my_points) %>%
  corr_check()

# Example 4: Custom thresholds for high correlation (>0.7) and VIF (>5)
result <- par_set(country = "Italy") %>%
  chelsa(vars = c("bio1", "bio12")) %>%
  corr_check(pearson = 0.7, vif = 5)
```

`earthenvlandcover` *Download and process EarthEnv land cover variables*

Description

This function downloads, processes, and extracts variables from the EarthEnv Consensus Land Cover dataset. Each variable corresponds to a global raster representing a specific land cover class at 1-km resolution.

Usage

```
earthenvlandcover(x, vars, discover = TRUE, ...)
```

Arguments

`x`	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
`vars`	Character vector of one or more variables to download and process.
`discover`	Logical. If 'TRUE' (default), downloads the version integrated with the DISCover dataset. If 'FALSE', downloads the version without DISCover integration.
`...`	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

- 1 - "consensus_full_class_1" ("evergreen deciduous needleleaf trees", "needleleaf trees", "needleleaf", "conifer")
- 2 - "consensus_full_class_2" ("evergreen broadleaf trees", "evergreen broadleaf", "broadleaf evergreen")
- 3 - "consensus_full_class_3" ("deciduous broadleaf trees", "deciduous broadleaf", "broadleaf deciduous")
- 4 - "consensus_full_class_4" ("mixed other trees", "mixed trees", "other trees", "mixed forest")
- 5 - "consensus_full_class_5" ("shrubs", "shrubland", "shrub")
- 6 - "consensus_full_class_6" ("herbaceous vegetation", "herbaceous", "grassland", "grass", "herbs")
- 7 - "consensus_full_class_7" ("cultivated and managed vegetation", "cultivated", "managed vegetation", "agriculture", "crops", "cropland")
- 8 - "consensus_full_class_8" ("regularly flooded vegetation", "flooded vegetation", "flooded", "wetland")

- 9 - "consensus_full_class_9" ("urban built up", "urban", "built up", "built-up", "artificial surface")
- 10 - "consensus_full_class_10" ("snow ice", "snow", "ice", "glacier", "permafrost")
- 11 - "consensus_full_class_11" ("barren", "barren land", "bare ground", "bare")
- 12 - "consensus_full_class_12" ("open water", "water", "water bodies")

Citation:
Tuanmu MN, Jetz W (2014). "A global 1-km consensus land-cover product for biodiversity and ecosystem modeling." Global Ecology and Biogeography 23, 1031-1045. https://doi.org/10.1111/geb.12182

Note: Users should verify the terms of use for EarthEnv data provided at https://www.earthenv.org/

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
processed <- par_set(country= "Italy", crs=3035) %>%
earthenvlandcover(vars=c("snow ice"))
```

Europe *Europe*

Description

Europe shapefile from the GADM, the Global Administrative Areas database.

Usage

```
Europe
```

Format

An sf object :

name Europe

geometry sfc_MULTIPOLYGON geometry

Source

GADM

`extr_check` *Check for Environmental Extrapolation*

Description

This function evaluates whether environmental conditions in the study area fall outside the range of conditions observed at calibration points. This helps identify areas where Species Distribution Model (SDM) predictions may be unreliable due to extrapolation.

Usage

```
extr_check(
  x,
  calib_points,
  calib_crs = "EPSG:4326",
  type = c("strict", "combinatorial")
)
```

Arguments

`x`	A 'SpatRaster', 'data.frame', or a list containing the output from previous pipeline steps (e.g., from 'corr_check()').
`calib_points`	A 'data.frame' with columns 'X' and 'Y' containing the coordinates of the calibration points. These are the locations where presence/absence or occurrence data were collected for model training.
`calib_crs`	Character or numeric. The Coordinate Reference System of the calibration points. Can be an EPSG code (e.g., '"EPSG:4326"', '4326'), an ESRI code, a PROJ4 string, or WKT. Default is '"EPSG:4326"' (WGS84).
`type`	Character vector specifying the type(s) of extrapolation to check. Options are '"strict"', '"combinatorial"', or 'c("strict", "combinatorial")' (default). • '"strict"': Uses 1 bin per variable (detects univariate extrapolation). • '"combinatorial"': Uses 5 bins per variable (detects multivariate novelty).

Details

Extrapolation types

Extrapolation can occur in two ways:

- **Strict extrapolation**: At least one variable is outside the range found in the calibration data.
- **Combinatorial extrapolation**: Each variable is within the calibration range, but the combination of predictors is new.

This function uses the environmental overlap mask approach from Zurell et al. (2012), here implemented natively within 'envar' (adapted from the 'eo_mask' method described in that work). It uses 1 bin per variable for strict extrapolation detection and 5 bins per variable for combinatorial extrapolation detection.

Citations:

Elith J, Kearney M, Phillips S (2010). "The art of modelling range-shifting species." Methods in Ecology and Evolution 1, 330-342. https://doi.org/10.1111/j.2041-210X.2010.00036.x

Zurell D, Elith J, Schroeder B (2012). "Predicting to new environments: tools for visualizing model behaviour and impacts on mapped distributions." Diversity and Distributions 18, 628-634. https://doi.org/10.1111/j.1472-4642.2012.00887.x

Value

A 'list' containing:

- All elements from the input if it was already a list (e.g., from 'corr_check()').
- 'extrapolation': Either a 'SpatRaster' with layer(s) named "strict" and/or "combinatorial" (if input was raster-based), or a 'data.frame' with additional column(s) named "strict" and/or "combinatorial" (if input was point-based). Values of 1 indicate extrapolation (novel environments), values of 0 indicate analog environments.

Examples

```
# extr_check() runs offline on the small example raster bundled with the
# package (a real WorldClim extract for Switzerland). Here we flag cells whose
# climate is novel relative to the Apollo occurrences that fall in Switzerland:
switzerland <- terra::rast(
  system.file("extdata", "switzerland.tif", package = "envar")
)
calib <- subset(Apollo, X >= 5.9 & X <= 10.5 & Y >= 45.8 & Y <= 47.8)
checked <- extr_check(switzerland, calib_points = calib, type = "strict")
checked$extrapolation


# Example 1: Check extrapolation after getting environmental variables
result <- par_set(country = "Italy") %>%
  melc(vars = c("tree", "water")) %>%
  chelsa(vars = "bio1", years = "1981-2010", month = 1) %>%
  extr_check(calib_points = my_occurrence_data)

# Example 2: Chain with corr_check()
result <- par_set(pointsdf = Apollo[1:10,]) %>%
  melc(vars = c("ice")) %>%
  chelsa(vars = "bio1", years = "1981-2010", month = 1) %>%
  corr_check() %>%
  extr_check(calib_points = calibration_data, type = "strict")

# Example 3: Check only combinatorial extrapolation
result <- par_set(country = "Germany") %>%
  worldclim(vars = c("bio1", "bio12")) %>%
  extr_check(calib_points = occ_points, type = "combinatorial")
```

`freshwater` *Download and process EarthEnv Freshwater Environmental Variables*

Description

This function downloads, processes, and extracts variables from the Near-global freshwater-specific environmental variables dataset. These variables are available at 1 km resolution and capture upstream catchment characteristics, including topography, land cover, soil, and climate.

