Package: tidysdm (via r-universe)

<center>November 20, 2024</center>

Title Sprecies Distribution Modelling with Tidymodels

Version 0.0.0.9000

Description Provides functions and classes to help create Species
Distribution Models (SDMs) within the {tidymodels} framework.
Provides new {parsnip} models, new {recipes} steps, and
{yardstick} metrics. This package provides access to the
powerful {tidymodels} ecosystem for SDMs, especially the
package {spatialsample} for spatially aware cross validation
and model evaluation, as well as opening up all classification
models implemented in {parsnip} for presence-only SDMs, using
the pseudo-absence approach.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Depends parsnip, R (>= 2.10), recipes

Imports dplyr, fasterize, nngeo, purrr, rlang (>= 0.4.11), sf, stars

Suggests knitr, rmarkdown, roxygen2, ENMTools, tidyverse, tidymodels,
spatialsample, lwgeom, ranger, vip

VignetteBuilder knitr

URL <https://rdinnager.github.io/tidysdm/>

LazyData true

Remotes danlwarren/ENMTools

Config/pak/sysreqs libgdal-dev gdal-bin libgeos-dev libicu-dev
libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository https://rdinnager.r-universe.dev

RemoteUrl https://github.com/rdinnager/tidysdm

RemoteRef HEAD

RemoteSha 5b82b81199bdd661ef21d594b43157081c8ee5df

Contents

---

add_env_vars *Add environmental variables*

Description

Add environmental variables

Usage

```
add_env_vars(x, env)
```

Arguments

env

---

create_background *Make background from points*

Description

Make background from points

Usage

```
create_background(
  x,
 method = c("convex_hull", "point_buffer", "ecoregion", "concave_hull", "grid_fill",
    "user_fill"),
  buffer = 0,
  max_bg = NULL,
  grid_options = NULL,
  concave_options = NULL
)
```

Arguments

concave_options

---

create_prediction_grid

Generate a grid of values for making predictions

Description

Generate a grid of values for making predictions

Usage

```
create_prediction_grid(bg, n = 10000, square = TRUE, include_polygons = FALSE)
```

Arguments

include_polygons

---

get_spatial_neighbours

Get Spatial Neighbours

Description

Get Spatial Neighbours

Usage

```
get_spatial_neighbours(
  x,
  k = 8,
  maxdist = 1e+05,
  progress = TRUE,
  parallel = 1
)
```

Arguments

x	An sf object
k	The number of nearest neighbours to retrieve
maxdist	The maximum distance to get nearest neighbours from
progress	Print progess bar?
parallel	How many parallel cores to use.

Value

A list of neighbour indices

---

po_spatial_buffer_vfold_cv

Spatial Cross Validation Designed for Presence-Only Data

Description

Spatial Cross Validation Designed for Presence-Only Data

Usage

```
po_spatial_buffer_vfold_cv(
  data,
  radius = NULL,
  buffer = 0.1,
  v = 10,
  repeats = 1,
  presence = NULL,
  pool = 0.1,
  ...
)
```

Arguments

```
...
```

---

sdm_data   *Convert presence points and a background area into an* sdm_data *object suitable for a* tidysdm *workflow.*

Description

Convert presence points and a background area into an sdm_data object suitable for a tidysdm workflow.

Usage

```
sdm_data(pres, bg, n = 500, abs = NULL, sample_options = list(), ...)
```

Arguments

pres	Points representing species occurrences. Should be an `sf` object with `crs` information
bg	A background area as an `sf` polygon or a raster mask (in `stars` or `raster` format)
n	Integer specifying the number of background or pseudo-absence points to sample from the background as specified by bg.
abs	Optionally an `sf` object with true absence data in the form of points.
...	

Value

A `sdm_data` object inheriting from a `tibble`

---

step_sample_pseudo_absences
Title

Description

Title

Usage

```
step_sample_pseudo_absences(
  recipe,
  bg_geometry = NULL,
  role = NA,
  trained = FALSE,
  size,
  options = list(type = "random", exact = FALSE),
  skip = FALSE,
  id = rand_id("sample_pseudo_absences")
)
```

Arguments

id

Index