# Package ‘neonOS’

**June 14, 2024**

**Title**  
Basic Data Wrangling for NEON Observational Data

**Version**  
1.1.0

**Date**  
2024-06-14

**Description**  
NEON observational data are provided via the NEON Data Portal [https://www.neonscience.org](https://www.neonscience.org) and NEON API, and can be downloaded and reformatted by the 'neonUtilities' package. NEON observational data (human-observed measurements, and analyses derived from human-collected samples, such as tree diameters and algal chemistry) are published in a format consisting of one or more tabular data files. This package provides tools for performing common operations on NEON observational data, including checking for duplicates and joining tables.

**Depends**  
R (>= 4.0)

**Imports**  
utils, data.table, httr, curl, jsonlite

**Suggests**  
testthat, neonUtilities

**License**  
AGPL-3

**URL**  
https://github.com/NEONScience/NEON-OS-data-processing

**BugReports**  
https://github.com/NEONScience/NEON-OS-data-processing/issues

**Encoding**  
UTF-8

**LazyData**  
true

**RoxygenNote**  
7.3.1

**NeedsCompilation**  
no

**Author**  
Claire Lunch [aut, cre, ctb] (https://orcid.org/0000-0001-8753-6593),  
Eric Sokol [aut, ctb] (https://orcid.org/0000-0001-5923-0917),  
Natalie Robinson [aut, ctb] (https://orcid.org/0000-0002-1144-418X),  
NEON (National Ecological Observatory Network) [aut]

**Maintainer**  
Claire Lunch <clunch@battelleecology.org>

**Repository**  
CRAN

**Date/Publication**  
2024-06-14 20:50:02 UTC
Contents

brd_countdata .................................................. 2
brd_perpoint ................................................... 3
cfc_lignin_test_dups ......................................... 5
cfc_lignin_variables .......................................... 5
dupProcess ..................................................... 6
getAPI .......................................................... 7
getSampleTree .................................................. 7
getTaxonList ................................................... 8
idSampleParents ............................................... 10
joinTableNEON .................................................. 10
removeDups .................................................... 12

Index 14

<table>
<thead>
<tr>
<th>brd_countdata</th>
<th>Count data table from Breeding landbird point counts (DP1.10003.001)</th>
</tr>
</thead>
</table>

Description

An example set of NEON observational data. Contains the individual bird observations from Niwot Ridge (NIWO) in 2019, as published in RELEASE-2021.

Usage

brd_countdata

Format

A data frame with 472 rows and 26 columns

**uid** Unique ID within NEON database; an identifier for the record

**namedLocation** Name of the measurement location in the NEON database

**domainID** Unique identifier of the NEON domain

**siteID** NEON site code

**plotID** Plot identifier (NEON site code XXX)

**plotType** NEON plot type in which sampling occurred: tower, distributed or gradient

**pointID** Identifier for a point location

**startDate** The start date-time or interval during which an event occurred

**eventID** An identifier for the set of information associated with the event, which includes information about the place and time of the event

**pointCountMinute** The minute of sampling within the point count period

**targetTaxaPresent** Indicator of whether the sample contained individuals of the target taxa
taxonID  Species code, based on one or more sources
scientificName  Scientific name, associated with the taxonID. This is the name of the lowest level taxonomic rank that can be determined
taxonRank  The lowest level taxonomic rank that can be determined for the individual or specimen
vernacularName  A common or vernacular name
family  The scientific name of the family in which the taxon is classified
nativeStatusCode  The process by which the taxon became established in the location
observerDistance  Radial distance between the observer and the individual(s) being observed
detectionMethod  How the individual(s) was (were) first detected by the observer
visualConfirmation  Whether the individual(s) was (were) seen after the initial detection
sexOrAge  Sex of individual if detectable, age of individual if individual can not be sexed
clusterSize  Number of individuals in a cluster (a group of individuals of the same species)
clusterCode  Alphabetic code (A-Z) linked to clusters (groups of individuals of the same species) spanning multiple records
identifiedBy  An identifier for the technician who identified the specimen
publicationDate  Date of data publication on the NEON data portal
release  Identifier for data release

Source

https://data.neonscience.org/api/v0/products/DP1.10003.001

brd_perpoint  Per-point data table from Breeding landbird point counts (DP1.10003.001)

Description

An example set of NEON observational data. Contains the point metadata associated with bird observations from Niwot Ridge (NIWO) in 2019, as published in RELEASE-2021.

