### Package ‘pastclim’

November 2, 2023

**Type** Package  
**Title** Manipulate Time Series of Palaeoclimate Reconstructions  
**Version** 2.0.0  
**Maintainer** Andrea Manica <am315@cam.ac.uk>  
**Description** Methods to easily extract and manipulate palaeoclimate reconstructions for ecological and anthropological analyses, as described in Leonardi et al. (2023) [doi:10.1111/ecog.06481].  
**License** CC BY 4.0  
**Language** en-GB  
**BugReports** https://github.com/EvolEcolGroup/pastclim/issues  
**Encoding** UTF-8  
**LazyData** true  
**LazyDataCompression** xz  
**RoxygenNote** 7.2.3  
**Depends** R (>= 4.0.0), terra (>= 1.7.18)  
**Imports** curl, lubridate, methods, ncdf4, utils  
**Suggests** rmarkdown, knitr, ggplot2, sf, testthat (>= 3.0.0), spelling  
**VignetteBuilder** knitr  
**Config/testthat/edition** 3  
**NeedsCompilation** no  
**Author** Michela Leonardi [aut], Emily Y. Hallet [ctb], Robert Beyer [ctb], Mario Krapp [ctb], Andrea Manica [aut, cre]  
**Repository** CRAN  
**Date/Publication** 2023-11-02 14:10:02 UTC
R topics documented:

Barreto2023 ......................................................... 3
Beyer2020 ............................................................... 3
bioclim_vars ......................................................... 4
clean_data_path .................................................... 6
climate_for_locations ............................................. 6
climate_for_time_slice ........................................... 7
df_from_region_series .......................................... 7
df_from_region_slice ............................................. 8
distance_from_sea .................................................. 8
download_dataset .................................................. 9
Example ................................................................. 9
get_available_datasets .......................................... 10
get_biome_classes ............................................... 10
get_data_path ...................................................... 11
get_downloaded_datasets ....................................... 11
get_file_for_dataset ............................................. 12
get_ice_mask ......................................................... 12
get_land_mask ....................................................... 13
get_mis_time_steps ............................................... 13
get_time_bp_steps ................................................. 14
get_varname ......................................................... 14
get_vars_for_dataset ........................................... 15
HYDE_3.3_baseline ................................................. 16
is_region_series ................................................. 16
Krapp2021 ............................................................. 17
list_available_datasets ......................................... 18
location_series .................................................... 18
location_slice ...................................................... 20
location_slice_from_region_series ......................... 21
mis_boundaries ..................................................... 22
pastclim .............................................................. 23
region_extent ....................................................... 23
region_outline ..................................................... 24
region_outline_union ........................................... 24
region_series ....................................................... 25
region_slice ........................................................ 26
sample_region_series ........................................... 27
sample_region_slice ............................................ 27
set_data_path ........................................................ 28
slice_region_series .............................................. 29
time_bp .............................................................. 29
time_series_for_locations ..................................... 30
update_dataset_list ............................................. 31
validate_nc .......................................................... 31
var_labels ........................................................... 32
WorldClim_2.1 ......................................................... 33
Description

Spatio-temporal series of monthly temperature and precipitation and 17 derived bioclimatic variables covering the last 5 Ma (Pliocene–Pleistocene), at intervals of 1,000 years, and a spatial resolution of 1 arc-degrees (see Barreto et al., 2023 for details).

Details

PALEO-PGEM-Series is downscaled to 1 × 1 arc-degrees spatial resolution from the outputs of the PALEO-PGEM emulator (Holden et al., 2019), which emulates reasonable and extensively validated global estimates of monthly temperature and precipitation for the Plio-Pleistocene every 1 kyr at a spatial resolution of ~5 × 5 arc-degrees (Holden et al., 2016, 2019).

PALEO-PGEM-Series includes the mean and the standard deviation (i.e., standard error) of the emulated climate over 10 stochastic GCM emulations to accommodate aspects of model uncertainty. This allows users to estimate the robustness of their results in the face of the stochastic aspects of the emulations. For more details, see Section 2.4 in Barreto et al. (2023).

Note that this is a very large dataset, with 5001 time slices. It takes approximately 1 minute to set up each variable when creating a region_slice or region_series. However, once the object has been created, other operations tend to be much faster (especially if you subset the dataset to a small number of time steps of interest).

If you use this dataset, make sure to cite the original publications:


Description

This dataset covers the last 120k years, at intervals of 1/2 k years, and a resolution of 0.5 degrees in latitude and longitude.
Details

If you use this dataset, make sure to cite the original publication:
Beyer, R.M., Krapp, M. & Manica, A. High-resolution terrestrial climate, bioclimate and vegetation for the last 120,000 years. Sci Data 7, 236 (2020). doi:10.1038/s4159702005521

The version included in pastclim has the ice sheets masked, as well as internal seas (Black and Caspian Sea) removed. The latter are based on:

As there is no reconstruction of their depth through time, modern outlines were used for all time steps.

Also, for bio15, the coefficient of variation was computed after adding one to monthly estimates, and it was multiplied by 100 following https://pubs.usgs.gov/ds/691/ds691.pdf

Changelog
v1.1.0 Added monthly variables. Files can be downloaded from: https://zenodo.org/deposit/7062281

v1.0.0 Remove ice sheets and internal seas, and use correct formula for bio15. Files can be downloaded from: doi:10.6084/m9.figshare.19723405.v1

| bioclim_vars | Compute bioclimatic variables |

Description

Function to compute "bioclimatic" variables from monthly average temperature and precipitation data. For modern data, this variables are generally computed using min and maximum temperature, but for many palaeoclimatic reconstructions only average temperature is available. Most variables, with the exception of BIO02 and BIO03, can be rephrased meaningfully in terms of mean temperature. This function is a modified version of predicts::bcvars.

