Package ‘itsdm’

June 20, 2022

Title Isolation Forest-Based Presence-Only Species Distribution Modeling

Version 0.1.2

Description Collection of R functions to do purely presence-only species distribution modeling with isolation forest (iForest) and its variations such as Extended isolation forest and SCiForest. See the details of these methods in references: Liu, F.T., Ting, K.M. and Zhou, Z.H. (2008) <doi:10.1109/ICDM.2008.17>, Hariri, S., Kind, M.C. and Brunner, R.J. (2019) <doi:10.1109/TKDE.2019.2947676>, Liu, F.T., Ting, K.M. and Zhou, Z.H. (2010) <doi:10.1007/978-3-642-15883-4_18>, Guha, S., Mishra, N., Roy, G. and Schrijvers, O. (2016) <https://proceedings.mlr.press/v48/guha16.html>, Cortes, D. (2021) <arXiv:2110.13402>. Additionally, Shapley values are used to explain model inputs and outputs. See details in references: Shapley, L.S. (1953) <doi:10.1515/9781400881970-018>, Lundberg, S.M. and Lee, S.I. (2017) <https://dl.acm.org/doi/abs/10.5555/3295222.3295230>, Molnar, C. (2020) <ISBN:978-0-244-76852-2>, Štrumbelj, E. and Kononenko, I. (2014) <doi:10.1007/s10115-013-0679-x>. itsdm also provides functions to diagnose variable response, analyze variable importance, draw spatial dependence of variables and examine variable contribution. As utilities, the package includes a few functions to download bioclimatic variables including 'WorldClim' version 2.0 (see Fick, S.E. and Hijmans, R.J. (2017) <doi:10.1002/joc.5086>) and 'CMCC-BioClimInd' (see Noce, S., Caporaso, L. and Santini, M. (2020) <doi:10.1038/s41597-020-00726-5>.

License MIT + file LICENSE

URL https://github.com/LLeiSong/itsdm,
https://lleisong.github.io/itsdm/

BugReports https://github.com/LLeiSong/itsdm/issues

Depends R (>= 3.5.0)

Imports checkmate, dplyr, fastshap, ggplot2, isotree, methods, ncdf4, outliertree, patchwork, raster, rlang, ROCit, sf, stars, stats, stringr, tidyselect, utils

Suggests knitr, rmarkdown

VignetteBuilder knitr

Encoding UTF-8

Language en-US

LazyData true

RoxygenNote 7.2.0

NeedsCompilation no

Author Lei Song [aut, cre] (<https://orcid.org/0000-0002-4371-1473>),
Lyndon Estes [ths]

Maintainer Lei Song <lsong@clarku.edu>

Repository CRAN

Date/Publication 2022-06-20 18:20:07 UTC

R topics documented:

itsdm-package *Isolation forest-based presence-only species distribution modeling*

Description

This package is a wrapper for a few packages including `isotree`, `outliertree`, `fastshap`, etc. It does purely presence-only species distribution modeling with isolation forest and variations such as SCiForest and EIF. It also provides functions to make response curves, analyze variable importance, analyze variable dependence and analyze variable contribution. As utilities, the package includes a few functions to download bioclimatic variables including worldclim version 2.0 and CMCC-BioClimInd. There are also functions to detect outliers in the occurrence dataset to do data cleaning.

Details

This package provides multiple features.

1. Download bioclimatic variables and reduce their dimensions. This includes historic and future climatic indicators from two sources:
   - worldclim
   - CMCC-BioClimInd
2. Detect suspicous environmental outliers.
3. Fit a isolation forest-based SDM.
4. Make presence-only evaluation.
5. Generate response curves of environmental variables including marginal and independent responses and analyze interactions between environmental variables.
6. Analyze variable importance using Shapley values.
7. Convert predicted environmental suitability to presence-absence map.
8. Analyze variable contributions to any specific observations.

Author(s)

Lei Song <`lsong@clarku.edu`>

Maintainer: Lei Song <`lsong@clarku.edu`>

References

Please check references in R documentation of each specific function.

cmcc_bioclim *Download historic Bioclimatic indicators (BIOs) named CMCC-BioClimInd.*

Description

Parse historic CMCC-BioClimInd bioclimatic indicators optionally with a setting of boundary and a few other options.

Usage

```
cmcc_bioclim(bry = NULL, path = NULL, nm_mark = "clip", return_stack = TRUE)
```

Arguments

`bry`	(sf or sp) The boundary to mask the downloaded original data. If `NULL`, it would get global map. If not `NULL`, it can take sf, sfc, `SpatialPolygonsDataFrame`, `SpatialPolygons`, etc. The default is `NULL`.
`path`	(`character`) The path to save the downloaded imagery. If `NULL`, it would use the current working directory. The default is `NULL`.
`nm_mark`	(`character`) the name mark of clipped images. The default is "clip". It would be ignored if `bry` is `NULL`.
`return_stack`	(`logical`) if `TRUE`, stack the imagery together and return. If the area is large and resolution is high, it is better not to stack them. The default is `TRUE`.

Details

Web page page for this dataset

Value

if `return_stack` is `TRUE`, the images would be returned as a `stars`. Otherwise, nothing to return, but the user would receive a message of where the images are.

Note

The function is experimental at the moment, because the download server of this dataset is not as stable as Worldclim yet. If it fails due to slow internet, try to set a larger timeout option, e.g., using `options(timeout = 1e3)`.

References

Noce, Sergio, Luca Caporaso, and Monia Santini."A new global dataset of bioclimatic indicators. "*Scientific data* 7.1 (2020): 1-12. doi: 10.1038/s41597020007265

Examples

```
## Not run:
library(dplyr)
library(sf)
library(itsdm)
bry <- st_polygon(
  list(rbind(c(29.34, -11.72), c(29.34, -0.95),
             c(40.31, -0.95), c(40.31, -11.72),
             c(29.34, -11.72)))) %>%
  st_sfc(crs = 4326)

cmcc_bios <- cmcc_bioclim(bry = bry,
  nm_mark = 'tza', path = tempdir())

## End(Not run)
```

`convert_to_pa` *Convert predicted suitability to presence-absence map.*

Description

Use threshold-based, logistic or linear conversion method to convert predicted suitability map to presence-absence map.

Usage

```
convert_to_pa(
  suitability,
  method = "logistic",
  beta = 0.5,
  alpha = -0.05,
  a = 1,
  b = 0,
  species_prevalence = NA,
  threshold = 0.5,
  seed = 10L,
  visualize = TRUE
)
```

Arguments

`suitability` (`stars` or `RasterLayer`) The suitability raster.

`method` (`character`) The conversion method, must be one of 'threshold', 'logistic', and 'linear'. The default is 'logistic'.

`beta` (`numeric`) Works for 'threshold' or 'logistic' method. If `method` is threshold, then `beta` is the threshold value to cutoff. If `method` is logistic, it is the sigmoid midpoint. The default is `0.5`.

alpha (numeric) Works for logistic method. It is the logistic growth rate or steepness of the curve. The default is -.05.

a (numeric) Works for linear method. It is the slope of the line. The default is 1.

b (numeric) Works for linear method. It is the intercept of the line. The default is 0.

species_prevalence
(numeric or NA) Works for all three methods. It is the species prevalence to classify suitability map. It could be NA, when the will be calculated automatically based on other arguments. The default is NA.

threshold (numeric) The threshold used to convert probability of occurrence to presence-absence map. It ranges in [0, 1]. The default is 0.5.

seed (integer) The seed for random progress. The default is 10L

visualize (logical) If TRUE, plot map of suitability, probability of occurrence, and presence-absence together. The default is TRUE.

