Package ‘rtry’

October 14, 2022

Title  Preprocessing Plant Trait Data
Version  1.0.0
Description  Designed to support the application of plant trait data providing easy applicable functions for the basic steps of data preprocessing, e.g. data import, data exploration, selection of columns and rows, excluding trait data according to different attributes, geocoding, long- to wide-table transformation, and data export. ‘rtry’ was initially developed as part of the TRY R project to preprocess trait data received via the TRY database.
License  CC BY 4.0
Depends  R (>= 4.0.0)
Imports  utils, data.table, dplyr, tidyr, jsonlite, curl
URL  https://github.com/MPI-BGC-Functional-Biogeography/rtry,
     https://www.try-db.org/TryWeb/Home.php
BugReports  https://github.com/MPI-BGC-Functional-Biogeography/rtry/issues
Encoding  UTF-8
LazyData  true
RoxygenNote  7.1.2
Suggests  knitr, rmarkdown, testthat
VignetteBuilder  knitr
Maintainer  Olee Hoi Ying Lam <hlam9@wisc.edu>
NeedsCompilation  no
Author  Olee Hoi Ying Lam [aut, cre] (https://orcid.org/0000-0002-7731-3246),
        Susanne Tautenhahn [aut] (https://orcid.org/0000-0002-2753-3443),
        Gabriel Walther [aut] (https://orcid.org/0000-0003-3550-9045),
        Gerhard Boenisch [aut],
        Pramod Baddam [aut] (https://orcid.org/0000-0002-7858-9295),
        Jens Kattge [aut] (https://orcid.org/0000-0002-1022-8469),
        Fellowship Group Functional Biogeography at MPI-BGC, Jena, Germany [cph]
Repository  CRAN
Date/Publication  2022-01-17 09:02:42 UTC
### R topics documented:

- `data_coordinates` ................................................. 2
- `data_locations` .................................................. 3
- `data_TRY_15160` .................................................. 3
- `data_TRY_15161` .................................................. 4
- `rtry_bind_col` .................................................... 6
- `rtry_bind_row` .................................................... 7
- `rtry_exclude` ..................................................... 8
- `rtry_explore` ..................................................... 9
- `rtry_export` ...................................................... 10
- `rtry_geocoding` .................................................. 11
- `rtry_import` ...................................................... 12
- `rtry_join_left` ................................................... 13
- `rtry_join_outer` .................................................. 14
- `rtry_remove_col` .................................................. 16
- `rtry_remove_dup` .................................................. 17
- `rtry_revgeocoding` .............................................. 18
- `rtry_select_anc` .................................................. 19
- `rtry_select_col` .................................................. 20
- `rtry_select_row` .................................................. 21
- `rtry_trans_wider` ................................................ 22

### Index

<table>
<thead>
<tr>
<th>data_coordinates</th>
<th>Sample coordinates data</th>
</tr>
</thead>
</table>

### Description

A dataset containing 20 sets of latitudes and longitudes in WGS84 projection. The raw dataset (`data_coordinates.csv`) is also provided in the directory `testdata`.

### Usage

```r
data_coordinates
```

### Format

A data frame with 20 rows and 2 variables:

- **Latitude**  Latitude, in WGS84 projection.
- **Longitude** Longitude, in WGS84 projection.

### Value

A data frame with 20 rows and 2 variables of sample coordinates data.
data_locations

Sample locations data

Description
A dataset containing 20 location information. The raw dataset (data_locations.csv) is also provided in the directory testdata.

Usage
data_locations

Format
A data frame with 20 rows and 3 variables:
- **Country code** Country code.
- **Country** Full name of a country.
- **Location** Specific location, e.g. town or city name.

Value
A data frame with 20 rows and 3 variables of sample locations data.

data_TRY_15160

Sample TRY data (Request 15160)

Description
A dataset requested from the TRY Database. The request ID of this dataset is 15160, which contains TraitID: 3115, 3116 and AccSpeciesID: 10773, 35846, 45737. The raw dataset (data_TRY_15160.txt) is also provided in the directory testdata.

Usage
data_TRY_15160

Format
A data frame with 1782 rows and 28 variables:
- **LastName** Surname of data contributor.
- **FirstName** First name of data contributor.
- **DatasetID** Unique identifier of contributed dataset.
- **Dataset** Name of contributed dataset.
**SpeciesName**  Original name of species.

