# Package ‘tidypaleo’

January 18, 2023

**Title** Tidy Tools for Paleoenvironmental Archives

**Version** 0.1.3

**Description** Provides a set of functions with a common framework for age-depth model management, stratigraphic visualization, and common statistical transformations. The focus of the package is stratigraphic visualization, for which 'ggplot2' components are provided to reproduce the scales, geometries, facets, and theme elements commonly used in publication-quality stratigraphic diagrams. Helpers are also provided to reproduce the exploratory statistical summaries that are frequently included on stratigraphic diagrams. See Dunnington et al. (2021) <doi:10.18637/jss.v101.i07>.

**Depends** R (>= 3.4.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**Suggests** testthat, knitr, rmarkdown, vegan, patchwork, forcats, vdiffr

**URL** `https://paleolimbot.github.io/tidypaleo/`,
`https://github.com/paleolimbot/tidypaleo`

**BugReports** `https://github.com/paleolimbot/tidypaleo/issues`

**Imports** rlang, tidyselect, dplyr, vctrs, tibble, scales, ggplot2 (>= 3.0.0), styler, purrr, ggstance, stringr, withr, tidyr (>= 1.0.2), digest, rioja

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Dewey Dunnington [aut, cre, cph] (<`https://orcid.org/0000-0002-9415-4582`>)

**Maintainer** Dewey Dunnington <`dewey@fishandwhistle.net`>

**Repository** CRAN

**Date/Publication** 2023-01-18 08:20:03 UTC

# R topics documented:

---

## `age_depth_as_sec_axis` *Use an age depth model as a second ggplot axis*

---

### Description

Use an age depth model as a second ggplot axis

### Usage

```
age_depth_as_sec_axis(x, primary = c("depth", "age"), ...)
```

### Arguments

| | |
|---|---|
| `x` | An age_depth_model |
| `primary` | Specify the primary axis as 'age' or 'depth' |
| `...` | Passed to sec_axis |

### Value

A ggplot2 sec_axis for use in scale_x_continuous, scale_y_continuous, or their reverse variants.

### Examples

```
library(ggplot2)
alta_lake_adm <- age_depth_model(
  alta_lake_210Pb_ages,
  depth = depth_cm,
  age = age_year_ad
)

ggplot(alta_lake_210Pb_ages, aes(y = depth_cm, x = age_year_ad)) +
  geom_path() +
  geom_point() +
  scale_y_reverse(sec.axis = age_depth_as_sec_axis(alta_lake_adm))
```

---

## `age_depth_interpolate` *Age-depth model interpolators/extrapolators*

---

### Description

Age-depth model interpolators/extrapolators

### Usage

```
age_depth_interpolate(x, y)

age_depth_extrapolate(x, y, x0 = last, y0 = last, slope = NULL)

age_depth_exact(x, y)

age_depth_na(x, y)
```

### Arguments

| | |
|---|---|
| `x` | A paired vector of x values |
| `y` | A paired vector of y values |
| `x0` | The x value to anchor the transform |
| `y0` | The y value to anchor the transform |
| `slope` | The slope (in units of y/x) to use for the transform |

### Value

A list with component functions `trans` and `inverse`

### Examples

```
age_depth_model(
  alta_lake_210Pb_ages,
  depth = depth_cm, age = age_year_ad,
  age_max = age_year_ad + age_error_yr,
  age_min = age_year_ad - age_error_yr,
  extrapolate_age_below = ~age_depth_extrapolate(
    tail(.x, 3), tail(.y, 3), x0 = dplyr::last, y0 = dplyr::last
  ),
  extrapolate_age_above = ~age_depth_extrapolate(
    head(.x, 3), head(.y, 3), x0 = dplyr::first, y0 = dplyr::first
  )
)
```

---

## `age_depth_model` *Create age depth models*

### Description

Create age depth models

### Usage

```
age_depth_model(
  .data = NULL,
  depth,
  age,
  age_min = NA_real_,
  age_max = NA_real_,
  interpolate_age = age_depth_interpolate,
  extrapolate_age_below = ~age_depth_extrapolate(.x, .y, x0 = last, y0 = last),
 extrapolate_age_above = ~age_depth_extrapolate(.x, .y, x0 = first, y0 = first),
  interpolate_age_limits = age_depth_exact,
  extrapolate_age_limits_below = age_depth_na,
  extrapolate_age_limits_above = age_depth_na
)
```

### Arguments

| | |
|---|---|
| `.data` | A data frame |
| `depth, age, age_min, age_max` | Expressions evaluated in `.data` that provide the known depths, known ages, and error information if available. These expressions are evaluated like they are within mutate if `.data` is present. |

`interpolate_age`, `extrapolate_age_below`, `extrapolate_age_above`
: These arguments provide the rules for interpolating and extrapolating ages based on depths.

`interpolate_age_limits`, `extrapolate_age_limits_below`, `extrapolate_age_limits_above`
: These arguments provide the rules for interpolating and extrapolating age min and max values based on depths.

### Value

An age depth model object.

### Examples

```
age_depth_model(
  alta_lake_210Pb_ages,
  depth = depth_cm, age = age_year_ad,
  age_max = age_year_ad + age_error_yr,
  age_min = age_year_ad - age_error_yr
)
```

---

## `alta_lake_210Pb_ages` *Alta Lake Lead-210 and Carbon-14 Ages*

### Description

The `alta_lake_210Pb_ages` object contains raw depths and ages for Alta Lake (Whistler, British Columbia, Canada) core AL-GC2 (Dunnington et al. 2016; Dunnington 2015). For these values, ages were calculated from Lead-210 activities using the constant rate of supply (CRS) model (Appleby and Oldfield 1983). The `alta_lake_14C_ages` object contains one uncalibrated Carbon-14 measurement from the same core. The `alta_lake_bacon_ages` object contains the combined result of the Lead-210 and the Carbon-14 ages as modelled by the rbacon package (Blaauw and Christen 2011).

