

Package: LSVI (via r-universe)

September 24, 2026

Title Rekenmodule Lokale Staat Van Instandhouding van habitattypen

Version 0.1.4

Description Het package LSVI bundelt een aantal functies in verband met het ophalen van informatie over en het berekenen van de Lokale Staat Van Instandhouding (LSVI) van habitattypes.

License GPL-3

URL <https://github.com/inbo/LSVI>

BugReports <https://github.com/inbo/LSVI/issues>

Depends R (>= 3.5.0)

Imports assertthat, DBI, dplyr, methods, odbc, plyr, pool, rgbif, rlang, rmarkdown, RSQLite, stringr, tibble, tidyr, tidyselect, utils

Suggests knitr, readr, testthat, xtable

VignetteBuilder knitr

Config/checklist/communities inbo

Config/checklist/keywords habitats; local conservation status

Encoding UTF-8

Language nl-BE

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Collate 'berekenLSVibasis.R' 'berekenStatus.R'
'berekenVerschilscores.R' 'berekenVoorwaarde.R'
'combinerenVerschilscore.R' 'combinerenVoorwaarden.R'
'connecteerMetLSVIdb.R' 'connecteerMetLSVIdbServer.R'
'controleerInvoerwaarde.R' 'deselecteerKenmerkenInOpname.R'
'geefInfoHabitatfiche.R' 'geefInvoervereisten.R'
'geefSoortenlijst.R' 'geefSoortenlijstVoorIDs.R'
'geefUniekeWaarden.R' 'geefVersieInfo.R' 'geefVertaallijst.R'
'invoercontroleData_habitat.R'

```
'invoercontroleData_soortenKenmerken.R'
'invoercontroleData_voorwaarden.R'
'invoercontroleKwaliteitsniveau.R' 'invoercontroleVersie.R'
'logDatabankfouten.R' 'maakConnectiePool.R'
'maakConnectiePoolServer.R' 'maakHabitatfiches.R'
'maakLSVIrapport.R' 'parseTaxonnaam.R' 'postparseTaxonnaam.R'
'preparseTaxonnaam.R' 's4_AnalyseVariabele.R' 's4_aandeel.R'
's4_bedekking.R' 's4_aandeelKruidlaag.R' 's4_aandeelLaagExcl.R'
's4_aantal.R' 's4_aantalGroepen.R' 's4_analyseVariabele_c.R'
's4_bedekkingExcl.R' 's4_bedekkingLaag.R'
's4_bedekkingLaagExcl.R' 's4_bedekkingLaagPlus.R'
's4_bedekkingSom.R' 's4_maxBedekking.R' 's4_maxBedekking2s.R'
's4_maxBedekkingExcl.R' 's4_meting.R' 's4_scoresom.R'
'selecteerIndicatoren.R' 'selecteerKenmerkenInOpname.R'
'vertaalIntervalUitvoer.R' 'vertaalInvoerInterval.R' 'zzz.R'
```

Config/pak/sysreqs cmake make libicu-dev libuv1-dev libxml2-dev unixodbc-dev libssl-dev

Repository <https://inbo.r-universe.dev>

Date/Publication 2026-05-04 13:19:16 UTC

RemoteUrl <https://github.com/inbo/lsvi>

RemoteRef HEAD

RemoteSha c7325cf2a3e64af9771735d9fea970e33f404d91

Contents

berekenLSVibasis	3
berekenStatus	6
berekenVerschilscores	7
berekenVoorwaarde	7
combinerenVerschilscore	8
combinerenVoorwaarden	9
connecteerMetLSVIdb	10
controleerInvoerwaarde	11
deselecteerKenmerkenInOpname	12
geefInfoHabitatfiche	13
geefInvoervereisten	14
geefSoortenlijst	17
geefSoortenlijstVoorIDs	19
geefUniekeWaarden	20
geefVersieInfo	21
geefVertaallijst	22
invoercontroleData_habitat	23
invoercontroleData_soortenKenmerken	23
invoercontroleData_voorwaarden	24
invoercontroleKwaliteitsniveau	25
invoercontroleVersie	25
logDatabankfouten	26

maakConnectiePool	26
maakHabitatfiches	27
maakLSVIrapport	28
parseTaxonnaam	30
selecteerIndicatoren	31
selecteerKenmerkenInOpname	32
vertaalIntervalUitvoer	33
vertaalInvoerInterval	34

Index	36
--------------	-----------

berekenLSVIbasis	<i>Berekent de LSVI op basis van VoorwaardeID en opgegeven waarden</i>
------------------	--

Description

Deze functie bepaalt de Lokale Staat van Instandhouding en biotische indices op basis van gegevens, die in het juiste formaat moeten aangeleverd worden. Zie hiervoor de beschrijving bij de parameters ("Arguments") en de tabellen van het voorbeeld. In principe is enkel de parameter Data_habitat verplicht om op te geven, maar extra datasets zijn uiteraard wel nodig om een resultaat te bekomen. Welke datasets relevant zijn, is afhankelijk van de opgegeven habitattypes: voor een aantal habitattypes kan een tabel met observaties en hun bedekking of aanwezigheid (=parameter Data_soortenKenmerken) volstaan, voor bossen zijn bv. bijkomend gegevens nodig over dood hout.

De Lokale Staat van Instandhouding wordt weergegeven in de kolom Status met als mogelijke waarden TRUE (= gunstig) en FALSE (= ongunstig).

De biotische indices zijn afgeleid van het verschil tussen een geobserveerde waarde en de referentiewaarde voor elke indicator. Deze verschillen werden herschaald tussen +1 en -1, waarbij een positieve en negatieve waarde overeenkomt met respectievelijk een gunstige en ongunstige score. Deze verschillen worden geaggregeerd, eerst voor de indicatoren die tot eenzelfde criterium behoren, vervolgens worden deze geaggregeerde scores verder geaggregeerd om tot een globale index te komen. Er worden drie verschillende globale indices berekend waarbij de naamgeving aangeeft welk aggregatie achtereenvolgens gebruikt werd: "index_min_min", "index_min_harm" en "index_harm_harm". Een naam met "min" duidt op minimum van de scores als aggregatie; bij "harm" werd het harmonisch gemiddelde berekend.

Usage

```
berekenLSVIbasis(
  Versie = "alle",
  Kwaliteitsniveau = "alle",
  Data_habitat,
  Data_voorwaarden = data.frame(ID = character(), Criterium = character(), Indicator =
    character(), Voorwaarde = character(), Waarde = character(), Type = character(),
    WaardeMin = double(), WaardeMax = double(), stringsAsFactors = FALSE),
  Data_soortenKenmerken = data.frame(ID = character()),
  Aggregatiemethode = "1-out-all-out",
```

```

ConnectieLSVIhabitats = NULL,
LIJST = geefVertaallijst(ConnectieLSVIhabitats),
na.rm = FALSE
)

```

Arguments

Versie	De versie van het LSVI-rapport op basis waarvan de berekening gemaakt wordt, bv. "Versie 2.0" of "Versie 3". Bij de default "alle" wordt de LSVI volgens de verschillende versies berekend.
Kwaliteitsniveau	Voor elke versie van de LSVI zijn er een of meerdere kwaliteitsniveaus gedefinieerd in de databank. Zo is er bij Versie 2.0 een onderscheid gemaakt tussen goede staat (A), voldoende staat (B) en gedegradeerde staat (C). Hier duidt kwaliteitsniveau 1 de grens tussen voldoende (B) en gedegrademd (C) aan en kwaliteitsniveau 2 het onderscheid tussen goed (A) en voldoende (B). Bij Versie 3 duidt kwaliteitsniveau 1 op het onderscheid tussen ongunstig en gunstig en kwaliteitsniveau 2 op de streefwaarde (uiteindelijk niet opgenomen in rapport). De betekenissen van de 2 kwaliteitsniveaus voor de verschillende versies is weergegeven in de tabel Versie in de databank en kan opgevraagd met de functie geefVersieInfo(). Geef als parameter Kwaliteitsniveau op op basis van welk kwaliteitsniveau de berekening gemaakt moet worden. (Strikt genomen is de berekening van de LSVI de berekening volgens kwaliteitsniveau 1.)
Data_habitat	Een opsomming van de te analyseren opnamen met opgave van het aanwezige habitatype (= het habitatype volgens welke criteria de beoordeling moet gebeuren). Deze info moet doorgegeven worden in de vorm van een dataframe met minimum de velden ID en Habitatype, waarbij ID een groeperende variabele is voor een opname (plaats en tijdstip). Habitatype moet overeenkomen met de naamgeving in de LSVI-databank (op te zoeken door geefUniekeWaarden("Habitatype", "Code")). Eventuele extra velden zullen overgenomen worden bij de uitvoer.
Data_voorwaarden	Gegevens over de opgemeten indicatoren in de vorm van een dataframe met velden ID, Criterium, Indicator, Voorwaarde, Waarde, Type, Invoertype en Eenheid, waarbij ID de groeperende variabele voor een opname is die ook bij Data_habitat opgegeven is. Criterium, Indicator en Voorwaarde moeten overeenkomen met de waarde in de databank (op te zoeken via de functie geefInvoervereisten()). Waarde is de waarde die voor die voorwaarde geobserveerd of gemeten is en Type het soort variabele (zie geefUniekeWaarden("TypeVariabele", "Naam") voor de mogelijke waarden). Ingeval van een categorische variabele moet bij Invoertype de naam van de lijst opgegeven worden waaruit deze waarde komt (bv. welke schaal gebruikt is, zie geefUniekeWaarden("Lijst", "Naam") voor alle mogelijkheden). Als een indicator rechtstreeks op het veld ingeschat is, kan deze ingevoerd worden door in deze tabel de kolom voorwaarde leeg te laten (wat in R aangeduid wordt door NA) en als waarde "TRUE" of "FALSE" in te geven. In dit geval moeten Type, Invoertype en Eenheid niet ingevoerd worden.
Data_soortenKenmerken	Gegevens van soorten en kenmerken en hun bedekking (m.a.w. enkel kenmerken waarvan een bedekking gemeten is, horen in deze tabel). Deze dataframe moet