Usage

bioclim_vars(prec, tavg, ...)

## S4 method for signature 'numeric,numeric'
bioclim_vars(prec, tavg)

## S4 method for signature 'SpatRaster,SpatRaster'
bioclim_vars(prec, tavg, filename = "", ...)

## S4 method for signature 'SpatRasterDataset,SpatRasterDataset'
bioclim_vars(prec, tavg, filename = "", ...)

## S4 method for signature 'matrix,matrix'
bioclim_vars(prec, tavg)
**Arguments**

- **prec**  
  monthly precipitation

- **tavg**  
  monthly average temperatures

- **...**  
  additional variables for specific methods

- **filename**  
  filename where the raster can be stored.

**Details**

The variables are:

- BIO01 = Annual Mean Temperature
- BIO04 = Temperature Seasonality (standard deviation x 100)
- BIO05 = Max Temperature of Warmest Month
- BIO06 = Min Temperature of Coldest Month
- BIO07 = Temperature Annual Range (P5-P6)
- BIO08 = Mean Temperature of Wettest Quarter
- BIO09 = Mean Temperature of Driest Quarter
- BIO10 = Mean Temperature of Warmest Quarter
- BIO11 = Mean Temperature of Coldest Quarter
- BIO12 = Annual Precipitation
- BIO13 = Precipitation of Wettest Month
- BIO14 = Precipitation of Driest Month
- BIO15 = Precipitation Seasonality (Coefficient of Variation)
- BIO16 = Precipitation of Wettest Quarter
- BIO17 = Precipitation of Driest Quarter
- BIO18 = Precipitation of Warmest Quarter
- BIO19 = Precipitation of Coldest Quarter

These summary Bioclimatic variables are after:


and expanded following the ANUCLIM manual

**Value**

the bioclim variables
clean_data_path

Clean the data path

Description
This function deletes old reconstructions that have been superseded in the data_path. It assumes that the only files in data_path are part of pastclim (i.e. there are no custom datasets stored in that directory).

Usage
clean_data_path(ask = TRUE)

Arguments
ask boolean on whether the user should be asked before deleting

Value
TRUE if files are deleted successfully

climatic_for_locations Extract local climate for one or more locations for a given time slice.

Description
Deprecated version of location_slice()

Usage
climate_for_locations(...) climate_for_locations(...) arguments to be passed to location_slice()

Value
a data.frame with the climatic variables of interest
a data.frame with the climatic variables of interest
climate_for_time_slice

Extract a climate slice for a region

Description

Deprecated version of region_slice()

Usage

climate_for_time_slice(...)

Arguments

... arguments to be passed to region_slice()

Value

a SpatRaster terra::SpatRaster object, with each variable as a layer.

df_from_region_series

Extract data frame from a region series

Description

Extract the climatic information from a region series and organise them as a data frame.

Usage

df_from_region_series(x, xy = TRUE)

Arguments

x climate time series generated with region_series()
xy a boolean whether x and y coordinates should be added to the dataframe (default to TRUE)

Details

To extract a data frame from a region slice, see df_from_region_slice().

Value

a data.frame where each cell each raster layer (i.e. timestep) is a row, and the available variables are columns.
df_from_region_slice  Extract data frame from a region slice

Description
Extract the climatic information from a region slice and organise it as a data frame. This is just a
wrapper around terra::as.data.frame().

Usage
df_from_region_slice(x, xy = TRUE)

Arguments
- x: climate time slice (i.e. a terra::SpatRaster) generated with region_slice()
- xy: a boolean whether x and y coordinates should be added to the dataframe (default to TRUE)

Details
To extract a data frame from a region series, see df_from_region_series().

Value
a data.frame where each cell the raster is a row, and the available variables are columns.

distance_from_sea  Compute a raster of distances from the sea for each land pixel.

Description
Get the land mask for a dataset at a given time point, and compute distance from the sea for each
land pixel.

Usage
distance_from_sea(time_bp = NULL, time_ce = NULL, dataset)

Arguments
- time_bp: time slice in years before present (negative)
- time_ce: time slice in years CE. Only one of time_bp or time_ce should be used.
- dataset: string defining the dataset to use (a list of possible values can be obtained with
list_available_datasets()). This function will not work on custom datasets.
Value

A `terra::SpatRaster` of distances from the coastline in km.

### Description

This function downloads palaeoclimate reconstructions. Files will be stored in the data path of `pastclim`, which can be inspected with `get_data_path()` and changed with `set_data_path()`.

### Usage

```r
download_dataset(dataset, bio_variables = NULL, annual = TRUE, monthly = FALSE)
```

### Arguments

- **dataset**: string defining dataset to be downloaded (a list of possible values can be obtained with `list_available_datasets()`). This function will not work on custom datasets.
- **bio_variables**: one or more variable names to be downloaded. If left to NULL, all variables available for this dataset will be downloaded (the parameters `annual` and `monthly`, see below, define which types).
- **annual**: boolean to download annual variables
- **monthly**: boolean to download monthly variables

### Value

TRUE if the dataset(s) was downloaded correctly.

### Example

Documentation for the Example dataset

### Description

This dataset is a subset of Beyer2020, used for the vignette of pastclim. Do not use this dataset for any real work, as it might not reflect the most up-to-date version of Beyer2020.
get_available_datasets

Get the available datasets.

Description

List the datasets available in pastclim, which can be passed to functions in pastclim as values for the parameter dataset. Most functions can also be used on custom datasets by setting dataset="custom"

Usage

get_available_datasets()

Details

This function provides a user-friendly list, summarising the many datasets available from WorldClim. A comprehensive list of all available datasets can be obtained with list_available_datasets.

Value

a character vector of the available datasets

gte_biome_classes

Get the biome classes for a dataset.