### Usage

```
alta_lake_210Pb_ages

alta_lake_14C_ages

alta_lake_bacon_ages

alta_lake_geochem
```

## Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 9 rows and 5 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 1 rows and 5 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 58 rows and 5 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 192 rows and 9 columns.

## References

Appleby, P. G., and F. Oldfield. "The Assessment of 210Pb Data from Sites with Varying Sediment Accumulation Rates." Hydrobiologia 103, no. 1 (July 1, 1983): 29–35. doi:10.1007/BF00028424.

Blaauw, Maarten, and J. Andrés Christen. "Flexible Paleoclimate Age-Depth Models Using an Autoregressive Gamma Process." Bayesian Analysis 6, no. 3 (September 2011): 457–74. doi:10.1214/ba/1339616472.

Dunnington, Dewey W., Ian S. Spooner, Chris E. White, R. Jack Cornett, Dave Williamson, and Mike Nelson. "A Geochemical Perspective on the Impact of Development at Alta Lake, British Columbia, Canada." Journal of Paleolimnology 56, no. 4 (November 2016): 315–330. doi:10.1007/s109330169919x.

Dunnington, Dewey W. "A 500-Year Applied Paleolimnological Assessment of Environmental Change at Alta Lake, Whistler, British Columbia, Canada." M.Sc. Thesis, Acadia University, 2015. `https://scholar.acadiau.ca/islandora/object/theses:411`.

## Examples

```
alta_lake_210Pb_ages
alta_lake_14C_ages
alta_lake_bacon_ages
```

---

# `as_trans_factory` *Coerce and validate transforms and functions that produce them*

## Description

Coerce and validate transforms and functions that produce them

## Usage

```
as_trans_factory(factory, env = parent.frame())

validate_trans_factory(factory, x = 1:3, y = 1:3)

validate_trans(trans, x = 1:3, y = 1:3)
```

### Arguments

| | |
|---|---|
| `factory` | A function that produces a transform object |
| `env` | The calling environment, for transform factories that are calls or rlang lambda-style functions. |
| `x` | The test x data |
| `y` | The test y data |
| `trans` | A transform object |

### Value

The input, invisibly.

### Examples

```
as_trans_factory(age_depth_interpolate)
```

---

## `facet_abundanceh` *Facet for relative abundance data*

### Description

Provides a number of modifications to the plot that are necessary for relative abundance plots of a number of species. See scale_x_abundance, facet_grid, facet_grid, label_species, label_geochem, and rotated_facet_labels rotated_axis_labels for examples of how to customize the default behaviour.

### Usage

```
facet_abundanceh(
  taxon,
  grouping = NULL,
  rotate_facet_labels = 45,
  labeller = label_species,
  scales = "free_x",
  space = "free_x",
  dont_italicize = c("\\(.*?\\)", "spp?\\.", "-complex", "[Oo]ther"),
  ...
)

facet_abundance(
  taxon,
  grouping = NULL,
  rotate_facet_labels = 0,
  labeller = label_species,
  scales = "free_y",
```

```
  space = "free_y",
  dont_italicize = c("\\(.*?\\)", "spp?\\.", "-complex", "[Oo]ther"),
  ...
)

facet_geochem_wraph(
  param,
  grouping = NULL,
  rotate_axis_labels = 90,
  scales = "free_x",
  labeller = label_geochem,
  renamers = c(`^d([0-9]+)([HCNOS])$` = "paste(delta ^ \\1, \\2)", `^210Pb$` =
    "paste({}^210, Pb)", `^Pb210$` = "paste({}^210, Pb)"),
  units = character(0),
  default_units = NA_character_,
  ...
)

facet_geochem_wrap(
  param,
  grouping = NULL,
  scales = "free_y",
  labeller = label_geochem,
  renamers = c(`^d([0-9]+)([HCNOS])$` = "paste(delta ^ \\1, \\2)", `^210Pb$` =
    "paste({}^210, Pb)", `^Pb210$` = "paste({}^210, Pb)"),
  units = character(0),
  default_units = NA_character_,
  ...
)

facet_geochem_grid(
  param,
  grouping = NULL,
  rotate_axis_labels = 0,
  scales = "free_y",
  space = "fixed",
  labeller = label_geochem,
  renamers = c(`^d([0-9]+)([HCNOS])$` = "paste(delta ^ \\1, \\2)", `^210Pb$` =
    "paste({}^210, Pb)", `^Pb210$` = "paste({}^210, Pb)"),
  units = character(0),
  default_units = NA_character_,
  ...
)

facet_geochem_gridh(
  param,
  grouping = NULL,
  rotate_axis_labels = 90,
```

```
  scales = "free_x",
  space = "fixed",
  labeller = label_geochem,
  renamers = c(`^d([0-9]+)([HCNOS])$` = "paste(delta ^ \\1, \\2)", `^210Pb$` =
    "paste({}^210, Pb)", `^Pb210$` = "paste({}^210, Pb)"),
  units = character(0),
  default_units = NA_character_,
  ...
)
```

