

Package ‘funbiogeo’

July 29, 2026

Type Package

Title Streamlining Functional Biogeography Analyses

Version 0.0.1

Description Helps users with analyses in functional biogeography by loading and combining data, computing trait coverage, as well as computing functional diversity indices, drawing maps, correlating them with the environment, and upscaling assemblages.

URL <https://frbcesab.github.io/funbiogeo/>,
<https://github.com/frbcesab/funbiogeo>

BugReports <https://github.com/frbcesab/funbiogeo/issues>

License GPL (>= 2)

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

Imports ggplot2 (>= 3.4.0), grid, rlang, rnatualearth, scales, sf,
stats, terra, tidyr, tidyselect, tools, utils, xfun

Suggests dplyr, fs, fundiversity, ggridges, knitr, mFD, mockery,
patchwork, rmarkdown, rstudioapi, testthat (>= 3.0.0), vdiff, withr

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Nicolas Casajus [aut, cre, cph] (ORCID:
<<https://orcid.org/0000-0002-5537-5294>>),
Matthias Grenié [aut] (ORCID: <<https://orcid.org/0000-0002-4659-7522>>)

Maintainer Nicolas Casajus <nicolas.casajus@fondationbiodiversite.fr>

Repository CRAN

Date/Publication 2026-07-29 16:40:19 UTC

Contents

fb_aggregate_site_data	3
fb_count_sites_by_species	4
fb_count_species_by_site	5
fb_count_species_by_trait	5
fb_count_traits_by_species	6
fb_cwm	7
fb_filter_sites_by_species_coverage	8
fb_filter_sites_by_trait_coverage	9
fb_filter_species_by_site_coverage	10
fb_filter_species_by_trait_coverage	11
fb_filter_traits_by_species_coverage	12
fb_format_site_locations	13
fb_format_site_species	14
fb_format_species_categories	15
fb_format_species_traits	16
fb_get_all_trait_coverages_by_site	17
fb_get_environment	18
fb_get_trait_combination_coverage	18
fb_get_trait_coverage_by_site	20
fb_make_report	21
fb_map_raster	23
fb_map_site_data	24
fb_map_site_traits_completeness	25
fb_plot_distribution_site_trait_coverage	26
fb_plot_number_sites_by_species	27
fb_plot_number_species_by_trait	28
fb_plot_number_traits_by_species	29
fb_plot_site_environment	30
fb_plot_site_traits_completeness	31
fb_plot_species_traits_completeness	32
fb_plot_species_traits_missingness	33
fb_plot_trait_combination_frequencies	34
fb_plot_trait_correlation	35
fb_table_trait_summary	36
woodiv_categories	37
woodiv_locations	37
woodiv_site_species	38
woodiv_traits	39

Index

40

fb_aggregate_site_data

Aggregate Site Data Along Coarser Spatial Grid

Description

This function helps aggregating site data along a coarser grid. The idea is that you have any type of data at the site scale (diversity metrics, environmental data, etc.) but you would like to work on it at a coarser scale, or you want to visualize it at that scale. This function helps you do exactly that.

Usage

```
fb_aggregate_site_data(site_locations, site_data, agg_geom, fun = mean, ...)
```

Arguments

site_locations	a sf object with the spatial geometries of sites. NOTE: the first column should be named "site" and indicate site names.
site_data	a matrix or data.frame containing values per site to aggregate along the provided grid. It can contain one or several columns (variables to aggregate). The first column must contain site names as provided in the first argument site_locations.
agg_geom	a terra::SpatRaster or an sf object. This defines the geometry along which to aggregate the initial data. See more in the Details section.
fun	the function used to aggregate point values when there are multiple points in one cell. Default is mean.
...	additional argument(s) passed to the provided function fun

Details

The agg_geom object will condition the type of object output by the function. It can be of any sort of a SpatRaster or a sf object. Depending on the need, it could be a regular square grid or hexagonal grid; it could also be irregular polygons like biomes or ecoregions, or points, and even lines (such as when aggregating across transects or trajectories).

Value

An object of the same type as the agg_geom input with as many layers (if SpatRaster) or columns (if sf) as columns provided in the input site_data.

Examples

```
library("sf")
## Raster grid
tavg <- system.file(
  "extdata", "annual_mean_temp.tif", package = "funbiogeo"
)
```

```

tavg <- terra::rast(tavg)

# Rasterize 3 first species counts
fb_aggregate_site_data(
  head(wooddiv_locations), wooddiv_site_species[, 1:4], tavg, fun = sum
)

## Irregular polygons
countries <- readRDS(system.file(
  "extdata", "countries_sf.rds", package = "funbiogeo"
))
# Aggregate occurrence per country
fb_aggregate_site_data(
  head(wooddiv_locations, n = 20), wooddiv_site_species[, 1:4], countries,
  fun = sum
)

```

fb_count_sites_by_species

Count Number of Sites Occupied by Species

Description

This function computes the number and proportion of sites occupied by each species (distribution value higher than 0 and non-NA).

Usage

```
fb_count_sites_by_species(site_species)
```

Arguments

site_species a data.frame with sites in rows and species in columns. **NOTE:** the first column should be named "site" and indicate site names. The other columns should be named according to species names.

Value

A three-column data.frame with:

- **species:** the name of the species;
- **n_sites:** the number of sites where the species is present;
- **coverage:** the percentage of sites where the species is present.

Examples

```

site_coverage_by_species <- fb_count_sites_by_species(wooddiv_site_species)
head(site_coverage_by_species)

```

`fb_count_species_by_site`*Count Number of Species per Site*

Description

This function computes for each site the number and proportion of species present (distribution value higher than 0 and non-NA) compared to all species provided. For example, a site could contain only 20% of all species provided.

Usage

```
fb_count_species_by_site(site_species)
```

Arguments

`site_species` a data.frame with sites in rows and species in columns. **NOTE:** the first column should be named "site" and indicate site names. The other columns should be named according to species names.

Value

A three-column data.frame with:

- `site`: the name of the site;
- `n_species`: the number of present species;
- `coverage`: the percentage of present species.