Usage

brd_perpoint

Format

A data frame with 54 rows and 31 columns

uid  Unique ID within NEON database; an identifier for the record
namedLocation  Name of the measurement location in the NEON database
domainID  Unique identifier of the NEON domain
siteID  NEON site code
plotID  Plot identifier (NEON site code_XXX)
plotType  NEON plot type in which sampling occurred: tower, distributed or gradient
pointID  Identifier for a point location
nlcdClass  National Land Cover Database Vegetation Type Name
decimalLatitude  The geographic latitude (in decimal degrees, WGS84) of the geographic center of the reference area
decimalLongitude  The geographic longitude (in decimal degrees, WGS84) of the geographic center of the reference area
geodeticDatum  Model used to measure horizontal position on the earth
coordinateUncertainty  The horizontal distance (in meters) from the given decimalLatitude and decimalLongitude describing the smallest circle containing the whole of the Location. Zero is not a valid value for this term
elevation  Elevation (in meters) above sea level
elevationUncertainty  Uncertainty in elevation values (in meters)
startDate  The start date-time or interval during which an event occurred
samplingImpracticalRemarks  Technician notes; free text comments accompanying the sampling impractical record
samplingImpractical  Samples and/or measurements were not collected due to the indicated circumstance
eventID  An identifier for the set of information associated with the event, which includes information about the place and time of the event
startCloudCoverPercentage  Observer estimate of percent cloud cover at start of sampling
endCloudCoverPercentage  Observer estimate of percent cloud cover at end of sampling
startRH  Relative humidity as measured by handheld weather meter at the start of sampling
endRH  Relative humidity as measured by handheld weather meter at the end of sampling
observedHabitat  Observer assessment of dominant habitat at the sampling point at sampling time
observedAirTemp  The air temperature measured with a handheld weather meter
kmPerHourObservedWindSpeed  The average wind speed measured with a handheld weather meter, in kilometers per hour
laboratoryName  Name of the laboratory or facility that is processing the sample
samplingProtocolVersion  The NEON document number and version where detailed information regarding the sampling method used is available; format NEON.DOC.#####vX
remarks  Technician notes; free text comments accompanying the record
measuredBy  An identifier for the technician who measured or collected the data
publicationDate  Date of data publication on the NEON data portal
release  Identifier for data release

Source
https://data.neonscience.org/api/v0/products/DP1.10003.001
cfc_lignin_test_dups  Lignin data table from Plant foliar traits (DP1.10026.001)

Description
An example set of NEON observational data, containing duplicate records. NOT APPROPRIATE FOR ANALYTICAL USE. Contains foliar lignin data from Moab and Toolik in 2017, with artificial duplicates introduced to demonstrate the removeDups() function.

Usage
cfc_lignin_test_dups

Format
A data frame with 26 rows and 25 columns
The variable names, descriptions, and units can be found in the cfc_lignin_variables table

Source
https://data.neonscience.org/api/v0/products/DP1.10026.001

cfc_lignin_variables  Variables file, subset to lignin table, from Plant foliar traits (DP1.10026.001)

Description
The foliar lignin table’s variables file from NEON observational data. Example to illustrate use of removeDups().

Usage
cfc_lignin_variables

Format
A data frame with 26 rows and 8 columns

table  The table name of the NEON data table
fieldName Field name within the table; corresponds to column names in cfc_lignin_test_dups
description Description for each field name
dataType Type of data for each field name
units  Units for each field name
downloadPkg  Is the field published in the basic or expanded data package?

pubFormat  Publication formatting, e.g. date format or rounding

primaryKey  Fields indicated by Y, when combined, should identify a unique record. Used by removeDups() to identify duplicate records.

Source

https://data.neonscience.org/api/v0/products/DP1.10026.001

dupProcess  Helper function to remove duplicates from a data table; assumes input data are a duplicated set.

Description

Helper function to carry out duplicate removal on a data table of duplicates.

Usage

dupProcess(data, data.dup, table)

Arguments

data  A data frame containing original duplicated data. [data frame]
data.dup  A data frame containing lowercase conversion of the duplicated data. [character]
table  The table name for the input data frame

Details

Helper function to carry out the flagging and removal for removeDups().