## Arguments

| | |
|---|---|
| `taxon, param` | A call to vars, defining the column that identifies the taxon (parameter). |
| `grouping` | A call to vars, identifying additional grouping columns |
| `rotate_facet_labels, rotate_axis_labels` | Facet (axis) label rotation (degrees) |
| `labeller` | Labeller to process facet names. Use label_species to italicize species names, label_geochem to perform common formatting and units, or label_value to suppress. |
| `space, scales` | Modify default scale freedom behaviour |
| `dont_italicize` | Regular expressions that should not be italicized |
| `...` | Passed to facet_grid (abundance) or facet_wrap (geochem). |
| `renamers` | Search and replace operations to perform in the form search = replace. Replace text can (should) contain backreferences, and will be parsed as an expression (see plotmath). Use NULL to suppress renaming. |
| `units` | A named list of values = unit |
| `default_units` | The default units to apply |

## Value

A subclass of `ggplot2::facet_grid()` or `ggplot2::facet_wrap()`.

## Examples

```
library(ggplot2)

ggplot(keji_lakes_plottable, aes(x = rel_abund, y = depth)) +
  geom_col_segsh() +
  scale_y_reverse() +
  facet_abundanceh(vars(taxon), grouping = vars(location)) +
  labs(y = "Depth (cm)")

ggplot(keji_lakes_plottable, aes(y = rel_abund, x = depth)) +
  geom_col_segs() +
  scale_x_reverse() +
  facet_abundance(vars(taxon), grouping = vars(location)) +
  labs(x = "Depth (cm)")
```

```
ggplot(alta_lake_geochem, aes(x = value, y = depth)) +
  geom_lineh() +
  geom_point() +
  scale_y_reverse() +
  facet_geochem_wrap(vars(param), units = c(C = "%", Cu = "ppm", Ti = "ppm"), nrow = 1) +
  labs(x = NULL, y = "Depth (cm)")

ggplot(alta_lake_geochem, aes(x = value, y = depth)) +
  geom_lineh() +
  geom_point() +
  scale_y_reverse() +
  facet_geochem_gridh(vars(param), units = c(C = "%", Cu = "ppm", Ti = "ppm")) +
  labs(x = NULL, y = "Depth (cm)")

ggplot(alta_lake_geochem, aes(y = value, x = depth)) +
  geom_line() +
  geom_point() +
  scale_x_reverse() +
  facet_geochem_grid(vars(param), units = c(C = "%", Cu = "ppm", Ti = "ppm")) +
  labs(y = NULL, x = "Depth (cm)")
```

---

## `geom_col_segsh` *Useful geometries for strat diagrams*

### Description

Useful geometries for strat diagrams

### Usage

```
geom_col_segsh(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  xend = 0,
  arrow = NULL,
  arrow.fill = NULL,
  lineend = "butt",
  linejoin = "round",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_col_segs(
  mapping = NULL,
```

```
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  yend = 0,
  arrow = NULL,
  arrow.fill = NULL,
  lineend = "butt",
  linejoin = "round",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

GeomColSegsh

GeomColSegs

GeomLineh
```

## Arguments

`mapping, data, stat, position, arrow, arrow.fill, lineend, linejoin, na.rm, show.legend, inherit.aes, ...`
See geom_segment.

| | |
|---|---|
| `xend, yend` | The end of the horizontal or vertical segment bars, respectively. |

## Format

An object of class `GeomColSegsh` (inherits from `GeomSegment`, `Geom`, `ggproto`, `gg`) of length 4.

An object of class `GeomColSegs` (inherits from `GeomSegment`, `Geom`, `ggproto`, `gg`) of length 4.

An object of class `GeomLineh` (inherits from `GeomPath`, `Geom`, `ggproto`, `gg`) of length 2.

## Value

A ggplot2 layer

---

# `geom_lineh` *Connect observations in the vertical direction*

## Description

Connect observations in the vertical direction

### Usage

```
geom_lineh(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)
```

### Arguments

| | |
|---|---|
| `mapping, data, stat, position, na.rm, show.legend, inherit.aes, ...` | See geom_line. |

### Value

A ggplot2 layer.

---

## `geom_point_exaggerate` *Exaggerated geometries that do not train scales*

### Description

Exaggerated geometries that do not train scales

### Usage

```
geom_point_exaggerate(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  exaggerate_x = 1,
  exaggerate_y = 1,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_line_exaggerate(
  mapping = NULL,
  data = NULL,
  stat = "identity",
```

```
  position = "identity",
  ...,
  exaggerate_x = 1,
  exaggerate_y = 1,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_lineh_exaggerate(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  exaggerate_x = 1,
  exaggerate_y = 1,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_area_exaggerate(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  exaggerate_x = 1,
  exaggerate_y = 1,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

geom_areah_exaggerate(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  exaggerate_x = 1,
  exaggerate_y = 1,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)
```

## Arguments

| | |
|---|---|
| `mapping, data, stat, position, na.rm, show.legend, inherit.aes, ...` | See parent geometries |
| `exaggerate_x, exaggerate_y` | The factor by which to exaggerate x or y values |

## Value

A subclass of ggplot2::Geom.