Usage

```
freshwater(x, vars, year = NULL, month = NULL, algorithm = NULL, ...)
```

Arguments

`x`	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
`vars`	Character vector of one or more variables to download and process.
`year`	Numeric. Selected year(s) for extraction. Note that most EarthEnv freshwater variables are static layers (2015 version); this argument is primarily for consistency.
`month`	Numeric. Selected month(s) (1-12) for extraction. Only applicable to monthly variables (e.g., tmin, tmax, prec).
`algorithm`	Character. Aggregation method to filter specific bands. • For **Elevation** and **Slope**: "min" (Band 1), "max" (Band 2), "range" (Band 3), "avg" (Band 4). • For **Flow Accumulation**: "length" (Band 1), "acc" (Band 2). • For other variables: matches the string in the layer name.
`...`	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Temperature

- 1 - "monthly_tmin_average.nc" ("monthly minimum temperature average", "min temp average", "tmin avg", "tmin")
- 2 - "monthly_tmax_average.nc" ("monthly maximum temperature average", "max temp average", "tmax avg", "tmax")
- 3 - "monthly_tmin_weighted_average.nc" ("monthly minimum temperature weighted", "min temp weighted", "tmin weighted")
- 4 - "monthly_tmax_weighted_average.nc" ("monthly maximum temperature weighted", "max temp weighted", "tmax weighted")

Precipitation

- 5 - "monthly_prec_sum.nc" ("monthly upstream precipitation sum", "precipitation sum", "precip sum", "prec")
- 6 - "monthly_prec_weighted_sum.nc" ("monthly upstream precipitation weighted", "precipitation weighted", "precip weighted")

Hydroclimatic

- 7 - "hydroclim_average+sum.nc" ("hydroclimatic variables average", "hydroclim average", "hydroclim")
- 8 - "hydroclim_weighted_average+sum.nc" ("hydroclimatic variables weighted", "hydroclim weighted")

Topography

- 9 - "elevation.nc" ("upstream elevation", "elevation", "dem")
- 10 - "slope.nc" ("upstream slope", "slope")
- 11 - "flow_acc.nc" ("stream length", "flow accumulation", "flow")

Land cover

- 12 - "landcover_minimum.nc" ("upstream landcover minimum", "landcover min")
- 13 - "landcover_maximum.nc" ("upstream landcover maximum", "landcover max")
- 14 - "landcover_range.nc" ("upstream landcover range", "landcover range")
- 15 - "landcover_average.nc" ("upstream landcover average", "landcover avg", "landcover")
- 16 - "landcover_weighted_average.nc" ("upstream landcover weighted", "landcover weighted")

Geology & Soil

- 17 - "geology_weighted_sum.nc" ("upstream geology", "geology weighted", "geology")
- 18 - "soil_minimum.nc" ("upstream soil minimum", "soil min")
- 19 - "soil_maximum.nc" ("upstream soil maximum", "soil max")
- 20 - "soil_range.nc" ("upstream soil range", "soil range")
- 21 - "soil_average.nc" ("upstream soil average", "soil avg", "soil")
- 22 - "soil_weighted_average.nc" ("upstream soil weighted", "soil weighted")

Quality control

- 23 - "quality_control.nc" ("quality control", "qc")

Citation:
Domisch S, Amatulli G, Jetz W (2015). "Near-global freshwater-specific environmental variables for biodiversity analyses in 1 km resolution." Scientific Data 2, 150073. https://doi.org/10.1038/sdata.2015.73

Note: Please cite original sources of primary datasets where appropriate.

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
# Topography with algorithm filtering (keeping only the average band)
processed <- par_set(country = "Switzerland", crs = 3035) %>%
  freshwater(vars = c("elevation", "slope"), algorithm = "avg")

# Monthly Climate (January and July)
processed <- par_set(country = "Italy", crs = 3035) %>%
  freshwater(vars = "tmin", month = c(1, 7))
```

---

gcamlandcover *Download and process Global Future Land Use/Cover layers (2020-2100)*

Description

This function downloads, processes, and extracts simulated global land use and land cover (LULC) data for the period 2020-2100.

Usage

```
gcamlandcover(x, vars = "landcover", ssp = 126, year = 2020, ...)
```

Arguments

x	The output from ‘par_set()‘ defining the area or locations for extraction, the reference system, and the buffer.
vars	Character. Currently unused/ignored as this function returns the landcover map defined by ‘year‘ and ‘ssp‘, but kept for consistency. Default is "landcover".
ssp	Numeric or Character. The SSP scenario code (126, 245, 370, 434, 585). Ignored if ‘year‘ is 2020.
year	Numeric. The year of simulation (2020, 2030, 2050, 2070, 2100).
...	Additional arguments (currently unused).

Details

The data represents 1 km resolution LULC maps. The original data is in World Mercator projection and will be automatically reprojected to the CRS defined in ‘par_set()‘.

Land cover codes

- 1 - Cropland

- 2 - Forest
- 3 - Grassland
- 4 - Urban
- 5 - Barren
- 6 - Water

Available Years

- 2020, 2030, 2050, 2070, 2100

Available SSPs (Shared Socioeconomic Pathways)

- 126 (SSP1-2.6)
- 245 (SSP2-4.5)
- 370 (SSP3-7.0)
- 434 (SSP4-3.4)
- 585 (SSP5-8.5)

Citation:
Zhang T, Cheng C, Wu X (2023). "Mapping the spatial heterogeneity of global land use and land cover from 2020 to 2100 at a 1 km resolution." Scientific Data 10, 748. https://doi.org/10.1038/s41597-023-02637-7

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack. If ‘par_set()‘ contained spatial points without buffer: a ‘data.frame‘.

Examples

```
# Get Baseline (2020)
processed <- par_set(country= "Italy", crs=4326) %>%
  gcamlandcover(year = 2020)

# Get Future (SSP5-8.5 in 2050)
processed <- par_set(country= "Italy", crs=4326) %>%
  gcamlandcover(ssp = 585, year = 2050)
```

`gdppast` *Download and process Historical Real GDP and Electricity Consumption*

Description

This function downloads, processes, and extracts variables from the global 1km gridded revised real gross domestic product and electricity consumption dataset (1992–2019).

Usage

```
gdppast(x, vars, year, ...)
```

Arguments

`x`	The output from ‘par_set()‘ defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use ‘par_set()‘ to define parameters for download.
`vars`	Character vector of variables to download ("gdp" or "electricity").
`year`	Numeric vector of years to download. Available range: 1992-2019.
`...`	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Economic Metrics

- "gdp" ("gross domestic product", "real gdp", "economy", "economic output", "gross product")

Energy Metrics

- "electricity" ("electricity consumption", "energy", "energy consumption", "power", "ec", "electric")

Years available

- 1992 to 2019.

Citation:
Chen J, Gao M, Cheng S et al (2022). "Global 1 km x 1 km gridded revised real gross domestic product and electricity consumption during 1992–2019 based on calibrated nighttime light data." Scientific Data 9, 202. https://doi.org/10.1038/s41597-022-01322-5

Note: Please cite original sources of primary datasets where appropriate.

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
# Get GDP for 2000 and 2010
processed <- par_set(country= "Italy", crs=3035) %>%
  gdppast(vars="gdp", year=c(2000, 2010))

# Get Electricity and GDP for 2019
processed <- par_set(country= "Vietnam") %>%
  gdppast(vars=c("electricity", "gdp"), year=2019)
```

`geososlandcover` *Download and process Global Simulation Land Use/Cover (GEOSOS) 1 km variables*

Description

This function downloads, processes, and extracts variables from the Global Land-Use and Land-Cover Change Product (2010-2100). The dataset provides global 1 km resolution rasters based on different IPCC scenarios.

