Package: rWCVP (via r-universe)

November 12, 2026

Type Package

Title Working with Data from the World Checklist of Vascular Plants

Version 1.3.1

Maintainer Matilda Brown <matilda.brown@kew.org>

Description A companion to the World Checklist of Vascular Plants (WCVP). It includes functions to generate maps and species lists, as well as match names to the WCVP. For more details and to cite the package, see: Brown M.J.M., Walker B.E., Black N., Govaerts R., Ondo I., Turner R., Nic Lughadha E. (2023). ``rWCVP: A companion R package to the World Checklist of Vascular Plants". New Phytologist, 240: 1355-1365.

License GPL (>= 3)

Encoding UTF-8

LazyData true

LazyDataCompression xz

Imports cli, dplyr, ggplot2, gt, htmltools, lifecycle, magrittr, phonics, purrr, RecordLinkage, rlang, rmarkdown, sf, stringr, tibble, tidyr

Suggests rWCVPdata (>= 0.1.0), knitr, testthat (>= 3.0.0), spdep

Additional_repositories https://matildabrown.github.io/drat

VignetteBuilder knitr

URL https://github.com/matildabrown/rWCVP, https://matildabrown.github.io/rWCVP/

BugReports https://github.com/matildabrown/rWCVP/issues

Config/testthat/edition 3

Depends R (>= 3.5.0)

Config/roxygen2/version 8.1.1

NeedsCompilation no

Author Matilda Brown [aut, cre], Barnaby Walker [aut], Freya Read [ctb]

Repository https://cran.r-universe.dev

Date/Publication 2026-11-11 22:51:15 UTC

RemoteUrl https://github.com/cran/rWCVP

RemoteRef HEAD

RemoteSha 3b4aec332258d0d7687f020d391133fc9a7576b2

Contents

get_area_name . . . 2
get_wgsrpd3_codes . . . 3
powo_map . . . 4
powo_pal . . . 4
redlist_example . . . 6
taxonomic_mapping . . . 6
wcvp_checklist . . . 7
wcvp_distribution . . . 9
wcvp_distribution_map . . . 10
wcvp_match_exact . . . 11
wcvp_match_fuzzy . . . 12
wcvp_match_names . . . 13
wcvp_occ_mat . . . 15
wcvp_summary . . . 17
wcvp_summary_gt . . . 18
wgsrpd_mapping . . . 19
wgsrpd3 . . . 19

Index **21**

`get_area_name` *Get area description from vector of area codes*

Description

Get area description from vector of area codes

Usage

```
get_area_name(area_codes)
```

Arguments

`area_codes`	Character vector containing the set of codes to be mapped to a name.

Details

Usually used as an inverse function for `get_wgsrpd3_codes`. Useful for condensing sets of codes for e.g. file names, plotting and table formatting.

Value

Character. Either a vector of length one, with a name for the set of Level 3 areas, or (if no name exists for that set of areas) the input vector of codes.

Examples

```
get_area_name(get_wgsrpd3_codes("Brazil"))
```

`get_wgsrpd3_codes` *Extract WGSRPD Level 3 (area) codes.*

Description

Extract WGSRPD Level 3 (area) codes.

Usage

```
get_wgsrpd3_codes(geography, include_equatorial = NULL)
```

Arguments

`geography`	Character. The geography to convert into Level 3 codes. May be a WGSRPD area (Level 3), region (Level 2) or continent (Level 1), country (political) or hemisphere ("Northern Hemisphere", "Southern Hemisphere" or "Equatorial")
`include_equatorial`	Logical. Include Level 3 areas that span the equator? Defaults to `NULL`, which generates a message and includes these areas. Ignored if geography is not a hemisphere.

Details

Country mapping follows Gallagher et al. (2020). Importantly, this means that some overseas territories are not considered part of the country in this system, e.g. the Canary Islands are designated as their own Level 3 area, rather than part of Spain in this mapping. Where this is ambiguous, the mapping can be explored using `View(wgsrpd_mapping)`.