Value

A modified data frame with resolveable duplicates removed and a flag field added and populated.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007
getAPI  

Get the data from API

Description

Accesses the API with options to use the user-specific API token generated within neon.datascience user accounts.

Usage

getAPI(apiURL, token = NA_character_)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>apiURL</td>
<td>The API endpoint URL</td>
</tr>
<tr>
<td>token</td>
<td>User specific API token (generated within neon.datascience user accounts). Optional.</td>
</tr>
</tbody>
</table>

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

getSampleTree  

Find all relatives (parents, children, and outward) of a given sample.

Description

Find all samples in the sample tree of a given sample.

Usage

getSampleTree(
    sampleNode,
    idType = "tag",
    sampleClass = NA_character_,
    token = NA_character_
)
getTaxonList

getTaxonList (getTaxonList) Get NEON taxon table

Description

This is a function to retrieve a taxon table from the NEON data portal for a given taxon type and provide it in a tractable format.

Args

sampleNode A NEON sample identifier. [character]
idType Is sampleNode a tag, barcode, or guid? Defaults to tag. [character]
sampleClass The NEON sampleClass of sampleNode. Required if sampleNode is a tag and there are multiple valid classes. [character]
token User specific API token (generated within neon.datascience user accounts). Optional. [character]

Details

Related NEON samples can be connected to each other in a parent-child hierarchy. Parents can have one or many children, and children can have one or many parents. Sample hierarchies can be simple or complex - for example, particulate mass samples (dust filters) have no parents or children, whereas water chemistry samples can be subsampled for dissolved gas, isotope, and microbial measurements. This function finds all ancestors and descendants of the focal sample (the sampleNode), and all of their relatives, and so on recursively, to provide the entire hierarchy. See documentation for each data product for more specific information.

Value

A table of sample identifiers, their classes, and their parent samples.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

# Find related samples for a soil nitrogen transformation sample
## Not run:
soil_samp <- getSampleTree(sampleNode="B0000012358", idType="barcode")
## End(Not run)
getTaxonList

Usage

getTaxonList(
    taxonType = NA,
    recordReturnLimit = NA,
    stream = "true",
    verbose = "false",
    token = NA
)

Arguments

taxonType  The taxonTypeCode to access. Must be one of ALGAE, BEETLE, BIRD, FISH, HERPETOLOGY, MACROINVERTEBRATE, MOSQUITO, MOSQUITO_PATHOGENS, SMALL_MAMMAL, PLANT, TICK [character]

recordReturnLimit  The number of items to limit the result set to. If NA (the default), will return either the first 100 records, or all records in the table, depending on the value of ‘stream’. Use ‘stream=’true’’ to get all records. [integer]

stream  True or false, obtain the results as a stream. Utilize for large requests. Note this is lowercase true and false as character strings, not logical. [character]

verbose  True or false, include all possible taxonomic parameters. Defaults to false, only essential parameters. Note this is lowercase true and false as character strings, not logical. [character]

token  User specific API token (generated within neon.datascience.org user account) [character]

Value

Data frame with selected NEON taxonomic data

Author(s)

Eric R. Sokol <esokol@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

# taxonTypeCode must be one of
# ALGAE, BEETLE, BIRD, FISH,
# HERPETOLOGY, MACROINVERTEBRATE,
# MOSQUITO, MOSQUITO_PATHOGENS,
# SMALL_MAMMAL, PLANT, TICK
# get the first 4 fish taxa
taxa_table <- getTaxonList('FISH', recordReturnLimit = 4)
idSampleParents  

Reformat sample data to indicate the parents of a focal sample.

Description
Reformat table of sample identifiers to include parent sample identifiers. Used in getSampleTree().

Usage
idSampleParents(sampleUuid, token = NA_character_)

Arguments
- sampleUuid: A NEON sample UUID. [character]
- token: User specific API token (generated within neon.datascience user accounts). Optional. [character]

Value
A table of sample identifiers, their classes, and their parent samples.

Author(s)
Claire Lunch <clunch@battelleecology.org>

References
License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

joinTableNEON  

Join two data tables from NEON Observational System

Description
NEON observational data are published in multiple tables, usually corresponding to activities performed in different times or places. This function uses the fields identified in NEON Quick Start Guides to join tables containing related data.