Examples

```
species_coverage_by_site <- fb_count_species_by_site(wooddiv_site_species)
head(species_coverage_by_site)
```

`fb_count_species_by_trait`*Count Number of Species for Each Trait*

Description

This function computes, for each trait, the number and proportion of species without missing trait value (NA).

Usage

```
fb_count_species_by_trait(species_traits)
```

Arguments

`species_traits` a `data.frame` with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

Value

A three-column `data.frame` with:

- `trait`: the name of the trait;
- `n_species`: the number of species with non-missing value for the trait;
- `coverage`: the percentage of species with non-missing value for the trait.

Examples

```
species_coverage_by_trait <- fb_count_species_by_trait(wooddiv_traits)
head(species_coverage_by_trait)
```

```
fb_count_traits_by_species
```

Compute Number of Known Trait(s) per Species

Description

For each species, this function computes the number and proportion of traits without NA (missing trait values).

Usage

```
fb_count_traits_by_species(species_traits)
```

Arguments

`species_traits` a `data.frame` with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

Value

A three-column `data.frame` with:

- `species`: the name of the species;
- `n_traits`: the number of traits with non-missing value for the species;
- `coverage`: the percentage of traits with non-missing value for the species.

Examples

```
trait_coverage_by_species <- fb_count_traits_by_species(wooddiv_traits)
head(trait_coverage_by_species)
```

fb_cwm

*Compute community-weighted means (CWM) of trait values***Description**

This function returns the community-weighted mean of provided trait values. It only works with quantitative traits and will warn you otherwise. It will remove species that either have NA values in the `site_species` input or NA values as their trait in the provided trait object.

Usage

```
fb_cwm(site_species, species_traits)
```

Arguments

`site_species` a data.frame with sites in rows and species in columns. **NOTE:** the first column should be named "site" and indicate site names. The other columns should be named according to species names.

`species_traits` a data.frame with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

Details

The community-weighted mean is a site-based trait mean weighted by the abundance of the species. It can be written with the following equation:

$$CWM_k = \sum_{i=1}^S p_{ik} \times t_{ik}$$

with CWM_k the CWM of site k , p_{ik} the relative abundance of species i in site k , and t_{ik} the trait of species i in site k .

Value

A data.frame with sites in rows and the following variables:

- `site`, the site label,
- `trait`, the trait label as provided in `species_traits`,
- and `cwm`, the community-weighted means of quantitative traits values.

Examples

```
site_cwm <- fb_cwm(head(wooddiv_site_species), wooddiv_traits)
head(site_cwm)
```

`fb_filter_sites_by_species_coverage`*Filter sites with a given species coverage threshold*

Description

Selects sites (rows) for which the proportion of species present (distribution value higher than 0 and non-NA) is higher than the user-defined threshold.

Usage

```
fb_filter_sites_by_species_coverage(  
  site_species,  
  threshold_species_proportion = 0  
)
```

Arguments

`site_species` a data.frame with sites in rows and species in columns. **NOTE:** the first column should be named "site" and indicate site names. The other columns should be named according to species names.

`threshold_species_proportion` a numeric of length 1 between 0 and 1. The threshold of species coverage under which to exclude the sites.

Value

A subset of `site_species` with sites with at least `threshold_species_proportion%` species present.

Examples

```
# Get sites with more than 40% of the species  
new_site_species <- fb_filter_sites_by_species_coverage(  
  woodiv_site_species,  
  threshold_species_proportion = 0.4  
)  
# There are now only 148 sites  
dim(new_site_species)  
new_site_species[1:3, 1:4]
```

`fb_filter_sites_by_trait_coverage`*Filter sites with a given trait coverage threshold*

Description

Select sites (rows of the `site_species` object) with all given traits available for at least the user-defined proportion of species `threshold_traits_proportion`. If a single trait is given, then the threshold applies to a single trait, if more than one trait is provided, then the function considers a threshold across all traits taken together.

Usage

```
fb_filter_sites_by_trait_coverage(  
  site_species,  
  species_traits,  
  threshold_traits_proportion = 1  
)
```

Arguments

`site_species` a `data.frame` with sites in rows and species in columns. **NOTE:** the first column should be named "site" and indicate site names. The other columns should be named according to species names.

`species_traits` a `data.frame` with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

`threshold_traits_proportion`
a numeric of length 1 between 0 and 1. The percentage trait coverage threshold

Value

A subset of `site_species` with sites covered by X% of abundance/coverage considering all provided traits.

Examples

```
# Filter all the sites where all species have known traits  
new_site_species <- fb_filter_sites_by_trait_coverage(  
  wooddiv_site_species, wooddiv_traits  
)  
  
# There is only one such site  
nrow(new_site_species)  
  
# Filter sites where at least 80% of species have known traits  
new_site_species_2 <- fb_filter_sites_by_trait_coverage(  
  wooddiv_site_species, wooddiv_traits, threshold_traits_proportion = 0.8
```

```
)

# There are now four sites
nrow(new_site_species_2)
```

```
fb_filter_species_by_site_coverage
```

Filter species with a given sites coverage threshold

Description

Selects species (columns) for which the percentage of sites where the species is present (distribution value higher than 0 and non-NA) is higher than a user-defined threshold.

Usage

```
fb_filter_species_by_site_coverage(
  site_species,
  threshold_sites_proportion = 0
)
```

Arguments

site_species a data.frame with sites in rows and species in columns. **NOTE:** the first column should be named "site" and indicate site names. The other columns should be named according to species names.

threshold_sites_proportion a numeric of length 1 between 0 and 1. The percentage of sites coverage threshold.

Value

A subset of **site_species** with species with a prevalence higher than **threshold_sites_proportion**.

Examples

```
# Filter species present in at least 10% of the sites
new_site_species <- fb_filter_species_by_site_coverage(
  woodiv_site_species,
  threshold_sites_proportion = 0.1
)

new_site_species[1:3, 1:4]

# There are now only 84 species (instead of 149)
ncol(new_site_species)
```

`fb_filter_species_by_trait_coverage`*Filter species with a given traits coverage threshold*

Description

Selects species (rows) for which the proportion of traits without NA (missing trait values) is higher than the user-defined threshold. It considers as many traits as the ones provided to filter given the threshold.