Usage

```
geososlandcover(x, vars, scenario = "A1B", year = 2010, discover = TRUE, ...)
```

Arguments

`x`	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
`vars`	Character vector of one or more variables to download and process.
`scenario`	Character. The IPCC scenario: "A1B", "A2", "B1", or "B2". Ignored if year is 2010.
`year`	Numeric or character. The year of the product: 2010, 2050, or 2100.
`discover`	Logical. Kept for compatibility (currently unused).
`...`	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Land Cover Classification

- 1 - "landcover" (Categorical: 1=Water, 2=Forest, 3=Grassland, 4=Farmland, 5=Urban, 6=Barren) ("lc", "cover", "land cover", "land cover class", "land use", "lulc", "classes")

Simulation Parameters

- **Years:** 2010, 2050, 2100 (Note: 2010 is the MODIS baseline).

- **Scenarios:** "A1B", "A2", "B1", "B2" (Ignored if year is 2010).

Citation:
Li X, Chen G, Liu X, Liang X, Wang S, Chen Y, Pei F, Xu X (2017). "A new global land-use and land-cover change product at a 1-km resolution for 2010 to 2100 based on human–environment interactions." Annals of the American Association of Geographers 107(5), 1040–1059. https://doi.org/10.1080/24694452.2017.

Value

If 'par_set()' contained a raster/polygon/points with buffer: a 'SpatRaster' stack of processed variables. If 'par_set()' contained spatial points or data.frame of points without buffer: a 'data.frame' of x, y, and extracted values.

Examples

```
# Example 1: Download land cover for Italy in 2050 under scenario A1B
processed <- par_set(country = "Italy", crs = 3035) %>%
  geososlandcover(vars = c("land cover"), year = 2050, scenario = "A1B")

# Example 2: Extract baseline (2010) values for specific points
points_df <- data.frame(ID = 1:2, x = c(12, 13), y = c(42, 43))
extracted <- par_set(data = points_df, crs = 4326) %>%
  geososlandcover(vars = "lc", year = 2010)
```

---

habitat *Download and process IUCN Habitat Classification layers*

Description

This function downloads, processes, and extracts variables from the IUCN Global Habitat Classification Fractions dataset (Jung et al., 2020). The data is available at Level 1 (broad) and Level 2 (detailed) classifications.

Usage

```
habitat(x, vars, level = 1, ...)
```

Arguments

x	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
vars	Character vector of one or more variables to download and process.
level	Integer. The classification level to download. 1 (broad) or 2 (detailed). Defaults to 1.
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Level 1 (Broad Categories)

- "100_Forest" ([Fraction]) ("forest", "100")
- "200_Savanna" ([Fraction]) ("savanna", "200")
- "300_Shrubland" ([Fraction]) ("shrubland", "300")
- "400_Grassland" ([Fraction]) ("grassland", "400")
- "500_Wetlands inland" ([Fraction]) ("wetlands inland", "wetlands", "inland wetlands", "500")
- "600_Rocky Areas" ([Fraction]) ("rocky areas", "rocky", "600")
- "800_Desert" ([Fraction]) ("desert", "800")
- "900_Marine - Neritic" ([Fraction]) ("marine neritic", "neritic", "900")
- "1000_Marine - Oceanic" ([Fraction]) ("marine oceanic", "oceanic", "1000")
- "1100_Marine - Deep Ocean Floor" ([Fraction]) ("marine deep ocean floor", "deep ocean floor", "1100")
- "1200_Marine - Intertidal" ([Fraction]) ("marine intertidal", "intertidal", "1200")
- "1400_Artificial - Terrestrial" ([Fraction]) ("artificial terrestrial", "artificial", "terrestrial artificial", "1400")

Level 2 (Detailed Categories - Selection)

- "101_Forest - Boreal" ([Fraction]) ("forest boreal", "boreal forest", "101")
- "104_Forest - Temperate" ([Fraction]) ("forest temperate", "temperate forest", "104")
- "105_Forest - Subtropical-tropical dry" ([Fraction]) ("dry forest", "tropical dry forest", "105")
- "106_Forest - Subtropical-tropical moist lowland" ([Fraction]) ("moist lowland forest", "tropical moist forest", "106")
- "107_Forest - Subtropical-tropical mangrove vegetation" ([Fraction]) ("mangrove", "mangroves", "107")
- "201_Savanna - Dry" ([Fraction]) ("dry savanna", "201")
- "303_Shrubland - Boreal" ([Fraction]) ("boreal shrubland", "303")
- "308_Shrubland - Mediterranean-type" ([Fraction]) ("mediterranean shrubland", "308")
- "401_Grassland - Tundra" ([Fraction]) ("tundra", "401")
- "1401_Arable Land" ([Fraction]) ("arable land", "cropland", "1401")
- "1405_Urban Areas" ([Fraction]) ("urban areas", "urban", "city", "1405")
- (See function code for full list of Level 2 variables)

Citation:
Jung M, Dahal PR, Butchart SHM, Donald PF, De Lamo X, Lesiv M, Kapos V, Rondinini C, Visconti P (2020). "A global map of terrestrial habitat types." Scientific Data 7, 256. https://doi.org/10.1038/s41597-020-00599-8

Note: Please cite original sources of primary datasets where appropriate.