**AccSpeciesID**  Unique identifier of consolidated species name.

**AccSpeciesName**  Consolidated species name.

**ObservationID**  Unique identifier for each observation in TRY.

**ObsDataID**  Unique identifier for each row in the TRY data table, either trait record or ancillary data.

**TraitID**  Unique identifier for traits (only if the record is a trait).

**TraitName**  Name of trait (only if the record is a trait).

**DataID**  Unique identifier for each DataName (either sub-trait or ancillary data).

**DataName**  Name of sub-trait or ancillary data.

**OriglName**  Original name of sub-trait or ancillary data.

**OrigValueStr**  Original value of trait or ancillary data.

**OrigUnitStr**  Original unit of trait or ancillary data.

**ValueKindName**  Value kind (single measurement, mean, median, etc.).

**OrigUncertaintyStr**  Original uncertainty.

**UncertaintyName**  Kind of uncertainty (standard deviation, standard error, etc.).

**Replicates**  Number of replicates.

**StdValue**  Standardized trait value: available for frequent continuous traits.

**UnitName**  Standard unit: available for frequent continuous traits.

**RelUncertaintyPercent**  Relative uncertainty in %.

**OrigObsDataID**  Unique identifier for duplicate trait records.

**ErrorRisk**  Indication for outlier trait values: distance to mean in standard deviations.

**Reference**  Reference to be cited if trait record is used in analysis.

**Comment**  Explanation for the OrigIName in the contributed dataset.

**V28**  Empty, an artifact due to different interpretation of column separator by MySQL and R.

### Value

A data frame with 1782 rows and 28 variables of sample TRY data (Request 15160).

---

**data_TRY_15161**  Sample TRY data (Request 15161)

---

### Description

A dataset requested from the TRY Database. The request ID of this dataset is 15161, which contains TraitID: 3117 and AccSpeciesID: 10773, 35846, 45737. The raw dataset (data_TRY_15161.txt) is also provided in the directory testdata.
Usage

data_TRY_15161

Format

A data frame with 4627 rows and 28 variables:

- **LastName**: Surname of data contributor.
- **FirstName**: First name of data contributor.
- **DatasetID**: Unique identifier of contributed dataset.
- **Dataset**: Name of contributed dataset
- **SpeciesName**: Original name of species.
- **AccSpeciesID**: Unique identifier of consolidated species name.
- **AccSpeciesName**: Consolidated species name.
- **ObservationID**: Unique identifier for each observation in TRY.
- **ObsDataID**: Unique identifier for each row in the TRY data table, either trait record or ancillary data.
- **TraitID**: Unique identifier for traits (only if the record is a trait).
- **TraitName**: Name of trait (only if the record is a trait).
- **DataID**: Unique identifier for each DataName (either sub-trait or ancillary data).
- **DataName**: Name of sub-trait or ancillary data.
- **OriglName**: Original name of sub-trait or ancillary data.
- **OrigValueStr**: Original value of trait or ancillary data.
- **OrigUnitStr**: Original unit of trait or ancillary data.
- **ValueKindName**: Value kind (single measurement, mean, median, etc.).
- **OrigUncertaintyStr**: Original uncertainty.
- **UncertaintyName**: Kind of uncertainty (standard deviation, standard error, etc.).
- **Replicates**: Number of replicates.
- **StdValue**: Standardized trait value: available for frequent continuous traits.
- **UnitName**: Standard unit: available for frequent continuous traits.
- **RelUncertaintyPercent**: Relative uncertainty in %.
- **OrigObsDataID**: Unique identifier for duplicate trait records.
- **ErrorRisk**: Indication for outlier trait values: distance to mean in standard deviations.
- **Reference**: Reference to be cited if trait record is used in analysis.
- **Comment**: Explanation for the OriglName in the contributed dataset.
- **V28**: Empty, an artifact due to different interpretation of column separator by MySQL and R.

Value

A data frame with 1782 rows and 28 variables of sample TRY data (Request 15161).
rtry_bind_col  

Bind data by columns

Description
This function takes a list of data frames or data tables and combines them by columns. The data have to have the same number and sequence of rows.

