

Package: SISINTAR (via r-universe)

September 21, 2024

Title Lee y Manipula Datos de Suelo de SISINTA
Version 0.0.0.9000
Description Permite descargar y manipular datos de perfiles de suelo disponibles en la plataforma SISINTA.
License GPL (>= 3)
Encoding UTF-8
Language es
LazyData true
Roxygen list(markdown = TRUE)
RoxygenNote 7.2.3
Depends R (>= 2.10)
Imports aqp, progress, xml2, urltools, data.table, sf, openxlsx, httr, rvest, rappdirs, RCurl, jsonlite
Suggests rmarkdown, knitr, covr, testthat (>= 3.0.0), dplyr
Config/testthat/edition 3
VignetteBuilder knitr, dplyr
Repository <https://yabellini.r-universe.dev>
RemoteUrl <https://github.com/INTA-Suelos/SISINTAR>
RemoteRef HEAD
RemoteSha 359790580f73f88b3c588c3cb4a211ad14008914

Contents

anidar_horizontes	2
as_SoilProfileCollection	3
buscar_perfiles	3
check_sisinta	4
exportar_excel	5
exportar_shapefile	5
get_perfiles	6

imputar_profundidad_inferior	7
interpoliar_concatenar	8
interpoliar_perfiles	8
log_in	10
perfiles	10
separar_perfiles	11

Index	12
--------------	-----------

anidar_horizontes	<i>Anida datos de horizontes</i>
-------------------	----------------------------------

Description

La función `anidar_horizontes` permite organizar la información de uno o más perfiles y sus horizontes en un `data.frame`. Cada fila se corresponde con un perfil y los datos de los horizontes se guardan en una columna de `data.frames` llamada `horizonte`. `desanidar_horizontes` realiza la operación contraria.

Usage

```
anidar_horizontes(perfiles)

desanidar_horizontes(perfiles)
```

Arguments

`perfiles` un `data.frame` con perfiles (salida de `get_perfiles()`)

Value

Un `data.frame` anidado. El resultado tiene una fila por cada perfil y una columna por cada dato del perfil más una columna llamada "horizontes" que contiene un `data.frame` con los horizontes de cada perfil.

Examples

```
anidados <- anidar_horizontes(perfiles)

# igual a perfiles salvo posiblemente por el orden de las columnas
desanidados <- desanidar_horizontes(anidados)
```

as_SoilProfileCollection
Objeto SoilProfileCollection

Description

Transforma uno másperfiles de SISINTA en formato data.frame en un objeto SoilProfileCollection del paquete aqp.

Usage

```
as_SoilProfileCollection(perfiles)
```

Arguments

perfiles data.frame con perfiles descargados o leídos con la función [get_perfiles\(\)](#).

Value

un objeto [aqp::SoilProfileCollection](#)

Examples

```
coleccion <- as_SoilProfileCollection(perfiles)
```

buscar_perfiles *Buscar perfiles en base a ubicación y fechas*

Description

Buscar perfiles en base a ubicación y fechas

Usage

```
buscar_perfiles(  
  rango_lon = NULL,  
  rango_lat = NULL,  
  rango_fecha = NULL,  
  clase = NULL,  
  serie = NULL,  
  actualizar_cada = 30  
)
```

Arguments

rango_lon, rango_lat	vectores numéricos con los límites de longitud y latitud. La longitud tiene que estar entre -180° y 180° . Los límites se definen como el mínimo y el máximo valor de cada vector.
rango_fecha	vector tipo Date o que se puede cohercer a fecha con <code>as.Date()</code> que define el rango de fechas a buscar.
clase	vector de caracteres para filtrar la clase del perfil. La función lo trata como una expresión regular que no distingue mayúsculas y minúsculas. Si es un vector de longitud mayor a 1, se filtran las clases que coincidan con al menos uno de los elementos (es decir, filtra con O).
serie	vector con nombres de series. El nombre debe ser el mismo que aparece en la serie. Si alguna serie no se encuentra, la función tira error.
actualizar_cada	valor numérico que define cada cuantos días se actualiza el archivo con la información de la base de datos (no usado)