## Examples

```
library(ggplot2)

ggplot(keji_lakes_plottable, aes(x = rel_abund, y = depth)) +
  geom_lineh_exaggerate(exaggerate_x = 2, lty = 2) +
  geom_col_segsh() +
  scale_y_reverse() +
  facet_abundanceh(vars(taxon), grouping = vars(location)) +
  labs(y = "Depth (cm)")
```

---

# `geom_ribbonh` *Vertical ribbons and area plots*

## Description

Vertical ribbons and area plots

## Usage

```
geom_ribbonh(
  mapping = NULL,
  data = NULL,
  stat = "identity",
  position = "identity",
  ...,
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE
)

GeomRibbonh

geom_areah(
  mapping = NULL,
  data = NULL,
```

```
  stat = "identity",
  position = "stackv",
  na.rm = FALSE,
  show.legend = NA,
  inherit.aes = TRUE,
  ...
)

GeomAreah
```

## Arguments

| | |
|---|---|
| `mapping, data, stat, position, na.rm, show.legend, inherit.aes, ...` | See geom_ribbon. |

## Format

An object of class `GeomRibbonh` (inherits from `Geom`, `ggproto`, `gg`) of length 6.

An object of class `GeomAreah` (inherits from `GeomRibbonh`, `Geom`, `ggproto`, `gg`) of length 4.

## Examples

```
library(ggplot2)

# Generate data
huron <- data.frame(year = 1875:1972, level = as.vector(LakeHuron))
h <- ggplot(huron, aes(y = year))

h + geom_ribbonh(aes(xmin=0, xmax=level))
h + geom_areah(aes(x = level))

# Add aesthetic mappings
h +
  geom_ribbonh(aes(xmin = level - 1, xmax = level + 1), fill = "grey70") +
  geom_lineh(aes(x = level))
```

---

# `halifax_lakes_plottable`
*Halifax lakes water chemistry and top/bottom diatom counts*

## Description

A subset of well-labeled, clean diatom count data for 44 Halifax-area (Nova Scotia) lakes, an analysis of which has been published by Ginn et al. (2015).

## Usage

```
halifax_lakes_plottable
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 114 rows and 5 columns.

### Source

Neotoma paleoecology database (`https://www.neotomadb.org`)

### References

Ginn, Brian K., Thiyake Rajaratnam, Brian F. Cumming, and John P. Smol. "Establishing Realistic Management Objectives for Urban Lakes Using Paleolimnological Techniques: An Example from Halifax Region (Nova Scotia, Canada)." Lake and Reservoir Management 31, no. 2 (April 3, 2015): 92-108. doi:10.1080/10402381.2015.1013648.

### Examples

```
halifax_lakes_plottable
```

---

## `keji_lakes_plottable` *Keji lakes core diatom counts*

### Description

A subset of well-labeled, clean diatom count data for 3 Keji-area (Nova Scotia) lakes, which form part of the analysis in Ginn et al. (2007).

### Usage

```
keji_lakes_plottable
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 202 rows and 5 columns.

### Source

Neotoma paleoecology database (`https://www.neotomadb.org`)

### References

Ginn, Brian K., Brian F. Cumming, and John P. Smol. "Long-Term Lake Acidification Trends in High- and Low-Sulphate Deposition Regions from Nova Scotia, Canada." Hydrobiologia 586, no. 1 (July 1, 2007): 261-75. doi:10.1007/s1075000706443.

### Examples

```
keji_lakes_plottable
```

---

## `kellys_lake_cladocera` *Kellys Lake Data*

---

### Description

Geochemistry measurements and Cladocera counts from Kellys Lake, Cape Breton Island, Nova Scotia, Canada.

### Usage

```
kellys_lake_cladocera

kellys_lake_geochem

kellys_lake_ages
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 300 rows and 5 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 305 rows and 9 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 14 rows and 5 columns.

### References

Joshua Kurek, Ian Spooner, and Dewey Dunnington (unpublished data).

---

## `label_geochem` *Geochem facet labelers*

---

### Description

Geochem facet labelers

### Usage

```
label_geochem(
  labels,
  units = character(0),
  default_units = NA_character_,
  geochem_facet = 1,
  renamers = c(`^d([0-9]+)([HCNOS])$` = "paste(delta ^ \\1, \\2)", `^210Pb$` =
    "paste({}^210, Pb)", `^Pb210$` = "paste({}^210, Pb)"),
  multi_line = TRUE
)
```

### Arguments

| | |
|---|---|
| `labels` | A data.frame of facet label values |
| `units` | A named list of values = unit |
| `default_units` | The default units to apply |
| `geochem_facet` | Which facet to apply formatting |
| `renamers` | Search and replace operations to perform in the form search = replace. Replace text can (should) contain backreferences, and will be parsed as an expression (see plotmath). Use NULL to suppress renaming. |
| `multi_line` | See label_parsed |

### Value

A `ggplot2::labeller()`

### Examples

```
library(ggplot2)

ggplot(alta_lake_geochem, aes(x = value, y = depth)) +
  geom_lineh() +
  geom_point() +
  scale_y_reverse() +
  facet_wrap(
    vars(param),
    labeller = purrr::partial(label_geochem, geochem_facet = "param"),
    nrow = 1,
    scales = "free_x"
  ) +
  labs(x = NULL, y = "Depth (cm)")
```

---

## `label_species` *Species facet labellers*

### Description

Use these to label species with partial italic formatting. See label_parsed.

### Usage

```
label_species(
  labels,
  dont_italicize = c("\\(.*?\\)", "spp?\\.", "-complex", "[Oo]ther"),
  species_facet = 1,
  multi_line = TRUE
)
```

### Arguments

| | |
|---|---|
| `labels` | A data.frame of facet label values |
| `dont_italicize` | Regular expressions that should not be italicized |
| `species_facet` | Which facet(s) contain species values |
| `multi_line` | See label_parsed |

### Value

A `ggplot2::labeller()`

### Examples