Usage
rtry_bind_col(..., showOverview = TRUE)

Arguments
... A list of data frames or data tables to be combined by columns.
showOverview Default TRUE displays the dimension and column names of the combined data.

Value
An object of the same type as the first input.

Note
A common attribute is not necessary (difference to the function rtry_join_left and rtry_join_outer): the binding process simply puts the data side-by-side.

References
This function makes use of the bind_cols function within the dplyr package.

See Also
rtry_bind_row, rtry_join_left, rtry_join_outer

Examples
# Assuming a user has selected different columns as separated data tables
# and later on would like to combine them as one for further processing.
data1 <- rtry_select_col(data_TRY_15160,
  ObsDataID, ObservationID, AccSpeciesID, AccSpeciesName, ValueKindName,
  TraitID, TraitName, DataID, DataName, OrigObsDataID, ErrorRisk, Comment)

data2 <- rtry_select_col(data_TRY_15160,
  OriglName, OrigValueStr, OrigUnitStr, StdValue, UnitName)

data <- rtry_bind_col(data1, data2)

# Expected messages:
# dim:  1782 12
rtry_bind_row

# col: ObsDataID ObservationID AccSpeciesID AccSpeciesName ValueKindName TraitID
#      TraitName DataID DataName OrigObsDataID ErrorRisk Comment
#
# dim: 1782 5
# col: OriglName OrigValueStr OrigUnitStr StdValue UnitName
#
# dim: 1782 17
# col: ObsDataID ObservationID AccSpeciesID AccSpeciesName ValueKindName TraitID
#      TraitName DataID DataName OrigObsDataID ErrorRisk Comment OriglName
#      OrigValueStr OrigUnitStr StdValue UnitName

rtry_bind_row Bind data by rows

Description
This function takes a list of data frames or data tables and combines them by rows, it adds the rows
of the second data below the rows of the first one.

Usage
rtry_bind_row(..., showOverview = TRUE)

Arguments
...

A list of data frames or data tables to be combined by rows.

showOverview

Default TRUE displays the dimension and column names of the combined data.

Value
An object of the same type as the first input. The object will contain a column if that column appears
in any of the inputs.

Note
A common attribute is not necessary (difference to the function rtry_join_left and rtry_join_outer):
the binding process simply puts the data one after another while matching the column names, and
any missing columns will be filled with NA.

References
This function makes use of the bind_rows function within the dplyr package.

See Also
rtry_bind_col, rtry_join_left, rtry_join_outer
# Examples

```r
data <- rtry_bind_row(data_TRY_15160, data_TRY_15161)

# Expected message:
# dim:  6409  28
# col:  LastName  FirstName  DatasetID  Dataset  SpeciesName  AccSpeciesID  AccSpeciesName
#      ObservationID  ObsDataID  TraitID  TraitName  DataID  DataName  OrigName
#      OrigValueStr  OrigUnitStr  ValueKindName  OrigUncertaintyStr  UncertaintyName
#      Replicates  StdValue  UnitName  RelUncertaintyPercent  OrigObsDataID  ErrorRisk
#      Reference  Comment  V28
```

---

## rtry_exclude

### Exclude (remove) data

This function takes the input data frame or data table and excludes all records (rows) with the same value in the attribute specified in the argument `baseOn` if the criteria specified in the arguments for excluding (...) are fulfilled for one of those records.

### Usage

```r
rtry_exclude(input, ..., baseOn, showOverview = TRUE)
```

### Arguments

- **input**: Input data frame or data table.
- **...**: Criteria for excluding.
- **baseOn**: The attribute on which excluding is based on. If it is set to `ObservationID`, the function excludes all records with the respective `ObservationID` if the specified criteria for excluding is fulfilled for one record. Alternatively, use `ObsDataID` to exclude only the record (row) for which the specified criterion is fulfilled. Other reasonable parameter values are `TraitID`, `DataID` or `AccSpeciesID`.
- **showOverview**: Default `TRUE` displays the dimension of the data after excluding.

### Value

An object of the same type as the input data after excluding.