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
# Example 1: Level 1 extraction (Forest and Artificial)
processed <- par_set(country = "Italy", crs = 3035) %>%
  habitat(vars = c("Forest", "Artificial"), level = 1)

# Example 2: Level 2 extraction (Specific biomes)
processed_l2 <- par_set(country = "Brazil", crs = 3035) %>%
  habitat(vars = c("Mangrove", "Tropical moist lowland forest"), level = 2)
```

---

heterogeneity *Download and process EarthEnv habitat heterogeneity layers*

Description

This function downloads, processes, and extracts variables from the EarthEnv habitat heterogeneity dataset (1-km resolution). Each variable corresponds to a global Cloud-Optimized GeoTIFF (COG) representing different metrics of habitat heterogeneity derived from remote sensing data.

Usage

```
heterogeneity(x, vars, ...)
```

Arguments

x	The output from ‘par_set()‘ defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use ‘par_set()‘ to define parameters for download.
vars	Character vector of one or more variables to download and process.
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

First-order statistics

- "cv" ("coefficient of variation", "coeff of variation")
- "evenness" ("even")
- "range"
- "shannon" ("shannon index", "shannon entropy")

- "simpson" ("simpson index", "simpson diversity")
- "std" ("standard deviation", "std dev")

Second-order statistics (texture metrics)

- "Contrast" ("contrast")
- "Correlation" ("correlation", "corr")
- "Dissimilarity" ("dissimilarity")
- "Entropy" ("entropy", "texture entropy")
- "Homogeneity" ("homogeneity")
- "Maximum" ("maximum", "max")
- "Uniformity" ("uniformity", "uniform")
- "Variance" ("variance", "var")

Citation:
Tuanmu M-N, Jetz W (2015). "A global, remote sensing-based characterization of terrestrial habitat heterogeneity for biodiversity and ecosystem modeling." Global Ecology and Biogeography, 24, 1329-1339. https://doi.org/10.1111/geb.12365

Note: Please cite original sources of primary datasets where appropriate.

Value

If 'par_set()' contained a raster/polygon/points with buffer: a 'SpatRaster' stack of processed variables. If 'par_set()' contained spatial points or data.frame of points without buffer: a 'data.frame' of x, y, and extracted values.

Examples

```
# Example 1: Download Shannon Index and Coefficient of Variation for the Alps
processed <- par_set(shape = Alps, crs = 3035) %>%
  heterogeneity(vars = c("shannon", "cv"))
```

---

hybridlandcover *Download and process Hybrid Global Annual 1-km IGBP Land Cover Maps*

Description

This function downloads, processes, and extracts land cover variables from the Hybrid Global Annual 1-km IGBP Land Cover Maps dataset (Luo et al., 2024). The data covers the period from 2000 to 2020.

Usage

```
hybridlandcover(x, vars = NULL, year = 2000, ...)
```

Arguments

x	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
vars	Character vector of one or more variables to download and process.
year	Integer. The year for which to download the land cover map (2000-2020). Defaults to 2000.
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

- 1 - "landcover" ("cover", "land", "lc", "igbp", "hybrid")

Citation:
Luo Y, Zhu Z, Zhao W, Li M, Chen J, Zhao P, Sun L, Zhang Y, Duanmu Z, Chen J (2024). "Hybrid Global Annual 1-km IGBP Land Cover Maps for the Period 2000–2020." Journal of Remote Sensing, 4, 0122. https://doi.org/10.34133/remotesensing.0122

Note: You must specify the 'year' argument (integer between 2000 and 2020).

Value

If 'par_set()' contained a raster/polygon/points with buffer: a 'SpatRaster' stack of processed variables. If 'par_set()' contained spatial points or data.frame of points without buffer: a 'data.frame' of x, y, and extracted values.

Examples

```
processed <- par_set(country= "Italy", crs=3035) %>%
  hybridlandcover(vars="landcover", year=2015)
```

---

melc *Download and process MacroEcological Land Cover (MELC)*

Description

This function downloads, processes, and extracts variables from the MELC 1 km Land Cover dataset, designed for biogeography and ecological modelling. Each variable corresponds to a global raster representing a specific land cover class or diversity index derived from very high-resolution imagery of the ESA.

Usage

```
melc(x, vars, discover = TRUE, ...)
```

Arguments

x	The output from ‘par_set()‘ defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use ‘par_set()‘ to define parameters for download.
vars	Character vector of one or more variables to download and process.
discover	Logical. If TRUE, creates a discovery map (unused in current implementation but kept for compatibility).
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Land Cover Classes

- 1 - "wetland" ("wetlands", "swamp", "marsh", "bog", "fen")
- 2 - "bare" ("bare ground", "bare soil", "desert", "unvegetated")
- 3 - "built" ("built area", "built up", "urban", "artificial", "impervious")
- 4 - "cropland" ("agriculture", "agricultural", "crop", "crops", "farming")
- 5 - "grass" ("grassland", "grass land", "meadow", "pasture", "prairie")
- 6 - "ice" ("snow", "snow and ice", "glacier", "ice", "permafrost")
- 8 - "mangrove" ("mangroves")
- 9 - "moss" ("mosses", "lichen", "lichens", "moss and lichen")
- 10 - "shrub" ("shrubland", "scrub", "bush", "thicket")
- 11 - "tree" ("trees", "forest", "woodland", "canopy", "canopy cover")
- 12 - "water" ("surface water", "lake", "river", "freshwater")

Diversity & Metrics

- 7 - "land_perc" ("percentage of land", "land percentage", "land cover fraction", "land fraction")
- 13 - "simpson" ("simpson index", "diversity simpson", "simpson diversity")
- 14 - "shannon" ("shannon index", "entropy", "shannon entropy", "shannon diversity")
- 15 - "evenness" ("evenness index", "pielou", "pielou evenness", "species evenness")

Citation:
Lo Parrino E, Simoncini A, Ficetola GF, Falaschi M (2025). "Global 1 km land cover for macroecological modelling from very high resolution imagery." Figshare. https://doi.org/10.6084/m9.figshare.30665069

Note: Users should verify the terms of use provided at https://figshare.com/s/4e7dee46628b530aee03

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
processed <- par_set(country= "Italy", crs=3035) %>%
  melc(vars=c("tree", "water"))
```

`par_set` *Initialize the Environmental Variable Retrieval Pipeline*

Description

`par_set()` is the entry point for the **envar** package workflow. It defines the spatial extent, resolution, and coordinate reference system (CRS) for the study area.

Usage

```
par_set(
  country = NULL,
  continent = NULL,
  shape = NULL,
  ecoregion = NULL,
  biome = NULL,
  realm = NULL,
  zooregion = NULL,
  zoorealm = NULL,
  mountain_region = NULL,
  mountain_region_cmec = NULL,
  glacier_region_19 = NULL,
  glacier_region_20 = NULL,
  freshwater_ecoregion = NULL,
  marine_ecoregion = NULL,
  marine_realm = NULL,
  marine_province = NULL,
  pelagic_province = NULL,
  pelagic_biome = NULL,
  pelagic_realm = NULL,
  pointsdf = NULL,
  alpha_hull = FALSE,
  buffer = 0,
  res = NULL,
  path = NULL,
  crs = "EPSG:4326",
  set_na = FALSE,
  scale = "medium",
  land = FALSE,
  cache = TRUE
)
```