Description

Get a full list of biomes and how their id as coded in the biome variable for a given dataset.

Usage

gte_biome_classes(dataset)

Arguments

- dataset: string defining dataset to be downloaded (a list of possible values can be obtained with list_available_datasets()). This function will not work on custom datasets.

Value

a data.frame with columns id and category.
get_data_path

Get the data path where climate reconstructions are stored.

Description
This function returns the path where climate reconstructions are stored.

Usage
get_data_path(silent = FALSE)

Arguments
   silent boolean on whether a message is returned when data_path is not set (i.e. equal to NULL)

Details
The path is stored in an option for pastclim named data_path. If a configuration file was saved when using set_data_path(), the path is retrieved from a file named "pastclim_data.txt", which is found in the directory returned by tools::R_user_dir("pastclim","config") (i.e. the default configuration directory for the package as set in R >= 4.0).

Value
the data path

get_downloaded_datasets

Get the variables downloaded for each dataset.

Description
List the downloaded variable for each dataset.

Usage
get_downloaded_datasets(data_path = NULL)

Arguments
data_path leave it to NULL to use the default data_path

Value
a list of variable names per dataset.
get_file_for_dataset  
*Get the file details for a variable and dataset.*

**Description**

Internal getter function

**Usage**

```r
get_file_for_dataset(variable, dataset)
```

**Arguments**

- **variable**: one or more variable names to be downloaded
- **dataset**: string defining dataset to be downloaded (a list of possible values can be obtained with `list_available_datasets()`). This function will not work on custom datasets.

**Value**

the filename for that variable and dataset

---

get_ice_mask  
*Get the ice mask for a dataset.*

**Description**

Get the ice mask for a dataset, either for the whole series or for specific time points.

**Usage**

```r
get_ice_mask(time_bp = NULL, dataset)
```

**Arguments**

- **time_bp**: time slices in years before present (negative values represent time before present, positive values time in the future). This parameter can be a vector of times (the slices need to exist in the dataset), a list with a min and max element setting the range of values, or left to NULL to retrieve all time steps. To check which slices are available, you can use `get_time_bp_steps()`.
- **dataset**: string defining dataset to be downloaded (a list of possible values can be obtained with `list_available_datasets()`). This function will not work on custom datasets.

**Value**

a binary `terra::SpatRaster` with the ice mask as 1s
### get_land_mask

Get the land mask for a dataset.

**Description**

Get the land mask for a dataset, either for the whole series or for specific time points.

**Usage**

```r
get_land_mask(time_bp = NULL, time_ce = NULL, dataset)
```

**Arguments**

- **time_bp**: time slices in years before present (negative values represent time before present, positive values time in the future). This parameter can be a vector of times (the slices need to exist in the dataset), a list with a min and max element setting the range of values, or left to NULL to retrieve all time steps. To check which slices are available, you can use `get_time_bp_steps()`.
- **time_ce**: time in years CE as an alternative to `time_bp`. Only one of `time_bp` or `time_ce` should be used. For available time slices in years CE, use `get_time_ce_steps()`.
- **dataset**: string defining dataset to be downloaded (a list of possible values can be obtained with `list_available_datasets()`). This function will not work on custom datasets.

**Value**

A binary `terra::SpatRaster` with the land mask as 1s.

### get_mis_time_steps

Get time steps for a given MIS.

**Description**

Get the time steps available in a given dataset for a MIS.

**Usage**

```r
get_mis_time_steps(mis, dataset, path_to_nc = NULL)
```

**Arguments**

- **mis**: string giving the mis; it must use the same spelling as used in `mis_boundaries`.
- **dataset**: string defining dataset to be downloaded (a list of possible values can be obtained with `list_available_datasets()`). If set to "custom", then a single nc file is used from "path_to_nc".
- **path_to_nc**: the path to the custom nc file containing the palaeoclimate reconstructions. All the variables of interest need to be included in this file.
### get_time_bp_steps

**Description**
Get time steps for a given dataset.

**Usage**
```
get_time_bp_steps(dataset, path_to_nc = NULL)
get_time_ce_steps(dataset, path_to_nc = NULL)
get_time_steps(dataset, path_to_nc = NULL)
```

**Arguments**
- **dataset**: string defining dataset to be downloaded (a list of possible values can be obtained with `list_available_datasets()`). If set to "custom", then a single nc file is used from "path_to_nc".
- **path_to_nc**: the path to the custom nc file containing the palaeoclimate reconstructions. All the variables of interest need to be included in this file.

**Value**
a vector of time steps (in time_bp or time_ce)

### get_varname

**Description**
Get a the varname for this variable

**Usage**
```
get_varname(variable, dataset)
```

**Arguments**
- **variable**: string defining the variable name
- **dataset**: string defining dataset to be downloaded
get_vars_for_dataset

Value
the name of the variable

---

**get_vars_for_dataset**  
*Get a list of variables for a given dataset.*

Description

This function lists the variables available for a given dataset. Note that the spelling and use of capitals in names might differ from the original publications, as pastclim harmonises the names of variables across different reconstructions.

Usage

```r
get_vars_for_dataset(
  dataset,
  path_to_nc = NULL,
  details = FALSE,
  annual = TRUE,
  monthly = FALSE
)
```

Arguments

- **dataset**: string defining dataset to be downloaded (a list of possible values can be obtained with `list_available_datasets()`).
- **path_to_nc**: the path to the custom nc file containing the palaeoclimate reconstructions. If a custom nc file is given, 'details', 'annual' and 'monthly' are ignored.
- **details**: boolean determining whether the output should include information including long names of variables and their units.
- **annual**: boolean to show annual variables
- **monthly**: boolean to show monthly variables

Value

- a vector of variable names
HYDE_3.3_baseline  Documentation for HYDE 3.3 dataset

Description

This database presents an update and expansion of the History Database of the Global Environment (HYDE, v 3.3) and replaces former HYDE 3.2 version from 2017. HYDE is an internally consistent combination of updated historical population estimates and land use. Categories include cropland, with a new distinction into irrigated and rain fed crops (other than rice) and irrigated and rain fed rice. Also grazing lands are provided, divided into more intensively used pasture, converted rangeland and non-converted natural (less intensively used) rangeland. Population is represented by maps of total, urban, rural population and population density as well as built-up area.