Gallagher, R. V., Allen, S., Rivers, M. C., Allen, A. P., Butt, N., Keith, D., & Adams, V. M. (2020). Global shortfalls in extinction risk assessments for endemic flora. bioRxiv, 2020.2003.2012.984559. https://doi.org/10.1101/2020.03.12.984559

Value

Character with area codes (Level 3) that fall within the geography.

Examples

```
get_wgsrpd3_codes("Brazil")
```

`powo_map` *Plot a POWO style map for given range and range centroids.*

Description

Plot a POWO style map for given range and range centroids.

Usage

```
powo_map(range_sf, centroids_sf)
```

Arguments

`range_sf`	A simple features (`sf`) data frame of range polygons
`centroids_sf`	A simple features (`sf`) data frame of range centroids

Value

A ggplot map of the range

`powo_pal` *POWO colour palette for range maps*

Description

Range maps displayed on the POWO website have a fixed, discrete colour palette based on the type of taxon occurrence in a region.

Usage

```
powo_pal()

scale_color_powo(...)

scale_colour_powo(...)

scale_fill_powo(...)
```

Arguments

...	Arguments passed on to `discrete_scale` `limits` One of: • `NULL` to use the default scale values • A character vector that defines possible values of the scale and their order • A function that accepts the existing (automatic) values and returns new ones. Also accepts rlang lambda function notation. `drop` Should unused factor levels be omitted from the scale? The default, `TRUE`, uses the levels that appear in the data; `FALSE` includes the levels in the factor. Please note that to display every level in a legend, the layer should use `show.legend = TRUE`. `na.translate` Unlike continuous scales, discrete scales can easily show missing values, and do so by default. If you want to remove missing values from a discrete scale, specify `na.translate = FALSE`. `name` The name of the scale. Used as the axis or legend title. If `waiver()`, the default, the name of the scale is taken from the first mapping used for that aesthetic. If `NULL`, the legend title will be omitted. `minor_breaks` One of: • `NULL` for no minor breaks • `waiver()` for the default breaks (none for discrete, one minor break between each major break for continuous) • A numeric vector of positions • A function that given the limits returns a vector of minor breaks. Also accepts rlang lambda function notation. When the function has two arguments, it will be given the limits and major break positions. `labels` One of the options below. Please note that when `labels` is a vector, it is highly recommended to also set the `breaks` argument as a vector to protect against unintended mismatches. • `NULL` for no labels • `waiver()` for the default labels computed by the transformation object • A character vector giving labels (must be same length as `breaks`) • An expression vector (must be the same length as breaks). See ?plotmath for details. • A function that takes the breaks as input and returns labels as output. Also accepts rlang lambda function notation. `guide` A function used to create a guide or its name. See `guides()` for more information. `call` The `call` used to construct the scale for reporting messages. `super` The super class to use for the constructed scale

Value

Character. Vector of names and HEX values to match those of POWO.

redlist_example *Example dataset for name matching*

Description

A dataset containing 20 sampled Red List assessments for name-matching

Usage

```
redlist_example
```

Format

A data frame with 20 rows and 4 variables:

assessmentId Red List identifier

scientificName Taxon name.

redlistCategory Red List threat category

authority Taxon author/s.

Source

Downloaded and sampled from https://www.iucnredlist.org/

taxonomic_mapping *Data for mapping plant family to order or higher classification*

Description

A dataset containing the higher classification (Angiosperms, Gymnosperms, Ferns and Lycophytes) and Order for each family in the WCVP.