```
library(ggplot2)

ggplot(keji_lakes_plottable, aes(x = rel_abund, y = depth)) +
  geom_col_segsh() +
  scale_y_reverse() +
  facet_grid(
    cols = vars(taxon),
    rows = vars(location),
    scales = "free_x",
    space = "free_x",
    labeller = purrr::partial(label_species, species_facet = "taxon")
  ) +
  labs(y = "Depth (cm)")
```

---

## `layer_dendrogram` *Add a dendrogram as a layer or facet*

### Description

Add a dendrogram as a layer or facet

### Usage

```
layer_dendrogram(
  object,
  mapping,
  ...,
  colour = "black",
  size = 0.5,
  linetype = 1,
  alpha = NA,
  sequential_facets = TRUE
)
```

```
plot_layer_dendrogram(object, mapping, ..., panel_label = "CONISS")

layer_zone_boundaries(
  object,
  mapping,
  ...,
  linetype = 2,
  alpha = 0.7,
  colour = "black",
  size = 0.5
)
```

## Arguments

| | |
|---|---|
| `object` | A nested_hclust object. |
| `mapping` | Map at least one axis (x or y) to a qualifier, like `aes(x = depth)` or similar. |
| `...` | Use facet_var = "CONISS" or similar to name the panel |
| `linetype, alpha, colour, size` | Customize the apperance of boundary/dendrogram segment lines |
| `sequential_facets` | TRUE will result in the panel containing the dendrogram added to the right of the plot. |
| `panel_label` | Use to label a pane on a stanalone dendrogram plot |

## Value

A `ggplot2::layer()`

## Examples

```
library(ggplot2)
library(dplyr, warn.conflicts = FALSE)

alta_coniss <- nested_data(
  alta_lake_geochem,
  qualifiers = c(age, depth, zone),
  key = param,
  value = value,
  trans = scale
) %>%
  nested_chclust_coniss()

ggplot() +
  layer_dendrogram(alta_coniss, aes(y = depth)) +
  scale_y_reverse()
```

---

## `layer_scores` *Add scores to a plot*

---

### Description

Add scores to a plot

### Usage

```
layer_scores(
  object,
  mapping = NULL,
  which = "PC1",
  key = "param",
  value = "value",
  scores_geom = list(ggplot2::geom_path(), ggplot2::geom_point()),
  sequential_facets = TRUE
)

plot_layer_scores(
  object,
  mapping,
  which = "PC1",
  key = "param",
  value = "value",
  ...
)
```

### Arguments

| | |
|---|---|
| `object` | A nested_prcomp or similar object |
| `mapping` | A mapping created with aes |
| `which` | Which principal components to plot |
| `key` | The column name to use for the principal component names |
| `value` | The column name to use for the principal component score values |
| `scores_geom` | One or more geometries to which scores should be applied. |
| `sequential_facets` | TRUE will result in the panel containing the dendrogram added to the right of the plot. |
| `...` | Passed to layer_scores() |

### Value

A `list()` that can be addeed to a `ggplot2::ggplot()`

### Examples

```
library(ggplot2)
library(dplyr, warn.conflicts = FALSE)

alta_pca <- nested_data(
  alta_lake_geochem,
  qualifiers = c(age, depth, zone),
  key = param,
  value = value,
  trans = scale
) %>%
  nested_prcomp()

ggplot() +
  layer_scores(alta_pca, aes(value, depth), which = "PC1") +
  scale_y_reverse()

plot_layer_scores(alta_pca, aes(y = depth), which = c("PC1", "PC2")) +
  scale_y_reverse()
```

---

## `long_lake_14C_ages` *Long Lake Carbon-14 Ages*

### Description

This object contains several uncalibrated Carbon-14 measurements from Long Lake (Nova Scotia-New Brunswick Border Region, Canada) core LL-PC2 (Dunnington et al. 2017; White 2012). The `long_lake_bacon_ages` object contains the result of the Carbon-14 ages as modelled by the rbacon package (Blaauw and Christen 2011).

### Usage

```
long_lake_14C_ages

long_lake_bacon_ages

long_lake_plottable
```

### Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 5 rows and 5 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 86 rows and 5 columns.

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 51 rows and 10 columns.

### References

Blaauw, Maarten, and J. Andrés Christen. "Flexible Paleoclimate Age-Depth Models Using an Autoregressive Gamma Process." Bayesian Analysis 6, no. 3 (September 2011): 457–74. doi:10.1214/ba/1339616472.

Dunnington, Dewey W., Hilary White, Ian S. Spooner, Mark L. Mallory, Chris White, Nelson J. O'Driscoll, and Nic R. McLellan. "A Paleolimnological Archive of Metal Sequestration and Release in the Cumberland Basin Marshes, Atlantic Canada." FACETS 2, no. 1 (May 23, 2017): 440–60. doi:10.1139/facets20170004.

White, Hilary E. "Paleolimnological Records of Post-Glacial Lake and Wetland Evolution from the Isthmus of Chignecto Region, Eastern Canada." M.Sc. Thesis, Acadia University, 2012. `https://scholar.acadiau.ca/islandora/object/theses:247`.

### Examples

```
long_lake_14C_ages
long_lake_bacon_ages
long_lake_plottable
```

---

## `nested_analysis` *Perform an analysis on a nested data matrix*

### Description

Perform an analysis on a nested data matrix

### Usage