### References

This function makes use of the `subset` function within the `base` package.
**Examples**

# Example 1: Exclude observations on juvenile plants or unknown state:
# Identify observations where the plant developmental status (DataID 413) is either
# "juvenile" or "unknown", and exclude the whole observation

data_filtered <- rtry_exclude(data_TRY_15160,
(DataID %in% 413) & (OrigValueStr %in% c("juvenile", "unknown")),
baseOn = ObservationID)

# Expected message:
# dim: 1618 28

# Example 2: Exclude outliers:
# Identify the outliers, i.e. trait records where the ErrorRisk is larger than 4
# and exclude these records (not the whole observation)
data_filtered <- rtry_exclude(data_TRY_15160,
ErrorRisk > 4,
baseOn = ObsDataID)

# Expected message:
# dim: 1778 28

# Learn more applications of the excluding function via the vignette (Workflow for
# general data preprocessing using rtry): vignette("rtry-workflow-general").

---

**rtry_explore**  

**Explore data**

**Description**

This function takes a data frame or data table and converts it into a grouped data frame of unique values based on the specified column names. A column (Count) is added, which shows the number of records within each group. The data are grouped by the first attribute if not specified with the argument sortBy.

**Usage**

```
rtry_explore(input, ..., sortBy = "", showOverview = TRUE)
```

**Arguments**

- `input`: Data frame or data table, e.g. from rtry_import().
- `...`: Attribute names to group together.
- `sortBy`: (Optional) Default "" indicates no sorting is applied to the grouped data. Specify the attribute name used to re-order the rows in ascending order.
- `showOverview`: Default TRUE displays the dimension of the result data table.

**Value**

A data frame of unique values grouped and sorted by the specified attribute(s).
References

This function makes use of the `group_by`, `summarise` and `arrange` functions within the `dplyr` package.

Examples

```r
# Explore the unique values in the provided sample data (data_TRY_15160)
# based on the attributes AccSpeciesID, AccSpeciesName, TraitID, TraitName, DataID
# and DataName, sorted by TraitID
data_explore <- rtry_explore(data_TRY_15160,
                              AccSpeciesID, AccSpeciesName, TraitID, TraitName, DataID, DataName,
                              sortBy = TraitID)

# Expected message:
# dim: 235 7

# Learn more applications of the explore function via the vignette (Workflow for
# general data preprocessing using rtry): vignette("rtry-workflow-general").
```

---

**rtry_export**

*Export preprocessed data*

**Description**

This function exports the preprocessed data as comma separated values to a `.csv` file. If the specified output directory does not exist, it will be created.

**Usage**

```r
rtry_export(data, output, quote = TRUE, encoding = "UTF-8")
```

**Arguments**

- `data`: The data to be saved.
- `output`: Output path.
- `quote`: Default `TRUE` inserts double quotes around any character or factor columns.
- `encoding`: Default "UTF-8". File encoding.

**Value**

No return value, called for exporting a `.csv` file.

**References**

This function makes use of the `write.csv` function within the `utils` package.
**rtry_geocoding**

**Examples**

```r
data_TRY_15160

# Export the preprocessed data to a specific location
rtry_export(data_TRY_15160, file.path(tempdir(), "TRYdata_unprocessed.csv"))

# Expected message:
# File saved at: C:\Users\user\AppData\Local\Temp\Rtmp4wJAvQ/TRYdata_unprocessed.csv
```

**rtry_geocoding**

**Perform geocoding**

**Description**

This function uses Nominatim, a search engine for OpenStreetMap (OSM) data, to perform geocoding, i.e. converting an address into coordinates (latitudes, longitudes). The data provided by OSM is free to use for any purpose, including commercial use, and is governed by the distribution license ODbL.

**Usage**

```r
rtry_geocoding(address, email)
```

**Arguments**

- **address**
  - String of an address.
- **email**
  - String of an email address.

**Value**

A data frame that contains latitudes (lat) and longitudes (lon) in WGS84 projection.

**See Also**

- `rtry_revgeocoding`

**Examples**

```r
# Convert the address of MPI-BGC ("Hans-Knoell-Strasse 10, 07745 Jena, Germany")
# into coordinates in latitudes and longitudes
# Note: Please change to your own email address when executing this function
rtry_geocoding("Hans-Knoell-Strasse 10, 07745 Jena, Germany",
  email = "john.doe@example.com")

# Expected message:
# lat  lon
# 1 50.91011 11.56682

# Learn to perform geocoding to a list of locations via the vignette (Workflow for
# geocoding using rtry): vignette("rtry-workflow-geocoding").
```
rtry_import

Import data

Description
This function imports a data file as a data.table for further processing. The default arguments are set to import tabular or delimited data files in text format (.txt) exported from the TRY database. It can also be used to import other file formats, such as .csv files with comma separated values.

