

Package: fellingdater (via r-universe)

August 30, 2026

Title Tree-ring dating and estimating felling dates of historical timbers

Version 1.2.1

Description fellingdater provides a comprehensive suite of functions for dendrochronological and dendroarchaeological analysis, covering a workflow from tree-ring ****data processing**** and ****crossdating****, up to the estimation and reporting of ****felling dates****. Originally designed to assist dendrochronologists in inferring felling dates from dated tree-ring series with partially preserved sapwood, the package has been expanded to include a workflow for tree-ring dating, analysis, and visualization.

URL <https://docs.ropensci.org/fellingdater>,
<https://github.com/ropensci/fellingdater>,
<https://ropensci.github.io/fellingdater>

BugReports <https://github.com/ropensci/fellingdater/issues>

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Depends R (>= 4.1.0)

LazyData true

Imports dplyr, plyr, ggplot2 (>= 3.5.0), ggtext, MASS, tidyr, utils, dplyr, stats, rlang

Suggests covr, knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

Config/pak/sysreqs cmake make libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev

Repository <https://ropensci.r-universe.dev>

Date/Publication 2026-04-15 06:10:08 UTC

RemoteUrl <https://github.com/ropensci/fellingdater>

RemoteRef main

RemoteSha 0b5082d0f732d9584dd68dee70e056702489baff

Contents

Brathen_1982	3
fd_report	4
fellingdater	6
fh_header	7
hdi	8
Hollstein_1980	9
Miles_1997_NM	9
Miles_1997_SC	10
Miles_1997_WBC	11
mov_av	12
Pilcher_1987	13
read_fh	13
sgc_for_plot	15
Sohar_2012_ELL_c	16
Sohar_2012_ELL_t	17
Sohar_2012_FWE_c	17
Sohar_2012_FWE_t	18
sw_combine	19
sw_combine_plot	21
sw_data_info	22
sw_data_overview	23
sw_example0	23
sw_example1	24
sw_example2	24
sw_example3	25
sw_example4	25
sw_example5	26
sw_example6	27
sw_example7	27
sw_interval	28
sw_interval_plot	29
sw_model	30
sw_model_plot	31
sw_sum	32
sw_sum_plot	34
trs_crossdate	35
trs_end_date	38
trs_plot_dated	39
trs_plot_rwl	40
trs_pseudo_rwl	42

<i>Brathen_1982</i>	3
trs_pv	43
trs_remove	44
trs_select	45
trs_tbp	46
trs_tbp_transform	48
trs_tho	48
trs_tho_transform	50
trs_trim	51
trs_tSt	52
trs_zscore	53
vanDaalen_NLBE	54
vanDaalen_Norway	55
Wazny_1990	55
Weitz_2025	56
Index	58

Brathen_1982	<i>Brathen 1982 sapwood data set.</i>
--------------	---------------------------------------

Description

Sapwood data set for Western Sweden published by Bräthen in 1982.

Usage

Brathen_1982

Format

- A tibble of 2 variables:
- n_sapwood** number of sapwood rings
- count** number of times n_sapwood was observed

Details

sample size = 69 observations

Source

Bräthen A. 1982. A tree-ring chronology from the western part of Sweden. Sapwood and a dating problem, in: Hackens T., Mejdahl V. (Eds.), Second Nordic Conference on the Application of Scientific Methods in Archaeology, PACT 7(1). pp. 27–35.

Examples

```
sw_data_info("Brathen_1982")

sw_model("Brathen_1982", plot = TRUE)
```

fd_report

Report felling dates of individual tree-ring series

Description

This function reports the felling date estimate of individual tree-ring series, based on the presence/absence of sapwood and/or waney edge. There are three possible modes of reporting:

- a *terminus post quem* or *earliest possible felling date*: when only heartwood rings have been observed and measured
- a felling date range or interval: when sapwood rings have been recorded, but no bark or waney edge is present.
- an exact felling date: when bark or waney edge is present on the measured sample.

Usage

```
fd_report(
  x,
  series = "series",
  last = "last",
  n_sapwood = "n_sapwood",
  waneyedge = "waneyedge",
  sw_data = "Hollstein_1980",
  cred_mass = 0.954,
  densfun = "lognormal"
)
```

Arguments

x	Name of a data.frame with at least four columns, providing information on • the id's of the tree-ring series ("series") • the number of sapwood rings observed ("n_sapwood") • the presence of waney edge ("waneyedge") • the year assigned to the last measured ring ("last"). Optionally, a column specifying the sapwood data set (sw_data) can also be included.
series	Name of the column in x where id's of the tree-ring series are listed as character values.

last	Name of the column in x which lists the calendar year assigned to the last measured ring (should be a numeric vector).
n_sapwood	Name of the column in x where the number of observed sapwood rings are listed (should be numeric vector).
waneyedge	Name of the column in x indicating the presence (TRUE)/absence (FALSE) of waney edge (should be a logical vector).
sw_data	There are two options: • A character string providing the name of the sapwood data set to use for modelling. It should be one of the data sets listed in sw_data_overview(), • or the name of a data.frame with sapwood data in columns n_sapwood and count, or • or character string naming a column in x that lists for each series the sapwood model to use, e.g. sw_data = "sapwood_model_column".
cred_mass	A numeric scalar [0, 1] specifying the mass within the credible interval (default = .954).
densfun	Name of the density function to fit to the sapwood distribution. Should be one of: • <i>lognormal</i> (the default value), • <i>normal</i>, • <i>weibull</i>, • <i>gamma</i>.