Usage

```
fb_filter_species_by_trait_coverage(  
  species_traits,  
  threshold_traits_proportion = 0  
)
```

Arguments

`species_traits` a data.frame with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

`threshold_traits_proportion`
a numeric of length 1 between 0 and 1. The percentage of traits coverage threshold.

Value

A subset of `species_traits` with species covered by X% of traits.

Examples

```
# Filter species that have at least 60% of the traits described  
new_species_traits <- fb_filter_species_by_trait_coverage(  
  wooddiv_traits,  
  threshold_traits_proportion = 0.6  
)  
  
# There are now only 93 species  
nrow(new_species_traits)
```

`fb_filter_traits_by_species_coverage`*Filter traits with a given species coverage threshold*

Description

Selects traits (columns) for which the percentage of species without NA (missing trait values) is higher than a user-defined threshold.

Usage

```
fb_filter_traits_by_species_coverage(  
  species_traits,  
  threshold_species_proportion = 0  
)
```

Arguments

`species_traits` a data.frame with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

`threshold_species_proportion`
numeric(1) [default = NULL]
between 0 and 1. The percentage of species coverage threshold.

Value

A subset of `species_traits` with traits for the specified proportion of species.

Examples

```
# Filter traits that have at least 60% non-missing values  
new_species_traits <- fb_filter_traits_by_species_coverage(  
  woodiv_traits,  
  threshold_species_proportion = 0.6  
)  
  
# There are now only 2 traits  
ncol(new_species_traits)
```

fb_format_site_locations

Extract site x locations information from long format data

Description

Convert a flat `data.frame` with site coordinates into a proper `sf` object that can then be used by other functions. This function assumes by default that the coordinates are given in WGS84 (longitude vs. latitude), but this can be changed through the `crs` argument. The function automatically removes repeated coordinates from the input dataset.

Usage

```
fb_format_site_locations(
  data,
  site,
  longitude,
  latitude,
  crs = sf::st_crs(4326),
  na_rm = FALSE
)
```

Arguments

<code>data</code>	a <code>data.frame</code> in a long format (see example).
<code>site</code>	a character of length 1. Name of the column with site labels.
<code>longitude</code>	a character of length 1. Name of the column with longitude. The function assumes coordinates are WGS84 (EPSG:4326).
<code>latitude</code>	a character of length 1. Name of the column with latitude. The function assumes coordinates are WGS84 (EPSG:4326).
<code>crs</code>	a character of length 1 or an object of class <code>crs</code> . Coordinate Reference System (CRS) of the specified coordinates. The CRS should be a valid CRS in R . It can either be a character like <code>"+proj=longlat +datum=WGS84 +no_defs"</code> or as specified using <code>sf::st_crs()</code> like <code>sf::st_crs(4326)</code> the default value.
<code>na_rm</code>	a logical value. If <code>TRUE</code> remove sites with incomplete coordinates. Default is <code>FALSE</code> .

Value

An `sf` object with a `site` column specifying site coordinates.

Examples

```
filename <- system.file(
  "extdata", "wooddiv_raw_data.csv",
  package = "funbiogeo"
```

```

)
all_data <- read.csv(filename)

head(all_data)

site_locations <- fb_format_site_locations(
  all_data, "site", "longitude",
  "latitude"
)
head(site_locations)

```

fb_format_site_species

Extract site x species information from long format data

Description

Convert a flat `data.frame` with species occurrence/abundance at site level into a proper `data.frame` object that can then be used by other functions. The final output contains sites in rows and species in columns.

Usage

```
fb_format_site_species(data, site, species, value, na_to_zero = TRUE)
```

Arguments

<code>data</code>	a <code>data.frame</code> in a long format (see example).
<code>site</code>	a character of length 1. Name of the column with site labels.
<code>species</code>	a character of length 1. Name of the column with species names.
<code>value</code>	a character of length 1. Name of the column with species occurrence/abundance.
<code>na_to_zero</code>	a logical value. If TRUE (default) NA are replaced by 0.

Value

A `data.frame` with sites in rows and species in columns. The first column is named "site" and contains the name of the sites.

Examples

```

filename <- system.file(
  "extdata", "wooddiv_raw_data.csv",
  package = "funbiogeo"
)
all_data <- read.csv(filename)

head(all_data)

```

```
site_species <- fb_format_site_species(all_data, "site", "species", "count")
site_species[1:3, 1:4]
```

fb_format_species_categories

Extract species x categories information from long format data

Description

Convert a flat data.frame with species names and species (supra-)category (e.g. family, order, endemism status, etc.) into a proper species_categories object that can then be used by other functions. The final output contains species in rows and two columns (species name and species category).

Usage

```
fb_format_species_categories(data, species, category)
```

Arguments

data	a data.frame in a long format (see example).
species	a character of length 1. Name of the column with species names.
category	a character of length 1. Name of the column with species category.

Value

A data.frame with species in rows and two columns (species name and species category).

Examples

```
filename <- system.file(
  "extdata", "wooddiv_raw_data.csv",
  package = "funbiogeo"
)
all_data <- read.csv(filename)

head(all_data)

species_categories <- fb_format_species_categories(
  all_data, "species", "genus"
)
species_categories[1:6, ]
```

`fb_format_species_traits`*Extract species x traits information from long format data*

Description

Convert a flat data.frame with traits values for different species into a proper species_traits object that can then be used by other functions. The final output contains species in rows and traits in columns.

Usage

```
fb_format_species_traits(data, species, traits)
```

Arguments

data	a data.frame in a long format (see example).
species	a character of length 1. Name of the column with species names.
traits	a character of length ≥ 1 . Name(s) of trait column(s).

Value

A data.frame with species in rows and traits in columns, with the first column names "species" containing the species names.

Examples

```
filename <- system.file(
  "extdata", "wooddiv_raw_data.csv", package = "funbiogeo"
)
all_data <- read.csv(filename)

head(all_data)

traits <- c("plant_height", "seed_mass", "sla", "wood_density")

species_traits <- fb_format_species_traits(all_data, "species", traits)
head(species_traits)
```

`fb_get_all_trait_coverages_by_site`*Compute Trait Coverage per Site for Each Trait*

Description

Compute trait coverage for all sites, i.e., the proportion of total abundance/presence of species that have traits data compared to total species. This function assumes that all species provided in the traits dataset have all their traits specified (meaning that all species have either known or NA values reported as their traits). The coverage of each trait separately is returned as well as all traits taken together if wanted.