de velden ID, Vegetatielaag, Kenmerk, TypeKenmerk, Waarde, Type, Invoertype en Eenheid bevatten, waarbij ID de groeperende variabele voor een opname is die ook bij Data_habitat opgegeven is. Kenmerk bevat een soortnaam of een naam die voorkomt in de lijst gegenereerd door `geefUniekeWaarden("StudieItem", "Waarde")` en TypeKenmerk geeft een beschrijving voor dat kenmerk: "studiegroep", "soort_Latijn", "soort_NL" of "soort_gbif". Waarde is de geobserveerde bedekking en Type het soort variabele dat voor de bedekking gebruikt is (zie `geefUniekeWaarden("TypeVariabele", "Naam")` voor de mogelijke waarden). Ingeval van een categorische variabele moet bij Invoertype de naam van de lijst opgegeven worden welke schaal gebruikt is (zie `geefUniekeWaarden("Lijst", "Naam")` voor alle mogelijkheden).

Aggregatiemethode

Keuze van de methode om tot één beoordeling per criterium of per habitatvlek of meetpunt te komen. Er zijn twee opties: (1) "RapportageHR": de beoordeling is gunstig als meer dan 50 procent van de indicatoren gunstig zijn EN als geen enkele zeer belangrijke indicator ongunstig is; (2) "1-out-all-out": de beoordeling is gunstig als alle indicatoren gunstig zijn. "1-out-all-out" is default.

ConnectieLSVIhabitats

Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie `connecteerMetLSVIdb()`.

LIJST

Dataframe met lijst die weergeeft hoe de vertaling moet gebeuren van categorische variabelen naar numerieke waarden (en omgekeerd). Default worden deze waarden uit de databank met LSVI-indicatoren gehaald d.m.v. de functie `vertaalInvoerInterval()`. Aangeraden wordt om deze default te gebruiken (dus parameter niet expliciet invullen), of deze waar nodig aan te vullen met eigen schalen. Omdat er ook een omzetting moet gebeuren voor grenswaarden uit de databank, kan het niet doorgeven van een gedeelte van deze lijst problemen geven.

na.rm

Hier geef je aan hoe de berekening moet omgaan met NA-waarden. Default is FALSE. Dit betekent dat NA-waarden niet worden verwijderd. Hierdoor zal de indexberekening resulteren in een NA zodra één van de indicatoren NA is. Voor de berekening van de status zal dit enkel resulteren in een NA indien minstens één van de indicatoren NA is en minstens één van de indicatoren status TRUE (= gunstig) heeft. Indien `na.rm = TRUE` worden eventuele NA-waarden verwijderd zodat status en de indices een resultaat hebben. Doordat deze dan mogelijk niet op de volledige set van indicatoren gebaseerd zijn, moet hiermee rekening gehouden worden afhankelijk van de context waarvoor de resultaten gebruikt worden.

Value

Deze functie genereert de resultaten in de vorm van een list met 4 tabellen: een eerste met de beoordelingen per kwaliteitsniveau, een tweede met de beoordelingen per criterium en kwaliteitsniveau, een derde met de beoordelingen per indicator en kwaliteitsniveau, en een vierde met de detailgegevens inclusief meetwaarden.

Examples

```
# Omwille van de iets langere lange duurtijd van de commando's staat bij
```

```
# onderstaand voorbeeld de vermelding 'dontrun' (om problemen te vermijden
# bij het testen van het package). Maar het voorbeeld werkt en kan zeker
# uitgetest worden.
## Not run:
library(LSVI)
maakConnectiePool()
library(readr)
Data_habitat <-
  read_csv2(system.file("vbdata/Opname4030habitat.csv", package = "LSVI"),
    col_types = list(col_character(), col_character(), col_character()))
Data_voorwaarden <-
  read_csv2(
    system.file("vbdata/Opname4030voorwaardenv2.csv", package = "LSVI"))
Data_soortenKenmerken <-
  read_csv2(
    system.file("vbdata/Opname4030soortenKenmerken.csv", package = "LSVI"))
berekenLSVibasis(Versie = "Versie 2.0",
  Kwaliteitsniveau = "1", Data_habitat,
  Data_voorwaarden, Data_soortenKenmerken)

## End(Not run)
```

berekenStatus
Berekent de Status voor de records van een een opgegeven tabel

Description

Deze functie, die bedoeld is als hulpfunctie voor de hoofdfunctie `berekenLSVibasis()`, evalueert de status van de records van een opgegeven Statustabel met velden Waarde, Referentiewaarde en Operator.

Usage

```
berekenStatus(Statustabel)
```

Arguments

Statustabel Dataframe met velden Rijnr, RefMin, RefMax, Operator, WaardeMin en WaardeMax.

Value

Deze functie geeft een tabel terug met velden Rijnr en Status

berekenVerschilscores	<i>Berekent de verschilscores voor de records van een opgegeven tabel</i>
-----------------------	---

Description

Deze functie, die bedoeld is als hulpfunctie voor de hoofdfunctie berekenLSVIbasis(), berekent de verschilscores van de records van een opgegeven Statustabel met velden Rijnr, RefMin, RefMax, Operator, WaardeMin, WaardeMax en TheoretischMaximum. De verschilscores hebben een waarde tussen -1 en +1 en geven negatieve of positieve afwijking ten opzichte van de referentiewaarde.

Usage

```
berekenVerschilscores(Statustabel)
```

Arguments

Statustabel	Dataframe met velden Rijnr, RefMin, RefMax, Operator, WaardeMin, WaardeMax, TheoretischMaximum en TypeVariabele.
-------------	--

Value

Deze functie geeft een tabel terug met velden Rijnr en Verschilscore

berekenVoorwaarde	<i>Berekent de voorwaarde op basis van datasets</i>
-------------------	---

Description

Deze hulpfunctie berekent de waarde voor een opgegeven voorwaarde (verwijzend naar de "rekenregels" in de LSVI-indicatorendatabank) op basis van opgegeven datasets. Ze doet dit voor 1 enkele voorwaarde en 1 enkele opname (datum en locatie). Deze functie test NIET of de datasets zich beperken tot een enkele opname, dus het is aan de gebruiker om enkel gegevens van 1 locatie mee te geven. Voor een berekening van meerdere opnamen (en ook de volledige LSVI i.p.v. enkel 1 voorwaarde) verwijzen we naar de functie berekenLSVIbasis().

Usage

```
berekenVoorwaarde(
  OpnameID,
  VoorwaardeID,
  Kenmerken,
  ConnectieLSVIhabitats,
  LIJST
)
```

Arguments

OpnameID	nummer van de opname
VoorwaardeID	ID-nummer (uit LSVI-indicatorendatabank) van de voorwaarde die moet berekend worden
Kenmerken	Dataframe met soorten of kenmerken en hun bedekking voor 1 opname met minimum de velden ID, Vegetatielaag, Kenmerk, TypeKenmerk, Waarde, Type, Invoertype en Eenheid (data identiek aan Data_soortenKenmerken in berekenLSVIbasis())
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie connecteerMetLSVIDb().
LIJST	Dataframe met lijst die weergeeft hoe de vertaling moet gebeuren van categorische variabelen naar numerieke waarden (en omgekeerd). Default worden deze waarden uit de databank met LSVI-indicatoren gehaald d.m.v. de functie vertaalInvoerInterval(). Aangeraden wordt om deze default te gebruiken (dus parameter niet expliciet invullen), of deze waar nodig aan te vullen met eigen schalen. Omdat er ook een omzetting moet gebeuren voor grenswaarden uit de databank, kan het niet doorgeven van een gedeelte van deze lijst problemen geven.