Details

The period covered is 10 000 BCE to 2023 CE. Spatial resolution is 5 arc minutes (approx. 85 km2 at the equator). The full HYDE 3.3 release contains: a Baseline estimate scenario, a Lower estimate scenario and an Upper estimate scenario. Currently only the baseline scenario is available in pastclim

If you use this dataset, make sure to cite the original publication for the HYDE 3.2 (there is no current publication for 3.3):


is_region_series  Check the object is a valid region series

Description

A region series is a terra::SpatRasterDataset for which each sub-dataset is a variable, and all variables have the same number of time steps.

Usage

is_region_series(x, strict = FALSE)

Arguments

x  a terra::SpatRasterDataset representing a time series of regional reconstructions obtained from region_series().

strict  a boolean defining whether to perform a thorough test (see description above for details).
Details

The standard test only checks that all sub-datasets (each of which is a terra::SpatRaster) have the same number of layers. The more thorough test (obtained with strict=TRUE) actually checks that all variables have the same identical time steps by comparing the result of terra::time() applied to each variable.

Value

TRUE if the object is a region series

Description

This dataset covers the last 800k years, at intervals of 1k years, and a resolution of 0.5 degrees in latitude and longitude.

Details

The units of several variables have been changed to match what is used in WorldClim.

If you use this dataset, make sure to cite the original publication:


The version included in pastclim has the ice sheets masked.

Note that, for bio15, we use the corrected version, which follows https://pubs.usgs.gov/ds/691/ds691.pdf

Changelog

v1.4.0 Change units to match WorldClim. Fix variable duplication found on earlier versions of the dataset. https://zenodo.org/records/8415273

v1.1.0 Added monthly variables. Files can be downloaded from: https://zenodo.org/record/7056055

v1.0.0 Remove ice sheets and use correct formula for bio15. Files can be downloaded from: doi:10.6084/m9.figshare.19733680.v1
### list_available_datasets

List all the available datasets.

**Description**

List the datasets available in pastclim. The list is comprehensive, and includes all combinations of models and future scenarios for WorldClim. For a more user-friendly list, use `get_available_datasets()`. Most functions can also be used on custom datasets by setting `dataset="custom"`.

**Usage**

```r
list_available_datasets()
```

**Value**

a character vector of the available datasets

### location_series

Extract a time series of bioclimatic variables for one or more locations.

**Description**

This function extract a time series of local climate for a set of locations. Note that this function does not apply any interpolation (as opposed to `location_slice()`). If you have a coastal location that just falls into the water for the reconstructions, you will have to amend the coordinates to put it more firmly on land.

**Usage**

```r
location_series(
  x,
  time_bp = NULL,
  time_ce = NULL,
  coords = NULL,
  bio_variables,
  dataset,
  path_to_nc = NULL,
  nn_interpol = FALSE,
  buffer = FALSE,
  directions = 8
)
```
Arguments

- **x**
a data.frame with columns of x and y coordinates (and an optional name column), or a vector of cell numbers. See coords for standard coordinate names, or how to use custom ones.

- **time_bp**
time slices in years before present (negative values represent time before present, positive values time in the future). This parameter can be a vector of times (the slices need to exist in the dataset), a list with a min and max element setting the range of values, or left to NULL to retrieve all time steps. To check which slices are available, you can use `get_time_bp_steps()`.

- **time_ce**
time slice in years CE (see time_bp for options). For available time slices in years CE, use `get_time_ce_steps()`. Only one of time_bp or time_ce should be used.

- **coords**
a vector of length two giving the names of the "x" and "y" coordinates, as found in data. If left to NULL, the function will try to guess the columns based on standard names `c("x", "y")`, `c("X","Y")`, `c("longitude", "latitude")`, or `c("lon", "lat")`

- **bio_variables**
vector of names of variables to be extracted.

- **dataset**
string defining the dataset to use. If set to "custom", then a single nc file is used from "path_to_nc"

- **path_to_nc**
the path to the custom nc file containing the palaeoclimate reconstructions. All the variables of interest need to be included in this file.

- **nn_interpol**
boolean determining whether nearest neighbour interpolation is used to estimate climate for cells that lack such information (i.e. they are under water or ice). By default, interpolation is only performed from the first ring of nearest neighbours; if climate is not available, NA will be returned for that location. The number of neighbours can be changed with the argument directions. `nn_interpol` defaults to FALSE (this is DIFFERENT from `location_slice()`).

- **buffer**
boolean determining whether the variable will be returned as the mean of a buffer around the focal cell. If set to TRUE, it overrides `nn_interpol` (which provides the same estimates as buffer but only for locations that are in cells with an NA). The buffer size is determined by the argument directions. `buffer` defaults to FALSE.

- **directions**
character or matrix to indicate the directions in which cells are considered connected when using `nn_interpol` or `buffer`. The following character values are allowed: "rook" or "4" for the horizontal and vertical neighbours; "bishop" to get the diagonal neighbours; "queen" or "8" to get the vertical, horizontal and diagonal neighbours; or "16" for knight and one-cell queen move neighbours. If directions is a matrix it should have odd dimensions and have logical (or 0, 1) values.