Usage

```
fb_get_all_trait_coverages_by_site(  
  site_species,  
  species_traits,  
  all_traits = TRUE  
)
```

Arguments

<code>site_species</code>	a data.frame with sites in rows and species in columns. NOTE: the first column should be named "site" and indicate site names. The other columns should be named according to species names.
<code>species_traits</code>	a data.frame with species in rows and traits as columns. NOTE: The first column should be named "species" and contain species names. The other columns should be named according to trait names.
<code>all_traits</code>	a logical (default = TRUE) which tell if the coverage considering all provided traits should be provided in an additional column <code>all_traits</code>

Value

a data.frame with a column with sites and one column per provided trait giving its coverage (percent species per site, weighted by abundance that have trait data), and, when argument `all_traits = TRUE`, an additional column named `all_traits` considering the coverage of all traits taken together.

Examples

```
site_trait_cov <- fb_get_all_trait_coverages_by_site(  
  wooddiv_site_species, wooddiv_traits  
)  
  
head(site_trait_cov)
```

fb_get_environment	<i>Extract Raster Values at Location of Sites</i>
--------------------	---

Description

This function uses a `site_locations` object and a given `SpatRaster`. It extracts the mean value of the provided raster for each site.

Usage

```
fb_get_environment(site_locations, environment_raster)
```

Arguments

`site_locations` a `sf` object with the spatial geometries of sites. **NOTE:** the first column should be named "site" and indicate site names.

`environment_raster` a `SpatRaster` object (package `terra`). A single or multi-layers environmental raster.

Value

A `data.frame` with average environmental values (columns) per site (rows), with the first column being "site", indicating site names.

Examples

```
## Import climate rasters ----
prec <- system.file("extdata", "annual_tot_prec.tif", package = "funbiogeo")
tavg <- system.file("extdata", "annual_mean_temp.tif", package = "funbiogeo")

layers <- terra::rast(c(tavg, prec))

fb_get_environment(head(wooddiv_locations), layers)
```

fb_get_trait_combination_coverage	<i>Compute site trait coverage for each trait combination</i>
-----------------------------------	---

Description

This function computes trait coverage for each site for different trait. The function will not run if the total number of combinations given is over 10,000.

Usage

```
fb_get_trait_combination_coverage(  
  site_species,  
  species_traits,  
  comb_size = NULL  
)
```

Arguments

- site_species** a data.frame with sites in rows and species in columns. **NOTE:** the first column should be named "site" and indicate site names. The other columns should be named according to species names.
- species_traits** a data.frame with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.
- comb_size** an integer vector defining one or more sizes of combinations over which to compute trait coverage. (default: NULL)

Value

a data.frame with the following columns:

- **site** with the site indices from **site_species**,
- **combination_length** with the number of traits in given combinations,
- **combination_name** with the name of the trait combination (concatenated trait names with **---**),
- **trait_coverage** the corresponding trait coverage for the given trait combination and site.

Examples

```
# Compute Coverages using All Trait Combinations  
all_combinations <- fb_get_trait_combination_coverage(  
  wooddiv_site_species, wooddiv_traits  
)  
  
# Get only combinations of 3 traits  
three_traits <- fb_get_trait_combination_coverage(  
  wooddiv_site_species, wooddiv_traits, 3  
)  
  
# Combinations of 2, 3, and 4 traits  
two_to_four <- fb_get_trait_combination_coverage(  
  wooddiv_site_species, wooddiv_traits, 2:4  
)
```

`fb_get_trait_coverage_by_site`*Compute Trait Coverage For Each Site Weighted by Abundance*

Description

Compute trait coverage for all sites, i.e., the proportion of total abundance/presence of species that have traits data compared to all species. This function assumes that all species provided in the trait dataset have all their traits specified (meaning that all species have either known or NA values reported as their traits). **NB:** this function returns trait coverage using all traits provided in the input `species_traits` `data.frame`.

Usage

```
fb_get_trait_coverage_by_site(site_species, species_traits)
```

Arguments

`site_species` a `data.frame` with sites in rows and species in columns. **NOTE:** the first column should be named "site" and indicate site names. The other columns should be named according to species names.

`species_traits` a `data.frame` with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

Value

A `data.frame` with `n` rows (where `n` is the number of sites) and two columns: `site`, the site label, and `trait_coverage`, the percent of total abundance/presence of species that have traits data.

Examples

```
site_trait_cov <- fb_get_trait_coverage_by_site(
  wooddiv_site_species, wooddiv_traits
)

head(site_trait_cov)
```

Description

Creates an R Markdown (.Rmd) report from a template to explore and summarize user data in (functional) biogeography through the use of the site-species, the species-traits, and the site-locations objects. Users can modify this report and use the function `rmarkdown::render()` (or click the *Render* of the RStudio IDE) to convert this .Rmd in different formats:

- HTML document (`output_format = "bookdown::html_document2"`);
- PDF document (`output_format = "bookdown::pdf_document2"`);
- Word document (`output_format = "bookdown::word_document2"`);
- HTML, PDF and Word documents (`output_format = "all"`).

Usage

```
fb_make_report(
  path,
  filename = NULL,
  title = NULL,
  author = NULL,
  species_traits,
  site_species,
  site_locations,
  species_categories = NULL,
  overwrite = FALSE,
  open = TRUE
)
```

Arguments

path	a character of length 1. The directory in which the .Rmd and .rds files will be created. This directory must exist. Note that subdirectories funbiogeo/ and funbiogeo/data/ will be created.
filename	a character of length 1. The name of the .Rmd file to be created. If NULL (default) the .Rmd file will be named from the title (if provided) or funbiogeo_report.Rmd otherwise.
title	a character of length 1. The title of the report. If NULL (default) the title will be named from the title (if provided) or funbiogeo Report otherwise.
author	a character of length 1. The author(s) of the report. If NULL (default) no author will be added.
species_traits	a data.frame with species in rows and traits as columns. NOTE: The first column should be named "species" and contain species names. The other columns should be named according to trait names.

site_species	a data.frame with sites in rows and species in columns. NOTE: the first column should be named "site" and indicate site names. The other columns should be named according to species names.
site_locations	a sf object with the spatial geometries of sites. NOTE: the first column should be named "site" and indicate site names.
species_categories	(default = NULL) 2-columns data.frame giving species categories, with the first column describing the species name, and the second column giving their corresponding categories
overwrite	a logical. If the .Rmd file (or any .rds dataset) is already present and overwrite = TRUE, the .Rmd file (and all .rds files) will be replaced. Default is FALSE.
open	a logical. If TRUE (default), the .Rmd file will be opened in the text editor.