Value

Een vector van 3 waarden die het resultaat is van de berekening, namelijk het minimum en het maximum van het interval waartussen de berekende waarde ligt en het theoretisch maximum. Als het resultaat een exacte waarde is en geen interval, bevat het minimum en maximum tweemaal dezelfde waarde. (Het resultaat is een interval als de brondata categorische variabelen zijn, bv. bedekkingen volgens de beheermonitoringschaal of Tansley-schaal.)

combinerenVerschilscore

combineert de Verschilscores van voorwaarden die via EN of OF logische operatoren gelinkt zijn

Description

Technische hulpfunctie die in een formule de VoorwaardeID's vervangt door opgegeven waarden en het resultaat van de formule teruggeeft.

Usage

```
combinerenVerschilscore(Formule, VoorwaardeID, Verschilscore)
```

Arguments

Formule	string van VoorwaardeID's gecombineerd met EN en OF, bijvoorbeeld "(720 EN 721) OF 15"
VoorwaardeID	vector van alle VoorwaardeID's die voorkomen in de Formule
Verschilscore	vector met voor elke VoorwaardeID een overeenkomstige verschilscore

Value

gecombineerde verschilscore waarbij EN gecombineerd wordt via het minimum van beide verschilscores en OF gecombineerd wordt via het maximum van beide verschilscores

Examples

```
#onderstaand voorbeeld geeft problemen bij het testen van het package door
#devtools, maar buiten deze context werkt het wel
## Not run:
combinerenVerschilscore(
  "(720 AND 721) OR 15",
  c(720, 721, 15),
  c(0.5, -0.3, 0.8)
)

## End(Not run)
```

`combinerenVoorwaarden` *combineert de Status van voorwaarden via de opgegeven formule*

Description

Technische hulpfunctie die in een formule de VoorwaardeID's vervangt door opgegeven logische waarden en het resultaat van de formule teruggeeft.

Usage

```
combinerenVoorwaarden(Formule, VoorwaardeID, Status)
```

Arguments

Formule	string van VoorwaardeID's gecombineerd met EN en OF, bijvoorbeeld "(720 EN 721) OF 15"
VoorwaardeID	vector van alle VoorwaardeID's die voorkomen in de Formule
Status	vector met voor elke VoorwaardeID een overeenkomstige logische waarde status (TRUE of FALSE)

Value

logische waarde (TRUE of FALSE) die de uitkomst van de Formule is (gecombineerd met VoorwaardeID en Status)

Examples

```
#onderstaand voorbeeld geeft problemen bij het testen van het package door
#devtools, maar buiten deze context werkt het wel
## Not run:
combinerenVoorwaarden(
  "(720 AND 721) OR 15",
  c(720, 721, 15),
  c(TRUE, FALSE, TRUE)
)

## End(Not run)
```

connecteerMetLSVIdb	<i>Connecteer met de databank met LSVI-indicatoren in het package</i>
---------------------	---

Description

Deze functie maakt een connectie met de in het package toegevoegde databank met LSVI-indicatoren, wat nodig is om de functies te kunnen gebruiken. Deze connectie moet als argument meegegeven worden bij elke functie die informatie uit de databank ophaalt. Alternatief is om eenmalig een connectiepool aan te maken met de functie `maakConnectiePool()`.

Usage

```
connecteerMetLSVIdb()
```

Value

Deze functie geeft een open odbc-connectie naar de SQLite-databank in de installatie-file van het package.

Examples

```
library(LSVI)
ConnectieLSVIhabitats <- connecteerMetLSVIdb()
geefVersieInfo(ConnectieLSVIhabitats)
library(DBI)
dbGetQuery(ConnectieLSVIhabitats,
  "SELECT VersieLSVI, Referentie FROM Versie")
dbDisconnect(ConnectieLSVIhabitats)
```

`controleerInvoerwaarde`*Hulpfunctie voor het uitvoeren van foutcontroles*

Description

Deze technische hulpfunctie bevat een standaardroutine om te controleren of de door een gebruiker ingevoerde waarde(n) voorkomen in een gespecificeerde tabel in de databank. Indien niet, dan geeft de functie een informatieve error (welke waarde is fout, en welk zijn de mogelijke invoerwaarden voor de variabele?).

Usage

```
controleerInvoerwaarde(  
  Beschrijving,  
  Invoerwaarden,  
  Tabelnaam,  
  Veldnaam,  
  ConnectieLSVIhabitats,  
  Tolower = TRUE  
)
```

Arguments

Beschrijving	Hoe de invoerwaarde beschreven moet worden in de error
Invoerwaarden	De waarden die de gebruiker ingevoerd heeft
Tabelnaam	De naam van de tabel waarin het veld zich bevindt (String)
Veldnaam	De naam van het veld (in de bij Tabelnaam opgegeven tabel) waarvan de waarden moeten opgezocht worden (String)
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie connecteerMetLSVIdb().
Tolower	default (als TRUE) wordt tolower() uitgevoerd op de invoerwaarden en gegevens uit de databank vooraleer de vergelijking uitgevoerd wordt, FALSE zorgt dat dit niet uitgevoerd wordt, maar idealiter worden deze stap voor stap vervangen door Tolower = TRUE

Value

Deze functie geeft geen waarde terug, maar gooit een error als er een foute waarde ingevoerd is

deselecteerKenmerkenInOpname

Controle van de ingevoerde opname

Description

Deze hulpfunctie voor de s4-lassen aantal en bedekking selecteert soorten of kenmerken uit een opname die niet tot de soortgroep of studiegroep van een bepaalde voorwaarde behoren. Op basis hiervan kunnen de s4-lassen bedekkingExcl en maxBedekkingExcl berekend worden (bv. dominantie van een soort: maximale bedekking van soorten in een opname exclusief de sleutelsoorten).

Usage

```
deselecteerKenmerkenInOpname(
  Kenmerken,
  Soortengroep,
  Studiegroep,
  SubAnalyseVariabele,
  SubRefMin,
  SubRefMax,
  SubOperator
)
```

Arguments

Kenmerken	dataframe met alle opgegeven kenmerken, met velden Kenmerk, TypeKenmerk, WaardeMin en WaardeMax
Soortengroep	dataframe met de soortenlijst die uit Kenmerken gedeselecteerd moet worden
Studiegroep	dataframe met de lijst kenmerken die uit Kenmerken gedeselecteerd moet worden. Als ook Soortengroep opgegeven is, geeft Studiegroep aan welke kenmerken wel behouden moeten blijven na deselecteren van de soortengroep.
SubAnalyseVariabele	heeft waarde "bedekking" als er een subvoorwaarde is voor de bedekking van de geselecteerde soorten of kenmerken
SubRefMin	minimumwaarde van de grenswaarde voor de bedekking
SubRefMax	maximumwaarde van de grenswaarde voor de bedekking
SubOperator	operator voor deze subvoorwaarde: moet de bedekking hoger of lager liggen dan de opgegeven referentiewaarde?

Value

Deze functie geeft een aangepaste tabel Data_soorten terug waarin enkel de soorten uit de soortenlijst(en) opgenomen zijn en die bovendien gekoppeld is aan de gegevens van de soortenlijst.

geefInfoHabitatfiche	<i>Geeft tabel met info uit de LSVI-rapporten voor de opgegeven parameters</i>
----------------------	--

Description

Deze functie geeft de inhoud van de tabellen habitatkarakteristieken en beoordelingsmatrix uit de rapporten van de Lokale Staat van Instandhouding voor de habitattypes die voldoen aan de opgegeven parameters. Volledigheidshalve geeft ze ook de uitgebreide namen van de habitattypes en habitatsubtypes. De uitvoer van deze functie kan gebruikt worden om rapportages op te maken (bv. rapport samenstellen met LSVI-criteria,...). Een "afgewerkt rapport" kan gegenereerd worden met de functie `maakHabitatfiches()`.

De parameters kunnen enkel de hieronder gespecificeerde waarden bevatten en moeten als string opgegeven worden. Default is telkens "alle", waarbij de soortenlijsten voor alle mogelijke waarden van die parameter weergegeven worden (m.a.w. er is geen selectie voor deze parameter).

Usage

```
geefInfoHabitatfiche(
  Versie = "alle",
  Habitatgroep = "alle",
  Habitatype = "alle",
  Criterium = "alle",
  Indicator = "alle",
  Stijl = c("Rmd", "tekst"),
  ConnectieLSVIhabitats = NULL
)
```

Arguments

Versie	De versie van het LSVI-rapport, bv. "Versie 2" of "Versie 3". Bij de default "alle" worden de gegevens voor de verschillende versies gegeven. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Versie", "VersieLSVI")</code> of <code>geefVersieInfo()</code> .
Habitatgroep	Parameter waarmee alle habitats van een bepaalde habitatgroep kunnen geselecteerd worden, bv. "Bossen", "Heiden", "(Half-)natuurlijke graslanden", "Zoete en brakke wateren",... en "alle" (=default). Deze waarde moet niet gespecificeerd worden als een bepaald habitat(sub)type geselecteerd wordt. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitatgroep", "Naam")</code> .
Habitatype	Parameter waarmee een habitatype of habitatsubtype kan geselecteerd worden. Als dit een habitatype betreft met meerdere subtypes, zullen de gegevens van alle subtypes van dit habitatype weergegeven worden. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitatype", "Code")</code> . Er is voor deze parameter ook de mogelijkheid om een vector van meerdere habitat(sub)typen op te geven.