Usage
rtry_import(
  input,
  separator = "\t",
  encoding = "Latin-1",
  quote = "",
  showOverview = TRUE
)

Arguments
- **input**: Path to the data file.
- **separator**: Default "\t" for the TRY data output. Data separator.
- **encoding**: Default "Latin-1". File encoding.
- **quote**: Default ";" reads the fields as is. If the fields in the data file are by a double quote, use "\"" instead.
- **showOverview**: Default TRUE displays the input path, the dimension and the column names of the imported data.

Value
A data.table.

References
This function makes use of the fread function within the data.table package.

Examples
# Example 1: Import data exported from the TRY database
# Specify file path to the raw data provided within the rtry package
input_path <- system.file("testdata", "data_TRY_15160.txt", package = "rtry")

# For own data and Windows users the path might rather look similar to this:
# input_path <- "C:/Users/User/Desktop/data_TRY_15160.txt"

# Import data file using rtry_import
input <- rtry_import(input_path)

# Explicit notation:
# input <- rtry_import(input_path, separator = "\t", encoding = "Latin-1",
# quote = "", showOverview = TRUE)

# Expected message:
# input: ~/R/R-4.0.3/library/rtry/testdata/data_TRY_15160.txt
# dim: 1782 28
# col: LastName FirstName DatasetID Dataset SpeciesName AccSpeciesID AccSpeciesName
# ObservationID ObsDataID TraitID TraitName DataID DataName OriglName
# OrigValueStr OrigUnitStr ValueKindName OrigUncertaintyStr UncertaintyName
# Replicates StdValue UnitName RelUncertaintyPercent OrigObsDataID ErrorRisk
# Reference Comment V28

# Example 2: Import CSV file
# Specify file path to the raw data provided within the rtry package
input_path <- system.file("testdata", "data_locations.csv", package = "rtry")

# Import data file using rtry_import
input <- rtry_import(input_path, separator = ",", encoding = "UTF-8",
quote = "\\", showOverview = TRUE)

# Expected message:
# input: ~/R/R-4.0.3/library/rtry/testdata/data_locations.csv
# dim: 20 3
# col: Country code Country Location

rtry_join_left

Left join for two data frames

Description

This function merges two data frames or data tables based on a specified common column and
returns all records from the left data frame (x) together with the matched records from the right data
frame (y), while discards all the records in the right data frame that does not exist in the left data
frame. In other words, this function performs a left join on the two provided data frames or data
tables.

Usage

rtry_join_left(x, y, baseOn, showOverview = TRUE)

Arguments

x
A data frame or data table to be coerced and will be considered as the data on
the left.

y
A data frame or data table to be coerced and will be considered as the data on
the right.
baseOn  The common column used for merging.
showOverview Default TRUE displays the dimension and column names of the merged data.

Value
An object of the same type of the input data. The merged data is by default lexicographically sorted
on the common column. The columns are the common column followed by the remaining columns
in x and then those in y.

References
This function makes use of the merge function within the base package.

See Also
rtry_join_outer, rtry_bind_col, rtry_bind_row

Examples
# Assume a user has obtained two unique data tables, one with the ancillary data
# Longitude and one with Latitude (e.g. using rtry_select_anc()), and would like to
# add a column Latitude to the data table with Longitude based on the common
# identifier ObservationID
lon <- rtry_select_anc(data_TRY_15160, 60)
lat <- rtry_select_anc(data_TRY_15160, 59)

georef <- rtry_join_left(lon, lat, baseOn = ObservationID)

# Expected messages:
# dim: 97 2
# col: ObservationID Longitude
#
# dim: 98 2
# col: ObservationID Latitude
#
# dim: 97 3
# col: ObservationID Longitude Latitude

---

**rtry_join_outer**  
*Outer join for two data frames*

**Description**
This function merges two data frames or data tables based on a specified common column and
returns all rows from both data, join records from the left (x) which have matching keys in the
right data frame (y). In order words, this functions performs an outer join on the two provided data
frames, i.e. the join table will contain all records from both data frames or data tables.
rtry_join_outer

**Usage**

```r
rtry_join_outer(x, y, baseOn, showOverview = TRUE)
```

**Arguments**

- **x**: A data frame or data table to be coerced and will be considered as the data on the left.
- **y**: A data frame or data table to be coerced and will be considered as the data on the right.
- **baseOn**: The common column used for merging.
- **showOverview**: Default TRUE displays the dimension and column names of the merged data.