Details

Note that a copy of user data will be saved as .rds files in path/funbiogeo/data/ (where path is the directory defined by the user).

Value

No return value.

Examples

```
if (interactive()) {

# Create temporary folder (optional) ----
temp_path <- tempdir()

# Create report ----
fb_make_report(
  path          = temp_path,
  author        = "Casajus N. and Grenié M.",
  species_traits = woodiv_traits,
  site_species  = woodiv_site_species,
  site_locations = woodiv_locations,
  open          = FALSE
)

# Open Rmd file ----
utils::file.edit(file.path(temp_path, "funbiogeo", "funbiogeo_report.Rmd"))

# Render Rmd file ----
rmarkdown::render(
  input          = file.path(temp_path, "funbiogeo", "funbiogeo_report.Rmd"),
  output_format = "all"
)
}
```

fb_map_raster	<i>Map a Single Raster Layer</i>
---------------	----------------------------------

Description

This is a helper function to plot a map of an environmental raster. The raster is plotted as is, with its given coordinate reference system. The function can provide a background map if the background argument is toggled.

Usage

```
fb_map_raster(x, background = FALSE, ...)
```

Arguments

x	a SpatRaster object (package terra). A raster of one single layer
background	a logical. If TRUE adds a layer of country boundaries from Natural Earth.
...	other parameters passed to <code>ggplot2::theme()</code>

Value

A ggplot object.

Examples

```
library(ggplot2)

## Load raster ----
tavg <- system.file("extdata", "annual_mean_temp.tif", package = "funbiogeo")
tavg <- terra::rast(tavg)

## Default map ----
fb_map_raster(tavg)

## Map with a background ----
fb_map_raster(tavg, background = TRUE)

## Map with custom theme ----
fb_map_raster(tavg, legend.position = "bottom")

## Advanced customization ----
my_map <- fb_map_raster(tavg) +
  scale_fill_distiller("Temperature", palette = "Spectral") +
  theme(legend.position = "bottom") +
  ggtitle("Mean annual temperature in Western Europe")

my_map

## Map w/o annotation ----
```

```
fb_map_raster(tavg) +
  theme_void() +
  theme(legend.position = "none")
```

fb_map_site_data	<i>Map Arbitrary Site Data</i>
------------------	--------------------------------

Description

This function helps to map arbitrary site data using the site-locations object and a dataset organized by site. The returned plot is as little customized as possible to let the user choose. The function can provide a basic background map if the background argument is toggled.

Usage

```
fb_map_site_data(site_locations, site_data, selected_col, background = FALSE)
```

Arguments

site_locations	a sf object with the spatial geometries of sites. NOTE: the first column should be named "site" and indicate site names.
site_data	data.frame() of additional site information containing the column "site" to merge with the site_locations argument
selected_col	character(1) name of the column to plot
background	a logical. If TRUE adds a layer of country boundaries from Natural Earth.

Value

a ggplot object.

Examples

```
site_rich <- fb_count_species_by_site(wooddiv_site_species)

# Map of Species Richness
rich_map <- fb_map_site_data(wooddiv_locations, site_rich, "n_species")
rich_map

# Customize the map
rich_map +
  ggplot2::scale_fill_viridis_c("Species Richness")

# Map w/ a background
fb_map_site_data(
  wooddiv_locations, site_rich, "n_species", background = TRUE
)
```

fb_map_site_traits_completeness
Map Trait Coverage Per Site

Description

Returns a ggplot2 map of sites colored by trait coverage (proportion of species having a known trait value). By default shows one plot for each trait and add an additional facet named "all_traits" considering the trait coverage with all provided traits taken together.

Usage

```
fb_map_site_traits_completeness(
  site_locations,
  site_species,
  species_traits,
  all_traits = TRUE,
  background = FALSE
)
```

Arguments

site_locations	a sf object with the spatial geometries of sites. NOTE: the first column should be named "site" and indicate site names.
site_species	a data.frame with sites in rows and species in columns. NOTE: the first column should be named "site" and indicate site names. The other columns should be named according to species names.
species_traits	a data.frame with species in rows and traits as columns. NOTE: The first column should be named "species" and contain species names. The other columns should be named according to trait names.
all_traits	a logical (default = TRUE) which tell if the coverage considering all provided traits should be provided in an additional column all_traits
background	a logical. If TRUE adds a layer of country boundaries from Natural Earth.

Value

a 'ggplot2' object

Examples

```
# Map without a background
fb_map_site_traits_completeness(
  wooddiv_locations, wooddiv_site_species, wooddiv_traits
)

# Map with a background
fb_map_site_traits_completeness(
```

```
wooddiv_locations, wooddiv_site_species, wooddiv_traits, background = TRUE
)
```

fb_plot_distribution_site_trait_coverage

Plot Distribution of Trait Coverages across all Sites

Description

Plots the distributions of trait coverage across site, i.e. the proportion of species weighted by abundance with known trait values, for each trait separately and all traits taken together. The trait distributions are ordered from the lowest to the highest average trait coverage (top to bottom). The top always displays a distribution named "all_traits" that contains the distribution of coverage all traits taken together.

Usage

```
fb_plot_distribution_site_trait_coverage(
  site_species,
  species_traits,
  species_categories = NULL,
  all_traits = TRUE
)
```

Arguments

site_species	a data.frame with sites in rows and species in columns. NOTE: the first column should be named "site" and indicate site names. The other columns should be named according to species names.
species_traits	a data.frame with species in rows and traits as columns. NOTE: The first column should be named "species" and contain species names. The other columns should be named according to trait names.
species_categories	(default = NULL) 2-columns data.frame giving species categories, with the first column describing the species name, and the second column giving their corresponding categories
all_traits	a logical (default = TRUE) which tell if the coverage considering all provided traits should be provided in an additional column all_traits

Details

Note: this function requires the installation of the ggridges package. If it isn't installed, the function will give an error suggesting installing it through `install.packages("ggridges")`.