Usage

```
taxonomic_mapping
```

Format

A data frame with 457 rows and 3 variables: `family`, `order` and `higher_classification`

Source

Fern and lycophyte taxonomy from PPG I (2016; doi:10.1111/jse.12229). Angiosperm taxonomy from APG IV (2016; doi:10.1111/boj.12385). Gymnosperm taxonomy from Forest et al (2018; doi:10.1038/s41598018243654)

wcvp_checklist *Generate a species checklist from WCVP*

Description

Generate a species checklist from WCVP

Usage

```
wcvp_checklist(
  taxon = NULL,
  taxon_rank = c("species", "genus", "family", "order", "higher"),
  area_codes = NULL,
  synonyms = TRUE,
  render_report = FALSE,
  native = TRUE,
  introduced = TRUE,
  extinct = TRUE,
  location_doubtful = TRUE,
  hybrids = FALSE,
  infraspecies = TRUE,
  report_filename = NULL,
  report_dir = NULL,
  report_type = c("alphabetical", "taxonomic"),
  wcvp_names = NULL,
  wcvp_distributions = NULL
)
```

Arguments

taxon	Character. Taxon to be included. Defaults to NULL (no taxonomic filter; all taxa).
taxon_rank	Character. One of "species", "genus", "family", "order" or "higher", giving the rank of the value/s in taxon. Must be specified unless taxon is NULL.
area_codes	Character. One or many WGSPRD level 3 region codes. Defaults to NULL (global).
synonyms	Logical. Include synonyms in checklist (see Details)? Defaults to TRUE.
render_report	Logical. Render the checklist as a markdown report? Defaults to FALSE.
native	Logical. Include species occurrences not flagged as introduced, extinct or doubtful? Defaults to TRUE.
introduced	Logical. Include species occurrences flagged as introduced? Defaults to TRUE.
extinct	Logical. Include species occurrences flagged as extinct? Defaults to TRUE.
location_doubtful	Logical. Include species occurrences flagged as location_doubtful? Defaults to TRUE.

`hybrids` Logical. Include hybrid species in checklist? Defaults to `FALSE`.

`infraspecies` Logical. Include hybrid species in checklist? Defaults to `TRUE`.

`report_filename`
Character. Name for the HTML file. Defaults to taxon_area_type.html

`report_dir` Character. Directory for the HTML file to be saved in. Must be provided by user.

`report_type` Character; one of "alphabetical" (the default) or "taxonomic". Should the generated checklist be sorted alphabetically, or by taxonomic status?

`wcvp_names` A data frame of taxonomic names from WCVP version 7 or later. If `NULL` (the default), names will be loaded from `rWCVPdata::wcvp_names`.

`wcvp_distributions`
A data frame of distributions from WCVP version 7 or later. If `NULL` (the default), distributions will be loaded from `rWCVPdata::wcvp_names`.

Details

The `synonyms` argument can be used to limit names to those that are Accepted. If `synonyms = TRUE` then invalid, illegitimate and other non-accepted names are also included (i.e., the checklist is not limited to names for which `taxon_status == "Synonym"`). Two styles of checklist are supported in `rWCVP` - alphabetical and taxonomic. In an alphabetical checklist, all names are arranged alphabetically with accepted names in bold, and synonyms are followed by their accepted name. For a taxonomic checklist, names are grouped by their accepted names, and synonyms are listed beneath. Both types of checklist include author, publication and distribution information, though note that family headings are only supported in alphabetical checklists (due to the additional grouping requirement of the taxonomic format).

Value

Data frame with filtered data and, if `render_report=TRUE`. a report HTML file.

Examples

```
# These examples take >10 seconds to run and require 'rWCVPdata'

if(requireNamespace("rWCVPdata")){
wcvp_checklist(taxon = "Myrtaceae", taxon_rank = "family", area = get_wgsrpd3_codes("Brazil"))
wcvp_checklist(taxon = "Ferns", taxon_rank = "higher", area = get_wgsrpd3_codes("New Zealand")) %>%
head()
}
```

`wcvp_distribution` *Generate spatial distribution objects for species, genera or families*

Description

Generate spatial distribution objects for species, genera or families

Usage

```
wcvp_distribution(
  taxon,
  taxon_rank = c("species", "genus", "family", "order", "higher"),
  native = TRUE,
  introduced = TRUE,
  extinct = TRUE,
  location_doubtful = TRUE,
  wcvp_names = NULL,
  wcvp_distributions = NULL
)
```