```
nested_analysis(
  .data,
  .fun,
  ...,
  .output_column = "model",
  .reserved_names = NULL,
  .env = parent.frame()
)
```

### Arguments

| | |
|---|---|
| `.data` | A data frame with a list column of data frames, possibly created using nested_data. |
| `.fun` | A model function |
| `...` | Passed to fun |
| `.output_column` | A column name in which the output of .fun should be stored. |
| `.reserved_names` | Names that should not be allowed as columns in any data frame within this object |
| `.env` | Passed to as_function |

### Value

.data with an additional list column of fun output

### Examples

```
nd <- nested_data(
  alta_lake_geochem,
  qualifiers = c(age, depth, zone),
  key = param,
  value = value,
  trans = scale
)

na <- nested_analysis(nd, vegan::rda, X = data)
plot(na)
```

---

## `nested_data` *Prepare a parameter-long data frame for statistical analysis*

### Description

Prepare a parameter-long data frame for statistical analysis

### Usage

```
nested_data(
  .data,
  qualifiers = NULL,
  key = NULL,
  value,
  fill = NA,
  select_if = ~TRUE,
  filter_all = any_vars(TRUE),
  trans = identity,
  groups = NULL
)

unnested_data(.data, ...)
```

### Arguments

| | |
|---|---|
| `.data` | Data in parameter-long form |
| `qualifiers` | Columns that add context to observations (e.g., depth, zone, core) |
| `key` | The column name that contains the column names of the data matrix |
| `value` | The column name that contains the values |

| | |
|---|---|
| `fill` | If a key/value combination doesn't exist in the input, this value will be assigned in the data matrix. Generally, using NA for geochemical data and 0 for relative abundance data is advised. |
| `select_if` | Use `~TRUE` to keep all columns; use `~all(is.finite(.))` to keep columns with all finite values. See select_if. |
| `filter_all` | Use `any_vars(TRUE)` to keep all observations; use `all_vars(is.finite(.))` to keep only observations with finite (non-missing) values. See filter_all. |
| `trans` | A function that will be applied to all columns, column-wise. Use identity to perform no transformation, use scale to scale each column to a mean of zero and variance of 1. See mutate_all. |
| `groups` | Use group_by or this argument to group by one or more columns (e.g., core or lake) |
| `...` | One or more columns to unnest. |

### Value

A nested data matrix, which is composed of a `tibble::tibble()` with tibble list-columns `data`, `discarded_rows`, `discarded_columns`, and `qualifiers`.

### Examples

```
nested_data(
  alta_lake_geochem,
  qualifiers = c(age, depth, zone),
  key = param,
  value = value,
  trans = scale
)
```

---

## `nested_hclust` *Nested (Constrained) hierarchical clustering*

### Description

Powered by chclust and hclust; broken stick using bstick.

### Usage

```
nested_hclust(
  .data,
  data_column = "data",
  qualifiers_column = "qualifiers",
  distance_fun = stats::dist,
  n_groups = NULL,
  ...,
```

```
  .fun = stats::hclust,
  .reserved_names = character(0)
)

nested_chclust_conslink(
  .data,
  data_column = "data",
  qualifiers_column = "qualifiers",
  distance_fun = stats::dist,
  n_groups = NULL,
  ...
)

nested_chclust_coniss(
  .data,
  data_column = "data",
  qualifiers_column = "qualifiers",
  distance_fun = stats::dist,
  n_groups = NULL,
  ...
)
```

## Arguments

| | |
|---|---|
| `.data` | A data frame with a list column of data frames, possibly created using nested_data. |
| `data_column` | An expression that evalulates to the data object within each row of .data |
| `qualifiers_column` | The column that contains the qualifiers |
| `distance_fun` | A distance function like dist or vegdist. |
| `n_groups` | The number of groups to use (can be a vector or expression using vars in .data) |
| `...` | Passed to chclust or hclust. |
| `.fun` | Function powering the clustering. Must return an hclust object of some kind. |
| `.reserved_names` | Names that should not be allowed as columns in any data frame within this object |

## Value

`.data` with additional columns

## References

Bennett, K. (1996) Determination of the number of zones in a biostratigraphic sequence. New Phytologist, 132, 155-170. doi:10.1111/j.14698137.1996.tb04521.x (Broken stick)

Grimm, E.C. (1987) CONISS: A FORTRAN 77 program for stratigraphically constrained cluster analysis by the method of incremental sum of squares. Computers & Geosciences, 13, 13-35. doi:10.1016/00983004(87)900227

Juggins, S. (2017) rioja: Analysis of Quaternary Science Data, R package version (0.9-15.1). (`https://cran.r-project.org/package=rioja`).

See hclust for hierarchical clustering references

### Examples

```
library(tidyr)
library(dplyr, warn.conflicts = FALSE)

nested_coniss <- keji_lakes_plottable %>%
  group_by(location) %>%
  nested_data(depth, taxon, rel_abund, fill = 0) %>%
  nested_chclust_coniss()

# plot the dendrograms using base graphics
plot(nested_coniss, main = location, ncol = 1)

# plot broken stick dispersion to verify number of plausible groups
library(ggplot2)

nested_coniss %>%
  select(location, broken_stick) %>%
  unnest(broken_stick) %>%
  tidyr::gather(type, value, broken_stick_dispersion, dispersion) %>%
  ggplot(aes(x = n_groups, y = value, col = type)) +
  geom_line() +
  geom_point() +
  facet_wrap(vars(location))
```

---

## `nested_prcomp` *Nested Principal Components Analysis (PCA)*

### Description

Powered by prcomp. When creating the nested_data, the data should be scaled (i.e, `trans = scale`) if all variables are not in the same unit.

### Usage