Value

a 'ggplot2' object

Examples

```
fb_plot_distribution_site_trait_coverage(wooddiv_site_species, wooddiv_traits)
```

```
fb_plot_number_sites_by_species
```

Plot Number of Sites by Species

Description

Represent all species in each function of the number of sites they occupy. The species are ordered from the ones that occupy the least number of sites from the ones that occupy the most. The number of site is indicated at the bottom x-axis, while the top x-axis represents the proportion of occupied sites. The left y-axis label species names and their rank by increasing prevalence. The user can supplied a threshold of sites to see how many species occupy more or less than the given proportion of sites.

Usage

```
fb_plot_number_sites_by_species(
  site_species,
  threshold_sites_proportion = NULL
)
```

Arguments

site_species a data.frame with sites in rows and species in columns. **NOTE:** the first column should be named "site" and indicate site names. The other columns should be named according to species names.

threshold_sites_proportion a numeric of length 1 between 0 and 1. The percentage of sites coverage threshold.

Value

a ggplot2 object

Examples

```
fb_plot_number_sites_by_species(wooddiv_site_species)

# Add a vertical cutoff line (40% of sites)
fb_plot_number_sites_by_species(wooddiv_site_species, 0.4)
```

`fb_plot_number_species_by_trait`*Plot Number of Species per Trait*

Description

Display a lollipop graph showing the number and proportion of species with non-NA trait for each trait ranked in decreasing order.

Usage

```
fb_plot_number_species_by_trait(  
  species_traits,  
  species_categories = NULL,  
  threshold_species_proportion = NULL  
)
```

Arguments

`species_traits` a `data.frame` with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

`species_categories`
(default = NULL) 2-columns `data.frame` giving species categories, with the first column describing the species name, and the second column giving their corresponding categories

`threshold_species_proportion`
`numeric(1)` [default = NULL]
between 0 and 1. The percentage of species coverage threshold.

Value

a `ggplot2` object

Examples

```
fb_plot_number_species_by_trait(wooddiv_traits)  
  
# Add a vertical cutoff line (12.5% of species)  
fb_plot_number_species_by_trait(wooddiv_traits, NULL, 1/8)
```

`fb_plot_number_traits_by_species`*Plot Number of Traits per Species*

Description

Display a graph showing the number (and proportion) of species having at least 0, 1, etc. number of traits. It provides a lollipop graph to examine which number of trait cover a certain proportion of the species. This plot doesn't show which traits are concerned.

Usage

```
fb_plot_number_traits_by_species(  
  species_traits,  
  species_categories = NULL,  
  threshold_species_proportion = NULL  
)
```

Arguments

`species_traits` a `data.frame` with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

`species_categories`
(default = NULL) 2-columns `data.frame` giving species categories, with the first column describing the species name, and the second column giving their corresponding categories

`threshold_species_proportion`
`numeric(1)` [default = NULL]
between 0 and 1. The percentage of species coverage threshold.

Value

a `ggplot2` object

Examples

```
fb_plot_number_traits_by_species(wooddiv_traits)  
  
# Add a vertical cutoff line (33% of the species)  
fb_plot_number_traits_by_species(  
  wooddiv_traits, threshold_species_proportion = 1/3  
)
```

fb_plot_site_environment

Plot Position of Sites in Environmental Space

Description

Plot a figure showing the average environmental space of given sites compared to a full environmental vector. For the sake of simplicity only represents the figure along two environmental axes. The average environmental value are extracted for each site.

Usage

```
fb_plot_site_environment(
  site_locations,
  environment_raster,
  first_layer = names(environment_raster)[1],
  second_layer = names(environment_raster)[2]
)
```

Arguments

site_locations	a sf object with the spatial geometries of sites. NOTE: the first column should be named "site" and indicate site names.
environment_raster	a SpatRaster object (package terra). A single or multi-layers environmental raster.
first_layer	character(1) the name of the first layer to use, by default uses the first layer of environment_raster
second_layer	character(1) the name of the second layer to use, by default uses the second layer of environment_raster

Value

a ggplot object

Examples

```
# Import climate rasters
prec <- system.file("extdata", "annual_tot_prec.tif", package = "funbiogeo")
tavg <- system.file("extdata", "annual_mean_temp.tif", package = "funbiogeo")

layers <- terra::rast(c(tavg, prec))

# Make plot (show environmental position of 6 first sites)
fb_plot_site_environment(head(wooddiv_locations), layers)
```

fb_plot_site_traits_completeness

Plot Trait Coverage per Site for each Trait

Description

Display a binary heatmap visualizing the site x traits matrix with colors displaying the proportion of occurring species with known trait values. Traits are ordered from the most to the least known (left to right). Sites are ordered from the ones with highest to lowest overall trait coverage (bottom to top). The site average proportion of species with known trait for each trait (across all sites) is shown in the x-axis labels. An additional column at the very right of the plot named "all_traits" shows a summary considering traits together.

Usage

```
fb_plot_site_traits_completeness(
  site_species,
  species_traits,
  species_categories = NULL,
  all_traits = TRUE
)
```

Arguments

site_species	a data.frame with sites in rows and species in columns. NOTE: the first column should be named "site" and indicate site names. The other columns should be named according to species names.
species_traits	a data.frame with species in rows and traits as columns. NOTE: The first column should be named "species" and contain species names. The other columns should be named according to trait names.
species_categories	(default = NULL) 2-columns data.frame giving species categories, with the first column describing the species name, and the second column giving their corresponding categories
all_traits	a logical (default = TRUE) which tell if the coverage considering all provided traits should be provided in an additional column all_traits

Value

a ggplot2 object

Examples

```
fb_plot_site_traits_completeness(wooddiv_site_species, wooddiv_traits)
```

`fb_plot_species_traits_completeness`*Plot Trait Coverage per Species for each Trait*

Description

Display a binary heatmap visualizing the species x traits matrix with colors displaying present and missing traits. Traits are ordered from the most to the least known (left to right). Species are ordered from the ones with most to the ones with least traits (bottom to top). The proportion of species with non-missing traits is shown on the x-axis labels. An additional column at the very right of the plot named "all_traits" shows a summary considering if all other traits are known.