Arguments

`taxon`	Character. The taxon to be mapped. Must be provided.
`taxon_rank`	Character. One of "species", "genus", "family", "order" or "higher", giving the rank of the value in `taxon`.
`native`	Logical. Include native range? Defaults to `TRUE`.
`introduced`	Logical. Include introduced range? Defaults to `TRUE`.
`extinct`	Logical. Include extinct range? Defaults to `TRUE`.
`location_doubtful`	Logical. Include occurrences that are thought to be doubtful? Defaults to `TRUE`.
`wcvp_names`	A data frame of taxonomic names from WCVP version 7 or later. If `NULL` (the default), names will be loaded from `rWCVPdata::wcvp_names`.
`wcvp_distributions`	A data frame of distributions from WCVP version 7 or later. If `NULL` (the default), distributions will be loaded from `rWCVPdata::wcvp_names`.

Details

Where `taxon_rank` is higher than species, the distribution of the whole group will be returned, not individual species within that group. This also applies when toggling options - for example, introduced occurrences will only be included if they are outside the native range, regardless of whether `native=TRUE` or `native=FALSE`. To identify extinctions, introductions or doubtful occurrences within the native range, the `wcvp_summary` and `wcvp_occ_mat` functions can be used.

Value

Simple features (sf) data frame containing the range polygon/s of the taxon.

Examples

```
 # this example requires 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
r <- wcvp_distribution("Callitris", taxon_rank = "genus")
p <- wcvp_distribution_map(r)
p
}
```

`wcvp_distribution_map` *Plot distribution maps for species, genera or families*

Description

Plot distribution maps for species, genera or families

Usage

```
wcvp_distribution_map(
  range,
  crop_map = FALSE,
  native = TRUE,
  introduced = TRUE,
  extinct = TRUE,
  location_doubtful = TRUE
)
```

Arguments

`range`	Simple features (`sf`) data frame of the type output by `wcvp_distribution()`.
`crop_map`	Logical. Crop map extent to distribution? Defaults to `FALSE`.
`native`	Logical. Include native range? Defaults to `TRUE`.
`introduced`	Logical. Include introduced range? Defaults to `TRUE`.
`extinct`	Logical. Include extinct range? Defaults to `TRUE`.
`location_doubtful`	Logical. Include occurrences that are thought to be doubtful? Defaults to `TRUE`.

Details

The colour scheme mirrors that used by Plants of the World (POWO), where green is native, purple is introduced, red is extinct and orange is doubtful. See Examples for how to use custom colours.

Value

A ggplot2::ggplot of the distribution.

Examples

```
 # these examples require 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
p <- wcvp_distribution_map(wcvp_distribution("Callitris", taxon_rank = "genus"))
p
# now only the native range, and cropped to range extent
p <- wcvp_distribution_map(wcvp_distribution("Callitris", taxon_rank = "genus"),
  introduced = FALSE, crop_map = TRUE
)
p
# now with different colours
# note that this taxon only has native and introduced occurrences, so only two colours are needed
p <- wcvp_distribution_map(wcvp_distribution("Callitris", taxon_rank = "genus"))
p +
  # for polygons
  ggplot2::scale_fill_manual(values = c("red", "blue")) +
  # for points (islands)
  ggplot2::scale_colour_manual(values = c("red", "blue"))
  }
```

---

wcvp_match_exact *Exact matching to WCVP.*

Description

Exact matching of names to the WCVP, optionally using the author string to refine results.

Usage

```
wcvp_match_exact(names_df, wcvp_names, name_col, author_col = NULL, id_col)
```

Arguments

names_df	Data frame of names for matching.
wcvp_names	Data frame of taxonomic names from WCVP version 7 or later. If NULL (the default), names will be loaded from rWCVPdata::wcvp_names.
name_col	Character. The column in names_df that has the taxon name for matching.
author_col	the column in names_df that has the name authority, to aid matching. Set to NULL to match with no author string.
id_col	the column in names_df that has the observation id.

Value

Match results from WCVP bound to the original data from `names_df`.