Details

Multiple methods and arguments could be used as a combination to do the conversion.

Value

(PAConversion) A list of

- suitability (stars) The input suitability map
- probability_of_occurrence (stars) The map of occurrence probability
- pa_conversion (list) A list of conversion arguments
- pa_map (stars) The presence-absence map

References

c onvertToPA in package virtualspecies

See Also

plot.PAConversion

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())
```

```
set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 3L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

# Threshold conversion
pa_thred <- convert_to_pa(mod$prediction,
  method = 'threshold', beta = 0.5)
pa_thred
plot(pa_thred)

# Logistic conversion
pa_log <- convert_to_pa(mod$prediction, method = 'logistic',
  beta = 0.5, alpha = -.05)

# Linear conversion
pa_lin <- convert_to_pa(mod$prediction, method = 'linear',
  a = 1, b = 0)
```

dim_reduce *Reduce dimensions using auto-correlation between numeric raster layers.*

Description

Reduce dimensions of raster layers based on Pearson correlation and a user-defined threshold. NOTE that it only works on numeric variables, does not work on categorical variables.

Usage

```
dim_reduce(
  img_stack = NULL,
  threshold = 0.5,
  preferred_vars = NULL,
  samples = NULL
)
```

Arguments

img_stack (stars or RasterStack) The image stack to work on.

threshold (numeric) The threshold number of Pearson correlation that indicates two variables are strongly correlated. The default is 0.5.

preferred_vars (vector of character) The preferred variables **in order** in dimension reduction. The preferred variables will move to the beginning before the reduction. So make sure they are placed in order. Furthermore, setting preferred_vars does not guarantee they can survive. For example, one preferred variable that is placed later has strong correlation with former preferred variable.

samples (sf or sp) The samples to reduce dimension. If not NULL, it can take sf, sfc, SpatialPointsDataFrame, SpatialPoints, etc. If NULL, the whole raster stack would be used. The default is NULL.

Value

(ReducedImageStack) A list of

- threshold (numeric) The threshold set in function inputs
- img_reduced (stars) The image stack after dimension reduction
- cors_original (data.frame) A table of Pearson correlations between all variables.
- cors_reduced (data.frame) A table of Pearson correlations between variables after dimension reduction.

Examples

```
library(sf)
library(itsdm)
library(stars)
library(dplyr)
env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars()
img_reduced <- dim_reduce(env_vars, threshold = 0.7,
  preferred_vars = c('bio1', 'bio12'))
```

evaluate_po *Evaluate the model based on presence-only data.*

Description

This function will calculate two major types of evaluation metrics in terms of presence-only data. The first type is presence-only customized metrics, such as Contrast Validation Index (CVI), continuous Boyce index (CBI), and ROC_ratio. The second type is presence-background evaluation metrics by extracting background points as pseudo absence observations.

Usage

```
evaluate_po(
  model,
  occ_pred,
  bg_pred = NULL,
  var_pred,
  threshold = NULL,
  visualize = FALSE
)
```

Arguments

`model` (`isolation_forest`) The extended isolation forest SDM. It could be the item `model` of `POIsotree` made by function `isotree_po`.

`occ_pred` (`vector of numeric`) A `vector` contains predicted values at occurrence locations.

`bg_pred` (`vector` of `numeric`) the vector contains predicted values with same number of background points.

`var_pred` (`vector` of `numeric`) the vector contains predicted values of the whole area.

`threshold` (`numeric` or `NULL`) The threshold to calculate threshold-based evaluation metrics. If `NULL`, a recommended threshold will be calculated based on optimal TSS value. The default is `NULL`.

`visualize` (`logical`) If `TRUE`, plot the evaluation figures. The default is `FALSE`.

Details

- **CVI** is the proportion of presence points falling in cells having a threshold (`0.5` for example) habitat suitability index minus the proportion of cells within this range of threshold of the model. Here we used varied thresholds: `0.25`, `0.5`, and `0.75`.
- **continuous Boyce index (CBI)** is made with a 100 resolution of moving windows and Kendall method.
- **ROC_ratio** curve plots the proportion of presences falling above a range of thresholds against the proportion of cells falling above the range of thresholds. The area under the modified ROC curve was then called **AUC_ratio**.
- **Sensitivity (TPR)** = TP/(TP + FN)
- **Specificity (TNR)** = TN/(TN + FP)
- **True skill statistic (TSS)** = Sensitivity + specificity - 1
- **Jaccard's similarity index** = TP/(FN + TP + FP)
- **Sørensen's similarity index (F-measure)** = 2TP/(FN + 2TP + FP)
- **Overprediction rate** = FP/(TP + FP)
- **Underprediction rate** = FN/(TP + FN)

Value

(POEvaluation) A list of

- **po_evaluation** is presence-only evaluation metrics. It is a list of
  - cvi (list) A list of CVI with 0.25, 0.5, and 0.75 as threshold
  - boyce (list) A list of items related to continuous Boyce index (CBI)
  - roc_ratio (list) A list of ROC ratio and AUC ratio
- **pb_evaluation** is presence-background evaluation metrics. It is a list of
  - confusion matrix (table) A table of confusion matrix. The columns are true values, and the rows are predicted values.
  - sensitivity (numeric) The sensitivity or TPR
  - specificity (numeric) The specificity or TNR
  - TSS (list) A list of info related to true skill statistic (TSS)
    - cutoff (vector of numeric) A vector of cutoff threshold values
    - tss (vector of numeric) A vector of TSS for each cutoff threshold
    - Recommended threshold (numeric) A recommended threshold according to TSS
    - Optimal TSS (numeric) The best TSS value
  - roc (list) A list of ROC values and AUC value
  - Jaccard's similarity index (numeric) The Jaccard's similarity index
  - Sørensen's similarity index (numeric) The Sørensen's similarity index or F-measure
  - Overprediction rate (numeric) The Overprediction rate
  - Underprediction rate (numeric) The Underprediction rate

References

- Peterson, A. Townsend, Monica Papeş, and Jorge Soberón. "Rethinking receiver operating characteristic analysis applications in ecological niche modeling." *Ecological modelling* 213.1 (2008): 63-72. doi: 10.1016/j.ecolmodel.2007.11.008
- Hirzel, Alexandre H., et al. "Evaluating the ability of habitat suitability models to predict species presences." *Ecological modelling* 199.2 (2006): 142-152. doi: 10.1016/j.ecolmodel.2006.05.017
- Hirzel, Alexandre H., and Raphaël Arlettaz. "Modeling habitat suitability for complex species distributions by environmental-distance geometric mean." *Environmental management* 32.5 (2003): 614-623. doi: 10.1007/s0026700300403
- Leroy, Boris, et al. "Without quality presence-absence data, discrimination metrics such as TSS can be misleading measures of model performance." *Journal of Biogeography* 45.9 (2018): 1994-2002. doi: 10.1111/jbi.13402

See Also

print.POEvaluation, plot.POEvaluation

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data(”occ_virtual_species”)
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 2L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

eval_train <- evaluate_po(mod$model,
  occ_pred = mod$pred_train$prediction,
  var_pred = na.omit(as.vector(mod$prediction[[1]])))
print(eval_train)
plot(eval_train)
```

`future_cmcc_bioclim` *Download future Bioclimatic indicators (BIOs) named CMCC-BioClimInd.*

Description

Parse future CMCC-BioClimInd bioclimatic indicators obtained by different Earth System Models (ESMs) optionally with a setting of boundary and a few other options.