```
nested_prcomp(.data, data_column = .data$data, ...)
```

### Arguments

| | |
|---|---|
| `.data` | A data frame with a list column of data frames, possibly created using nested_data. |
| `data_column` | An expression that evalulates to the data object within each row of .data |
| `...` | Passed to prcomp. |

## Value

.data with additional columns 'model', 'loadings', 'variance' and 'scores'

## Examples

```
library(dplyr, warn.conflicts = FALSE)

nested_pca <- alta_lake_geochem %>%
  nested_data(
    qualifiers = c(depth, zone),
    key = param,
    value = value,
    trans = scale
  ) %>%
  nested_prcomp()

# get variance info
nested_pca %>% unnested_data(variance)

# get loadings info
nested_pca %>% unnested_data(loadings)

# scores, requalified
nested_pca %>% unnested_data(c(qualifiers, scores))
```

---

# `plot.age_depth_model` *Plot an age depth model using base graphics*

## Description

Plot an age depth model using base graphics

## Usage

```
## S3 method for class 'age_depth_model'
plot(
  x,
  xlab = "depth",
  ylab = "age",
  xlim = NULL,
  ylim = NULL,
  add = FALSE,
  ...
)
```

### Arguments

| | |
|---|---|
| `x` | An age_depth_model |
| `xlab, ylab` | Axis labels |
| `xlim, ylim` | Axis limits |
| `add` | Pass TRUE to skip creating a new plot |
| `...` | Passed to points to customize points display |

### Value

The input, invisibly

### Examples

```
adm <- age_depth_model(
  alta_lake_210Pb_ages,
  depth = depth_cm, age = age_year_ad,
  age_max = age_year_ad + age_error_yr,
  age_min = age_year_ad - age_error_yr
)

plot(adm)
```

---

## `plot.nested_analysis` *Plot a nested analysis*

### Description

Calls plot or another (base) plotting function on all models, arranging the output in subplots.

### Usage

```
## S3 method for class 'nested_analysis'
plot(x, ..., main = "", nrow = NULL, ncol = NULL)

plot_nested_analysis(
  .x,
  .fun,
  ...,
  nrow = NULL,
  ncol = NULL,
  .model_column = .data$model,
  .output_column = NULL
)

## S3 method for class 'nested_prcomp'
biplot(x, ..., nrow = NULL, ncol = NULL)
```

### Arguments

| | |
|---|---|
| `x, .x` | A nested_analysis object (or subclass) |
| `...` | Passed to the plot function. Tidy evaluation is supported, and arguments are evaluated within a transposed version of x for each row. |
| `main` | The plot title |
| `nrow, ncol` | Force a number of rows or columns in the output |
| `.fun` | A function that produces graphical output |
| `.model_column` | The column containing the model |
| `.output_column` | The column in which the output of the plot function should be placed |

### Value

the input, invisibly

### Examples

```
nd <- nested_data(
  alta_lake_geochem,
  qualifiers = c(age, depth, zone),
  key = param,
  value = value,
  trans = scale
)

na <- nested_analysis(nd, vegan::rda, X = data)
plot(na)
```

---

## `predict.age_depth_model`

*Predict age and depth values*

### Description

Predict age and depth values

### Usage

```
## S3 method for class 'age_depth_model'
predict(object, .data = NULL, depth = NULL, age = NULL, ...)

predict_depth(object, age)

predict_age(object, depth)
```

### Arguments

| | |
|---|---|
| `object` | An age_depth_model object |
| `.data` | Optional input data frame |
| `depth, age` | Specify exactly one of these to predict the other. |
| `...` | Unused |

### Value

A data frame with the same number of observations as the input age or depth vector.

### Examples

```
adm <- age_depth_model(
  alta_lake_210Pb_ages,
  depth = depth_cm, age = age_year_ad,
  age_max = age_year_ad + age_error_yr,
  age_min = age_year_ad - age_error_yr
)

predict(adm, depth = 1:5)
```

---

## `rotated_facet_labels` *Common plot modifications for stratigraphic plots*

### Description

Common plot modifications for stratigraphic plots

### Usage

```
rotated_facet_labels(
  angle = 45,
  direction = "x",
  remove_label_background = TRUE
)

rotated_axis_labels(angle = 90, direction = "x")
```

### Arguments

| | |
|---|---|
| `angle` | The angle at which labels should be rotated |
| `direction` | The axes along which the operations should be performed |
| `remove_label_background` | Whether or not label backgrounds should be removed along rotated label axes |

## Value

An object or list of objects that can be added to a ggplot

## Examples

```
library(ggplot2)

ggplot(mpg, aes(cty, hwy)) +
  geom_point() +
  facet_wrap(vars(class)) +
  rotated_facet_labels(45, "x")
```

---

## `scale_x_abundance` *Scales for relative abundance values*

---

## Description

Continuous scales that (1) always start at 0, (2) always have the same breaks, and (3) expand using a constant rather than a percentage. These scales assume that data are in percentages (i.e., range 0 to 100 rather than 0 to 1).

## Usage