Criterium	Het LSVI-criterium waarvoor de gegevens geselecteerd worden: "Vegetatie", "Structuur", "Verstoring" of "alle".
Indicator	De indicator waarvoor de gegevens uit de databank gehaald worden. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Indicator", "Naam")</code> .
Stijl	Keuze uit "Rmd" en "tekst". Bij "Rmd" (default) worden soortgroepnamen voorafgegaan en gevolgd door "_" en Latijnse namen van soorten door "_", waardoor deze bij gebruik van RMarkdown worden omgezet naar resp. vet en <i>italics</i> . Bij "tekst" worden deze underscores weggelaten.
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie <code>connecteerMetLSVIDb()</code> .

Value

Deze functie genereert een tabel met alle gegevens die nodig zijn om de tabellen habitatkarakteristieken en beoordelingsmatrix uit de LSVI-rapporten te genereren.

Examples

```
# Omwille van de iets langere lange duurtijd van de commando's staat bij
# onderstaand voorbeeld de vermelding 'dontrun' (om problemen te vermijden
# bij het testen van het package). Maar het voorbeeld werkt en kan zeker
# uitgetest worden.
## Not run:
library(LSVI)
maakConnectiePool()
geefInfoHabitatfiche(Versie = "Versie 2.0", Habitatype = "4030")
library(pool)
poolClose(ConnectiePool)

## End(Not run)
```

`geefInvoervereisten` *Geeft de invoervereisten voor de waarde van een opname*

Description

Deze functie geeft alle informatie die nodig is om veldobservaties klaar te maken voor de berekening van de de Lokale Staat van Instandhouding met de functie `berekenLSVIbasis()`, alsook de berekeningsregels die gebruikt worden. Allereerst geeft ze de Voorwaarde die vermeld moet worden bij de observaties (zie `Data_voorwaarden` bij `berekenLSVIbasis()`), samen met informatie uit de LSVI-tabellen (vnl. beoordelingsmatrix) en een beschrijving van de voorwaarde (Voorwaarde) die zou moeten toelaten om de koppeling te maken.

Verder geeft ze informatie over de Waarde die verwacht wordt in de functie `berekenLSVIbasis()`. `AnalyseVariabele` is een korte omschrijving voor de variabele waarde, bv. "aantal" staat voor

het aantal soorten of klassen en "bedekking" voor de totale bedekking van de lijst soorten of klassen. (Zie vignette("berekeningen", package = "LSVI") voor een uitgebreide beschrijving van de achterliggende berekeningen.) Referentiewaarde en Operator geven respectievelijk de grenswaarde en de vergelijking aan op basis waarvan de beoordeling van de waarde zal gebeuren. Voor elke AnalyseVariabele wordt informatie gegeven over het formaat dat verwacht wordt voor Waarde: de Eenheid (die niet opgenomen moet worden in Waarde maar wel de grootte-orde van het verwachte getal aangeeft), het formaat van de variabele (TypeVariabele), en bij categorische variabelen het Invoertype en de Invoerwaarde (een naam voor de categorische variabele en de mogelijke waarden die deze kan aannemen).

Waar nodig, wordt een soortengroep of studiegroep opgegeven. Een studiegroep is eigenlijk equivalent aan een soortengroep: de verschillende klassen of fasen of ... waarvoor een bedekking of andere analysevariabele moet berekend worden. Voorbeelden zijn groeiklassen, vegetatielagen, ... Omwille van de overzichtelijkheid van de tabel is voor de Soortengroep enkel een ID gegeven, de volledige lijst kan opgevraagd worden met de functie geefSoortenlijstInvoerniveau().

Ingeval van de AnalyseVariabele aantal kan er ook een SubAnalyseVariabele vermeld zijn, meestal bedekking, die aangeeft aan welke voorwaarde elke soort of klasse afzonderlijk moet voldoen. Aan deze SubAnalysevariabele zijn dezelfde velden gekoppeld als aan AnalyseVariabele, nl. SubReferentiewaarde, SubOperator, SubEenheid, TypeSubVariabele, SubInvoertype en SubInvoerwaarde. Bijvoorbeeld, bij de voorwaarde "minimum 5 soorten minimum talrijk aanwezig" zal de AnalyseVariabele aantal zijn, de Referentiewaarde "5", de Operator ">=", TypeVariabele "Geheel getal", SubAnalysevariabele "bedekking", SubReferentiewaarde "T", SubOperator ">=", TypeSubVariabele "Categorie" en SubInvoertype "Beheermonitoringsschaal 2017".

Usage

```
geefInvoervereisten(
  Versie = "alle",
  Habitatgroep = "alle",
  Habitattype = "alle",
  Criterium = "alle",
  Indicator = "alle",
  Kwaliteitsniveau = "alle",
  Weergave = c("basis", "uitgebreid"),
  ConnectieLSVIhabitats = NULL
)
```

Arguments

Versie	De versie van het LSVI-rapport, bv. "Versie 2" of "Versie 3". Bij de default "alle" worden de gegevens voor de verschillende versies gegeven. De mogelijke waarden kunnen opgevraagd worden via geefUniekeWaarden("Versie", "VersieLSVI") of geefVersieInfo().
Habitatgroep	Parameter waarmee alle habitats van een bepaalde habitatgroep kunnen geselecteerd worden, bv. "Bossen", "Heiden", "(Half-)natuurlijke graslanden", "Zoete en brakke wateren",... en "alle" (=default). Deze waarde moet niet gespecificeerd worden als een bepaald habitat(sub)type geselecteerd wordt. De mogeli-

	jke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitatgroep", "Naam")</code> .
Habitattype	Parameter waarmee een habitattype of habitatsubtype kan geselecteerd worden. Als dit een habitattype betreft met meerdere subtypes, zullen de gegevens van alle subtypes van dit habitattype weergegeven worden. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitattype", "Code")</code> . Er is voor deze parameter ook de mogelijkheid om een vector van meerdere habitat(sub)typen op te geven.
Criterium	Het LSVI-criterium waarvoor de gegevens geselecteerd worden: "Vegetatie", "Structuur", "Verstoring" of "alle".
Indicator	De indicator waarvoor de gegevens uit de databank gehaald worden. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Indicator", "Naam")</code> .
Kwaliteitsniveau	Voor elke versie van de LSVI zijn er een of meerdere kwaliteitsniveaus gedefinieerd in de databank. Zo is er bij Versie 2.0 een onderscheid gemaakt tussen goede staat (A), voldoende staat (B) en gedegradeerde staat (C). Hier duidt kwaliteitsniveau 1 de grens tussen voldoende (B) en gedegradeerd (C) aan en kwaliteitsniveau 2 het onderscheid tussen goed (A) en voldoende (B). Bij Versie 3 duidt kwaliteitsniveau 1 op het onderscheid tussen ongunstig en gunstig en kwaliteitsniveau 2 op de streefwaarde (uiteindelijk niet opgenomen in rapport). De betekenissen van de 2 kwaliteitsniveaus voor de verschillende versies is weergegeven in de tabel Versie in de databank en kan opgevraagd met de functie <code>geefVersieInfo()</code> . Geef als parameter Kwaliteitsniveau op op basis van welk kwaliteitsniveau de berekening gemaakt moet worden. (Strikt genomen is de berekening van de LSVI de berekening volgens kwaliteitsniveau 1.)
Weergave	Wat moet er in de tabel weergegeven worden? De default "basis" geeft een meer overzichtelijke tabel waarbij mogelijke invoerwaarden gescheiden door een komma in 1 cel weergegeven worden, "uitgebreid" geeft deze invoerwaarden met alle bijhorende informatie weer in aparte records, waardoor de tabel groot en onoverzichtelijk is.
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie <code>connecteerMetLSVIDb()</code> .

Value

Deze functie geeft een tabel met de hierboven beschreven informatie uit de databank.

Examples

```
# Omwille van de iets langere lange duurtijd van de commando's staat bij
# onderstaande voorbeelden de vermelding 'dontrun' (om problemen te vermijden
# bij het testen van het package). Maar de voorbeelden werken en kunnen zeker
# uitgetest worden.
## Not run:
library(LSVI)
maakConnectiePool()
```

```

geefInvoervereisten(
  Versie = "Versie 2.0",
  Habitatype = "4030",
  Kwaliteitsniveau = "1"
)
library(pool)
poolClose(ConnectiePool)

## End(Not run)

```

geefSoortenlijst	<i>Genereert soortenlijst(en) LSVI op basis van de opgegeven parameters</i>
------------------	---

Description

Deze functie genereert soortenlijsten (met wetenschappelijke en Nederlandse namen) die gebruikt worden voor de bepaling van de Lokale Staat van Instandhouding van de opgegeven parameters. In feite genereert ze een tabel met velden Versie, Habitatype, Habitatsubtype, WetNaam, WetNaamKort en NedNaam en evt. Criterium, Indicator en/of Beschrijving waarin de gespecificeerde parameters uitgeselecteerd zijn en waar voor andere parameters alle waarden uit de databank weergegeven zijn.