Usage

```
future_cmcc_bioclim(
  bry = NULL,
  path = NULL,
  esm = "CMCC-CESM",
  rcp = 85,
  interval = "2040-2079",
  nm_mark = "clip",
  return_stack = TRUE
)
```

Arguments

`bry` (`sf` or `sp`) The boundary to mask the downloaded original data. If `NULL`, it would get global map. If not `NULL`, it can take `sf`, `sfc`, `SpatialPolygonsDataFrame`, `SpatialPolygons`, etc. The default is `NULL`.

`path` (`character`) The path to save the downloaded imagery. If `NULL`, it would use the current working directory. The default is `NULL`.

`esm` (`character`) The option for Earth System Models (ESMs). Should be one of "CMCC-CESM", 'GFDL-ESM2M', 'HadGEM2-ES', 'IPSL-CM5A-LR', 'MIROC-ESM-CHEM', 'NorESM1-M'. The default is CMCC-CESM.

`rcp` (`numeric`) The option of Representative Concentration Pathways (RCPs). Should be 45 or 85. Only 85 is available for CMCC-CESM. The default is 85.

`interval` (`character`) The option for time interval. Should be one of "2040-2079", "2060-2099". The default is "2040-2079".

`nm_mark` (`character`) the name mark of clipped images. The default is "clip". It would be ignored if `bry` is `NULL`.

`return_stack` (`logical`) if `TRUE`, stack the imagery together and return. If the area is large and resolution is high, it is better not to stack them. The default is `TRUE`.

Details

https://doi.pangaea.de/10.1594/PANGAEA.904278?format=html#download

Value

if `return_stack` is `TRUE`, the images would be returned as a `stars`. Otherwise, nothing to return, but the user would receive a message of where the images are.

Note

The function is experimental at the moment, because the download server of this dataset is not as stable as Worldclim yet. If it fails due to slow internet, try to set a larger timeout option, e.g., using `options(timeout = 1e3)`.

References

Noce, Sergio, Luca Caporaso, and Monia Santini."A new global dataset of bioclimatic indicators. "*Scientific data* 7.1 (2020): 1-12.doi: 10.1038/s41597020007265

Examples

```
## Not run:
library(itsdm)
future_cmcc_bioclim(path = tempdir(),
  esm = 'GFDL-ESM2M', rcp = 45,
  interval = "2040-2079", return_stack = FALSE)

## End(Not run)
```

---

future_worldclim2 *A function to parse the future climate from worldclim version 2.1.*

Description

This function allows you to parse worldclim version 2.1 future climatic files with a setting of boundary and a few other options.

Usage

```
future_worldclim2(
  var = "tmin",
  res = 10,
  gcm = "BCC-CSM2-MR",
  ssp = "ssp585",
  interval = "2021-2040",
  bry = NULL,
  path = NULL,
  nm_mark = "clip",
  return_stack = TRUE
)
```

Arguments

`var` (character) The option for the variable to download. Should be one of tmin, tmax, prec, bioc. The default is tmin.

`res` (numeric) The option for the resolution of image to download. Should be one of 2.5, 5, 10. The default is 10.

`gcm` (character) The option for global climate models. Should be one of "BCC-CSM2-MR", "CNRM-CM6-1","CNRM-ESM2-1", "CanESM5", "GFDL-ESM4", "IPSL-CM6A-LR","MIROC-ES2L", "MIROC6", "MRI-ESM2-0". The default is ’BCC-CSM2-MR’.

`ssp` (character) The option for Shared Socio-economic Pathways. Should be one of "ssp126", "ssp245", "ssp370", "ssp585". The default is "ssp585".

`interval` (character) The option for time interval. Should be one of "2021-2040", "2041-2060", "2061-2080", "2081-2100". The default is "2021-2040".

`bry` (sf or sp) The boundary to mask the downloaded original data. If NULL, it would get global map. If not NULL, it can take sf, sfc, SpatialPolygonsDataFrame, SpatialPolygons, etc. The default is NULL.

`path` (character) The path to save the downloaded imagery. If NULL, it would use the current working directory. The default is NULL.

`nm_mark` (character) the name mark of clipped images. The default is "clip". It would be ignored if bry is NULL.

`return_stack` (logical) if TRUE, stack the imagery together and return. If the area is large and resolution is high, it is better not to stack them. The default is TRUE.

Details

Web page page for this dataset

Value

if `return_stack` is `TRUE`, the images would be returned as a `stars`. Otherwise, nothing to return, but the user would receive a message of where the images are.

Note

If it fails due to slow internet, try to set a larger timeout option, e.g., using `options(timeout = 1e3)`.

References

Fick, Stephen E., and Robert J. Hijmans. "WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas." *International journal of climatology* 37.12 (2017): 4302-4315.doi: 10.1002/joc.5086

Examples

```
## Not run:
future_worldclim2("tmin", 10, "BCC-CSM2-MR",
  "ssp585", "2021-2040",
  path = tempdir(), return_stack = FALSE)

## End(Not run)
```

independent_response *Calculate independent responses of each variables.*

Description

Calculate the independent responses of each variables within the model.

Usage

```
independent_response(model, var_occ, variables, si = 1000, visualize = FALSE)
```

Arguments

model (Any predictive model). It is isolation_forest here. It could be the item model of POIsotree made by function isotree_po.

var_occ (data.frame, tibble) The data.frame style table that include values of environmental variables at occurrence locations.

variables (stars) The stars of environmental variables. It should have multiple attributes instead of dims. If you have raster object instead, you could use st_as_stars to convert it to stars or use read_stars directly read source data as a stars. You also could use item variables of POIsotree made by function isotree_po.

si (integer) The number of samples to generate response curves. If it is too small, the response curves might be biased. The default value is 1000.

visualize (logical) if TRUE, plot the response curves. The default is FALSE.

Details

The values show how each environmental variable independently affects the modeling prediction. They show how the predicted result only using this variable changes as it is varied.

Value

(IndependentResponse) A list of

- responses_cont (list) A list of response values of continuous variables
- responses_cat (list) A list of response values of categorical variables

References

- Elith, Jane, et al. "The evaluation strip: a new and robust method for plotting predicted responses from species distribution models." *Ecological modelling* 186.3 (2005): 280-289.doi: 10.1016/j.ecolmodel.2004.12.007

See Also

plot.IndependentResponse

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 2L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

independent_responses <- independent_response(
  model = mod$model,
  var_occ = mod$var_train %>% st_drop_geometry(),
  variables = mod$variables)
plot(independent_responses)
```

---

isotree_po *Function to run extended isolation forest as SDM.*

Description

Call isolation forest and its variations to do species distribution modeling and optionally do model explanation.

Usage