Value

a data.frame with the climatic variables of interest
location_slice

Extract local climate for one or more locations for a given time slice.

Description

This function extract local climate for a set of locations at the appropriate times (selecting the closest time slice available for the specific date associated with each location).

Usage

location_slice(
  x,
  time_bp = NULL,
  time_ce = NULL,
  coords = NULL,
  bio_variables,
  dataset,
  path_to_nc = NULL,
  nn_interpol = TRUE,
  buffer = FALSE,
  directions = 8
)

Arguments

x  a data.frame with columns x and y coordinates (see coords for standard coordinate names, or how to use custom ones), plus optional columns time_bp or time_ce (depending on the units used) and name. Alternatively, a vector of cell numbers.

time_bp used if no time_bp column is present in x: the dates in years before present (negative values represent time before present, i.e. 1950, positive values time in the future) for each location.

time_ce time in years CE as an alternative to time_bp. Only one of time_bp or time_ce should be used.

coords a vector of length two giving the names of the "x" and "y" coordinates, as found in data. If left to NULL, the function will try to guess the columns based on standard names c("x", "y"), c("X", "Y"), c("longitude", "latitude"), or c("lon", "lat")

bio_variables vector of names of variables to be extracted.

dataset string defining the dataset to use. If set to "custom", then a single nc file is used from "path_to_nc"

path_to_nc the path to the custom nc file containing the palaeoclimate reconstructions. All the variables of interest need to be included in this file.
**location_slice_from_region_series**

boolean determining whether nearest neighbour interpolation is used to estimate climate for cells that lack such information (i.e. they are under water or ice). By default, interpolation is only performed from the first ring of nearest neighbours; if climate is not available, NA will be returned for that location. The number of neighbours can be changed with the argument directions. **nn_interpol** defaults to TRUE.

boolean determining whether the variable will be returned as the mean of a buffer around the focal cell. If set to TRUE, it overrides **nn_interpol** (which provides the same estimates as buffer but only for locations that are in cells with an NA). The buffer size is determined by the argument directions. **buffer** defaults to FALSE.

character or matrix to indicate the directions in which cells are considered connected when using **nn_interpol** or **buffer**. The following character values are allowed: "rook" or "4" for the horizontal and vertical neighbours; "bishop" to get the diagonal neighbours; "queen" or "8" to get the vertical, horizontal and diagonal neighbours; or "16" for knight and one-cell queen move neighbours. If directions is a matrix it should have odd dimensions and have logical (or 0, 1) values.

**Value**

a data.frame with the climatic variables of interest.

---

**location_slice_from_region_series**  

*Extract local climate for one or more locations for a given time slice.*

**Description**

This function extract local climate for a set of locations at the appropriate times (selecting the closest time slice available for the specific date associated with each location).

**Usage**

```r
location_slice_from_region_series(
x,  
time_bp = NULL,  
time_ce = NULL,  
coords = NULL,  
region_series,  
nn_interpol = TRUE,  
buffer = FALSE,  
directions = 8)
```
Arguments

x a data.frame with columns x and y coordinates (see coords for standard coordinate names, or how to use custom ones), plus optional columns time_bp or time_ce (depending on the units used) and name. Alternatively, a vector of cell numbers.

time_bp used if no time_bp column is present in x: the dates in years before present (negative values represent time before present, i.e. 1950, positive values time in the future) for each location.

time_ce time in years CE as an alternative to time_bp. Only one of time_bp or time_ce should be used.

coords a vector of length two giving the names of the "x" and "y" coordinates, as found in data. If left to NULL, the function will try to guess the columns based on standard names c("x", "y"), c("X", "Y"), c("longitude", "latitude"), or c("lon", "lat")

region_series a SpatRasterDataset obtained with region_series()

nn_interpol boolean determining whether nearest neighbour interpolation is used to estimate climate for cells that lack such information (i.e. they are under water or ice). By default, interpolation is only performed from the first ring of nearest neighbours; if climate is not available, NA will be returned for that location. The number of neighbours can be changed with the argument directions. nn_interpol defaults to TRUE.

buffer boolean determining whether the variable will be returned as the mean of a buffer around the focal cell. If set to TRUE, it overrides nn_interpol (which provides the same estimates as buffer but only for locations that are in cells with an NA). The buffer size is determined by the argument directions. buffer defaults to FALSE.

directions character or matrix to indicate the directions in which cells are considered connected when using nn_interpol or buffer. The following character values are allowed: "rook" or "4" for the horizontal and vertical neighbours; "bishop" to get the diagonal neighbours; "queen" or "8" to get the vertical, horizontal and diagonal neighbours; or "16" for knight and one-cell queen move neighbours. If directions is a matrix it should have odd dimensions and have logical (or 0, 1) values.

Value

a data.frame with the climatic variables of interest.

description

mis_boundaries Time boundaries of marine isotope stages (MIS).

Description

A dataset containing the beginning and end of MIS.
**pastclim**

**Usage**

mis_boundaries

**Format**

A data frame with 24 rows and 2 variables:

- **mis** the stage, a string
- **start** the start of a given MIS, in kya
- **end** the start of a given MIS, in kya

**Description**

This R library is designed to provide an easy way to extract and manipulate palaeoclimate reconstructions for ecological and anthropological analyses.

**Details**

The functionalities of pastclim are described in Leonardi et al. (2023) doi:10.1111/ecog.06481. Please cite it if you use pastclim in your research.

On its dedicated website, you can find Articles giving you a step-by-step overview of the package, and a cheatsheet. There is also a version of the site updated for the dev version (on the top left, the version number is in red, and will be in the format x.x.x.9xxx, indicating it is a development version).

pastclim currently includes data from Beyer et al 2020, a reconstruction of climate based on the HadCM3 model for the last 120k years, and Krapp et al 2021, which covers the last 800k years. The reconstructions are bias-corrected and downscaled to 0.5 degree. More details on these datasets can be found here. There are also instructions on how to build and use custom datasets.