```
scale_x_abundance(
  ...,
  limits = c(0, NA),
  breaks = seq(10, 90, 30),
  minor_breaks = seq(0, 100, 10),
  expand = c(0, 1)
)

scale_y_abundance(
  ...,
  limits = c(0, NA),
  breaks = seq(10, 90, 30),
  minor_breaks = seq(0, 100, 10),
  expand = c(0, 1)
)
```

## Arguments

| | |
|---|---|
| `...` | Passed to scale_y_continuous or scale_x_continuous |
| `limits` | Limits for the scale |
| `breaks` | Where to place labels on the scale |
| `minor_breaks` | Where to place minor breaks |
| `expand` | A vector of expantion constants |

## Value

A scale_y_continuous or scale_x_continuous

## Examples

```
library(dplyr, warn.conflicts = FALSE)
library(ggplot2)

keji_lakes_plottable %>%
  filter(taxon == "Other", location == "Beaverskin Lake") %>%
  ggplot(aes(rel_abund, depth)) +
  geom_col_segsh() +
  scale_x_abundance() +
  scale_y_reverse()
```

---

# `scale_y_depth_age` *Age-depth scales*

## Description

Age-depth scales

## Usage

```
scale_y_depth_age(
  model = NULL,
  age_name = "age",
  age_breaks = waiver(),
  age_labels = waiver(),
  ...
)

scale_y_age_depth(
  model = NULL,
  reversed = FALSE,
  depth_name = "depth",
  depth_breaks = waiver(),
  depth_labels = waiver(),
  ...
)

scale_x_depth_age(
  model = NULL,
  age_name = "age",
  age_breaks = waiver(),
  age_labels = waiver(),
```

```
  ...
)

scale_x_age_depth(
  model = NULL,
  reversed = FALSE,
  depth_name = "depth",
  depth_breaks = waiver(),
  depth_labels = waiver(),
  ...
)
```

## Arguments

| | |
|---|---|
| `model` | An age-depth model, or NULL to suppress the second axis |
| `age_name, depth_name` | Label for the second axis |
| `age_breaks, depth_breaks` | Breaks for the second axis |
| `age_labels, depth_labels` | Labels for each break on the second axis |
| `...` | Passed to scale_y_continuous or scale_x_continuous |
| `reversed` | Reverse the primary age axis (for years BP or similar) |

## Value

A scale_y_continuous or scale_x_continuous

## Examples

```
library(ggplot2)
library(dplyr, warn.conflicts = FALSE)

adm <- age_depth_model(
  alta_lake_210Pb_ages,
  depth = depth_cm, age = age_year_ad
)

alta_lake_geochem %>%
  filter(param == "Cu") %>%
  ggplot(aes(value, depth)) +
  geom_point() +
  scale_y_depth_age(adm)
```

---

## `sequential_layer_facets` *Change facet ordering behaviour*

---

### Description

Normally, facets are ordered using as.factor on all values that occur within layer data, which means that when adding additional layers, any ordering is not preserved unless the factor levels are identical on all factors. This function changes this behaviour such that facet levels are combined in layer order. This is useful when adding standalone layers to a plot without disturbing the existing order.

### Usage

```
sequential_layer_facets(reverse = FALSE)
```

### Arguments

| | |
|---|---|
| `reverse` | Use TRUE to process layers in reverse order |

### Value

An object that can be added to a `ggplot2::ggplot()`

### Examples

```
library(ggplot2)

p <- ggplot(mapping = aes(x, y)) +
  geom_point(data = data.frame(x = 1:5, y = 1:5, facet = "b")) +
  geom_point(data = data.frame(x = 1:5, y = 1:5, facet = "a")) +
  facet_wrap(vars(facet))

p
p + sequential_layer_facets()
```

---

## `stat_nested_hclust` *Display a dendrogram as a ggplot2 layer*

---

### Description

Display a dendrogram as a ggplot2 layer

### Usage

```
stat_nested_hclust(
  mapping = NULL,
  data = NULL,
  geom = "segment",
  position = "identity",
  ...,
  inherit.aes = TRUE,
  show.legend = NA
)

StatNestedHclust
```

### Arguments

| | |
|---|---|
| `mapping` | A mapping created using aes. Must map x OR y to a qualifier. |
| `data` | A nested_hclust object |
| `geom` | Any geom that takes x, xend, y, and yend. Probably geom_segment is the only one that makes sense. |
| `position` | Position adjustment |
| `...` | Passed to the the stat/geom (see geom_segment) |
| `inherit.aes` | Inherit aesthetics from ggplot()? |
| `show.legend` | Show mapped aesthetics in the legend? |

### Format

An object of class `StatNestedHclust` (inherits from `Stat`, `ggproto`, `gg`) of length 4.

### Value

A ggplot2::Stat

### Examples

```
library(ggplot2)
library(dplyr, warn.conflicts = FALSE)

alta_coniss <- nested_data(
  alta_lake_geochem,
  qualifiers = c(age, depth, zone),
  key = param,
  value = value,
  trans = scale
) %>%
  nested_chclust_coniss()

ggplot(alta_coniss) +
  stat_nested_hclust(aes(model = model, y = depth)) +
```

```
  scale_y_reverse()
```

---

`theme_paleo` *A Paleo-friendly ggplot2 theme*

---

## Description

Essentially, this is `ggplot2::theme_bw()` with a few modifications

## Usage

```
theme_paleo(...)
```

## Arguments

| | |
|---|---|
| `...` | Passed to `ggplot2::theme_bw()` |

## Value

A complete `ggplot2::theme()`

## Examples

```
library(ggplot2)

ggplot(mpg, aes(cty, hwy)) +
  geom_point() +
  theme_paleo()
```

# Index