```
isotree_po(
    occ,
    occ_test = NULL,
    occ_crs = 4326,
    variables,
    categ_vars = NULL,
    ntrees = 100L,
    sample_size = 1,
    ndim = 1L,
    seed = 10L,
    ...,
    offset = 0,
    response = TRUE,
    spatial_response = TRUE,
    check_variable = TRUE,
    visualize = FALSE
  )
```

Arguments

`occ` (`data.frame`, `sf`, `SpatialPointsDataFrame`) The occurrence dataset for training. There must be column `x` and `y` for coordinates if it is a regular `data.frame`.

`occ_test` (`data.frame`, `sf`, `SpatialPointsDataFrame`, or `NULL`) The occurrence dataset for independent test. The same structure as `occ`. If not `NULL`, there must be column `x` and `y` for coordinates when it is a regular `data.frame`. If `NULL`, no independent test will be used. The default is `NULL`.

`occ_crs` (`numeric` or `crs`) The EPSG number or `crs` object of occurrence CRS. The default value is `4326`, which is the geographic coordinate system.

`variables` (`RasterStack` or `stars`) The stack of environmental variables.

`categ_vars` (`vector` of `character` or `NULL`) The names of categorical variables. Must be the same as the names in `variables`.

`ntrees` (`integer`) The number of trees for the isolation forest. It must be integer, which you could use function `as.integer` to convert to. The default is `100L`.

`sample_size` (`numeric`) It should be a rate for sampling size in `[0, 1]`. The default is `1.0`.

`ndim` (`integer`) ExtensionLevel for isolation forest. It must be integer, which you could use function `as.integer` to convert to. Also, it must be no smaller than the dimension of environmental variables. When it is 1, the model is a traditional isolation forest, otherwise the model is an extended isolation forest. The default is 1.

`seed` (`integer`) The random seed used in the modeling. It should be an integer. The default is `10L`.

`...` Other arguments that `isolation.forest` needs.

`offset` (`numeric`) The offset to adjust fitted suitability. The default is zero. Highly recommend to leave it as default.

`response` (`logical`) If `TRUE`, generate response curves. The default is `TRUE`.

`spatial_response`
: (`logical`) If `TRUE`, generate spatial response maps. The default is `TRUE` because it might be slow. NOTE that here SHAP-based map is not generated because it is slow. If you want it be mapped, you could call function `spatial_response` to make it.

`check_variable`
: (`logical`) If `TRUE`, check the variable importance. The default is `TRUE`.

`visualize`
: (`logical`) If `TRUE`, generate the essential figures related to the model. The default is `FALSE`.

Details

Please read details of algorithm `isolation.forest` on `https://github.com/david-cortes/isotree`, and the R documentation of function `isolation.forest`.

Value

(`POIsotree`) A list of

- model (`isolation.forest`) The threshold set in function inputs
- variables (`stars`) The formatted image stack of environmental variables
- pts_occ (`sf`) A `sf` of training occurrence dataset
- pts_bg_occ (`sf`) A `sf` of background points for training dataset evaluation or SHAP dependence plot
- pts_occ_test (`sf` or `NULL`) A `sf` of test occurrence dataset
- pts_bg_occ_test (`sf` or `NULL`) A `sf` of background points for test dataset evaluation or SHAP dependence plot
- var_train (`sf`) A `sf` with values of each environmental variables for training occurrence
- pred_train (`sf`) A `sf` with values of prediction for training occurrence
- eval_train (`POEvaluation`) A list of presence-only evaluation metrics based on training dataset. See details of `POEvaluation` in `evaluate_po`
- var_test (`sf` or `NULL`) A `sf` with values of each environmental variables for test occurrence
- pred_test (`sf` or `NULL`) A `sf` with values of prediction for test occurrence
- eval_test (`POEvaluation` or `NULL`) A list of presence-only evaluation metrics based on test dataset. See details of `POEvaluation` in `evaluate_po`
- prediction (`stars`) The predicted environmental suitability
- marginal_responses (`MarginalResponse` or `NULL`) A list of marginal response values of each environmental variables. See details in `marginal_response`
- offset (`numeric`) The offset value set as inputs.
- independent_responses (`IndependentResponse` or `NULL`) A list of independent response values of each environmental variables. See details in `independent_response`
- shap_dependences (`ShapDependence` or `NULL`) A list of variable dependence values of each environmental variables. See details in `shap_dependence`
- spatial_responses (`SpatialResponse` or `NULL`) A list of spatial variable dependence values of each environmental variables. See details in `shap_dependence`
- variable_analysis (`VariableAnalysis` or `NULL`) A list of variable importance analysis based on multiple metrics. See details in `variable_analysis`

References

- Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest." *2008 eighth ieee international conference on data mining*.IEEE, 2008. doi: 10.1109/ICDM.2008.17
- Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation-based anomaly detection." *ACM Transactions on Knowledge Discovery from Data (TKDD)* 6.1 (2012): 1-39. doi: 10.1145/2133360.2133363
- Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "On detecting clustered anomalies using SCiForest." *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, Berlin, Heidelberg, 2010. doi: 10.1007/9783642158834_18
- Ha riri, Sahand, Matias Carrasco Kind, and Robert J. Brunner. "Extended isolation forest." *IEEE Transactions on Knowledge and Data Engineering (2019)*. doi: 10.1109/TKDE.2019.2947676
- `https://github.com/david-cortes/isotree`
- References of related feature such as response curves and variable importance will be listed under their own functions

See Also

`evaluate_po`, `marginal_response`, `independent_response`, `shap_dependence`, `spatial_response`, `variable_analysis`, `isolation.forest`

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

# Prepare data
data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12))

# Modeling
mod_virtual_species <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 10,
  sample_size = 0.6, ndim = 1L,
  seed = 123L)

# Check results
## Evaluation based on training dataset
print(mod_virtual_species$eval_train)
plot(mod_virtual_species$eval_train)

## Response curves
plot(mod_virtual_species$marginal_responses)
plot(mod_virtual_species$independent_responses,
  target_var = c('bio1', 'bio5'))
plot(mod_virtual_species$shap_dependence)

## Relationships between target var and related var
plot(mod_virtual_species$shap_dependence,
  target_var = c('bio1', 'bio5'),
  related_var = 'bio12', smooth_span = 0)

# Variable importance
mod_virtual_species$variable_analysis
plot(mod_virtual_species$variable_analysis)
```

`mainland_africa` *Boundary of mainland Africa*

Description

The overall continental boundary of mainland Africa queried from `rnaturalearth` and get processed.

Usage

```
mainland_africa
```

Format

A `sf` with one rows and 2 fields

name (`character`) The name of the polygon: Africa

area (`units`) The united number of the overall area in km2. This is not a consensus area, but just a calculated area under this resolution.

geometry (`sfc`) The simple polygon feature of the boundary

Source

`rnaturalearth`

marginal_response *Calculate marginal responses of each variables.*

Description

Calculate the marginal responses of each variables within the model.

Usage

```
marginal_response(model, var_occ, variables, si = 1000, visualize = FALSE)
```

Arguments

`model`	(Any predictive model). In this package, it is `isolation_forest`. It could be the item `model` of `POIsotree` made by function `isotree_po`.
`var_occ`	(`data.frame`, `tibble`) The `data.frame` style table that include values of environmental variables at occurrence locations.
`variables`	(`stars`) The `stars` of environmental variables. It should have multiple `attributes` instead of `dims`. If you have `raster` object instead, you could use `st_as_stars` to convert it to `stars` or use `read_stars` directly read source data as a `stars`. You also could use item `variables` of `POIsotree` made by function `isotree_po`.
`si`	(`integer`) The number of samples to generate response curves. If it is too small, the response curves might be biased. The default value is `1000`.
`visualize`	(`logical`) if `TRUE`, plot the response curves. The default is `FALSE`.