**region_extent**

**Description**

A list of extents for major regions.

**Usage**

region_extent

**Format**

A list of vectors giving the extents.
region_outline

**Description**

An `sf::sf` object containing outlines for major regions. Outlines that span the antimeridian have been split into multiple polygons.

**Usage**

`region_outline`

**Format**

`sf::sf` of outlines.

- **name** names of regions

region_outline_union

**Description**

An `sf::sf` object containing outlines for major regions. Each outline is represented as a single polygon. If you want multiple polygons, use `region_outline`.

**Usage**

`region_outline_union`

**Format**

`sf::sf` of outlines.

- **name** names of regions
region_series

Extract a time series of climate variables for a region

Description

This function extracts a time series of one or more climate variables for a given dataset covering a region (or the whole world). The function returns a `terra::SpatRasterDataset` object, with each variable as a sub-dataset.

Usage

```r
region_series(
  time_bp = NULL,
  time_ce = NULL,
  bio_variables,
  dataset,
  path_to_nc = NULL,
  ext = NULL,
  crop = NULL
)
```

Arguments

- `time_bp` time slices in years before present (negative values represent time before present, positive values time in the future). This parameter can be a vector of times (the slices need to exist in the dataset), a list with a min and max element setting the range of values, or left to NULL to retrieve all time steps. To check which slices are available, you can use `get_time_bp_steps()`.
- `time_ce` time slices in years CE (see `time_bp` for options). For available time slices in years CE, use `get_time_ce_steps()`. Only one of `time_bp` or `time_ce` should be used.
- `bio_variables` vector of names of variables to be extracted
- `dataset` string defining the dataset to use. If set to "custom", then a single nc file is used from "path_to_nc”
- `path_to_nc` the path to the custom nc file containing the palaeoclimate reconstructions. All the variables of interest need to be included in this file.
- `ext` an extent, coded as numeric vector (length=4; order= xmin, xmax, ymin, ymax) or a `terra::SpatExtent` object. If NULL, the full extent of the reconstruction is given.
- `crop` a polygon used to crop the reconstructions (e.g. the outline of a continental mass). A `sf::sfg` or a `terra::SpatVector` object is used to define the polygon.

Value

A `terra::SpatRasterDataset` object, with each variable as a sub-dataset.
region_slice

**Extract a climate slice for a region**

**Description**

This function extracts a slice of one or more climate variables for a given dataset covering a region (or the whole world). The function returns a SpatRaster terra::SpatRaster object, with each variable as a layer.

**Usage**

```r
region_slice(
  time_bp = NULL,
  time_ce = NULL,
  bio_variables, 
  dataset, 
  path_to_nc = NULL,
  ext = NULL,
  crop = NULL
)
```

**Arguments**

- `time_bp`: the time slice in years before present (negative values represent time before present, positive values time in the future). The slice needs to exist in the dataset. To check which slices are available, you can use `get_time_bp_steps()`.
- `time_ce`: time slice in years CE. For available time slices in years CE, use `get_time_ce_steps()`. Only one of `time_bp` or `time_ce` should be used.
- `bio_variables`: vector of names of variables to be extracted
- `dataset`: string defining the dataset to use. If set to "custom", then a single nc file is used from "path_to_nc"
- `path_to_nc`: the path to the custom nc file containing the palaeoclimate reconstructions. All the variables of interest need to be included in this file.
- `ext`: an extent, coded as numeric vector (length=4; order= xmin, xmax, ymin, ymax) or a terra::SpatExtent object. If NULL, the full extent of the reconstruction is given.
- `crop`: a polygon used to crop the reconstructions (e.g. the outline of a continental mass). A sf::sfg or a terra::SpatVector object is used to define the polygon.

**Value**

a SpatRaster terra::SpatRaster object, with each variable as a layer.
**Sample points from a region time series**

**Description**

This function samples points from a region time series. Sampling can either be performed for the same locations at all time steps (if only one value is given for size), or for different locations for each time step (if size is a vector of length equal to the number of time steps). To sample the same number of points, but different locations, for each time step, provide a vector repeating the same value for each time step.

**Usage**

```r
sample_region_series(x, size, method = "random", replace = FALSE, na.rm = TRUE)
```

**Arguments**

- **x**: a `terra::SpatRasterDataset` returned by `region_series()`.  
- **size**: number of points sampled. A single value is used to sample the same locations across all time steps, a vector of values to sample different locations at each time step.  
- **method**: one of the sampling methods from `terra::spatSample()`; it defaults to "random".  
- **replace**: boolean determining whether we sample with replacement  
- **na.rm**: boolean determining whether NAs are removed

**Details**

This function wraps `terra::spatSample()` to appropriate sample the `terra::SpatRasters` in the `terra::SpatRasterDataset` returned by `region_series()`.

**Value**

A data.frame with the sampled cells and their respective values for the climate variables.

---

**Sample points from a region time slice**

**Description**

This function samples points from a region time slice (i.e. a time point).

**Usage**

```r
sample_region_slice(x, size, method = "random", replace = FALSE, na.rm = TRUE)
```
Arguments

- **x**
  - a `terra::SpatRaster` returned by `region_slice()`
- **size**
  - number of points sampled.
- **method**
  - one of the sampling methods from `terra::spatSample()`. It defaults to "random"
- **replace**
  - boolean determining whether we sample with replacement
- **na.rm**
  - boolean determining whether NAs are removed

Details

This function wraps `terra::spatSample()` to appropriate sample the `terra::SpatRaster` returned by `region_slice()`. You can also use `terra::spatSample()` directly on a slice (which is a standard `terra::SpatRaster`).