Voor de vorm van de soortenlijst zijn er meerdere opties: een soortenlijst met alle soorten per habitat(sub)type, ofwel gegroepeerd per criterium, indicator of voorwaarde. Dit kan opgegeven worden in de parameter Taxonlijstniveau.

Ook voor de weergave van de taxa zijn 2 opties: de taxa weergegeven zoals in de habitatfiches (op soortniveau, genusniveau of hoger niveau, zoals het in de habitatfiches vermeld is) of alle taxa op lagere niveaus ook weergegeven en dus bij soortengroepen alle mogelijke soorten van deze groep weergegeven. Deze opties kunnen opgegeven worden in de parameter Taxonlijsttype.

De parameters kunnen enkel de hieronder gespecificeerde waarden bevatten en moeten als string opgegeven worden. Default is telkens "alle", waarbij de soortenlijsten voor alle mogelijke waarden van die parameter weergegeven worden (m.a.w. er is geen selectie voor deze parameter).

Usage

```

geefSoortenlijst(
  Versie = "alle",
  Habitatgroep = "alle",
  Habitatype = "alle",
  Criterium = "alle",
  Indicator = "alle",
  Taxonlijstniveau = c("habitatype", "criterium", "indicator", "voorwaarde"),
  Taxonlijsttype = c("LSVIfiche", "alle"),
  ConnectieLSVIhabitats = NULL
)

```

Arguments

Versie	De versie van het LSVI-rapport, bv. "Versie 2" of "Versie 3". Bij de default "alle" worden de gegevens voor de verschillende versies gegeven. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Versie", "VersieLSVI")</code> of <code>geefVersieInfo()</code> .
Habitatgroep	Parameter waarmee alle habitats van een bepaalde habitatgroep kunnen geselecteerd worden, bv. "Bossen", "Heiden", "(Half-)natuurlijke graslanden", "Zoete en brakke wateren",... en "alle" (=default). Deze waarde moet niet gespecificeerd worden als een bepaald habitat(sub)type geselecteerd wordt. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitatgroep", "Naam")</code> .
Habitatype	Parameter waarmee een habitatype of habitatsubtype kan geselecteerd worden. Als dit een habitatype betreft met meerdere subtypes, zullen de gegevens van alle subtypes van dit habitatype weergegeven worden. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitatype", "Code")</code> . Er is voor deze parameter ook de mogelijkheid om een vector van meerdere habitat(sub)typen op te geven.
Criterium	Het LSVI-criterium waarvoor de gegevens geselecteerd worden: "Vegetatie", "Structuur", "Verstoring" of "alle".
Indicator	De indicator waarvoor de gegevens uit de databank gehaald worden. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Indicator", "Naam")</code> .
Taxonlijstniveau	Geeft aan op welk niveau de soortenlijst gegroepeerd is (en welke niveaus weergegeven worden in de soortenlijst), de mogelijke waarden zijn "habitatype", "criterium", "indicator" en "voorwaarde". Default is "habitatype".
Taxonlijsttype	"LSVIfiche" betekent dat de taxonlijst van de habitatfiche wordt overgenomen, "alle" betekent dat alle soorten en alle taxonomische groepen worden weergegeven die volledig in de groepen vallen die aan de parameters voldoen.
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie <code>connecteerMetLSVIDb()</code> .

Value

Deze functie geeft een tabel met velden Versie, Habitatype, Habitatsubtype, Criterium, Indicator, evt. Beschrijving, WetNaam, WetNaamKort en NedNaam (waarbij Beschrijving een omschrijving is voor een groep van taxa binnen eenzelfde indicator). WetNaam is de volledige Latijnse naam inclusief auteursnaam, WetNaamKort geeft de verkorte naam zonder auteursnaam.

Examples

```
# Omwille van de iets langere lange duurtijd van de commando's staat bij
# onderstaande voorbeelden de vermelding 'dontrun' (om problemen te vermijden
# bij het testen van het package). Maar de voorbeelden werken en kunnen zeker
# uitgetest worden.
## Not run:
```

```

maakConnectiePool()
geefSoortenlijst(Habitattype = "4030", Taxonlijsttype = "LSVIfiche")
geefSoortenlijst(Habitattype = "4030", Taxonlijsttype = "alle")
library(pool)
poolClose(ConnectiePool)

## End(Not run)

```

geefSoortenlijstVoorIDs

Genereert soorten(groep)lijst(en) LSVI op basis van TaxongroepID

Description

Deze functie genereert soortenlijsten (met wetenschappelijke en Nederlandse namen) uit de databank met de criteria en indicatoren voor de bepaling van de Lokale Staat van Instandhouding. Het is in feite een hulpfunctie die voor verschillende andere functies gebruikt wordt en die de complexe zoekfunctie in de tabellen met soorten uitvoert op basis van een opgegeven TaxongroepId (en in die zin iets minder gebruiksvriendelijk is). Voor een selectie van soortenlijsten op basis van specifieke parameters is de functie `geefSoortenlijst()` een beter alternatief.

Deze functie geeft standaard voor de gespecificeerde taxongroepen per groep een lijst van alle taxa zoals ze in de LSVI-habitatfiche vermeld zijn (genusniveau, soortniveau, ondersoort,...). Op basis van de parameter `soortenlijsttype` kan ook gekozen worden om een volledige lijst te geven van deze taxa en alle taxa die hieronder vallen (en opgenomen zijn in de onderliggende databank).

Usage

```

geefSoortenlijstVoorIDs(
  Taxongroeplijst,
  Taxonlijsttype = c("LSVIfiche", "alle"),
  ConnectieLSVIhabitats = NULL
)

```

Arguments

`Taxongroeplijst`

string waarin de TaxongroepId's na elkaar weergegeven worden, gescheiden door een komma. Eventueel mag dit ook een vector zijn van TaxongroepId's.

`Taxonlijsttype` "LSVIfiche" betekent dat de taxonlijst van de habitatfiche wordt overgenomen, "alle" betekent dat alle soorten en alle taxonomische groepen worden weergegeven die volledig in de groepen vallen die aan de parameters voldoen.

`ConnectieLSVIhabitats`

Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie `connecteerMetLSVIdb()`.

Value

Deze functie geeft een tabel met velden TaxongroepId, evt. Beschrijving, WetNaam, WetNaamKort en NedNaam (waarbij Beschrijving een omschrijving is voor een groep van taxons binnen eenzelfde indicator). WetNaam is de volledige Latijnse naam inclusief auteursnaam, WetNaamKort geeft de verkorte naam zonder auteursnaam.

Examples

```
# Omwille van de iets langere lange duurtijd van de commando's staat bij
# onderstaande voorbeelden de vermelding 'dontrun' (om problemen te vermijden
# bij het testen van het package). Maar de voorbeelden werken en kunnen zeker
# uitgetest worden.
## Not run:
maakConnectiePool()
geefSoortenlijstVoorIDs("434,88,565")
geefSoortenlijstVoorIDs("434,88,565","alle")
library(pool)
poolClose(ConnectiePool)

## End(Not run)
```

geefUniekeWaarden	<i>Geef alle unieke waarden van een veld uit de databank met de LSVI-indicatoren voorafgegaan door "alle"</i>
-------------------	---

Description

Deze hulpfunctie geeft een vector met alle verschillende waarden die in een gespecificeerd veld van een gespecificeerde tabel in de databank met LSVI-indicatoren staan, voorafgegaan door de (toegevoegde) waarde "alle". Deze functie wordt in verschillende functies van het package gebruikt om de invoer van parameters te controleren (waar de mogelijke invoer bestaat uit "alle" of een item uit het veld). In enkele gevallen wordt dit commando in de documentatie vermeld zodat ook de gebruiker in die specifieke gevallen de lijst met mogelijke invoerwaarden op een eenvoudige manier kan opvragen.

Gebruikers die de databankstructuur en bijhorende naamgeving niet kennen, kunnen dezelfde info het gemakkelijkst bekomen door een tabel voor de volledige dataset op te vragen, het gewenste veld te selecteren en hiervan de unieke waarden weer te geven (zie voorbeeld).

Usage

```
geefUniekeWaarden(Tabelnaam, Veldnaam, ConnectieLSVIhabitats = NULL)
```

Arguments

Tabelnaam	De naam van de tabel waarin het veld zich bevindt (String)
Veldnaam	De naam van het veld (in de bij Tabelnaam opgegeven tabel) waarvan de waarden moeten opgezocht worden (String)

ConnectieLSVIhabitats

Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie connecteerMetLSVIDb().