See Also

Other name matching functions: `wcvp_match_fuzzy()`, `wcvp_match_names()`

Examples

```
 # these examples require 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
wcvp_names <- rWCVPdata::wcvp_names

# including author string
wcvp_match_exact(redlist_example, wcvp_names, "scientificName",
  author_col = "authority",
  id_col = "assessmentId"
)

# without author string
wcvp_match_exact(redlist_example, wcvp_names, "scientificName", id_col = "assessmentId")
}
```

`wcvp_match_fuzzy` *Fuzzy (approximate) matching to the WCVP.*

Description

Fuzzy matching to names in the WCVP using phonetic matching and edit distance. The WCVP can be loaded for matching from rWCVPdata::wcvp_names.

Usage

```
wcvp_match_fuzzy(names_df = NULL, wcvp_names, name_col, progress_bar = TRUE)

phonetic_match(names_df, wcvp_names, name_col)

edit_match(names_df, wcvp_names, name_col)
```

Arguments

`names_df`	Data frame of names for matching.
`wcvp_names`	Data frame of taxonomic names from WCVP version 7 or later. If `NULL` (the default), names will be loaded from `rWCVPdata::wcvp_names`.
`name_col`	Character. The column in `names_df` that has the taxon name for matching.
`progress_bar`	Logical. Show progress bar when matching? Defaults to `TRUE`; should be changed to `FALSE` if used in a markdown report.

Details

The `wcvp_match_fuzzy` function uses phonetic matching first and then finds the closest match based on edit distance for any remaining names.

Phonetic matching uses phonics::metaphone encoding with a maximum code length of 20.

Edit distance matching finds the closest match based on Levenshtein similarity, calculated by RecordLinkage::levenshteinSim.

Value

Match results from WCVP bound to the original data from `names_df`.

See Also

Other name matching functions: `wcvp_match_exact()`, `wcvp_match_names()`

Examples

```
 # this example requires 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
wcvp_names <- rWCVPdata::wcvp_names
wcvp_match_fuzzy(redlist_example, wcvp_names, "scientificName")
}


 # this example requires 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
wcvp_names <- rWCVPdata::wcvp_names
phonetic_match(redlist_example, wcvp_names, "scientificName")
}


 # this example requires 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
wcvp_names <- rWCVPdata::wcvp_names
edit_match(redlist_example, wcvp_names, "scientificName")
}
```

`wcvp_match_names` *Match names to the WCVP.*

Description

Match names to WCVP, first using exact matching and then using fuzzy matching on any remaining unmatched names.

Usage

```
wcvp_match_names(
  names_df = NULL,
  wcvp_names = NULL,
  name_col = NULL,
  id_col = NULL,
  author_col = NULL,
  join_cols = NULL,
  fuzzy = TRUE,
  progress_bar = TRUE
)
```

Arguments

names_df	Data frame of names for matching.
wcvp_names	Data frame of taxonomic names from WCVP version 7 or later. If NULL (the default), names will be loaded from rWCVPdata::wcvp_names.
name_col	Character. The column in names_df that has the taxon name for matching.
id_col	Character. A column in names_df with a unique ID for each name. Will be created from the row number if not provided.
author_col	the column in names_df that has the name authority, to aid matching. Set to NULL to match with no author string.
join_cols	Character. A vector of name parts to make the taxon name, if name_col is not provided.
fuzzy	Logical; whether or not fuzzy matching should be used for names that could not be matched exactly.
progress_bar	Logical. Show progress bar when matching? Defaults to TRUE; should be changed to FALSE if used in a markdown report.

Details

By default, exact matching uses only the taxon name (supplied by name_col) unless a column specifying the author string is provided (as author_col).

Columns setting out name parts can be supplied as join_cols in place of a taxon name, but must be supplied in the order you want them joined (e.g. c("genus", "species", "infra_rank", "infra")).

Fuzzy matching uses a combination of phonetic and edit distance matching, and can optionally be turned off using fuzzy=FALSE.

The WCVP can be loaded for matching from rWCVPdata::wcvp_names.

See here for an example workflow.

Value

Match results from WCVP bound to the original data from names_df.