Usage

```
fb_plot_species_traits_completeness(  
  species_traits,  
  species_categories = NULL,  
  all_traits = TRUE  
)
```

Arguments

`species_traits` a `data.frame` with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

`species_categories`
(default = NULL) 2-columns `data.frame` giving species categories, with the first column describing the species name, and the second column giving their corresponding categories

`all_traits` a logical (default = TRUE) which tell if the coverage considering all provided traits should be provided in an additional column `all_traits`

Value

a `ggplot2` object

Examples

```
fb_plot_species_traits_completeness(wooddiv_traits)
```

`fb_plot_species_traits_missingness`*Plot Trait Coverage per Species for each Trait*

Description

Display a binary heatmap visualizing the species x traits matrix with colors displaying present and missing traits. Traits are ordered from the most to the least known (left to right). Species are ordered from the ones with most to the ones with least traits (bottom to top). The proportion of species with non-missing traits is shown on the x-axis labels. An additional column at the very right of the plot named "all_traits" shows a summary considering if all other traits are known.

Usage

```
fb_plot_species_traits_missingness(  
  species_traits,  
  species_categories = NULL,  
  all_traits = TRUE  
)
```

Arguments

`species_traits` a `data.frame` with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

`species_categories` 2-columns `data.frame` giving species categories NULL by default, with the first column describing the species name, and the second column giving their corresponding categories

`all_traits` a logical (default = TRUE) which tell if the coverage considering all provided traits should be provided in an additional column `all_traits`

Value

a `ggplot2` object

Examples

```
data(wooddiv_traits)  
fb_plot_species_traits_missingness(wooddiv_traits)
```

fb_plot_trait_combination_frequencies

Plot Present and Missing Trait Combinations Frequencies

Description

Display a figure of Present/Missing Trait Combinations where one square is represents one trait. The y-axis gives the frequency of the row (as well as its proportions compared to the total number of rows).

Usage

```
fb_plot_trait_combination_frequencies(
  species_traits,
  species_categories = NULL,
  order_by = c("number", "complete")
)
```

Arguments

species_traits a data.frame with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

species_categories (default = NULL) 2-columns data.frame giving species categories, with the first column describing the species name, and the second column giving their corresponding categories

order_by character(1) either "number" or "complete"
If "number" order rows by frequency so that most frequent rows are at the bottom. Otherwise order rows to order table by the number of non-missing traits then by the frequency of combinations

Value

a ggplot2 object

Examples

```
# Ordered by number by default
fb_plot_trait_combination_frequencies(wooddiv_traits)

# Order by present traits
fb_plot_trait_combination_frequencies(wooddiv_traits, NULL, "complete")
```

`fb_plot_trait_correlation`*Plot Trait Correlation Matrix*

Description

This function outputs a ggplot2 figure that outputs a trait correlation matrix. It only works on numerical traits and will output messages if non-numerical traits are found and errors if no numerical traits are found. Internally it uses the `stats::cor()` function and only works on complete observation (it removes any row that contains any NA). Use the `...` argument to pass options to the `cor()` function.

Usage

```
fb_plot_trait_correlation(species_traits, species_categories = NULL, ...)
```

Arguments

`species_traits` a data.frame with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

`species_categories`
(default = NULL) 2-columns data.frame giving species categories, with the first column describing the species name, and the second column giving their corresponding categories

`...` Additional options passed to `stats::cor()`

Value

a ggplot object

Examples

```
fb_plot_trait_correlation(wooddiv_traits)

# Plot Spearman's correlation
fb_plot_trait_correlation(wooddiv_traits, method = "spearman")
```

`fb_table_trait_summary`*Summary Table on Traits (Missingness, Range, etc.)*

Description

This function outputs a `data.frame` that summarises the species by trait table to have many information in a glance. This can then return a `data.frame` or a nicely formatted `knitr::kable()` for inclusion in an Rmarkdown document.

Usage

```
fb_table_trait_summary(species_traits, kable = FALSE)
```

Arguments

`species_traits` a `data.frame` with species in rows and traits as columns. **NOTE:** The first column should be named "species" and contain species names. The other columns should be named according to trait names.

`kable` TRUE or FALSE Should function returns a `knitr::kable()`? defaults FALSE

Value

a `data.frame` with the following columns:

- `trait_name`: a character column with the trait name as indicated in `species_traits`
- `trait_type`: the nature of the trait (numeric, categorical, or ordered)
- `number_non_missing`: the total number of non-NA trait values
- `proportion_non_missing`: the proportion of non-NA trait values
- `trait_range`: for numerical traits, the range of values
- `trait_mean_sd`: for numerical traits, the mean plus-minus the standard deviation
- `number_distinct`: for non-numerical traits, the number of categories
- `list_distinct`: for non-numerical traits, the list of categories

Examples

```
# Get a data.frame back
fb_table_trait_summary(wooddiv_traits)

# Get a kable (to use in Rmd documents)
fb_table_trait_summary(wooddiv_traits, TRUE)
```

woodiv_categories	<i>Species x Categories of some Mediterranean Conifers</i>
-------------------	--

Description

This dataset is derived from the WOODIV database (available at: <https://www.nature.com/articles/s41597-021-00873-3>). It contains information (classification, endemism, etc.) of the 24 Conifer tree species occurring in Portugal, Spain, France, and Italy (Mediterranean part). This information will be used as a species grouping factor.

This dataset shows the format of the argument `species_categories` used in several functions of `funbiogeo`. Note that species names must be the first column and must be named `species`.

Usage

```
woodiv_categories
```

Format

A data.frame with 24 rows (species) and 6 columns (species, family, genus, binomial, endemism, and cultivated).

References

Monnet AC, Cilleros K, Médail F *et al.* (2021) WOODIV, a database of occurrences, functional traits, and phylogenetic data for all Euro-Mediterranean trees. **Scientific Data**, 8, 89. DOI: [doi:10.1038/s41597021008733](https://doi.org/10.1038/s41597021008733).