Value

Deze functie geeft een vector bestaande uit "alle" en de verschillende waarden uit de gespecificeerde tabel.

Examples

```
# Omwille van de iets langere lange duurtijd van de commando's staat bij
# onderstaand voorbeeld de vermelding 'dontrun' (om problemen te vermijden
# bij het testen van het package). Maar het voorbeelden werkt en kan zeker
# uitgetest worden.
## Not run:
maakConnectiePool()
geefUniekeWaarden("Versie", "VersieLSVI")

#alternatieven om deze invoerlijst te bekomen:
unique(geefVersieInfo()$VersieLSVI)

library(dplyr)
geefVersieInfo() %>%
  select(VersieLSVI) %>%
  distinct()

library(pool)
poolClose(ConnectiePool)

## End(Not run)
```

geefVersieInfo	<i>Geeft informatie over de verschillende versies voor de berekening van de LSVI</i>
----------------	--

Description

Deze functie geeft een overzicht van alle versies die er zijn voor de berekening van de de Lokale Staat van Instandhouding, met naast de opsomming van de versies en de referenties een overzicht van de 2 kwaliteitsniveaus of types van beoordelingscriteria die in deze versie gedefinieerd zijn.

Usage

```
geefVersieInfo(ConnectieLSVIhabitats = NULL)
```

Arguments

ConnectieLSVIhabitats

Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie connecteerMetLSVIDb().

Value

Deze functie geeft de tabel Versie uit de databank.

Examples

```
maakConnectiePool()  
geefVersieInfo()  
library(pool)  
poolClose(ConnectiePool)
```

geefVertaallijst

Haalt vertaallijst op uit databank

Description

Deze functie haalt een lijst uit de databank van de gebruikte schalen (bv. Tansley-schaal en beheer-monitoringschaal uit 2017) met telkens de verschillende mogelijke categorieën en een overeenkomstige bedekkingswaarde. De functie berekenLSVIbasis() gebruikt deze waarden om de nodige omzettingen te doen tussen deze verschillende schalen

Usage

```
geefVertaallijst(ConnectieLSVIhabitats)
```

Arguments

ConnectieLSVIhabitats

Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie connecteerMetLSVIDb().

Value

Dataframe met Naam, Waarde, Volgnummer, Omschrijving, Ondergrens, Gemiddelde en Boven-grens. Telkens is een waarde tussen 0 en 1 opgegeven die afkomstig is van het delen van het percentage door 100)

`invoercontroleData_habitat`*Invoercontrole voor dataframe Data_habitat*

Description

Om te vermijden dat we meermaals dezelfde invoercontrole moeten uitvoeren en om de hoofdscrip-
ts overzichtelijk te houden, maken we voor elke invoercontrole een aparte hulpfunctie aan, die we
kunnen aanroepen. Deze wordt NIET geëxporteerd, dus deze functies kunnen niet als commando
gerund worden (maar worden wel gerund als de functie waarin ze voorkomen, aangeroepen wordt).

Usage

```
invoercontroleData_habitat(Data_habitat, ConnectieLSVIhabitats)
```

Arguments

`Data_habitat` dataframe waarop invoercontrole moet gebeuren.

`ConnectieLSVIhabitats`

Connectie met de databank met indicatoren voor de LSVI van habitats, in te
stellen d.m.v. functie `connecteerMetLSVIdb()`.

`invoercontroleData_soortenKenmerken`*Invoercontrole voor dataframe Data_soortenKenmerken*

Description

Om te vermijden dat we meermaals dezelfde invoercontrole moeten uitvoeren en om de hoofdscrip-
ts overzichtelijk te houden, maken we voor elke invoercontrole een aparte hulpfunctie aan, die we
kunnen aanroepen. Deze wordt NIET geëxporteerd, dus deze functies kunnen niet als commando
gerund worden (maar worden wel gerund als de functie waarin ze voorkomen, aangeroepen wordt).
Ingeval van `Data_soortenKenmerken` is ook de omzetting van soortnamen naar een `NbnTaxonVersionKey`
en de omzettingen van bedekkingen naar een interval opgenomen in de functie.

Usage

```
invoercontroleData_soortenKenmerken(  
  Data_soortenKenmerken,  
  ConnectieLSVIhabitats,  
  LIJST  
)
```

Arguments

Data_soortenKenmerken	dataframe waarop invoercontrole moet gebeuren.
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie connecteerMetLSVIdb().
LIJST	Dataframe met lijst die weergeeft hoe de vertaling moet gebeuren van categorische variabelen naar numerieke waarden (en omgekeerd). Default worden deze waarden uit de databank met LSVI-indicatoren gehaald d.m.v. de functie vertaalInvoerInterval(). Aangeraden wordt om deze default te gebruiken (dus parameter niet expliciet invullen), of deze waar nodig aan te vullen met eigen schalen. Omdat er ook een omzetting moet gebeuren voor grenswaarden uit de databank, kan het niet doorgeven van een gedeelte van deze lijst problemen geven.

invoercontroleData_voorwaarden

Invoercontrole voor dataframe Data_voorwaarden

Description

Om te vermijden dat we meermaals dezelfde invoercontrole moeten uitvoeren en om de hoofdscripts overzichtelijk te houden, maken we voor elke invoercontrole een aparte hulpfunctie aan, die we kunnen aanroepen. Deze wordt NIET geëxporteerd, dus deze functies kunnen niet als commando gerund worden (maar worden wel gerund als de functie waarin ze voorkomen, aangeroepen wordt). Ingeval van Data_voorwaarden is ook de omzetting van de voorwaarden naar een interval opgenomen in de functie.

Usage

```
invoercontroleData_voorwaarden(Data_voorwaarden, ConnectieLSVIhabitats, LIJST)
```

Arguments

Data_voorwaarden	dataframe waarop invoercontrole moet gebeuren.
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie connecteerMetLSVIdb().
LIJST	Dataframe met lijst die weergeeft hoe de vertaling moet gebeuren van categorische variabelen naar numerieke waarden (en omgekeerd). Default worden deze waarden uit de databank met LSVI-indicatoren gehaald d.m.v. de functie vertaalInvoerInterval(). Aangeraden wordt om deze default te gebruiken (dus parameter niet expliciet invullen), of deze waar nodig aan te vullen met eigen schalen. Omdat er ook een omzetting moet gebeuren voor grenswaarden uit de databank, kan het niet doorgeven van een gedeelte van deze lijst problemen geven.

`invoercontroleKwaliteitsniveau`*Invoercontrole voor waarde Kwaliteitsniveau*

Description

Om te vermijden dat we meermaals dezelfde invoercontrole moeten uitvoeren en om de hoofdscrip-
ts overzichtelijk te houden, maken we voor elke invoercontrole een aparte hulpfunctie aan, die we
kunnen aanroepen. Deze wordt NIET geëxporteerd, dus deze functies kunnen niet als commando
gerund worden (maar worden wel gerund als de functie waarin ze voorkomen, aangeroepen wordt).

Usage

```
invoercontroleKwaliteitsniveau(Kwaliteitsniveau, ConnectieLSVIhabitats)
```

Arguments

Kwaliteitsniveau

Waarde waarop invoercontrole moet gebeuren.

ConnectieLSVIhabitats

Connectie met de databank met indicatoren voor de LSVI van habitats, in te
stellen d.m.v. functie connecteerMetLSVIdb().

`invoercontroleVersie` *Invoercontrole voor waarde versie*

Description

Om te vermijden dat we meermaals dezelfde invoercontrole moeten uitvoeren en om de hoofdscrip-
ts overzichtelijk te houden, maken we voor elke invoercontrole een aparte hulpfunctie aan, die we
kunnen aanroepen. Deze wordt NIET geëxporteerd, dus deze functies kunnen niet als commando
gerund worden (maar worden wel gerund als de functie waarin ze voorkomen, aangeroepen wordt).

Usage

```
invoercontroleVersie(Versie, ConnectieLSVIhabitats)
```

Arguments

Versie

Waarde waarop invoercontrole moet gebeuren.

ConnectieLSVIhabitats

Connectie met de databank met indicatoren voor de LSVI van habitats, in te
stellen d.m.v. functie connecteerMetLSVIdb().

logDatabankfouten	<i>Lijst alle nog op te lossen databankfouten op</i>
-------------------	--

Description

Deze functie maakt een tabel met alle problemen die nog in de databank zitten. Enerzijds is er een beperkte tabel met problemen die op een hoger niveau opgelost kunnen worden en anderzijds een detail met alle Voorwaarden waar nog een fout in zit. Problemen die op beide niveaus kunnen opgelost worden (bv. benoemen van analysevariabelen), staan op beide niveaus vermeld.