Arguments

country	Character. The English name of a country (e.g., "Italy", "Viet Nam"). Used to generate the extent if shape is NULL.
continent	Character. The English name of a continent (e.g., "Europe", "Africa"). Used to generate the extent if shape and country are NULL.
shape	An sf object representing the study area. This can be: • **Polygons:** defining a region of interest. • **Points:** defining specific sampling locations. If shape is provided, country and continent are ignored.
ecoregion	Character. The name of a terrestrial ecoregion from Dinerstein et al. (2017). Uses the ECO_NAME column from the Ecoregions2017 dataset.
biome	Character. The name of a biome from Dinerstein et al. (2017). Uses the BIOME_NAME column from the Ecoregions2017 dataset.
realm	Character. The name of a biogeographic realm from Dinerstein et al. (2017). Uses the REALM column from the Ecoregions2017 dataset.
zooregion	Character. The name of a zoogeographic region from Holt et al. (2013). Uses the Regions column from the CMEC dataset.
zoorealm	Character. The name of a zoogeographic realm from Holt et al. (2013). Uses the Realm column from the CMEC newRealms dataset.
mountain_region	Character. The name of a mountain region from the GMBA Mountain Inventory v2.0 (Snethlage et al. 2022). Uses the MapName column.
mountain_region_cmec	Character. The name of a mountain region from the Center for Macroecology, Evolution, and Climate definition of mountain areas (Rahbek et al. 2019). Uses the Name column.
glacier_region_19	Character. The name of a glacier region based on RGI v6.0 (2017) first-order regions. Uses the RGI_CODE column.
glacier_region_20	Character. The name of a glacier region based on GTN-G 2023 first-order regions. Uses the o1region column.
freshwater_ecoregion	Character or Numeric. The FEOW_ID of a freshwater ecoregion from Abell et al. (2008). Uses the FEOW_ID column from the FEOW dataset.
marine_ecoregion	Character. The name of a marine ecoregion from Spalding et al. (2007). Uses the ECOREGION column from the MEOW dataset (TYPE == "MEOW").
marine_realm	Character. The name of a marine realm from Spalding et al. (2007). Uses the REALM column from the MEOW dataset (TYPE == "MEOW").
marine_province	Character. The name of a marine province from Spalding et al. (2007). Uses the PROVINC column from the MEOW dataset (TYPE == "MEOW").

`pelagic_province`
: Character. The name of a pelagic province from Spalding et al. (2012). Uses the `PROVINC` column from the PPOW dataset (`TYPE == "PPOW"`).

`pelagic_biome`
: Character. The name of a pelagic biome from Spalding et al. (2012). Uses the `BIOME` column from the PPOW dataset (`TYPE == "PPOW"`).

`pelagic_realm`
: Character. The name of a pelagic realm from Spalding et al. (2012). Uses the `REALM` column from the PPOW dataset (`TYPE == "PPOW"`).

`pointsdf`
: Data.frame with columns `X` and `Y` representing point coordinates.

`alpha_hull`
: Logical. If `TRUE`, creates an alpha hull polygon around the occurrence points using the `getDynamicAlphaHull` function from the **rangeBuilder** package (Rabosky et al. 2016) to model species distribution ranges. The `initialAlpha` is set to 2 and gradually increased until a polygon captures at least 99% of occurrence records. This method optimizes the balance between identifying distinct clusters as unique polygons and avoiding excessive fragmentation (Roll et al. 2017). Can be used in conjunction with `buffer` (applied after alpha hull creation) and `land` (intersects result with land boundary). Default is `FALSE`.

  References:

  - Rabosky ARD, et al. (2016). BAMMtools: an R package for the analysis of evolutionary dynamics on phylogenetic trees. Methods in Ecology and Evolution 7:701-707.
  - Roll U, et al. (2017). The global distribution of tetrapods reveals a need for targeted reptile conservation. Nature Ecology & Evolution 1:1677-1682.

`buffer`
: Numeric. A buffer distance in **kilometers** to expand or shrink the extent. The buffer is always specified in kilometers regardless of the target CRS - the function automatically converts to the appropriate units internally (degrees for geographic CRS like EPSG:4326, meters for projected CRS like EPSG:3035 or ESRI:54009).

  - **Positive values**: Expand the area outward by this distance.
  - **Negative values**: Shrink the area inward by this distance (useful for excluding coastal/border areas where data may have different characteristics).
  - For **points with positive buffer**: A circular buffer of this radius is drawn around each point, effectively converting the study area into polygons.
  - Default is `0`.

`res`
: Numeric. The target spatial resolution multiplier.

  - This controls the cell size of the output raster stack.
  - Must be a positive number `>= 1` (e.g., `1`, `5`, `10`). It is usually an integer, but fractional multipliers are allowed to match a dataset's native grid. For instance `biooracle()` requires `res = 6`, which reproduces Bio-ORACLE's native 0.05° (~5.5 km) grid exactly.
  - Default is `1` (30 arc-seconds or 0.008333333° at the equator).
  - Higher values will multiply the original 30 arcsec resolution by the specified factor.

`path`
: directory to store the result of the download/processing. Default to `NULL` (no output is stored locally). It works only if no `corr_check()` is specified. Specify the

path including the file name and the extension (e.g. `"../Out/rastername.tif"` if the final export is a `SpatRaster`; or `"../Out/extracteddataframe.csv"` if the output is a `data.frame`).

`crs` Character or Numeric. The Coordinate Reference System for the **final output**.

- Can be an EPSG code with or without prefix (e.g., `4326`, `3035`, `"EPSG:4326"`), an ESRI code (e.g., `54009`, `"ESRI:54009"`), a PROJ4 string, or WKT.
- If `NULL`, the pipeline uses the standard default WGS84 (`EPSG:4326`).
- If specified, all downstream environmental layers will be projected to this CRS after processing.
- Note: ESRI codes (53000-54999, 100000+) are automatically recognized and prefixed with "ESRI:" internally.

`set_na` Logical, with default `FALSE`. If `TRUE`, any cell that is `NA` in at least one raster is set to be `NA` in all rasters of the final `SpatRaster` object. It is useful only when the output is a `SpatRaster` and not a point extraction.

`scale` Character with value `"small"`, `"medium"`, or `"large"`. It represents the scale at which the country/continent shapefile are retrieved using the **rnaturalearthdata** package. Large implies a better definition of the borders of the shapefile (scale 1:10). The default is `"medium"`. It is useful only when setting the argument `country` or `continent`.

`land` Logical, with default `FALSE`. If `TRUE`, the extent is intersected with the global land boundary from Natural Earth (at the scale defined by the `scale` argument). This is useful for clipping marine/pelagic regions to land only, or for ensuring that buffered areas do not extend into the ocean. Note: This does not apply to point extractions (`pointsdf` without buffer).

`cache` Logical, with default `TRUE`. If `TRUE`, each source file downloaded by the downstream functions (e.g. `chelsa()`, `worldclim()`, `topography()`) is stored in a persistent per-user cache directory. If the download pipeline is interrupted (for example by a lost connection) and then re-launched, it resumes from where it stopped, reusing files that were already retrieved instead of downloading them again. Set to `FALSE` to use a temporary directory that is cleared at the end of the R session. The cache can be emptied at any time with `clear_cache`.