Examples

```
data("woodiv_categories")
class(woodiv_categories)
dim(woodiv_categories)
woodiv_categories[1:6, ]
```

woodiv_locations	<i>Sites x Locations of some Mediterranean Conifers</i>
------------------	---

Description

This dataset is derived from the WOODIV database (available at: <https://www.nature.com/articles/s41597-021-00873-3>). It contains the grid cells of sites (10 km x 10 km horizontal resolution) sampled in Portugal, Spain, France, and Italy (Mediterranean part) for which at least one of the 24 conifer tree species occurs.

This dataset exemplifies the argument `site_locations` used in several functions of `funbiogeo`. The variable `site` corresponds to the site labels (must match the same column in `site_species`) and the variable `country` will be used as a site grouping factor.

Usage

```
woodiv_locations
```

Format

An sf POLYGON object with 5,366 rows (grid cells) defined in the ETRS89-extended / LAEA Europe Coordinate Reference System (EPSG:3035).

References

Monnet AC, Cilleros K, Médail F *et al.* (2021) WOODIV, a database of occurrences, functional traits, and phylogenetic data for all Euro-Mediterranean trees. **Scientific Data**, 8, 89. DOI: [doi:10.1038/s41597021008733](https://doi.org/10.1038/s41597021008733).

Examples

```
data("woodiv_locations")
class(woodiv_locations)
dim(woodiv_locations)
head(woodiv_locations)
```

woodiv_site_species	<i>Sites x Species of some Mediterranean Conifers</i>
---------------------	---

Description

This dataset is derived from the WOODIV database (available at: <https://www.nature.com/articles/s41597-021-00873-3>). It contains the presence/absence of the 24 Conifer tree species occurring in Portugal, Spain, France, and Italy (Mediterranean part). The presence/absence is available at 5,366 sites (grid cells of 10 x 10 km horizontal resolution). Sites coordinates are provided in site_locations.

This dataset shows the format of the argument site_species used in several functions of funbiogeo. Note that site labels must be the first column and must be named site.

Usage

```
woodiv_site_species
```

Format

A data.frame with 5,366 rows (sites) and 25 columns (1 column for site label and 24 for species occurrence).

References

Monnet AC, Cilleros K, Médail F *et al.* (2021) WOODIV, a database of occurrences, functional traits, and phylogenetic data for all Euro-Mediterranean trees. **Scientific Data**, 8, 89. DOI: [doi:10.1038/s41597021008733](https://doi.org/10.1038/s41597021008733).

Examples

```
data("woodiv_site_species")
class(woodiv_site_species)
dim(woodiv_site_species)
woodiv_site_species[1:6, 1:6]
```

woodiv_traits

Species x Traits of some Mediterranean Conifers

Description

This dataset is derived from the WOODIV database (available at: <https://www.nature.com/articles/s41597-021-00873-3>). It contains the values of four functional traits for 24 Conifer tree species occurring in Portugal, Spain, France, and Italy (Mediterranean part).

This dataset shows the format of the argument `species_traits` used in several functions of `funbiogeo`. Note that species names must be the first column and must be named `species`.

Usage

```
woodiv_traits
```

Format

A `data.frame` with 24 rows (species) and the following five columns:

species species name corresponding to the columns of `site_species`

plant_height adult plant height (in *m*)

seed_mass seed mass (in *g*)

sla specific leaf area, i.e. the ratio between leaf area and dry mass (in $m^2 \cdot kg^{-1}$)

wood_density wood density, i.e. stem specific density (in $kg \cdot dm^{-3}$)

References

Monnet AC, Cilleros K, Médail F *et al.* (2021) WOODIV, a database of occurrences, functional traits, and phylogenetic data for all Euro-Mediterranean trees. **Scientific Data**, 8, 89. DOI: [doi:10.1038/s41597021008733](https://doi.org/10.1038/s41597021008733).

Examples

```
data("woodiv_traits")
class(woodiv_traits)
dim(woodiv_traits)
head(woodiv_traits)
```

Index

* datasets

- wooddiv_categories, [37](#)
 - wooddiv_locations, [37](#)
 - wooddiv_site_species, [38](#)
 - wooddiv_traits, [39](#)
-
- fb_aggregate_site_data, [3](#)
 - fb_count_sites_by_species, [4](#)
 - fb_count_species_by_site, [5](#)
 - fb_count_species_by_trait, [5](#)
 - fb_count_traits_by_species, [6](#)
 - fb_cwm, [7](#)
 - fb_filter_sites_by_species_coverage, [8](#)
 - fb_filter_sites_by_trait_coverage, [9](#)
 - fb_filter_species_by_site_coverage, [10](#)
 - fb_filter_species_by_trait_coverage, [11](#)
 - fb_filter_traits_by_species_coverage, [12](#)
 - fb_format_site_locations, [13](#)
 - fb_format_site_species, [14](#)
 - fb_format_species_categories, [15](#)
 - fb_format_species_traits, [16](#)
 - fb_get_all_trait_coverages_by_site, [17](#)
 - fb_get_environment, [18](#)
 - fb_get_trait_combination_coverage, [18](#)
 - fb_get_trait_coverage_by_site, [20](#)
 - fb_make_report, [21](#)
 - fb_map_raster, [23](#)
 - fb_map_site_data, [24](#)
 - fb_map_site_traits_completeness, [25](#)
 - fb_plot_distribution_site_trait_coverage, [26](#)
 - fb_plot_number_sites_by_species, [27](#)
 - fb_plot_number_species_by_trait, [28](#)
 - fb_plot_number_traits_by_species, [29](#)
 - fb_plot_site_environment, [30](#)
 - fb_plot_site_traits_completeness, [31](#)
 - fb_plot_species_traits_completeness, [32](#)
 - fb_plot_species_traits_missingness, [33](#)
 - fb_plot_trait_combination_frequencies, [34](#)
 - fb_plot_trait_correlation, [35](#)
 - fb_table_trait_summary, [36](#)
-
- knitr::kable(), [36](#)
 - rmarkdown::render(), [21](#)
 - stats::cor(), [35](#)
-
- wooddiv_categories, [37](#)
 - wooddiv_locations, [37](#)
 - wooddiv_site_species, [38](#)
 - wooddiv_traits, [39](#)