Package ‘prioritizrdata’

April 6, 2023

Type Package
Version 0.3.2
Title Conservation Planning Datasets
Description Conservation planning datasets for learning how to use the ‘prioritizr’ package <https://CRAN.R-project.org/package=prioritizr>.
Imports utils, terra (>= 1.6-53), sf (>= 1.0-12), tibble (>= 2.0.0)
Suggests testthat, roxygen2, knitr
Depends R (>= 4.1.0)
License GPL-3
Encoding UTF-8
Language en-US

BugReports https://github.com/prioritizr/prioritizrdata/issues
RoxygenNote 7.2.3

NeedsCompilation no

Author Jeffrey O Hanson [aut] (<https://orcid.org/0000-0002-4716-6134>),
Richard Schuster [aut, cre] (<https://orcid.org/0000-0003-3191-7869>),
Nina Morrell [aut],
Matthew Strimas-Mackey [aut] (<https://orcid.org/0000-0001-8929-7776>),
Matthew E Watts [aut],
Peter Arcese [aut] (<https://orcid.org/0000-0002-8097-482X>),
Joseph Bennett [aut] (<https://orcid.org/0000-0002-3901-9513>),
Hugh P Possingham [aut] (<https://orcid.org/0000-0001-7755-996X>)

Maintainer Richard Schuster <richard.schuster@glel.carleton.ca>

Repository CRAN

Date/Publication 2023-04-06 09:00:13 UTC
R topics documented:

prioritizrdata ................................................................. 2
salt_data ................................................................. 2
tas_data ................................................................. 4
wa_data ................................................................. 5

Index

prioritizrdata  prioritizrdata: Conservation Planning Datasets

Description

The prioritizrdata package is a supplemental package that contains example datasets for conservation planning. It is intended to be used alongside the prioritizr package.

Details

This package contains the following datasets:

- **tas_data** Conservation planning dataset for Tasmania, Australia.
- **salt_data** Conservation planning dataset for Salt Spring Island, Canada.
- **wa_data** Conservation planning dataset for Washington, The United States of America.

salt_data  Salt Spring Island data

Description

Conservation planning dataset for Salt Spring Island, Canada. It was obtained as part of an online Marxan-based planning tool created for the Coastal Douglas-fir Conservation Partnership (CDFCP; Schuster et al. 2017).

Usage

- `get_salt_pu()`
- `get_salt_features()`
- `get_salt_con()`

Format

- **salt_pu** `terra::rast()` object.
- **salt_features** `terra::rast()` object.
- **salt_con** `terra::rast()` object.
Details

The following functions are provided to import data:

get_salt_pu() Import planning unit data. The planning units are a single layer `terra::rast()` object. Cell values denote the monetary cost of acquiring different areas (e.g., a value of 1 = $100,000 CAD; BC Land Assessment 2015).

get_salt_features() Import biodiversity feature data. The feature data are a multi-layer `terra::rast()` object. It contains the spatial distribution of four key ecological communities. Each layer represents a different community type. These classes are (i) old forest, (ii) savanna, (iii) wetland, and (iv) shrub. For each layer, values indicate the composite probability of encountering the suite of bird species most commonly associated with that community type.

get_salt_con() Import connectivity data. The connectivity data are a single-layer `terra::rast()` object. It contains the inverse probability of occurrence of human commensal species. Based on the assumption that human modified areas impede connectivity for native fauna, cells with higher values have higher connectivity.

References


Examples

```r
# load packages
library(terra)
library(sf)

# import data
salt_pu <- get_salt_pu()
salt_features <- get_salt_features()

# preview planning units
print(salt_pu)
plot(salt_pu)

# preview features
print(salt_features)
plot(salt_features)

# preview connectivity data
salt_con <- get_salt_con()
print(salt_con)
plot(salt_con)
```
tas_data  Tasmania data

Description

Conservation planning dataset for Tasmania, Australia.