Details

The values show how each environmental variable affects the modeling prediction. They show how the predicted result changes as each environmental variable is varied while keeping all other environmental variables at average sample value. They might be hard to interpret if there are strongly correlated variables. The users could use `dim_reduce` function to remove the strong correlation from original environmental variable stack.

Value

(`MarginalResponse`) A nested list of

- responses_cont (`list`) A list of response values of continuous variables
- responses_cat (`list`) A list of response values of categorical variables

References

- Elith, Jane, et al. "The evaluation strip: a new and robust method for plotting predicted responses from species distribution models." *Ecological modelling* 186.3 (2005): 280-289.doi: 10.1016/j.ecolmodel.2004.12.007

See Also

plot.MarginalResponse

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 2L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

marginal_responses <- marginal_response(
  model = mod$model,
  var_occ = mod$var_train %>% st_drop_geometry(),
  variables = mod$variables)
plot(marginal_responses)
```

occ_virtual_species *Occurrence dataset of a virtual species*

Description

A pseudo presence-only occurrence dataset of a virtual species made by package virtualspecies.

Usage

```
occ_virtual_species
```

Format

A `data.frame` with 300 rows and 2 fields

x (`numeric`) The x coordinates of the records in geographic coordinate system

y (`numeric`) The y coordinates of the records

Details

The environmental niche of the virtual species is made by defining its response functions to annual temperature and annual precipitation in mainland Africa. The response function of annual temperature is normal distribution with mean = 22 and standard deviation = 5. The response function of annual precipitation is normal distribution with mean = 1000 and standard deviation = 200. Then the suitability is convert to presence-absence map by logistic conversion with beta = 0.7, alpha = -0.05, and species prevalence = 0.27. Finally 300 presence-only points are sampled across the whole region.

Source

`virtualspecies`

---

plot.EnvironmentalOutlier

Function to plot suspicious outliers in an observation dataset.

Description

Display observations and outliers in a dataset.

Usage

```
## S3 method for class 'EnvironmentalOutlier'
plot(x, overlay_raster = NULL, pts_alpha = 0.5, ...)
```

Arguments

`x`	(`EnvironmentalOutlier`) The PAConversion object to plot. It could be the return of function `suspicious_env_outliers`.
`overlay_raster`	(`RasterLayer` or `stars`) The environmental raster to plot together with points.
`pts_alpha`	(`numeric`) The `alpha` used by `geom_sf` to show points.
`...`	Not used.

Value

A ggplot2 figure of outliers distribution among all observations.

See Also

suspicious_env_outliers, print.EnvironmentalOutlier

Examples

```
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

occ_outliers <- suspicious_env_outliers(
  occ = occ_virtual_species, variables = env_vars,
  z_outlier = 3.5, outliers_print = 4L)

plot(occ_outliers)
plot(occ_outliers,
  overlay_raster = env_vars %>% slice('band', 1))
```

plot.IndependentResponse

Function to plot independent response curves.

Description

Plot independent response curves using ggplot2.

Usage

```
## S3 method for class 'IndependentResponse'
plot(x, target_var = NA, smooth_span = 0.3, ...)
```

Arguments

x	(IndependentResponse) The independent response curve object to plot. It could be the return of function independent_response.
target_var	(vector of character) The target variable to plot. It could be NA. If it is NA, all variables will be plotted.
smooth_span	(numeric) The span value for smooth fit in ggplot2. When it is 0, no smooth applied. The default is 0.3.
...	Not used.

Value

ggplot2 figure of response curves

See Also

independent_response

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 2L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)
```

```
independent_responses <- independent_response(
  model = mod$model,
  var_occ = mod$var_train %>% st_drop_geometry(),
  variables = mod$variables)
plot(independent_responses)
```

plot.MarginalResponse *Function to plot marginal response curves.*

Description

Plot marginal response curves using ggplot2.

Usage

```
## S3 method for class 'MarginalResponse'
plot(x, target_var = NA, smooth_span = 0.3, ...)
```

Arguments

x	(MarginalResponse) The marginal response curve object to plot. It could be the return of function marginal_response.
target_var	(vector of character) The target variable to plot. It could be NA. If it is NA, all variables will be plotted.
smooth_span	(numeric) The span value for smooth fit in ggplot2. When it is 0, no smooth applied. The default is 0.3.
...	Not used.

Value

ggplot2 figure of response curves

See Also

marginal_response

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)
```

```
data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 2L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

marginal_responses <- marginal_response(
  model = mod$model,
  var_occ = mod$var_train %>% st_drop_geometry(),
  variables = mod$variables)
plot(marginal_responses, target_var = 'bio1')
```

`plot.PAConversion` *Function to plot results of conversion to PA.*

Description

Display raster of suitability, probability of occurrence, presence-absence binary map from PA conversion.

Usage

```
## S3 method for class 'PAConversion'
plot(x, ...)
```

Arguments

x	(PAConversion) The PAConversion object to plot. It could be the return of function convert_to_pa.
...	Not used.

Value

A patchwork of ggplot2 figure of suitability, probability of occurrence, presence-absence binary map.

See Also

convert_to_pa, print.PAConversion

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 1L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

# Threshold conversion
pa_thred <- convert_to_pa(mod$prediction,
  method = 'threshold', beta = 0.5)
plot(pa_thred)
```

plot.POEvaluation *Function to plot presence-only evaluation.*

Description

Display informative and detailed figures of continuous Boyce index and AUC curves.

Usage

```
## S3 method for class 'POEvaluation'
plot(x, ...)
```

Arguments

x	(POEvaluation) The presence-only evaluation object to plot. It could be the return of function evaluate_po.
...	Not used.

Value

A patchwork of ggplot2 figure of AUC_ratio, AUC_background and CBI.

See Also

evaluate_po, print.POEvaluation

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 2L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

eval_train <- evaluate_po(mod$model,
  occ_pred = mod$pred_train$prediction,
  var_pred = na.omit(as.vector(mod$prediction[[1]])))

plot(eval_train)
```

plot.ShapDependence *Function to plot variable dependence obtained from SHAP test.*

Description

Plot variable dependence curves using ggplot2.

Usage

```
## S3 method for class 'ShapDependence'
plot(x, target_var = NA, related_var = NA, smooth_span = 0.3, ...)
```

Arguments

`x`	(`ShapDependence`) The variable dependence object to plot. It could be the return of function `shap_dependence`.
`target_var`	(`vector of character`) The target variable to plot. It could be `NA`. If it is `NA`, all variables will be plotted.
`related_var`	(`character`) The dependent variable to plot together with target variables. It could be `NA`. If it is `NA`, no related variable will be plotted.
`smooth_span`	(`numeric`) The span value for smooth fit in `ggplot2`. When it is `0`, no smooth applied. The default is `0.3`.
`...`	Not used.

Value

`ggplot2` figure of dependent curves

See Also

`shap_dependence`

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 3L,
  seed = 123L, response = FALSE,
  check_variable = FALSE)

var_dependence <- shap_dependence(
  model = mod$model,
  var_occ = mod$var_train %>% st_drop_geometry())
plot(var_dependence, target_var = 'bio1', related_var = 'bio12')
```

plot.SpatialResponse *Function to plot spatial variable dependence maps.*

Description

Plot spatial variable dependence maps using ggplot2.

Usage

```
## S3 method for class 'SpatialResponse'
plot(x, target_var = NA, ...)
```

Arguments

`x` (`SpatialResponse`) The spatial variable dependence object to plot. It could be the return of function `spatial_response`.