Value

Un `data.frame` con los perfiles que cumplen las condiciones de búsqueda

Examples

```
## Not run:
centro <- buscar_perfiles(rango_lat = c(-45, -30))
with(centro, plot(lon, lat))

recientes <- buscar_perfiles(rango_fecha = c("2010-01-01", "2025-01-01"))
with(recientes, plot(lon, lat))

# Perfiles donde la clase contiene "hapludol" o "natralbol"
buscar_perfiles(clase = c("hapludol", "natralbol"))

# Perfiles de la serie Ramallo
buscar_perfiles(serie = "Ramallo")

## End(Not run)
```

check_sisinta

Evalúa conexión con el servidor de SISINTA

Description

Evalúa conexión con el servidor de SISINTA

Usage

```
check_sisinta()
```

Value

Un valor lógico indicando si se puede establecer conexión con el servidor

exportar_excel	<i>Exporta perfiles a excel</i>
----------------	---------------------------------

Description

La función exporta uno o más perfiles de SISINTA descargados o leídos con [get_perfiles\(\)](#) a un archivo excel donde los datos de sitio están en una hoja llamada "sitios" y los horizontes, en una hoja llamada "horizontes".

Usage

```
exportar_excel(perfiles, archivo)
```

Arguments

perfiles	un data.frame con perfiles (salida de get_perfiles())
archivo	la ruta al archivo de salida.

Value

Devuelve la ruta al archivo de forma invisible.

Examples

```
archivo <- tempfile(fileext = ".xlsx")
exportar_excel(perfiles, archivo)
```

exportar_shapefile	<i>Exportar perfiles en shapefile</i>
--------------------	---------------------------------------

Description

La función exporta uno o más perfiles de SISINTA descargados o leídos con [get_perfiles\(\)](#) en formato shapefile. !!! REQUIERE NOMBRES DE COLUMNAS MÁS CORTOS

Usage

```
exportar_shapefile(perfiles, archivo)
```

Arguments

perfiles un data.frame con perfiles (salida de `get_perfiles()`)
 archivo la ruta al archivo de salida.

Examples

```
archivo <- tempfile(fileext = ".shp")
# exportar_shapefile(perfiles, archivo)
```

<code>get_perfiles</code>	<i>Obtiene perfiles de suelo</i>
---------------------------	----------------------------------

Description

Obtiene datos de perfiles. Por defecto, primero intenta leer datos existentes en `dir`, si no existen, usa los datos incluidos en el paquete, y si no existen, trata de descargarlos desde SISINTA. Esto garantiza reproducibilidad del proyecto o bajo mismas versiones del paquete y minimiza la dependencia de una conexión a internet.

Usage

```
get_perfiles(
  perfil_ids,
  dir = tempdir(),
  refresh = FALSE,
  parar_en_error = FALSE,
  credenciales = NULL
)
```

Arguments

`perfil_ids` vector numérico de ids de perfiles de suelo. Alternativamente puede ser un data.frame con una columna llamada `perfil_id` (la salida de `buscar_perfiles()`)

`dir` directorio donde se guardan los datos de perfiles o donde se leen perfiles ya descargados. Por defecto, los perfiles descargados se guardan en una carpeta temporal.

`refresh` booleano, fuerza descargar datos de SISINTA.

`parar_en_error` tirar un error si algún perfil no está disponible en vez de seguir intentando con los siguientes.

`credenciales` una lista con elementos "usuario" y "pass".

Value

Un data.frame.

Examples

```
## Not run:  
get_perfiles(c(3238, 4634, 4609))  
get_perfiles(4609, credenciales = list(usuario = "usuario",  
                                       pass = "pass"))  
  
## End(Not run)
```

imputar_profundidad_inferior

Imputa valores faltantes en profundidad inferior

Description

La función revisa si el perfil tiene valor faltante en la variable `profundidad_inferior` e imputa un valor dado por `profundidad_superior` más una constante definida por el argumento `profundidad`.

Usage

```
imputar_profundidad_inferior(perfiles, profundidad = 5)
```

Arguments

`perfiles` un data.frame con perfiles (salida de `get_perfiles()`)
`profundidad` la profundidad asumida de la ultima capa

Value

Un data.frame con las mismas filas y columnas que el objeto de entrada.

Examples

```
perfil <- subset(perfiles, perfil_id == 3238)  
imputar_profundidad_inferior(perfil)  
imputar_profundidad_inferior(perfil, profundidad = 10)
```

interpolar_concatenar *Métodos de interpolación*

Description

SISINTAR implementa dos métodos de interpolación para obtener horizontes estandarizados de las distintas variables disponibles. Estos métodos devuelven una función que recibe los límites superior e inferior de cada capa, las observaciones a interpolar y los horizontes o profundidades a los que se quiere interpolar cada variable.