Usage

get_tas_pu()

get_tas_features()

Format

**tas_pu**  
*sf::st_sf()* object.

**tas_features**  
*terra::rast()* object

Details

The following functions are provided to import data:

*get_tas_pu*  
Import planning unit data. The planning units are a *sf::st_sf()* simple features object. Each row corresponds to a different planning unit, and columns contain information about the planning units. It has columns that contain: (“id”) unique identifiers and (“cost”) unimproved land values for the planning units. It also contains columns (“locked_in” and “locked_out”) with logical values (i.e. TRUE or FALSE values) for locking in and locking out planning units. These data obtained from the *Introduction to Marxan* course and were originally generated as part of a larger spatial prioritization Resources (Klein *et al.* 2007).

*get_tas_features*  
Import biodiversity feature data. The feature data are a multi-layer *terra::rast()* object. classes. Each layer corresponds to a different vegetation class and contains binary cell values that indicate the presence or absence of the vegetation class. These data were obtained from the Australian Government’s National Vegetation Information System (Australian Government Department of Climate Change, Energy, the Environment and Water 2020).

References


Examples

```r
# load packages
library(terra)
library(sf)

# load data
tas_pu <- get_tas_pu()
tas_features <- get_tas_features()

# preview planning units
print(tas_pu)
plot(tas_pu)

# plot features
print(tas_features)
plot(tas_features)
```

---

**wa_data**

*Washington data*

---

**Description**


**Usage**

```r
get_wa_pu()

get_wa_locked_in()

get_wa_locked_out()

get_wa_features()
```

**Format**

- **get_wa_pu** `terra::rast()` object.
- **get_wa_locked_in** `terra::rast()` object.
- **get_wa_locked_out** `terra::rast()` object.
- **get_wa_features** `terra::rast()` object.

**Details**

The following functions are provided to import data:
get_wa_pu() Import planning unit data. The planning units are a single layer `terra::rast()` object. Cell values denote land acquisition costs. These data were originally obtained from Nolte (2020 a,b).

get_wa_locked_in() Import locked in data. The locked in data are a single layer `terra::rast()` object. Cell values denote binary values indicating if each cell is predominantly covered by protected areas (excluding those with no mandate for biodiversity protection). These data were originally obtained from USGS (2022).

get_wa_locked_out() Import locked out data. The locked out are a single layer `terra::rast()` object. Cell values denote binary values indicating if each cell is predominantly covered by urban areas. These data were originally obtained from the Commission for Environmental Cooperation (2020).

get_wa_features() Import biodiversity feature data. The feature data are a multi-layer `terra::rast()` object object. It contains the spatial distribution of 258 bird species. To account for migratory patterns, data are provided for the breeding and non-breeding distributions of species (indicated by "breeding" and "non-breeding" in the layer names). If a species is lacking such information, then the species is denoted with its full distribution (as indicated "full" in the layer names). These data were originally obtained from the eBird Status and Trends dataset (Fink et al. 2020).

References


Examples

```r
# load packages
library(terra)

# import data
wa_pu <- get_wa_pu()
wa_features <- get_wa_features()
wa_locked_in <- get_wa_locked_in()
wa_locked_out <- get_wa_locked_out()

# preview planning units
print(wa_pu)
plot(wa_pu)
```
wa_data

# preview locked in
print(wa_locked_in)
plot(wa_locked_in)

# preview locked out
print(wa_locked_out)
plot(wa_locked_out)

# preview features
print(wa_features)
plot(wa_features)
Index

* datasets
  salt_data, 2
  tas_data, 4
  wa_data, 5
  get_salt_con(salt_data), 2
  get_salt_features(salt_data), 2
  get_salt_pu(salt_data), 2
  get_tas_features(tas_data), 4
  get_tas_pu(tas_data), 4
  get_wa_features(wa_data), 5
  get_wa_locked_in(wa_data), 5
  get_wa_locked_out(wa_data), 5
  get_wa_pu(wa_data), 5
  prioritizrdata, 2
  salt_data, 2, 2
  salt_features(salt_data), 2
  salt_pu(salt_data), 2
  sf::st_sf(), 4
  tas_data, 2, 4
  tas_features(tas_data), 4
  tas_pu(tas_data), 4
  terra::rast(), 2–6
  wa_data, 2, 5
  wa_locked_in(wa_data), 5
  wa_locked_out(wa_data), 5
  wa_pu(wa_data), 5