Usage

```
logDatabankfouten(ConnectieLSVIhabitats = NULL)
```

Arguments

ConnectieLSVIhabitats

Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie connecteerMetLSVIdb().

Value

Deze functie geeft een list met 2 dataframes terug

maakConnectiePool	<i>Maak een connectiepool met de databank met LSVI-indicatoren in het package</i>
-------------------	---

Description

Deze functie maakt een connectiepool met de in het package toegevoegde databank met LSVI-indicatoren, wat nodig is om de functies te kunnen gebruiken. Deze connectiepool moet eenmalig aangemaakt worden, en functies zullen automatisch deze connectiepool gebruiken om te connecteren met de databank. Alternatief is om een connectie aan te maken met de functie connecteerMetLSVIdb() en deze bij elke functie mee te geven.

Usage

```
maakConnectiePool()
```

Value

Deze functie maakt een Environment-object aan dat de connecties regelt met de betreffende databank.

Examples

```
library(LSVI)
maakConnectiePool()
geefVersieInfo()
library(pool)
poolClose(ConnectiePool)
```

maakHabitatfiches	<i>Genereert habitatfiche(s) van LSVI op basis van de opgegeven parameters</i>
-------------------	--

Description

Deze functie genereert habitatfiches die gebruikt worden voor de bepaling van de Lokale Staat van Instandhouding van de habitattypes die voldoen aan de opgegeven parameters. (Om een tabel te genereren met deze informatie om zelf een fiche te kunnen samenstellen, wordt verwezen naar de functie `geefInfoHabitatfiche()`. Om een rapport samen te stellen met alle fiches na elkaar in 1 document, wordt verwezen naar de functie `maakLSVIRapport()`)

De parameters kunnen enkel de hieronder gespecificeerde waarden bevatten en moeten als string opgegeven worden. Default is telkens "alle", waarbij de soortenlijsten voor alle mogelijke waarden van die parameter weergegeven worden (m.a.w. er is geen selectie voor deze parameter).

Usage

```
maakHabitatfiches(
  Versie = "alle",
  Habitatgroep = "alle",
  Habitattype = "alle",
  ConnectieLSVIhabitats = NULL,
  verbose = TRUE
)
```

Arguments

Versie	De versie van het LSVI-rapport, bv. "Versie 2" of "Versie 3". Bij de default "alle" worden de gegevens voor de verschillende versies gegeven. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Versie", "VersieLSVI")</code> of <code>geefVersieInfo()</code> .
Habitatgroep	Parameter waarmee alle habitats van een bepaalde habitatgroep kunnen geselecteerd worden, bv. "Bossen", "Heiden", "(Half-)natuurlijke graslanden", "Zoete en brakke wateren",... en "alle" (=default). Deze waarde moet niet gespecificeerd worden als een bepaald habitat(sub)type geselecteerd wordt. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitatgroep", "Naam")</code> .

Habitattype	Parameter waarmee een habitattype of habitatsubtype kan geselecteerd worden. Als dit een habitattype betreft met meerdere subtypes, zullen de gegevens van alle subtypes van dit habitattype weergegeven worden. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitattype", "Code")</code> . Er is voor deze parameter ook de mogelijkheid om een vector van meerdere habitat(sub)typen op te geven.
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie <code>connecteerMetLSVIdb()</code> .
verbose	geeft de toestand van het systeem aan, om te zorgen dat boodschappen niet onnodig gegeven worden

Value

Deze functie genereert een rapport met habitatfiches in de vorm van een html-bestand dat in de working directory opgeslagen wordt.

Examples

```
# Omwille van de iets langere lange duurtijd van het commando staat bij
# onderstaand voorbeeld de vermelding 'dontrun' (om problemen te vermijden
# bij het testen van het package). Maar het voorbeeld werkt en mag zeker
# uitgetest worden.
## Not run:
maakConnectiePool()
maakHabitatfiches(Versie = "Versie 2.0", Habitattype = "4030")
library(pool)
poolClose(ConnectiePool)

## End(Not run)
```

maakLSVIrapport

Genereert LSVI-rapport op basis van de opgegeven parameters

Description

Deze functie genereert een rapport met habitatfiches die gebruikt worden voor de bepaling van de Lokale Staat van Instandhouding van de habitat(sub)types die voldoen aan de opgegeven parameters. (Om een tabel te genereren met deze informatie om zelf een rapport te kunnen samenstellen, wordt verwezen naar de functie `geefInfoHabitatfiche()`.)

De parameters kunnen enkel de hieronder gespecificeerde waarden bevatten en moeten als string opgegeven worden. Default is telkens "alle", waarbij de soortenlijsten voor alle mogelijke waarden van die parameter weergegeven worden (m.a.w. er is geen selectie voor deze parameter).

Usage

```
maakLSVIRapport(
  Bestandsnaam = "LSVIRapport.html",
  Versie = "alle",
  Habitatgroep = "alle",
  Habitatype = "alle",
  ConnectieLSVIhabitats = NULL,
  verbose = TRUE
)
```

Arguments

Bestandsnaam	Een naam voor het html-bestand dat gegenereerd wordt, bestaande uit een string die eindigt op .html
Versie	De versie van het LSVI-rapport, bv. "Versie 2" of "Versie 3". Bij de default "alle" worden de gegevens voor de verschillende versies gegeven. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Versie", "VersieLSVI")</code> of <code>geefVersieInfo()</code> .
Habitatgroep	Parameter waarmee alle habitats van een bepaalde habitatgroep kunnen geselecteerd worden, bv. "Bossen", "Heiden", "(Half-)natuurlijke graslanden", "Zoete en brakke wateren",... en "alle" (=default). Deze waarde moet niet gespecificeerd worden als een bepaald habitat(sub)type geselecteerd wordt. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitatgroep", "Naam")</code> .
Habitatype	Parameter waarmee een habitatype of habitatsubtype kan geselecteerd worden. Als dit een habitatype betreft met meerdere subtypes, zullen de gegevens van alle subtypes van dit habitatype weergegeven worden. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitatype", "Code")</code> . Er is voor deze parameter ook de mogelijkheid om een vector van meerdere habitat(sub)typen op te geven.
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie <code>connecteerMetLSVIdb()</code> .
verbose	geeft de toestand van het systeem aan, om te zorgen dat boodschappen niet onnodig gegeven worden

Value

Deze functie genereert habitatfiches in de vorm van html-bestanden die in de working directory opgeslagen worden.

Examples

```
# Omwille van de iets langere lange duurtijd van de commando's staat bij
# onderstaande voorbeelden de vermelding 'dontrun' (om problemen te vermijden
# bij het testen van het package). Maar de voorbeelden werken en kunnen zeker
# uitgetest worden.
## Not run:
```

```

maakConnectiePool()
maakLSVIRapport(
  Bestandsnaam = "LSVIRapport_heiden_versie3.html",
  Versie = "Versie 2.0", Habitatgroep = "Heiden"
)
maakLSVIRapport(
  Bestandsnaam = "LSVIRapport_4030.html",
  Habitattype = "4030"
)
library(pool)
poolClose(ConnectiePool)

## End(Not run)

```

parseTaxonnaam	<i>Vereenvoudig de soortnaam door de auteurs te verwijderen</i>
----------------	---

Description

Deze functie vereenvoudigt de opgegeven taxonnamen door de auteursnaam te verwijderen. Ze is gebaseerd op de functie `name_parse()` uit het package `rgbif`, maar ze vangt enkele specifieke situaties uit de INBO-plantenlijsten wel op die de originele functie niet correct opvangt, bv. v., an en den herkennen als deel van een auteursnaam en s.l. en meerdere soortnamen gescheiden door / (zonder spatie) vervangen door een achtervoegsel groep.

Usage

```
parseTaxonnaam(Taxonnaam, ParseType = "canonicalnamewithmarker")
```

Arguments

Taxonnaam	Wetenschappelijke naam of namen die vereenvoudigd moeten worden (String)
ParseType	Welk type parsing teruggegeven moet worden. Standaard is dit "canonicalnamewithmarker", andere opties zijn "canonicalname" en "canonicalnamecomplete"

Value

Deze functie geeft de vereenvoudigde naam of namen terug (als string of vector van strings)

Examples

```
parseTaxonnaam("Rosa canina L.")
```

selecteerIndicatoren *Selecteert indicatoren LSVI op basis van de opgegeven parameters*

Description

Deze hulpfunctie selecteert de indicatoren die gebruikt worden voor de bepaling van de Lokale Staat van Instandhouding voor de opgegeven parameters. Ze is bedoeld om te gebruiken als bouwsteen in andere functies waar de gegevens voor bijvoorbeeld een welbepaalde versie of welbepaalde habitattypes geselecteerd moeten kunnen worden.

De parameters kunnen enkel de hieronder gespecificeerde waarden bevatten en moeten als string opgegeven worden. Default is telkens "alle", waarbij de soortenlijsten voor alle mogelijke waarden van die parameter weergegeven worden (m.a.w. er is geen selectie voor deze parameter).