Details

This function does not download data itself. Instead, it creates a standardized spatial template (grid) or processes point locations that are passed to downstream functions (like `chelsa()`, `worldclim()`, `topography()`, etc.) to ensure all retrieved variables are perfectly aligned and stacked.

Value

A `list` object (class `envar_par`) containing:

- `grid`: A template `SpatRaster` defining the resolution and extent (for polygon input).
- `mask`: An `sf` object defining the exact study area boundaries (for polygon input).
- `res`: The resolution multiplier used.
- `bbox`: The bounding box of the study area.

- `crs`: The target coordinate reference system.
- `type`: The type of input (`"polygon"`, `"admin"`, or `"point"`).
- `is_global`: Logical, `TRUE` if processing global extent.
- `set_na`: Logical, `TRUE` if user wants to apply an NA mask.
- `path`: User-specified path to store the result.

How it works

1. **Extent Definition:** You can define the study area using a shapefile (`sf` object), a `country` name, a `continent` name, or various biogeographic boundary types.
2. **Resolution:** The `res` argument sets the target resolution as a multiplier of the base 30 arc-seconds (~1 km at equator).
3. **Buffering:** An optional buffer can be applied to expand the study area or create a sampling radius around points. The buffer is **always specified in kilometers**, regardless of the target CRS. The function automatically converts to the appropriate units (degrees for geographic CRS, meters for projected CRS).
4. **Output:**
   - If the input is a **polygon** (or country/continent), it returns a list containing a target `SpatRaster` grid and a vector mask.
   - If the input is **points** (without a buffer), it returns the point coordinates for extraction.
   - If the input is **points with a buffer**, it creates a polygon geometry around the points and returns a grid, allowing you to download raster data for the area surrounding your points.

Resampling and reprojection

Downstream functions align every layer to the target grid defined here using `terra::resample()`/`terra::project()`. Continuous layers are resampled with bilinear interpolation, while categorical (factor) layers automatically use nearest-neighbour to avoid creating invalid class codes. You can force a specific method for all layers with, e.g., `options(envar.resample_method = "near")` (accepted values are any `terra` resampling method, or `"auto"` for the default behaviour described above).

Examples

```
# Basic usage with a country
italy_grid <- par_set(country = "Italy")

# Download with a shapefile
processed_alps <- par_set(shape = "Alps") %>%
melc(vars=c("ice"))

# With a projected CRS and positive buffer (expand by 10 km)
italy_buffered <- par_set(country = "Italy", crs = 3035, buffer = 10)

# With a negative buffer (shrink by 10 km to exclude coastal areas)
italy_inland <- par_set(country = "Italy", crs = 3035, buffer = -10)

# Points with buffer to create extraction area
points_area <- par_set(pointsdf = Apollo, buffer = 10, crs = 4326)
```

```
# Using alpha hull to define species range from occurrence points
species_range <- par_set(pointsdf = species_occurrences, alpha_hull = TRUE)

# Alpha hull with buffer (buffer applied after alpha hull creation)
species_range_buffered <- par_set(pointsdf = species_occurrences, alpha_hull = TRUE, buffer = 50)

# Alpha hull clipped to land boundary
species_range_land <- par_set(pointsdf = species_occurrences, alpha_hull = TRUE, land = TRUE)

# Alpha hull with buffer and land intersection
species_range_full <- par_set(pointsdf = species_occurrences, alpha_hull = TRUE,
                              buffer = 25, land = TRUE)

# Using zoogeographic regions
palearctic <- par_set(zoorealm = "Palearctic")

# Using mountain regions
alps_gmba <- par_set(mountain_region = "European Alps")

# Using glacier regions
arctic_glaciers <- par_set(glacier_region_20 = "Arctic Canada North")

# Using freshwater ecoregions
danube <- par_set(freshwater_ecoregion = 404)

# Using marine ecoregions
mediterranean <- par_set(marine_realm = "Temperate Northern Atlantic")

# Using pelagic provinces
atlantic_pelagic <- par_set(pelagic_realm = "Atlantic Warm Water")

# Clip marine realm to land only
land_only <- par_set(marine_realm = "Temperate Northern Atlantic", land = TRUE)
```

`pftlandcover` *Download and process Global PFT Land Cover Projections (SSPs-RCPs)*

Description

This function downloads, processes, and extracts land cover variables from the Global 7-land-types LULC projection dataset based on plant functional types (PFT) with a 1-km resolution under socio-climatic scenarios (Chen et al., 2022).

Usage

```
pftlandcover(x, vars = NULL, year = 2025, ssp = 585, ...)
```

Arguments

x	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
vars	Character vector of one or more variables to download and process.
year	Integer. The year of the projection (2020-2100, step 5). Defaults to 2025.
ssp	Integer or Character. The SSP-RCP scenario code (119, 126, 245, 370, 434, 460, 534, 585). Defaults to 585.
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Land Cover

- 1 - "landcover" ("cover", "land", "lulc", "pft", "projection")

Citation:
Chen G, Li X, Liu X (2022). "Global land projection based on plant functional types with a 1-km resolution under socio-climatic scenarios." Scientific Data 9, 125. https://doi.org/10.1038/s41597-022-01208-6

Note: If the 'vars' argument is left empty, the function will default to downloading the land cover map.

Value

If 'par_set()' contained a raster/polygon/points with buffer: a 'SpatRaster' stack of processed variables. If 'par_set()' contained spatial points or data.frame of points without buffer: a 'data.frame' of x, y, and extracted values.

Examples

```
# Download SSP5-RCP8.5 projection for 2050
processed <- par_set(country= "Italy", crs=3035) %>%
  pftlandcover(vars="landcover", year=2050, ssp=585)
```

---

population *Download and process Global Population Projections (SSP)*

Description

This function downloads, processes, and extracts variables from the Global Population Projections dataset (Wang et al., 2022). It provides 1-km grid population distributions from 2020 to 2100 under five Shared Socioeconomic Pathways (SSPs).

Usage

```
population(x, vars, year = 2020, ssp = 1, ...)
```

Arguments

x	The output from ‘par_set()‘ defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use ‘par_set()‘ to define parameters for download.
vars	Character vector of one or more variables to download and process.
year	Numeric vector. Years to download. Available from 2020 to 2100 in 5-year intervals (e.g., c(2020, 2050)).
ssp	Numeric vector. Shared Socioeconomic Pathways to download (1, 2, 3, 4, or 5).
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

- 1 - "population" ([People]) ("pop", "inhabitants", "residents", "people", "count", "census")

Citation:
Wang X, Meng X, Long Y (2022). "Projecting 1 km-grid population distributions from 2020 to 2100 globally under shared socioeconomic pathways." Scientific Data 9, 563. doi:10.1038/s41597-02201675x

Note: Please cite original sources of primary datasets where appropriate.