`target_var` (`vector` of `character`) The target variable to plot. It could be `NA`. If it is `NA`, all variables will be plotted.

`...` Not used.

Value

`ggplot2` figure of dependent maps

See Also

`spatial_response`

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 3L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

spatial_responses <- spatial_response(
  model = mod$model,
  var_occ = mod$var_train %>% st_drop_geometry(),
  variables = mod$variables)
plot(spatial_responses)
plot(spatial_responses, target_var = 'bio1')
```

plot.VariableAnalysis *Function to plot variable importance.*

Description

Display informative and detailed figures of variable importance.

Usage

```
## S3 method for class 'VariableAnalysis'
plot(x, ...)
```

Arguments

x	(VariableAnalysis) The variable importance object to plot. It could be the return of function variable_analysis.
...	Not used.

Value

A patchwork of ggplot2 figure of variable importance according to multiple metrics.

See Also

variable_analysis, print.VariableAnalysis

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
```

```
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 1L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

var_analysis <- variable_analysis(
  model = mod$model,
  pts_occc = mod$pts_occ,
  pts_occ_test = mod$pts_occ_test,
  variables = mod$variables)
plot(var_analysis)
```

plot.VariableContribution

Function to plot variable contribution for target observations.

Description

Plot variable contribution for target observation separately or together using ggplot2.

Usage

```
## S3 method for class 'VariableContribution'
plot(x, plot_each_obs = FALSE, num_features = 5, ...)
```

Arguments

x	(VariableContribution) The VariableContribution object to plot. It could be the return of function variable_contrib.
plot_each_obs	(logical) The option of plot type. If TRUE, it will plot variable contribution for every observation. Otherwise, it will plot variable contribution violin plot for all observations.
num_features	(integer) A number of most important features to plot. Just work if plot_each_obs is TRUE.
...	Not used.

Value

ggplot2 figure of Variable Contribution.

See Also

variable_contrib

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 1L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

var_contribution <- variable_contrib(
  model = mod$model,
  var_occ = mod$var_train %>% st_drop_geometry(),
  var_occ_analysis = mod$var_train %>%
    st_drop_geometry() %>% slice(1:10))
plot(var_contribution,
  plot_each_obs = TRUE,
  num_features = 3)
plot(var_contribution)
```

print.EnvironmentalOutlier

Print summary information from PAConversion object.

Description

Display the most general and informative characteristics of a PAConversion object.

Usage

```
## S3 method for class 'EnvironmentalOutlier'
print(x, ...)
```

Arguments

`x`	(`EnvironmentalOutlier`) A `EnvironmentalOutlier` object to be messaged. It could be the return of function `suspicious_env_outliers`.
`...`	Not used.

Value

The same object that was passed as input.

See Also

`suspicious_env_outliers`, `plot.EnvironmentalOutlier`

Examples

```
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

occ_outliers <- suspicious_env_outliers(
  occ = occ_virtual_species, variables = env_vars,
  z_outlier = 5, outliers_print = 4L)

print(occ_outliers)
```

print.PAConversion *Print summary information from PAConversion object.*

Description

Display the most general and informative characteristics of a PAConversion object.

Usage

```
## S3 method for class 'PAConversion'
print(x, ...)
```

Arguments

x	(PAConversion) A PAConversion object to be messaged. It could be the return of function convert_to_pa.
...	Not used.

Value

The same object that was passed as input.

See Also

convert_to_pa, plot.PAConversion

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 1L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

# Threshold conversion
pa_thred <- convert_to_pa(mod$prediction, method = 'threshold', beta = 0.5)
print(pa_thred)
```

print.POEvaluation *Print summary information from presence-only evaluation object.*

Description

Display the most general and informative characteristics of a presence-only evaluation object.

Usage

```
## S3 method for class 'POEvaluation'
print(x, ...)
```

Arguments

x	(POEvaluation) A presence-only evaluation object to be messaged. It could be the return of function evaluate_po.
...	Not used.

Value

The same object that was passed as input.

See Also

evaluate_po, plot.POEvaluation

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 2L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

eval_train <- evaluate_po(mod$model,
  occ_pred = mod$pred_train$prediction,
  var_pred = na.omit(as.vector(mod$prediction[[1]])))

print(eval_train)
```

print.POIsotree *Print summary information from POIsotree object.*

Description

Display the most general and informative characteristics of a fitted POIsotree object.

Usage

```
## S3 method for class 'POIsotree'
print(x, ...)
```

Arguments

x	(POIsotree) The POIsotree object to be messaged. It could be the return of function isotree_po.
...	Not used.

Value

The same object that was passed as input.

See Also

isotree_po

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 1L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE)
print(mod)
```

print.ReducedImageStack

Print summary information from ReducedImageStack object.

Description

Display the most general and informative characteristics of a ReducedImageStack object.

Usage

```
## S3 method for class 'ReducedImageStack'
print(x, ...)
```

Arguments

x	(ReducedImageStack) A ReducedImageStack object to be messaged. It could be the return of function dim_reduce.
...	Not used.

Value

The same object that was passed as input.

See Also

dim_reduce

Examples

```
library(itsdm)
library(dplyr)
library(stars)
env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars()

img_reduced <- dim_reduce(env_vars, threshold = 0.7,
  preferred_vars = c('bio1', 'bio12'))

print(img_reduced)
```

print.VariableAnalysis

Print summary information from variable importance object.

Description

Display the most general and informative characteristics of a variable importance object.

Usage

```
## S3 method for class 'VariableAnalysis'
print(x, ...)
```

Arguments

x	(VariableAnalysis) A variable importance object to be messaged. It could be the return of function variable_analysis.
...	Not used.

Value

The same object that was passed as input.

See Also

variable_analysis, plot.VariableAnalysis

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 3L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

var_analysis <- variable_analysis(
  model = mod$model,
  pts_occ = mod$pts_occ,
  pts_occ_test = mod$pts_occ_test,
  variables = mod$variables)
```

```
print(variable_analysis)
```

probability — *Estimate suitability on stars object using trained* isolation.forest *model.*

Description

Apply an isolation.forest model on a stars object to calculate environmental suitability and do quantile stretch to [0, 1].

Usage

```
probability(x, vars, offset = 0)
```

Arguments

x	(isolation_forest). It could be the item model of POIsotree made by function isotree_po.
vars	(stars) The stack of environmental variables. More specifically, make sure it has x and y dimensions only, and distribute variables to attributes of this stars. Otherwise, the function would stop.
offset	(numeric) The offset to adjust fitted suitability. The default is zero. Highly recommend to leave it as default.

Value

a stars of predicted habitat suitability

See Also

isotree_po

Examples

```
## Not run:
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

# Prepare data
data("occ_virtual_species")
```

```
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

# Modeling
mod_virtual_species <- isotree_po(occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50, sample_size = 0.8, ndim = 2L,
  seed = 123L)
suit <- probability(mod_virtual_species$model,
  mod_virtual_species$variables)

## End(Not run)
```

shap_dependence *Calculate Shapley value based variable dependence.*

Description

Calculate the variable dependence using Shapley values.

Usage