**Value**

An object of the same type of the input data. The merged data is by default lexicographically sorted on the common column. The columns are the common column followed by the remaining columns in `x` and then those in `y`.

**References**

This function makes use of the `merge` function within the base package.

**See Also**

- `rtry_join_left`, `rtry_bind_col`, `rtry_bind_row`

**Examples**

```r
# Assume a user has obtained two unique data tables, one with the ancillary data
# Longitude and one with Latitude (e.g. using rtry_select_anc()), and would like to
# merge two data tables into one according to the common identifier ObservationID.
# It does not matter if either Longitude or Latitude data has no record
lon <- rtry_select_anc(data_TRY_15160, 60)
lat <- rtry_select_anc(data_TRY_15160, 59)

georef <- rtry_join_outer(lon, lat, baseOn = ObservationID)

# Expected messages:
# dim: 97 2
# col: ObservationID Longitude
# #
# dim: 98 2
# col: ObservationID Latitude
# #
# dim: 98 3
# col: ObservationID Longitude Latitude
```
Description

This function removes specified columns from the imported data for further processing.

Usage

rtry_remove_col(input, ..., showOverview = TRUE)

Arguments

input  
Input data frame or data table.

...  
Names of columns to be removed separated by commas. The operator : can be used for selecting a range of consecutive variables.

showOverview  
Default TRUE displays the dimension of the remaining data.

Value

An object of the same type as the input data.

References

This function makes use of the select function within the dplyr package.

See Also

rtry_select_col

Examples

# Remove certain columns from the provided sample data (data_TRY_15160)
data_rm_col <- rtry_remove_col(data_TRY_15160,
    LastName, FirstName, DatasetID, Dataset, SpeciesName,
    OrigUncertaintyStr, UncertaintyName, Replicates,
    RelUncertaintyPercent, Reference, V28)

# Expected message:
# dim: 1782 17
# col: AccSpeciesID AccSpeciesName ObservationID ObsDataID TraitID TraitName
# DataID DataName OrigName OrigValueStr OrigUnitStr ValueKindName
# StdValue UnitName OrigObsDataID ErrorRisk Comment
rtry_remove_dup

Remove duplicates in data

Description
This function removes the duplicates from the input data using the duplicate identifier OrigObsDataID provided within the TRY data. Once the function is called and executed, the number of duplicates removed will be displayed on the console as reference.

Usage
rtry_remove_dup(input, showOverview = TRUE)

Arguments
input Input data frame or data table.
showOverview Default TRUE displays the dimension of the data after removing the duplicates.

Value
An object of the same type as the input data after removing the duplicates.

Note
This function depends on the duplicate identifier OrigObsDataID listed in the data exported from the TRY database, therefore, if the column OrigObsDataID has been removed, this function will not work. Also, if the original value of an indicated duplicate is a restricted value, which has not been requested from the TRY database (if only public data were requested), the duplicate will be removed and this may result in data loss.

References
This function makes use of the \texttt{subset} function within the base package.

Examples
\begin{verbatim}
# Remove the duplicates within the provided sample data (data_TRY_15160)
data_rm_dup <- rtry_remove_dup(data_TRY_15160)

# Expected message:
# 45 duplicates removed.
# dim: 1737 28
\end{verbatim}
rtry_revgeocoding  Perform reverse geocoding

Description

This function uses Nominatim, a search engine for OpenStreetMap data, to perform reverse geocoding, i.e. converting coordinates (latitudes, longitudes) into an address. The data provided by OSM is free to use for any purpose, including commercial use, and is governed by the distribution license ODbL.

Usage

rtry_revgeocoding(lat_lon, email)

Arguments

lat_lon  A data frame containing latitude and longitude in WGS84 projection.
email  String of an email address.

Value

A data frame that contains address.