Usage

```
selecteerIndicatoren(
  Versie = "alle",
  Habitatgroep = "alle",
  Habitatype = "alle",
  Criterium = "alle",
  Indicator = "alle",
  HabitatnamenToevoegen = FALSE,
  ConnectieLSVIhabitats = NULL
)
```

Arguments

Versie	De versie van het LSVI-rapport, bv. "Versie 2" of "Versie 3". Bij de default "alle" worden de gegevens voor de verschillende versies gegeven. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Versie", "VersieLSVI")</code> of <code>geefVersieInfo()</code> .
Habitatgroep	Parameter waarmee alle habitats van een bepaalde habitatgroep kunnen geselecteerd worden, bv. "Bossen", "Heiden", "(Half-)natuurlijke graslanden", "Zoete en brakke wateren",... en "alle" (=default). Deze waarde moet niet gespecificeerd worden als een bepaald habitat(sub)type geselecteerd wordt. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitatgroep", "Naam")</code> .
Habitatype	Parameter waarmee een habitatype of habitatsubtype kan geselecteerd worden. Als dit een habitatype betreft met meerdere subtypes, zullen de gegevens van alle subtypes van dit habitatype weergegeven worden. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Habitatype", "Code")</code> . Er is voor deze parameter ook de mogelijkheid om een vector van meerdere habitat(sub)typen op te geven.
Criterium	Het LSVI-criterium waarvoor de gegevens geselecteerd worden: "Vegetatie", "Structuur", "Verstoring" of "alle".

Indicator	De indicator waarvoor de gegevens uit de databank gehaald worden. De mogelijke waarden kunnen opgevraagd worden via <code>geefUniekeWaarden("Indicator", "Naam")</code> .
HabitatnamenToevoegen	Moeten de namen van de habitattypes en habitatsubtypes toegevoegd worden als extra kolommen? (Bij FALSE worden enkel de habitatcodes toegevoegd, niet de volledige namen.)
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie <code>connecteerMetLSVIDb()</code> .

Value

Deze functie geeft een tabel met velden `Versie`, `Habitattype`, `Habitatsubtype`, `Criterium`, `Indicator`, `Indicator_habitatID`, `TaxongroepID` en `Indicator_beoordelingID`.

Examples

```
# Omwille van de iets langere lange duurtijd van de commando's staat bij
# onderstaande voorbeelden de vermelding 'dontrun' (om problemen te vermijden
# bij het testen van het package). Maar de voorbeelden werken en kunnen zeker
# uitgetest worden.
## Not run:
library(LSVI)
maakConnectiePool()
selecteerIndicatoren(Versie = "Versie 2.0", Habitattype = "4030")
selecteerIndicatoren(Versie = "Versie 2.0", Habitatgroep = "Heiden")
library(pool)
poolClose(ConnectiePool)

## End(Not run)
```

`selecteerKenmerkenInOpname`

Controle van de ingevoerde opname

Description

Deze hulpfunctie voor de s4-klassen `aantal` en `bedekking` zoekt soorten of kenmerken uit de voorwaarde in de opname en maakt een lijstje van de soorten die voldoen en in de opname voorkomen. Op basis hiervan kunnen de s4-klassen het totale aantal of de bedekking berekenen.

Usage

```
selecteerKenmerkenInOpname(
  Kenmerken,
  Soortengroep,
```

```

    Studiegroep,
    SubAnalyseVariabele,
    SubRefMin,
    SubRefMax,
    SubOperator
  )

```

Arguments

Kenmerken	dataframe met alle opgegeven kenmerken, met velden Vegetatielaag, Kenmerk, TypeKenmerk, WaardeMin en WaardeMax
Soortengroep	dataframe met de soortenlijst die uit Kenmerken gehaald moet worden
Studiegroep	dataframe met de lijst kenmerken die uit Kenmerken gehaald moet worden
SubAnalyseVariabele	heeft waarde "bedekking" als er een subvoorwaarde is voor de bedekking van de geselecteerde soorten of kenmerken
SubRefMin	minimumwaarde van de grenswaarde voor de bedekking
SubRefMax	maximumwaarde van de grenswaarde voor de bedekking
SubOperator	operator voor deze subvoorwaarde: moet de bedekking hoger of lager liggen dan de opgegeven referentiewaarde?

Value

Deze functie geeft een aangepaste tabel Data_soorten terug waarin enkel de soorten uit de soortenlijst(en) opgenomen zijn en die bovendien gekoppeld is aan de gegevens van de soortenlijst.

vertaalIntervalUitvoer

zet een interval om naar een waarde in de gevraagde eenheid

Description

Deze functie zet een interval bestaande uit minimumwaarde en maximumwaarde om naar een uitvoerwaarde in de opgegeven eenheid. De functie gebruikt Type, Eenheid en Invoertype om te bepalen welke omzetting eventueel nodig is. Als minimum en maximum niet dezelfde waarde hebben, geeft ze beide waarden weer, gescheiden door een "-".

Usage

```
vertaalIntervalUitvoer(Dataset, LIJST, ConnectieLSVIhabitats)
```

Arguments

Dataset	dataframe met velden Rijnr, Type, Min, Max, Eenheid en Invoertype
LIJST	Dataframe met lijst die weergeeft hoe de vertaling moet gebeuren van numerieke waarden naar categorische variabelen. Verschillend van andere functies die dezelfde lijst gebruiken, mogen hier geen overlappende categorieën voorkomen binnen eenzelfde schaal. Om zulke lijst te bekomen, moeten uit de lijst gegenereerd door de functie <code>vertaalInvoerInterval()</code> de records met Basisschaal 1 gefilterd worden.
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie <code>connecteerMetLSVIdb()</code> .

Value

Dataframe met velden Min

`vertaalInvoerInterval` zet ingevoerde gegevens om naar een interval

Description

Deze functie zet ingevoerde gegevens van meerdere types om naar een interval bestaande uit minimumwaarde en maximumwaarde. De functie gebruikt Type, Eenheid en Invoertype om te bepalen welke omzetting eventueel nodig is. Percentages worden bv. omgezet naar een decimaal getal (waarbij minimum en maximum dezelfde waarde zullen krijgen), en categorische variabelen met Invoertype Tansley worden omgezet naar de onder- en bovengrens die in de databank gegeven worden voor de betreffende categorie (op basis van parameter LIJST). Om een onderscheid te maken tussen numerieke waarden en aan-/afwezigheid, wordt voor deze laatste enkel de minimumwaarde ingevoerd (dus maximum = NA).

Usage

```
vertaalInvoerInterval(Dataset, LIJST, ConnectieLSVIhabitats)
```

Arguments

Dataset	dataframe met velden Rijnr, Type, Waarde, Eenheid en Invoertype
LIJST	Dataframe met lijst die weergeeft hoe de vertaling moet gebeuren van categorische variabelen naar numerieke waarden (en omgekeerd). Default worden deze waarden uit de databank met LSVI-indicatoren gehaald d.m.v. de functie <code>vertaalInvoerInterval()</code> . Aangeraden wordt om deze default te gebruiken (dus parameter niet expliciet invullen), of deze waar nodig aan te vullen met eigen schalen. Omdat er ook een omzetting moet gebeuren voor grenswaarden uit de databank, kan het niet doorgeven van een gedeelte van deze lijst problemen geven.
ConnectieLSVIhabitats	Connectie met de databank met indicatoren voor de LSVI van habitats, in te stellen d.m.v. functie <code>connecteerMetLSVIdb()</code> .

Value

Dataframe met velden Rijnr, Min en Max

Index

berekenLSVibasis, [3](#)
berekenStatus, [6](#)
berekenVerschilscores, [7](#)
berekenVoorwaarde, [7](#)

combinerenVerschilscore, [8](#)
combinerenVoorwaarden, [9](#)
connecteerMetLSVIdb, [10](#)
controleerInvoerwaarde, [11](#)

deselecteerKenmerkenInOpname, [12](#)

geefInfoHabitatfiche, [13](#)
geefInvoervereisten, [14](#)
geefSoortenlijst, [17](#)
geefSoortenlijstVoorIDs, [19](#)
geefUniekeWaarden, [20](#)
geefVersieInfo, [21](#)
geefVertaallijst, [22](#)

invoercontroleData_habitat, [23](#)
invoercontroleData_soortenKenmerken, [23](#)
invoercontroleData_voorwaarden, [24](#)
invoercontroleKwaliteitsniveau, [25](#)
invoercontroleVersie, [25](#)

logDatabankfouten, [26](#)

maakConnectiePool, [26](#)
maakHabitatfiches, [27](#)
maakLSVIRapport, [28](#)

parseTaxonnaam, [30](#)

selecteerIndicatoren, [31](#)
selecteerKenmerkenInOpname, [32](#)

vertaalIntervalUitvoer, [33](#)
vertaalInvoerInterval, [34](#)