```
shap_dependence(model, var_occ, shap_nsim = 100, visualize = FALSE, seed = 10L)
```

Arguments

`model` (`isolation_forest`). The isolation forest SDM. It could be the item `model` of `POIsotree` made by function `isotree_po`.

`var_occ` (`data.frame`, `tibble`) The `data.frame` style table that include values of environmental variables at occurrence locations.

`shap_nsim` (`integer`) The number of Monte Carlo repetitions in SHAP method to use for estimating each Shapley value. When the number of variables is large, a smaller shap_nsim could be used. See details in documentation of function `explain` in package `fastshap`. The default is 100.

`visualize` (`logical`) if `TRUE`, plot the variable dependence plots. The default is `FALSE`.

`seed` (`integer`) The seed for any random progress. The default is `10L`.

Details

The values show how each environmental variable independently affects the modeling prediction. They show how the Shapley value of each variable changes as its value is varied.

Value

(ShapDependence) A list of

- dependences_cont (list) A list of Shapley values of continuous variables
- dependences_cat (list) A list of Shapley values of categorical variables
- feature_values (data.frame) A table of feature values

References

- Strumbelj, Erik, and Igor Kononenko. "Explaining prediction models and individual predictions with feature contributions." *Knowledge and information systems* 41.3 (2014): 647-665.doi: 10.1007/s101150130679x
- Sundara rajan, Mukund, and Amir Najmi. "The many Shapley values for model explanation ." *International Conference on Machine Learning*. PMLR, 2020.
- https://github.com/bgreenwell/fastshap
- https://github.com/slundberg/shap

See Also

plot.ShapDependence explain in fastshap

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 1L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

var_dependence <- shap_dependence(
  model = mod$model,
  var_occ = mod$var_train %>% st_drop_geometry())
```

spatial_response — *Calculate spatial response or dependence plot.*

Description

Calculate spatially marginal, independence, and SHAP-based response plots.

Usage

```
spatial_response(
  model,
  var_occ,
  variables,
  shap_nsim = 0,
  seed = 10L,
  visualize = FALSE
)
```

Arguments

model — (isolation_forest). It could be the item model of POIsotree made by function isotree_po.

var_occ — (data.frame, tibble) The data.frame style table that include values of environmental variables at occurrence locations.

variables — (stars) The stars of environmental variables. It should have multiple attributes instead of dims. If you have raster object instead, you could use st_as_stars to convert it to stars or use read_stars directly read source data as a stars. You also could use item variables of POIsotree made by function isotree_po.

shap_nsim — (integer) The number of Monte Carlo repetitions in SHAP method to use for estimating each Shapley value. See details in documentation of function explain in package fastshap. Set it to 0 if you don't want to make SHAP-based spatial dependence. When the number of variables is large, a smaller shap_nsim could be used. Be cautious that making SHAP-based spatial dependence will be slow because of Monte-Carlo computation for all pixels. But it is worth the time because it is much more informative. See details in documentation of function `explain` in package `fastshap`. The default is 0. Usually a value 10 - 20 is enough.

`seed` (`integer`) The seed for any random progress. The default is `10L`.

`visualize` (`logical`) if `TRUE`, plot the response curves. The default is `FALSE`.

Details

The values show how each environmental variable affects the modeling prediction in space. These maps could help to answer questions of where in terms of environmental response. Compared to marginal dependence or independent dependence maps, SHAP-based maps are way more informative because SHAP-based dependence explain the contribution of each variable to final result.

Value

(`SpatialResponse`) A list of

- spatial_marginal_response (`list`) A list of `stars` object of spatially marginal response of all variables
- spatial_independent_response (`list`) A list of `stars` object of spatially independent response of all variables
- spatial_shap_dependence (`list`) A list of `stars` object of spatially SHAP-based response of all variables

See Also

`plot.SpatialResponse`

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 20,
  sample_size = 0.8, ndim = 2L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

spatial_responses <- spatial_response(
  model = mod$model,
  var_occ = mod$var_train %>% st_drop_geometry(),
  variables = mod$variables,
  shap_nsim = 1)
plot(spatial_responses)
```

---

suspicious_env_outliers

Function to detect suspicious outliers based on environmental variables.

Description

Run outlier.tree to detect suspicious outliers in observations.

Usage

```
suspicious_env_outliers(
  occ,
  occ_crs = 4326,
  variables,
  rm_outliers = FALSE,
  seed = 10L,
  ...
)
```

Arguments

occ	(data.frame, sf, SpatialPointsDataFrame) The occurrence dataset for training. There must be column x and y for coordinates if it is a regular data.frame.
occ_crs	(numeric or crs) The EPSG number or crs object of occurrence CRS. The default value is 4326, which is the geographic coordinate system.
variables	(RasterStack or stars) The stack of environmental variables.

`rm_outliers` (`logical`) The option to remove the suspicious outliers or not. The default is `FALSE`.

`seed` (`integer`) The random seed used in the modeling. It should be an integer. The default is `10L`.

`...` Other arguments passed to function `outlier.tree` in package `outliertree`.

Details

Please check more details in R documentation of function `outlier.tree` in package `outliertree` and their GitHub.

Value

(`EnvironmentalOutlier`) A list that contains

- outliers (`sf`) The `sf` points of outliers
- outlier_details (`tibble`) A table of outlier details returned from function `outlier.tree` in package `outliertree`
- pts_occ (`sf`) The `sf` points of occurrence. If `rm_outliers` is `TRUE`, outliers are deleted from points of occurrence. If `FALSE`, the full observations are returned.

References

- Cortes, David. "Explainable outlier detection through decision tree conditioning." *arXiv preprint arXiv:2001.00636* (2020).
- `https://github.com/david-cortes/outliertree`

See Also

`print.EnvironmentalOutlier`, `plot.EnvironmentalOutlier` `outlier.tree` in package `outliertree`

Examples

```
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

occ_outliers <- suspicious_env_outliers(
  occ = occ_virtual_species, variables = env_vars,
  z_outlier = 3.5, outliers_print = 4L)

occ_outliers
plot(occ_outliers)
```

variable_analysis *Function to evaluate relative importance of each variable.*

Description

Evaluate relative importance of each variable within the model using the following methods:

- Jackknife test based on AUC ratio and Pearson correlation between the result of model using all variables
- SHapley Additive exPlanations (SHAP) according to Shapley values

Usage