See Also

Other name matching functions: wcvp_match_exact(), wcvp_match_fuzzy()

Examples

```
 # these examples require 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
wcvp_names <- rWCVPdata::wcvp_names

# without author
wcvp_match_names(redlist_example, wcvp_names,
  name_col = "scientificName",
  id_col = "assessmentId"
)

# with author
wcvp_match_names(redlist_example, wcvp_names,
  name_col = "scientificName",
  id_col = "assessmentId", author_col = "authority"
)
}
```

---

wcvp_occ_mat *Generate occurrence matrix for taxa and areas*

Description

Generate occurrence matrix for taxa and areas

Usage

```
wcvp_occ_mat(
  taxon = NULL,
  taxon_rank = c("species", "genus", "family", "order", "higher"),
  area_codes = NULL,
  native = TRUE,
  introduced = TRUE,
  extinct = TRUE,
  location_doubtful = TRUE,
  wcvp_names = NULL,
  wcvp_distributions = NULL
)
```

Arguments

taxon	Character. One or many taxa to be included. Defaults to NULL (all species)
taxon_rank	Character. One of "species", "genus", "family", "order" or "higher", giving the rank of the value/s in taxon. Must be specified unless taxon is NULL.
area_codes	Character. One or many WGSPRD level 3 region codes. Defaults to NULL (global).
native	Logical. Include species occurrences not flagged as introduced, extinct or doubtful? Defaults to TRUE.
introduced	Logical. Include species occurrences flagged as introduced? Defaults to TRUE.
extinct	Logical. Include species occurrences flagged as extinct? Defaults to TRUE.
location_doubtful	Logical. Include species occurrences flagged as location doubtful? Defaults to TRUE.
wcvp_names	A data frame of taxonomic names from WCVP version 7 or later. If NULL, names will be loaded from rWCVPdata::wcvp_names.
wcvp_distributions	A data frame of distributions from WCVP version 7 or later. If NULL, distributions will be loaded from rWCVPdata::wcvp_names.

Details

See here for an example of how this output can be formatted for publication.

Value

A data.frame containing the taxon_name and plant_name_id of all species that are present in the area, plus one variable for each WGSPRD level 3 region in area, with species presences marked as 1 and absences marked as 0.

Examples

```
 # this example requires 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
wcvp_occ_mat(
  taxon = "Poa", taxon_rank = "genus",
  area = c("TAS", "VIC", "NSW"), introduced = FALSE
)
}
```

wcvp_summary *Generate a summary table from the WCVP*

Description

Generate a summary table from the WCVP

Usage

```
wcvp_summary(
  taxon = NULL,
  taxon_rank = c("species", "genus", "family", "order", "higher"),
  area_codes = NULL,
  grouping_var = c("area_code_l3", "genus", "family", "order", "higher"),
  hybrids = FALSE,
  wcvp_names = NULL,
  wcvp_distributions = NULL
)
```

Arguments

taxon	Character. Taxon to be included. Defaults to NULL (no taxonomic filter; all taxa).
taxon_rank	Character. One of "genus", "family", "order" or "higher", giving the rank of the value/s in taxon. Must be specified unless taxon is NULL.
area_codes	Character. One or many WGSPRD level 3 region codes. Defaults to NULL (global).
grouping_var	Character; one of "area_code_l3", "genus", "family","order" or "higher" specifying how the summary should be arranged. Defaults to area_code_l3.
hybrids	Logical. Include hybrid species in counts? Defaults to FALSE.
wcvp_names	A data frame of taxonomic names from WCVP version 7 or later. If NULL, names will be loaded from rWCVPdata::wcvp_names.
wcvp_distributions	A data frame of distributions from WCVP version 7 or later. If NULL, distributions will be loaded from rWCVPdata::wcvp_names.