See Also

rtry_geocoding

Examples

# Convert the coordinates of MPI-BGC (50.91012, 11.56674) into an address
# Note: Please change to your own email address when executing this function
rtry_revgeocoding(data.frame(50.91012, 11.56674),
email = "john.doe@example.com")

# Expected message:
# full_address town city country country_code
# 1 Jena, Thuringia, Germany NA Jena Germany de

# Learn to perform reverse geocoding to a list of coordinates via the vignette
# (Workflow for geocoding using rtry): vignette("rtry-workflow-geocoding").
### rtry_select_anc

**Select ancillary data in wide table format**

#### Description

This function selects one specified ancillary data together with the ObservationID from the imported data and transforms it into a wide table format for further processing. It works for only one ancillary data in DataID at a time.

#### Usage

```r
rtry_select_anc(input, id, showOverview = TRUE)
```

#### Arguments

- **input**: Input data frame or data table.
- **id**: The ID of the ancillary data (DataID in the TRY data) to be selected.
- **showOverview**: Default `TRUE` displays the dimension and column names of the selected data.

#### Value

An object of the same type as the input data.

#### References

This function makes use of the `subset` and `distinct` functions within the base and `dplyr` packages respectively. It also uses the functions `rtry_select_col` and `rtry_remove_col`.

#### Examples

```r
# Obtain a list of ObservationID and the corresponding ancillary data of interest using the specified DataID (e.g. DataID 59 for latitude) from the provided sample data (e.g. data_TRY_15160)
lat <- rtry_select_anc(data_TRY_15160, 59)
```

# Expected message:
# dim: 98 2
# col: ObservationID Latitude
rtry_select_col  Select columns

Description
This function selects the specified columns from the input data.

Usage
rtry_select_col(input, ..., showOverview = TRUE)

Arguments
- input: Input data frame or data table.
- ...: Column names to be selected.
- showOverview: Default TRUE displays the dimension and column names of the selected columns.

Value
An object of the same type as the input data.

References
This function makes use of the select function within the dplyr package.

See Also
rtry_remove_col

Examples
# Select certain columns from the provided sample data (data_TRY_15160)
data_selected <- rtry_select_col(data_TRY_15160,
ObsDataID, ObservationID, AccSpeciesID, AccSpeciesName,
ValueKindName, TraitID, TraitName, DataID, DataName, OriglName,
OrigValueStr, OrigUnitStr, StdValue, UnitName, OrigObsDataID,
ErrorRisk, Comment)

# Expected message:
# dim: 1782 17
# col: ObsDataID ObservationID AccSpeciesID AccSpeciesName ValueKindName TraitID
# TraitName DataID DataName OriglName OrigValueStr OrigUnitStr StdValue
# UnitName OrigObsDataID ErrorRisk Comment
**rtry_select_row**

---

**Select rows**

---

**Description**

This function selects rows based on specified criteria and the corresponding **ObservationID** from the imported data for further processing.

**Usage**

```r
rtry_select_row(
  input, ...
  getAncillary = FALSE,
  rmDuplicates = FALSE,
  showOverview = TRUE
)
```

**Arguments**

- `input`  Input data frame or data table.
- `...`  Criteria for row selection.
- `getAncillary`  Default FALSE, set to TRUE selects all ancillary data based on the row selection criteria.
- `rmDuplicates`  Default FALSE, set to TRUE calls the function `rtry_remove_dup`.
- `showOverview`  Default TRUE displays the dimension of the data after row selection.

**Value**

An object of the same type as the input data.

**Note**

This function by default filters data based on the unique identifier **ObservationID** listed in the TRY data, therefore, if the column **ObservationID** has been removed, this function will not work.

**References**

This function makes use of the `unique` and `subset` functions within the base package. It also uses the function `rtry_remove_dup`. 
Examples

# Within the provided sample data (data_TRY_15160) select the georeferenced traits
# records together with records for Latitude and Longitude (DataID 59 and 60) and
# exclude duplicate trait records
data_selected <- rtry_select_row(data_TRY_15160,
  (TraitID > 0) | (DataID %in% c(59, 60)),
  getAncillary = TRUE,
  rmDuplicates = TRUE)

# Expected message:
# 45 duplicates removed.
# dim: 1737 28

rtry_trans_wider  Transform data from long to wide table

Description

This function transforms the original long table format of the data into a wide table format.