Usage

```
interpolar_concatenar(sep = "|")
```

```
interpolar_spline(lambda = 0.1)
```

```
interpolar_promedio_ponderado()
```

Arguments

sep separador utilizado para concatenar los valores únicos de cada capa.

lambda parámetro lambda de suavizado.

Details

interpolar_promedio_ponderado utiliza un promedio ponderado para calcular los valores interpolados en las capas definidas.

interpolar_spline utiliza la funcione spline que preserva la masa de acuerdo a Bishop T.F.A. et al. (1999).

References

Bishop T.F.A. et al. (1999) Modelling soil attribute depth functions with equal-area quadratic smoothing splines. [https://doi.org/10.1016/S0016-7061\(99\)00003-8](https://doi.org/10.1016/S0016-7061(99)00003-8)

interpolar_perfiles *Interpola perfiles*

Description

La función genera perfiles normalizados a partir de horizontes estandarizados utilizando alguno de los métodos disponibles.

Usage

```
interpolar_perfiles(
  perfiles,
  variables,
  horizontes = 30,
  metodo = interpolar_promedio_ponderado(),
  metodo_categorico = interpolar_concatenar(),
  parar_en_error = FALSE
)
```

Arguments

`perfiles` un data.frame con datos de perfiles

`variables` un vector de texto con los nombres de las variables a interpolar

`horizontes` un vector numérico que determina los horizontes a usar para la interpolación o un numérico único que determina la resolución en centímetros.

`metodo, metodo_categorico` el método de interpolación utilizado para variables continuas y categóricas, respectivamente. Ver [interpolar_promedio_ponderado\(\)](#).

`parar_en_error` tirar un error si algún perfil tiene una profundidad máxima que es menor a la indicada. Si es FALSE, interpola hasta la máxima profundidad disponible y tira un warning.

Value

Un data.frame con los datos interpolados.

Examples

```
interpolar_perfiles(perfiles, c("sum_bases", "cic"))

## Not run:
# interpolar_spline() no acepta valores faltantes en las profundidades.
# Para imputar, ver imputar_profundidad_inferior().
interpolar_perfiles(perfiles, c("sum_bases", "cic"),
  metodo = interpolar_spline())

## End(Not run)
# Horizontes cada 10 centímetros entre 0 y 100.
interpolar_perfiles(perfiles, c("sum_bases", "cic"), seq(0, 100, 10))
```

log_in	<i>Inicial sesión en la página de SISINTA</i>
--------	---

Description

Esta función interna es utilizada por `get_perfiles()` para acceder a la web de SISINTA con usuario y contraseña y poder descargar los perfiles no públicos.

Usage

```
log_in(usuario, pass)
```

Arguments

usuario	string, usualmente el mail.
pass	string, contraseña asociada.

perfiles	<i>Perfiles de ejemplo</i>
----------	----------------------------

Description

Perfiles de ejemplo

Usage

```
perfiles
```

Format

An object of class `data.frame` with 18 rows and 105 columns.

separar_perfiles	<i>Separar datos de sitio y horizonte en dos tablas</i>
------------------	---

Description

Esta función permite separar el data.frame que genera `get_perfiles()` que contiene la información del sitio y sus horizontes en dos data.frames organizados en una lista.

Usage

```
separar_perfiles(perfiles)
```

Arguments

perfiles un data.frame con perfiles (salida de `get_perfiles()`)

Value

Una lista con dos elementos. El elemento `sitios`, es un data.frame con la información de los sitios y el elemneto `horizontes` es un data.frame con la información de horizontes.

Examples

```
separar_perfiles(perfiles)
```

Index

* datasets

- perfiles, [10](#)

- anidar_horizontes, [2](#)
- aqp::SoilProfileCollection, [3](#)
- as_SoilProfileCollection, [3](#)

- buscar_perfiles, [3](#)
- buscar_perfiles(), [6](#)

- check_sisinta, [4](#)

- desanidar_horizontes
 (anidar_horizontes), [2](#)

- exportar_excel, [5](#)
- exportar_shapefile, [5](#)

- get_perfiles, [6](#)
- get_perfiles(), [2](#), [3](#), [5–7](#), [10](#), [11](#)

- imputar_profundidad_inferior, [7](#)
- interpolar_concatenar, [8](#)
- interpolar_perfiles, [8](#)
- interpolar_promedio_ponderado
 (interpolar_concatenar), [8](#)
- interpolar_promedio_ponderado(), [9](#)
- interpolar_spline
 (interpolar_concatenar), [8](#)

- log_in, [10](#)

- perfiles, [10](#)

- separar_perfiles, [11](#)