Value

- a data.frame with the sampled cells and their respective values for the climate variables.

---

set_data_path  
*Set the data path where climate reconstructions will be stored*

Description

This function sets the path where climate reconstructions will be stored. This information is stored in a file names "pastclim_data.txt", which is found in the directory returned by `tools::R_user_dir("pastclim","config")` (i.e. the default configuration directory for the package as set in R >= 4.0).

Usage

```r
set_data_path(
  path_to_nc = NULL,
  ask = TRUE,
  write_config = TRUE,
  copy_example = TRUE,
  on_CRAN = FALSE
)
```

Arguments

- **path_to_nc**
  - the path to the file that contains the downloaded reconstructions. If left unset, the default location returned by `tools::R_user_dir("pastclim","data")` will be used
- **ask**
  - boolean on whether the user should be asked to confirm their choices
- **write_config**
  - boolean on whether the path should be saved in a config file
- **copy_example**
  - boolean on whether the example dataset should be saved in the data_path
- **on_CRAN**
  - boolean; users should NOT need this parameters. It is used to set up a data path in the temporary directory for examples and tests to run on CRAN.
slice_region_series

Value

TRUE if the path was set correctly

Description

This function extracts a time slice from time series of one or more climate variables for a given dataset covering a region (or the whole world).

Usage

slice_region_series(x, time_bp = NULL, time_ce = NULL)

Arguments

x climate time series generated with region_series()

<table>
<thead>
<tr>
<th>time_bp</th>
<th>time slice in years before present (i.e. 1950, negative integers for values in the past). The slices need to exist in the dataset. To check which slices are available, you can use time_bp(x).</th>
</tr>
</thead>
<tbody>
<tr>
<td>time_ce</td>
<td>time slice in years CE. Only one of time_bp or time_ce should be used.</td>
</tr>
</tbody>
</table>

Value

a SpatRaster of the relevant slice.

time_bp

Extract and set time in years before present for SpatRaster and SpatRasterDataset

Description

This functions extracts and sets time in years BP (i.e. from 1950) for a terra::SpatRaster or a terra::SpatRasterDataset. In a terra::SpatRaster object, time is stored with unit "years", which are years from 0AD. This means that, when a summary of the terra::SpatRaster is inspected, the times will appear as time_bp+1950. The same applies when the function terra::time() is used instead of time_bp().
time_series_for_locations

Usage

time_bp(x)

## S4 method for signature 'SpatRaster'
time_bp(x)

## S4 method for signature 'SpatRasterDataset'
time_bp(x)

time_bp(x) <- value

## S4 replacement method for signature 'SpatRaster'
time_bp(x) <- value

## S4 replacement method for signature 'SpatRasterDataset'
time_bp(x) <- value

Arguments

x a terra::SpatRaster
value a numeric vector of times in years BP

Value

a date in years BP (where negative numbers indicate a date in the past)

description

Deprecated version of location_series()

Usage

time_series_for_locations(...)

Arguments

... arguments to be passed to location_series()

Value

a data.frame with the climatic variables of interest
update_dataset_list  

Update the dataset list

Description

If a newer dataset list (which includes all the information about the files storing the data for pastclim), download it and start using it as `dataset_list_included.csv` in `tools::R_user_dir("pastclim","config")`. If the latter is present, the last column, named 'dataset_list_v', provides the version of this table, and the most advanced table is used.

Usage

```r
update_dataset_list(on_cran = FALSE)
```

Arguments

- **on_cran** boolean to make this function run on ci tests using tempdir

Value

TRUE if the dataset was updated

validate_nc  

Validate an netcdf file for pastclim

Description

This function validates a netcdf file as a potential dataset for pastclim. The key checks are: a) that the dimensions (longitude, latitude and time) have been set correctly. b) that all variables have the appropriate metadata (longname and units)

Usage

```r
validate_nc(path_to_nc)
```

Arguments

- **path_to_nc** path to the nc file of interest

Value

TRUE if the file is valid.
var_labels  

Generate pretty variable labels for plotting

Description

Generate pretty labels (in the form of an expression) that can be used for plotting

Usage

var_labels(x, dataset, with_units = TRUE, abbreviated = FALSE)

Arguments

x  
either a character vector with the names of the variables, or a terra::SpatRaster generated with [region_slice()]
[region_slice(): R:region_slice()]
dataset  
string defining dataset to be downloaded (a list of possible values can be obtained with list_available_datasets()). This function will not work on custom datasets.
with_units  
boolean defining whether the label should include units
abbreviated  
boolean defining whether the label should use abbreviations for the variable

Value

a expression that can be used as a label in plots

Examples

var_labels("bio01", dataset = "Example")

# set the data_path for this example to run on CRAN
# users don’t need to run this line
set_data_path(on_CRAN = TRUE)

# for a SpatRaster
climate_20k <- region_slice(
  time_bp = -20000,
  bio_variables = c("bio01", "bio10", "bio12"),
  dataset = "Example"
)
terra::plot(climate_20k, main = var_labels(climate_20k, dataset = "Example"))
terra::plot(climate_20k, main = var_labels(climate_20k, abbreviated = TRUE))
Description

WorldClim version 2.1 is a database of high spatial resolution global weather and climate data, covering both the present and future projections. If you use this dataset, make sure to cite the original publication:

Details


Present-day reconstructions are based on the mean for the period 1970-2000, and are available at multiple resolutions of 10 arc-minutes, 5 arc-minutes, 2.5 arc-minute and 0.5 arc-minutes. The resolution of interest can be obtained by changing the ending of the dataset name WorldClim_2.1_RESm, e.g. WorldClim_2.1_10m or WorldClim_2.1_5m (currently, only 10m and 5m are currently available in pastclim). In pastclim, the datasets are given a date of 1985 CE (the mid-point of the period of interest), corresponding to a time_bp of 35. There are 19 “bioclimatic” variables, as well as monthly estimates for minimum, mean, and maximum temperature, and precipitation.