```
variable_analysis(
  model,
  pts_occ,
  pts_occ_test = NULL,
  variables,
  shap_nsim = 100,
  visualize = FALSE,
  seed = 10
)
```

Arguments

model (isolation_forest) The extended isolation forest SDM. It could be the item model of POIsotree made by function isotree_po.

pts_occ (sf) The sf style table that include training occurrence locations.

pts_occ_test (sf, or NULL) The sf style table that include occurrence locations of test. If NULL, it would be set the same as var_occ. The default is NULL.

variables (stars) The stars of environmental variables. It should have multiple attributes instead of dims. If you have raster object instead, you could use st_as_stars to convert it to stars or use read_stars directly read source data as a stars.

shap_nsim (integer) The number of Monte Carlo repetitions in SHAP method to use for estimating each Shapley value. See details in documentation of function explain in package fastshap.

visualize (logical) If TRUE, plot the analysis figures. The default is FALSE.

seed (integer) The seed for any random progress. The default is 10L.

Details

Jackknife test of variable importance is reflected as the decrease in a model performance when an environmental variable is used singly or is excluded from the environmental variable pool. In this function, we used Pearson correlation and AUC ratio.

Pearson correlation is the correlation between the predictions generated by different variable importance evaluation methods and the predictions generated by the full model as the assessment of mode performance.

The area under the ROC curve (AUC) is a threshold-independent evaluator of model performance, which needs both presence and absence data. A ROC curve is generated by plotting the proportion of correctly predicted presence on the y-axis against 1 minus the proportion of correctly predicted absence on x-axis for all thresholds. Multiple approaches have been used to evaluate accuracy of presence-only models. Peterson et al. (2008) modified AUC by plotting the proportion of correctly predicted presence against the proportion of presences falling above a range of thresholds against the proportion of cells of the whole area falling above the range of thresholds. This is the so called **AUC ratio** that is used in this package.

SHapley Additive exPlanations (SHAP) uses Shapley values to evaluate the variable importance. The larger the absolute value of Shapley value, the more important this variable is. Positive Shapley values mean positive affect, while negative Shapely values mean negative affect. Please check references for more details if you are interested in.

Value

(`VariableAnalysis`) A list of

- variables (`vector` of `character`) The names of environmental variables
- pearson_correlation (`tibble`) A table of Jackknife test based on Pearson correlation
- full_AUC_ratio (`tibble`) A table of AUC ratio of training and test dataset using all variables, that act as references for Jackknife test
- AUC_ratio (`tibble`) A table of Jackknife test based on AUC ratio
- SHAP (`tibble`) A table of Shapley values of training and test dataset separately

References

- Peterson, A. Townsend, Monica Papeş, and Jorge Soberón. "Rethinking receiver operating characteristic analysis applications in ecological niche modeling." *Ecological modelling* 213.1 (2008): 63-72.doi: 10.1016/j.ecolmodel.2007.11.008
- Strumbelj, Erik, and Igor Kononenko. "Explaining prediction models and individual predictions with feature contributions." *Knowledge and information systems* 41.3 (2014): 647-665.doi: 10.1007/s101150130679x
- Sundara rajan, Mukund, and Amir Najmi. "The many Shapley values for model explanation ." *International Conference on Machine Learning*. PMLR, 2020.
- `https://github.com/bgreenwell/fastshap`
- `https://github.com/slundberg/shap`

See Also

`plot.VariableAnalysis`, `print.VariableAnalysis` `explain` in `fastshap`

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data("occ_virtual_species")
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 2L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

var_analysis <- variable_analysis(
  model = mod$model,
  pts_occ = mod$pts_occ,
  pts_occ_test = mod$pts_occ_test,
  variables = mod$variables)
plot(var_analysis)
```

variable_contrib *Evaluate variable contributions for targeted observations.*

Description

Evaluate variable contribution for targeted observations according to SHapley Additive exPlanations (SHAP).

Usage

```
variable_contrib(
  model,
  var_occ,
  var_occ_analysis,
  shap_nsim = 100,
  visualize = FALSE,
  seed = 10
)
```

Arguments

`model` (`isolation_forest`) The isolation forest SDM. It could be the item `model` of `POIsotree` made by function `isotree_po`.

`var_occ` (`data.frame`, `tibble`) The `data.frame` style table that include values of environmental variables at occurrence locations.

`var_occ_analysis` (`data.frame`, `tibble`) The `data.frame` style table that include values of environmental variables at occurrence locations for analysis. It could be either `var_occ` or its subset, or any new dataset.

`shap_nsim` (`integer`) The number of Monte Carlo repetitions in SHAP method to use for estimating each Shapley value. See details in documentation of function `explain` in package `fastshap`.

`visualize` (`logical`) if `TRUE`, plot the response curves. The default is `FALSE`.

`seed` (`integer`) The seed for any random progress. The default is `10L`.

Value

(`VariableContribution`) A list of

- shapley_values (`data.frame`) A table of Shapley values of each variables for all observations
- feature_values (`tibble`) A table of values of each variables for all observations

References

- https://github.com/ bgreenwell/fastshap
- https://github.com/slundberg/shap

See Also

`plot.VariableContribution` `explain` in `fastshap`

Examples

```
# Using a pseudo presence-only occurrence dataset of
# virtual species provided in this package
library(dplyr)
library(sf)
library(stars)
library(itsdm)

data(”occ_virtual_species”)
occ_virtual_species <- occ_virtual_species %>%
  mutate(id = row_number())

set.seed(11)
occ <- occ_virtual_species %>% sample_frac(0.7)
occ_test <- occ_virtual_species %>% filter(! id %in% occ$id)
occ <- occ %>% select(-id)
occ_test <- occ_test %>% select(-id)

env_vars <- system.file(
  'extdata/bioclim_tanzania_10min.tif',
  package = 'itsdm') %>% read_stars() %>%
  slice('band', c(1, 5, 12, 16))

mod <- isotree_po(
  occ = occ, occ_test = occ_test,
  variables = env_vars, ntrees = 50,
  sample_size = 0.8, ndim = 3L,
  seed = 123L, response = FALSE,
  spatial_response = FALSE,
  check_variable = FALSE)

var_contribution <- variable_contrib(
  model = mod$model,
  var_occ = mod$var_train %>% st_drop_geometry(),
  var_occ_analysis = mod$var_train %>%
    st_drop_geometry() %>% slice(1:10))
## Not run:
plot(var_contribution,
  num_features = 3,
  plot_each_obs = TRUE)

# Plot together
plot(var_contribution)

## End(Not run)
```

worldclim2 *Download environmental variables made by worldclim version 2.1.*

Description

Parse historic worldclim version 2.1 variables with a setting of boundary and a few other options.

Usage

```
worldclim2(
  var = "tmin",
  res = 10,
  bry = NULL,
  path = NULL,
  nm_mark = "clip",
  return_stack = TRUE
)
```

Arguments

`var` (`character`) The option for the variable to download, should be one of tvag, tmin, tmax, prec, srad, wind, vapr and bio. The default is 'tmin'.

`res` (`numeric`) The option for the resolution of image to download. Should be one of 0.5, 2.5, 5, 10 in minute degree. The default is 10.

`bry` (`sf` or `sp`) The boundary to mask the downloaded original data. If `NULL`, it would get global map. If not `NULL`, it can take `sf`, `sfc`, `SpatialPolygonsDataFrame`, `SpatialPolygons`, etc. The default is `NULL`.

`path` (`character`) The path to save the downloaded imagery. If `NULL`, it would use the current working directory. The default is `NULL`.

`nm_mark` (`character`) the name mark of clipped images. The default is "clip". It would be ignored if `bry` is `NULL`.

`return_stack` (`logical`) if `TRUE`, stack the imagery together and return. If the area is large and resolution is high, it is better not to stack them. The default is `TRUE`.

Details

Web page page for this dataset

Value

if `return_stack` is `TRUE`, the images would be returned as a `stars`. Otherwise, nothing to return, but the user would receive a message of where the images are.

Note

If it fails due to slow internet, try to set a larger timeout option, e.g., using `options(timeout = 1e3)`.

References

Fick, Stephen E., and Robert J. Hijmans. "WorldClim 2: new 1-km spatial resolution climate surfaces for global land areas." *International journal of climatology* 37.12 (2017): 4302-4315.doi: 10.1002/joc.5086

Examples

```
## Not run:
library(sf)
library(itsdm)

bry <- sf::st_polygon(
  list(rbind(c(29.34, -11.72), c(29.34, -0.95),
             c(40.31, -0.95), c(40.31, -11.72),
             c(29.34, -11.72)))) %>%
  st_sfc(crs = 4326)

bios <- worldclim2(var = "tmin", res = 10,
  bry = bry, nm_mark = 'exp', path = tempdir())

## End(Not run)
```

Index