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
processed <- par_set(country = "Italy", crs = 3035) %>%
  population(vars = "population", year = 2050, ssp = 2)
```

`protection` *Download and process WDPA Protected Area layers*

Description

This function downloads, processes, and extracts variables from the World Database of Protected Areas (WDPA). Each variable corresponds to a global raster representing different IUCN Management Categories of protected areas.

Usage

```
protection(x, vars, ...)
```

Arguments

`x`	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
`vars`	Character vector of one or more variables to download and process.
`...`	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

IUCN Management Categories

- "WDPA_IA" ("strict nature reserve", "strict reserve", "1a", "ia", "Ia")
- "WDPA_IB" ("wilderness area", "wilderness", "1b", "ib", "Ib")
- "WDPA_II" ("national park", "park", "2", "ii", "II")
- "WDPA_III" ("natural monument", "monument", "3", "iii", "III")
- "WDPA_IV" ("habitat species management", "habitat management", "4", "iv", "IV")
- "WDPA_V" ("protected landscape", "protected seascape", "landscape", "5", "v", "V")
- "WDPA_VI" ("sustainable use", "natural resources", "6", "vi", "VI")
- "WDPA_ALL" ("all", "combined", "full", "total", "all protected areas")

Citation:
Protected Planet (2025). "World Database of Protected Areas (WDPA)." [UNEP-WCMC]. https://www.protectedplanet.net/e

Note: Users should ensure they comply with the terms of use of the WDPA when using these data for commercial or research purposes.

Value

If 'par_set()' contained a raster/polygon/points with buffer: a 'SpatRaster' stack of processed variables. If 'par_set()' contained spatial points or data.frame of points without buffer: a 'data.frame' of x, y, and extracted values.

Examples

```
# Example 1: Download National Parks and All Protected Areas for Italy
processed <- par_set(country= "Italy", crs=3035) %>%
protection(vars=c("national park", "WDPA_ALL"))
```

`soil` *Download and process Harmonized World Soil Database v2.0*

Description

This function downloads, processes, and extracts variables from the Harmonized World Soil Database v2.0 (HWSD v2.0). The variable corresponds to a global raster file at 1 km resolution representing soil types.

Usage

```
soil(x, vars = NULL, ...)
```

Arguments

`x`	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
`vars`	Character vector of one or more variables to download and process. Defaults to "hwsd" if left empty.
`...`	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Soil Data

- 1 - "hwsd" ("soil", "type", "soiltype", "soil type")

Citation:
FAO, IIASA (2023). "Harmonized World Soil Database v2.0." Food and Agriculture Organization of the United Nations, Rome and International Institute for Applied Systems Analysis, Laxenburg, Austria. https://www.fao.org/soils-portal/data-hub/soil-maps-and-databases/harmonized-world-soil-database-v20/en/

Value

If 'par_set()' contained a raster/polygon/points with buffer: a 'SpatRaster' stack of processed variables. If 'par_set()' contained spatial points or data.frame of points without buffer: a 'data.frame' of x, y, and extracted values.

Examples

```
processed <- par_set(country= "Italy", crs=3035) %>%
soil(vars=c("soil"))
```

`soilclimate` *Download and process Soil Bioclimatic and Temperature layers*

Description

This function downloads, processes, and extracts soil bioclimatic variables and monthly soil temperatures from the Global Soil Temperature dataset (Lembrechts et al., 2022).

Usage

```
soilclimate(x, vars, depth = "0-5", ...)
```

Arguments

`x`	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
`vars`	Character vector of one or more variables to download and process.
`depth`	Character string defining the soil depth range. Options are "0-5" (default) or "5-15".
`...`	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Bioclimatic Variables

- SBIO1 - "SBIO1" [°C] ("annual mean temperature", "annual mean", "amt")
- SBIO2 - "SBIO2" [°C] ("mean diurnal range", "mean diurnal", "mdr")
- SBIO3 - "SBIO3" [Ratio] ("isothermality")
- SBIO4 - "SBIO4" [SD] ("temperature seasonality", "seasonality")
- SBIO5 - "SBIO5" [°C] ("max temperature warmest month", "max temp", "warmest month")
- SBIO6 - "SBIO6" [°C] ("min temperature coldest month", "min temp", "coldest month")
- SBIO7 - "SBIO7" [°C] ("temperature annual range", "annual range", "tar")
- SBIO8 - "SBIO8" [°C] ("mean temperature wettest quarter", "wettest quarter")
- SBIO9 - "SBIO9" [°C] ("mean temperature driest quarter", "driest quarter")
- SBIO10 - "SBIO10" [°C] ("mean temperature warmest quarter", "warmest quarter")
- SBIO11 - "SBIO11" [°C] ("mean temperature coldest quarter", "coldest quarter")

Monthly Mean Soil Temperatures

- soilT01 - "soilT01" [°C] ("january mean", "january", "jan")
- soilT02 - "soilT02" [°C] ("february mean", "february", "feb")
- soilT03 - "soilT03" [°C] ("march mean", "march", "mar")
- soilT04 - "soilT04" [°C] ("april mean", "april", "apr")
- soilT05 - "soilT05" [°C] ("may mean", "may")
- soilT06 - "soilT06" [°C] ("june mean", "june", "jun")
- soilT07 - "soilT07" [°C] ("july mean", "july", "jul")
- soilT08 - "soilT08" [°C] ("august mean", "august", "aug")
- soilT09 - "soilT09" [°C] ("september mean", "september", "sep")
- soilT10 - "soilT10" [°C] ("october mean", "october", "oct")
- soilT11 - "soilT11" [°C] ("november mean", "november", "nov")
- soilT12 - "soilT12" [°C] ("december mean", "december", "dec")

Citation:
Lembrechts JJ et al. (2022)."Global maps of soil temperature." Global Change Biology 28, 3110-3144. doi:10.1111/gcb.16060

Note: Please cite original sources of primary datasets where appropriate.

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
processed <- par_set(country= "Italy", crs=3035) %>%
soilclimate(vars=c("SBIO1", "SBIO10"), depth="5-15")
```

`spectre` *Download and process SPECTRE environmental threat layers*

Description

This function downloads, processes, and extracts variables from the SPECTRE — Spatially Explicit ECosysTem ThREats dataset. Each variable corresponds to a global Cloud-Optimized GeoTIFF (COG) representing a different anthropogenic or climatic threat.