Usage

rtry_trans_wider(
  input,
  names_from = NULL,
  values_from = NULL,
  values_fn = NULL,
  showOverview = TRUE
)

Arguments

input Input data frame or data table.
names_from The column(s) from which the output column names to be obtained.
values_from The column(s) from which the output values to be obtained.
values_fn (Optional) Function to be applied to the output values.
showOverview Default TRUE displays the dimension of the wide table.

Value

A data frame of the transformed wide table.

References

This function makes use of the pivot_wider function within the tidyr package.
See Also

rtry_select_row, rtry_select_col, rtry_select_anc, rtry_join_left

Examples

# Provide the standardized trait values per observation, together with species names
# and the georeferences of the sampling site (Latitude and Longitude), if availalbe,
# in a wide table format. Several steps are necessary:

# 1. Select only the trait records that have standardized numeric values.
#    The complete.cases() is used to ensure the cases are complete, i.e. have no
#    missing values.
num_traits <- rtry_select_row(data_TRY_15160,
                            complete.cases(TraitID) & complete.cases(StdValue))

# 2. Select the relevant columns for transformation.
num_traits <- rtry_select_col(num_traits,
                             ObservationID, AccSpeciesID, AccSpeciesName, TraitID, TraitName,
                             StdValue, UnitName)

# 3. Extract the values of georeferences and the corresponding ObservationID.
lat <- rtry_select_anc(data_TRY_15160, 59)
lon <- rtry_select_anc(data_TRY_15160, 60)

# 4. Merge the relevant data frames based on the ObservationID using rtry_join_left().
num_traits_georef <- rtry_join_left(num_traits, lat, baseOn = ObservationID)
num_traits_georef <- rtry_join_left(num_traits_georef, lon, baseOn = ObservationID)

# 5. Perform wide table transformation of TraitID, TraitName and UnitName based on
#    ObservationID, AccSpeciesID and AccSpeciesName with cell values from StdValue.
#    If several records with StdValue were provided for one trait with the same
#    ObservationID, AccSpeciesID and AccSpeciesName, calculate their mean.
num_traits_georef_wider <- rtry_trans_wider(num_traits_georef,
                                            names_from = c(TraitID, TraitName, UnitName),
                                            values_from = c(StdValue),
                                            values_fn = list(StdValue = mean))

# Expected messages:
# dim: 150 28
# dim: 150 7
# col: ObservationID AccSpeciesID AccSpeciesName TraitID TraitName
#      StdValue UnitName
# # dim: 98 2
# col: ObservationID Latitude
# # dim: 97 2
# col: ObservationID Longitude
# # dim: 150 8
# col: ObservationID AccSpeciesID AccSpeciesName TraitID TraitName
#      StdValue UnitName Latitude
# dim: 150 9
# col: ObservationID AccSpeciesID AccSpeciesName TraitID TraitName
#      StdValue UnitName Latitude Longitude
#
# dim: 146 7

# Learn more via the vignette (Workflow for general data preprocessing using rtry):
# vignette("rtry-workflow-general")
Index

* datasets
  data_coordinates, 2
  data_locations, 3
  data_TRY_15160, 3
  data_TRY_15161, 4

arrange, 10
bind_cols, 6
bind_rows, 7

data_coordinates, 2
data_locations, 3
data_TRY_15160, 3
data_TRY_15161, 4
distinct, 19

fread, 12

group_by, 10

merge, 14, 15

pivot_wider, 22
rtry_bind_col, 6, 7, 14, 15
rtry_bind_row, 6, 7, 14, 15
rtryExclude, 8
rtry_explor, 9
rtry_export, 10
rtry_geocoding, 11, 18
rtry_import, 12
rtry_join_left, 6, 7, 13, 15, 23
rtry_join_outer, 6, 7, 14, 14
rtry_remove_col, 16, 19, 20
rtry_remove_dup, 17, 21
rtry_revgeocoding, 11, 18
rtry_select_anc, 19, 23
rtry_select_col, 16, 19, 20, 23
rtry_select_row, 21, 23
rtry_trans_wider, 22

select, 16, 20
subset, 8, 17, 19, 21
summarise, 10
unique, 21
write.csv, 10