Details

Valid values for rank 'higher' are 'Angiosperms', 'Gymnosperms', 'Ferns' and 'Lycophytes'. Note that grouping variable (if taxonomic) should be of a lower level than taxon and taxon_rank to produce a meaningful summary (i.e., it does not make sense to group a genus by genus, family or higher classification). Additionally, if the grouping variable is taxonomic then species occurrences are aggregated across the input area. This means that if a species is native to any of the input area (even if it is introduced or extinct in other parts) it is counted as 'Native'. Similarly, introduced occurrences take precedence over extinct occurrences. Note that in this type of summary table, 'Endemic' means endemic to the input area, not necessarily to a single WGSRPD Level 3 Area within the input area.

Value

Data.frame with filtered data, or a `gt` table

Examples

```
 # this example requires 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
ferns <- wcvp_summary("Ferns", "higher", get_wgsrpd3_codes("New Zealand"), grouping_var = "family")
wcvp_summary_gt(ferns)
}
```

`wcvp_summary_gt` *Render a summary table from wcvp_summary*

Description

Render a summary table from wcvp_summary

Usage

```
wcvp_summary_gt(x)
```

Arguments

`x`	List.

Value

gt table

Examples

```
 # this example requires 'rWCVPdata'
if(requireNamespace("rWCVPdata")){
ferns <- wcvp_summary("Ferns", "higher", get_wgsrpd3_codes("New Zealand"), grouping_var = "family")
wcvp_summary_gt(ferns)
}
```

wgsrpd_mapping *Data for mapping WGSRPD geography to other levels*

Description

A dataset containing the area (Level 3), #' region (Level 2), continent (Level 1), country (political) and hemisphere category for each Level 3 area. Country mapping follows Gallagher et al. (2020).

Usage

```
wgsrpd_mapping
```

Format

A data frame with 370 rows and 7 variables:

HEMISPHERE Northern, Southern or Equatorial (spanning the equator).

LEVEL1_COD Continent code.

LEVEL1_NAM Continent.

LEVEL2_COD Region code.

LEVEL2_NAM Region.

COUNTRY Country (political; from Gallagher et al., 2020)

LEVEL3_COD Area code.

LEVEL3_NAM Area.

Source

Modified from data available at https://github.com/tdwg/wgsrpd

wgsrpd3 *Biodiversity Information Standards (TDWG) World Geographical Scheme for Recording Plant Distributions (WGSRPD)*

Description

Spatial data for WGSRPD Level 3, for plotting maps

Usage

```
wgsrpd3
```

Format

An 'sf' object with 20 rows and 4 variables:

LEVEL3_NAM Region name

LEVEL3_COD Region code

LEVEL2_COD Level 2 code

LEVEL1_COD Level 1 code (continent)

geometry sf geometry

fillcol Used for mapping.

Source

https://github.com/tdwg/wgsrpd/tree/master/level3

Index

∗ **datasets**
redlist_example, 6
taxonomic_mapping, 6
wgsrpd3, 19
wgsrpd_mapping, 19
∗ **name matching functions**
wcvp_match_exact, 11
wcvp_match_fuzzy, 12
wcvp_match_names, 13

discrete_scale, *5*

edit_match *(*wcvp_match_fuzzy*)*, 12

get_area_name, 2
get_wgsrpd3_codes, 3
ggplot2::ggplot, *11*
gt, *18*
guides(), *5*

lambda, *5*

phonetic_match *(*wcvp_match_fuzzy*)*, 12
phonics::metaphone, *13*
powo_map, 4
powo_pal, 4

RecordLinkage::levenshteinSim, *13*
redlist_example, 6

scale_color_powo *(*powo_pal*)*, 4
scale_colour_powo *(*powo_pal*)*, 4
scale_fill_powo *(*powo_pal*)*, 4

taxonomic_mapping, 6

wcvp_checklist, 7
wcvp_distribution, 9
wcvp_distribution(), *10*
wcvp_distribution_map, 10
wcvp_match_exact, 11
wcvp_match_exact(), *13*, *15*
wcvp_match_fuzzy, 12
wcvp_match_fuzzy(), *12*, *15*
wcvp_match_names, 13
wcvp_match_names(), *12, 13*
wcvp_occ_mat, 15
wcvp_summary, 17, *18*
wcvp_summary_gt, 18
wgsrpd3, 19
wgsrpd_mapping, 19