Usage
joinTableNEON(
  table1,
  table2,
  name1 = NA_character_,
  name2 = NA_character_,
  location.fields = NA,
  left.join = NA
)
Arguments

- **table1**: A data frame containing data from a NEON observational data table.
- **table2**: A second data frame containing data from a NEON observational data table.
- **name1**: The name of the first table. Defaults to the object name of table1.
- **name2**: The name of the second table. Defaults to the object name of table2.
- **location.fields**: Should standard location fields be included in the list of linking variables, to avoid duplicating those fields? For most data products, these fields are redundant, but there are a few exceptions. This parameter defaults to NA, in which case the Quick Start Guide is consulted. If QSG indicates location fields shouldn’t be included, value is updated to FALSE, otherwise to TRUE. Enter TRUE or FALSE to override QSG defaults.
- **left.join**: Should the tables be joined in a left join? This parameter defaults to NA, in which case the Quick Start Guide is consulted. If the QSG does not specify, a full join is performed. Enter TRUE or FALSE to override the default behavior, including using FALSE to force a full join when a left join is specified in the QSG. Forcing a left join is not generally recommended; remember you will likely be discarding data from the second table.

Details

The "Table joining" section of NEON Quick Start Guides (QSGs) provides the field names of the linking variables between related NEON data tables. This function uses the QSG information to join tables. Tables are joined using a full join unless the QSG specifies otherwise. If you need to remove duplicates as well as joining, run removeDups() before running joinTableNEON(). Tables that don’t appear together in QSG instructions can’t be joined here. Some tables may not be straightforwardly joinable, such as tables of analytical standards run as unknowns. Theoretically, these data could be joined to analytical results by a combination of laboratory and date, but in general, a table join is not the best way to analyze this type of data. If a pair of tables is omitted from QSG instructions that you expected to find, contact NEON.

Value

A single data frame created by joining table1 and table2 on the fields identified in the quick start guide.

Author(s)

Claire Lunch <clunch@battelleecology.org>

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007
removeDups

Remove duplicates from a data table based on a provided primary key; flag duplicates that can’t be removed.

Description

NEON observational data may contain duplicates; this function removes exact duplicates, attempts to resolve non-exact duplicates, and flags duplicates that can’t be resolved.

Usage

removeDups(data, variables, table = NA_character_, ncores = 1)

Arguments

data A data frame containing data from a NEON observational data table [data frame]
variables The NEON variables file containing metadata about the data table in question [data frame]
table The name of the table. Must match one of the table names in `variables` [character]
ncores The maximum number of cores to use for parallel processing. Defaults to 1. [numeric]

Details

Duplicates are identified based on exact matches in the values of the primary key. For records with identical keys, these steps are followed, in order: (1) If records are identical except for NA or empty string values, the non-empty values are kept. (2) If records are identical except for uid, remarks, and/or personnel (xxxxBy) fields, unique values are concatenated within each field, and the merged version is kept. (3) For records that are identical following steps 1 and 2, one record is kept and flagged with duplicateRecordQF=1. (4) Records that can’t be resolved by steps 1-3 are flagged with duplicateRecordQF=2. Note that in a set of three or more duplicates, some records may be resolveable and some may not; if two or more records are left after steps 1-3, all remaining records are flagged with duplicateRecordQF=2. In some limited cases, duplicates can’t be unambiguously identified, and these records are flagged with duplicateRecordQF=-1.

Value

A modified data frame with resolveable duplicates removed and a flag field added and populated.

Author(s)

Claire Lunch <clunch@battelleecology.org>
removeDups

References

License: GNU AFFERO GENERAL PUBLIC LICENSE Version 3, 19 November 2007

Examples

# Resolve and flag duplicates in a test dataset of foliar lignin
lig_dup <- removeDups(data=cfc_lignin_test_dups,
                         variables=cfc_lignin_variables,
                         table="cfc_lignin")
Index

* datasets
  brd_countdata, 2
  brd_perpoint, 3
  cfc_lignin_test_dups, 5
  cfc_lignin_variables, 5

brd_countdata, 2
brd_perpoint, 3

cfc_lignin_test_dups, 5
cfc_lignin_variables, 5

dupProcess, 6
getAPI, 7
getSampleTree, 7
getTaxonList, 8

idSampleParents, 10
joinTableNEON, 10
removeDups, 12