Usage

```
spectre(x, vars, ...)
```

Arguments

x	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
vars	Character vector of one or more variables to download and process.
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Land Use and Human Pressure

- 1 - "1_1_MINING_AREA_cog" ("mining area", "mining_area", "mining")
- 2 - "1_2_HAZARD_POTENTIAL_cog" ("hazard potential", "hazard")
- 3 - "1_3_HUMAN_DENSITY_cog" ("human density", "population", "pop")
- 4 - "1_4_BUILT_AREA_cog" ("built area", "built")
- 5 - "1_5_ROAD_DENSITY_cog" ("road density", "roads", "road")
- 6 - "1_6_FOOTPRINT_PERC_cog" ("human footprint", "footprint")
- 7 - "1_7_IMPACT_AREA_cog" ("impacted area", "impact area")
- 8 - "1_8_MODIF_AREA_cog" ("modified area", "modif area")
- 9 - "1_9_HUMAN_BIOMES_cog" ("human biomes", "biomes")
- 10 - "1_10_FIRE_OCCUR_cog" ("fires", "fire")
- 11 - "1_11_CROP_PERC_UNI_cog" ("crops uni", "crop uni", "crop")
- 12 - "1_12_CROP_PERC_IIASA_cog" ("crops iiasa", "iiasa crops")
- 13 - "1_13_LIVESTOCK_MASS_cog" ("livestock", "livestock mass")

Forest Loss

- 14 - "2_1_FOREST_LOSS_PERC_cog" ("forest loss")
- 15 - "2_2_FOREST_TREND_cog" ("forest trend")

Light Pollution

- 16 - "3_1_LIGHT_MCDM2_cog" ("light at night", "night light", "light")

Climate Change

- 17 - "5_1_TEMP_TRENDS_cog" ("temperature trends", "temp trends")
- 18 - "5_2_TEMP_SIGNIF_cog" ("temperature significance", "temp signif")
- 19 - "5_3_CLIM_EXTREME_cog" ("climate extremes")
- 20 - "5_4_CLIM_VELOCITY_cog" ("climate velocity", "velocity")
- 21 - "5_5_ARIDITY_TREND_cog" ("aridity trend", "aridity")

Citation:
Branco VV, Capinha C, Rocha J, Correia L, Cardoso P (2024). "SPECTRE: standardized global spatial data on terrestrial SPecies and ECosystems ThREats." Global Ecology and Biogeography, 34, e13949. doi:10.1111/geb.13949

Note: Many SPECTRE variables are derived from external primary datasets. Users should consult and cite the original sources listed in the SPECTRE supplementary materials.

Value

If 'par_set()' contained a raster/polygon/points with buffer: a 'SpatRaster' stack of processed variables. If 'par_set()' contained spatial points or data.frame of points without buffer: a 'data.frame' of x, y, and extracted values.

Examples

```
processed <- par_set(country= "Italy", crs=3035) %>%
  spectre(vars=c("forest loss", "light at night"))
```

---

topography *Download and process EarthEnv Topography layers*

Description

This function downloads, processes, and extracts variables from the EarthEnv Topography dataset. This dataset provides global, cross-scale topographic variables suitable for biodiversity and ecosystem modeling.

Usage

```
topography(x, vars, algorithm = "md", topo_source = "GMTED", ...)
```

Arguments

x	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
vars	Character vector of one or more variables to download and process.
algorithm	Character. The aggregation method/algorithm to use. • Common options: "md" (median, default), "mn" (mean), "min", "max", "sd". • Note: These codes directly affect the downloaded filename (e.g., _1KMmd_).
topo_source	Character. The source of the data. • "GMTED" (Global Multi-resolution Terrain Elevation Data) - Default. • "SRTM" (Shuttle Radar Topography Mission).
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

[Topography Variables]

- 1 - "elevation" ([m]) ("dem", "height", "alt", "altitude")
- 2 - "slope" ([degrees])
- 3 - "aspect" ([degrees])
- 4 - "roughness" ([Index]) ("rough")
- 5 - "tri" ([Index]) ("terrain ruggedness index", "ruggedness")
- 6 - "tpi" ([Index]) ("topographic position index", "position")
- 7 - "vrm" ([Index]) ("vector ruggedness measure")
- 8 - "pcurv" ([radians/m]) ("profile curvature", "profile curve")
- 9 - "tcurv" ([radians/m]) ("tangential curvature", "tangential curve")
- 10 - "eastness" ([Index]) ("east")
- 11 - "northness" ([Index]) ("north")

Citation:
Amatulli G, Domisch S, Tuanmu M-N, Parmentier B, Ranipeta A, Malczyk J, Jetz W (2018). "A suite of global, cross-scale topographic variables for environmental and biodiversity modeling." Scientific Data 5, 180040. https://doi.org/10.1038/sdata.2018.40

Note: Please cite original sources of primary datasets where appropriate.

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
# Example 1: Download elevation and slope for Italy
processed <- par_set(country = "Italy", crs = 3035) %>%
  topography(vars = c("elevation", "slope"))
```

`worldclim` *Download and process WorldClim Climate Data (Historical & Future)*

Description

This function downloads, processes, and extracts variables from the WorldClim climate dataset. Each variable corresponds to a global raster representing climate variables at approximately 1-km resolution. It supports both Historical (v2.1, 1970-2000) and Future (CMIP6) data.

Usage

```
worldclim(
  x,
  vars,
  years = NULL,
  months = NULL,
  gcm = NULL,
  rcp = NULL,
  ssp = NULL,
  ...
)
```

Arguments

x	The output from 'par_set()' defining the area or locations for extraction, the reference system, and the buffer. Leave this empty and use 'par_set()' to define parameters for download.
vars	Character vector of one or more variables to download and process.
years	Character vector of years or periods. For Historical: use "1970-2000" or "historical". For Future: use "2021-2040", "2041-2060", "2061-2080", "2081-2100".
months	Numeric vector (1-12) specifying which months to download. Only applies to monthly historical variables. Ignored for bioclimatic variables, elevation, or future data.
gcm	Character vector of General Circulation Models (for Future data).
rcp	Numeric or character vector of Representative Concentration Pathways, given as the radiative-forcing level (e.g., `2.6`, `4.5`, `7.0`, `8.5`). Combined with `ssp` to build the CMIP6 scenario code (e.g., `ssp = 5` and `rcp = 8.5` download the `ssp585` scenario). If `NULL`, `ssp` is assumed to already encode the full scenario (e.g., `"585"`).
ssp	Numeric or character vector of Shared Socioeconomic Pathway families (e.g., `1`, `2`, `5`). Combined with `rcp` as described above. A complete code such as `"585"` may also be supplied directly (with `rcp = NULL`).
...	Additional arguments (currently unused).

Details

Available variables (working synonyms in parentheses):

Temperature

- "tmin" ("min temp")
- "tmax" ("max temp")
- "tavg" ("average temp")

Precipitation

- "prec" ("precipitation", "pr")

Physical

- "srad" ("solar radiation")
- "wind" ("wind speed")
- "vapr" ("water vapor")
- "elev" ("elevation")

Bioclimatic

- "bio" (all 19 bioclimatic variables), or specific e.g., "bio1", "bio12"

Citation:
Fick SE, Hijmans RJ (2017). "WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas." International Journal of Climatology 37: 4302-4315. https://doi.org/10.1002/joc.5086

Value

If ‘par_set()‘ contained a raster/polygon/points with buffer: a ‘SpatRaster‘ stack of processed variables. If ‘par_set()‘ contained spatial points or data.frame of points without buffer: a ‘data.frame‘ of x, y, and extracted values.

Examples

```
processed <- par_set(country = "Italy", crs = 3035) %>%
  worldclim(vars = c("tmin", "bio1"), years = "1970-2000")
```

Index