Future projections are based on the models in CMIP6, downscaled and de-biased using WorldClim 2.1 for the present as a baseline. Monthly values of minimum temperature, maximum temperature, and precipitation, as well as 19 bioclimatic variables were processed for 23 global climate models (GCMs), and for four Shared Socio-economic Pathways (SSPs): 126, 245, 370 and 585. Model and SSP can be chosen by changing the ending of the dataset name WorldClim_2.1_GCM_SSP_RESm.

Available values for GCM are: "ACCESS-CM2", "BCC-CSM2-MR", "CMCC-ESM2", "EC-Earth3-Veg", "FIO-ESM-2-0", "GFDL-ESM4", "GISS-E2-1-G", "HadGEM3-GC31-LL", "INM-CM5-0", "IPSL-CM6A-LR", "MIROC6", "MPI-ESM1-2-HR", "MRI-ESM2-0", and "UKESM1-0-LL". For SSP use: "ssp126", "ssp245", "ssp370", and "ssp585". RES takes the same values as for present reconstructions (i.e. "10m", "5m", "2.5m", and "0.5m"). Example dataset names are WorldClim_2.1_ACCESS-CM2_ssp245_10m and WorldClim_2.1_MRI-ESM2-0_ssp370_5m

The dataset are averages over 20 year periods (2021-2040, 2041-2060, 2061-2080, 2081-2100). In pastclim, the midpoints of the periods (2030, 2050, 2070, 2090) are used as the time stamps. All 4 periods are automatically downloaded for each combination of GCM model and SSP, and are selected as usual by defining the time in functions such as region_slice().

ybp2date

Convert years BP from pastclim to lubridate date, or vice versa

Description

These functions convert between years BP as used by pastclim (negative numbers going into the past, positive into the future) and standard POSIXct date objects.
Usage

\texttt{ybp2date(x)}

\texttt{date2ybp(x)}

Arguments

\texttt{x} \hspace{1cm} \text{a time in years BP using the pastclim convention of negative numbers indicating years into the past, or a \texttt{POSIXct} date object}

Value

a \texttt{POSIXct} date object, or a vector

Examples

\texttt{ybp2date(-10000)}
\texttt{ybp2date(0)}
\texttt{# back and forth}
\texttt{date2ybp(ybp2date(-10000))}
Index

* datasets
  mis_boundaries, 22
  region_extent, 23
  region_outline, 24
  region_outline_union, 24

Barreto2023, 3
Beyer2020, 3
bioclim_vars, 4
bioclim_vars, matrix, matrix-method
(bioclim_vars), 4
bioclim_vars, numeric, numeric-method
(bioclim_vars), 4
bioclim_vars, SpatRaster, SpatRaster-method
(bioclim_vars), 4
bioclim_vars, SpatRasterDataset, SpatRasterDataset-method
(bioclim_vars), 4

clean_data_path, 6
clima_for_locations, 6
clima_for_time_slice, 7

date2ybp (ybp2date), 33
df_from_region_series, 7
df_from_region_series(), 8
df_from_region_slice, 7
df_from_region_slice(), 7
distance_from_sea, 8
download_dataset, 9

Example, 9
expression, 32

get_available_datasets, 10
get_available_datasets(), 18
get_biome_classes, 10
gedata_path, 11
gedata_path(), 9
gdownloaded_datasets, 11
gfile_for_dataset, 12
gice_mask, 12

g_land_mask, 13
g_mis_time_steps, 13
g_time_bp_steps, 14
g_time_bp_steps(), 12, 13, 19, 25, 26
g_time_ce_steps (get_time_bp_steps), 14

g_time_ce_steps(), 13, 19, 25, 26
g_time_steps (get_time_bp_steps), 14
g_varname, 14
g_vars_for_dataset, 15

HYDE_3.3_baseline, 16
is_region_series, 16
Kraepel2021, 17

list_available_datasets, 10, 18
list_available_datasets(), 8-10, 12-15, 32
location_series, 18
location_series(), 30
location_slice, 20
location_slice(), 6, 18, 19
location_slice_from_region_series, 21

mis_boundaries, 13, 22

pastclim, 23
pastclim-package (pastclim), 23

region_extent, 23
region_outline, 24, 24
region_outline_union, 24
region_series, 25
region_series(), 7, 16, 22, 27, 29
region_slice, 26
region_slice(), 7, 8, 28, 33

sample_region_series, 27
sample_region_slice, 27

35
set_data_path, 28
set_data_path(), 9, 11
sf::sf, 24
sf::sfg, 25, 26
slice_region_series, 29
SpatRaster, 29
SpatRasterDataset, 22
terra::as.data.frame(), 8
terra::SpatExtent, 25, 26
terra::SpatRaster, 7–9, 12, 13, 17, 26–30, 32
terra::SpatRasterDataset, 16, 25, 27, 29
terra::spatSample(), 27, 28
terra::SpatVector, 25, 26
terra::time(), 17, 29
time_bp, 29
time_bp(), 29
time_bp, SpatRaster-method(time_bp), 29
time_bp, SpatRasterDataset-method (time_bp), 29
time_bp<-(time_bp), 29
time_bp<-, SpatRaster-method(time_bp), 29
time_bp<-, SpatRasterDataset-method (time_bp), 29
time_series_for_locations, 30
update_dataset_list, 31
validate_nc, 31
var_labels, 32
WorldClim_2.1, 33
WorldClim_2.1_10m(WorldClim_2.1), 33
WorldClim_2.1_5m(WorldClim_2.1), 33
ybp2date, 33