Value

A data.frame with felling date estimates per tree-ring series. Columns include:

- series: series identifier
- last: last measured ring
- n_sapwood: number of sapwood rings
- waneyedge: TRUE/FALSE for waney edge
- lower, upper: numeric bounds for the estimated felling date
- felling_date: a character summary (e.g., "between 1500 and 1510")
- sapwood_model: the sapwood data/model used

See Also

[sw_interval\(\)](#), [sw_data_overview\(\)](#), [sw_interval_plot\(\)](#)

Examples

```
df <- data.frame(
  id = c("trs1", "trs2", "trs3", "trs4"),
  swr = c(7, 1, 10, 12),
  waneyedge = c(FALSE, FALSE, FALSE, TRUE),
  end = c(1482, 1475, 1490, 1498)
```

```

)
fd_report(df,
  series = "id",
  n_sapwood = "swr",
  last = "end",
  sw_data = "Wazny_1990"
)

# Example with different sw_model for individual series
# You can add a user-defined sapwood dataset as well.

sapwood_model_column <- c(
  "Sohar_2012_ELL_c",
  "Wazny_1990",
  "Hollstein_1980",
  "vanDaalen_Norway"
)

df2 <- cbind(df, sw_data = sapwood_model_column)

fd_report(df2,
  series = "id",
  n_sapwood = "swr",
  last = "end",
  sw_data = "sw_data"
)
#' @importFrom rlang .data

```

fellingdater

fellingdater: Tree-ring dating and estimating felling dates of historical timbers

Description

fellingdater provides a comprehensive suite of functions for dendrochronological and dendroarchaeological analysis, covering a workflow from tree-ring **data processing** and **crossdating**, up to the estimation and reporting of **felling dates**. Originally designed to assist dendrochronologists in inferring felling dates from dated tree-ring series with partially preserved sapwood, the package has been expanded to include a workflow for tree-ring dating, analysis, and visualization.

Author(s)

Maintainer: Kristof Haneca <Kristof.Haneca@vlaanderen.be> ([ORCID](#)) [copyright holder]

Other contributors:

- Koen Van Daele ([ORCID](#)) [contributor]
- Ronald Visser ([ORCID](#)) [contributor]
- Antonio Jesus Pérez-Luque ([ORCID](#)) [reviewer]
- Nicholas John Tierney ([ORCID](#)) [reviewer]

See Also

Useful links:

- <https://docs.ropensci.org/fellingdater>
- <https://github.com/ropensci/fellingdater>
- <https://ropensci.github.io/fellingdater>
- Report bugs at <https://github.com/ropensci/fellingdater/issues>

fh_header

Retrieve the HEADER fields of a Heidelberg format (.fh) file

Description

This function reports the HEADER fields from a Heidelberg format (.fh) ring-width file. The header fields are harvested from the .fh-file by the read_fh() function, which stores the HEADER fields from the .fh file as attributes of the data.frame with the measurement data it returns.

Usage

```
fh_header(x)
```

Arguments

x The output of read_fh(x, header = TRUE), a data.frame of class rwl.

Value

A data.frame with 29 header fields.

Examples

```
Doel1 <- system.file("extdata", "DOEL1.fh", package = "fellingdater")
Doel1_trs <- read_fh(Doel1, verbose = FALSE)
fh_header(Doel1_trs)
```

hdi

*Compute the highest posterior density interval (hdi)***Description**

This function returns a matrix with 'upper' and 'lower' limits of the hdi (*highest density interval*), and the associated probability 'p'. The function first sorts the input `data.frame` - with columns 'n_sapwood' and 'p' (the associated probability) - by column 'p' in decreasing order and then it calculates the HDI by finding the first value of the sorted probabilities higher than the specified `cred_mass`. It then finds the indices of the values that are greater than or equal to this threshold, and uses these indices to find the 'upper' and 'lower' limits of the hdi. The function also calculates the probability of the interval. The final result is returned as a matrix with 'lower', 'upper', and 'p' values.

This function is applied in functions [sw_model](#) and [sw_interval](#).

Usage

```
hdi(x, a = "n_sapwood", b = "p", cred_mass = 0.954)
```

Arguments

x	A <code>data.frame</code> with columns <code>n_sapwood</code> and <code>p</code> (the associated probability). <code>x</code> is computed in functions sw_model , sw_interval , and sw_combine .
a	The name of the column in <code>x</code> that lists the number of sapwood rings.
b	The name of the column in <code>x</code> that lists the the associated probability of having <code>n</code> sapwood rings.
cred_mass	A scalar $[0, 1]$ specifying the mass within the credible interval (default = .954).

Value

A matrix with 'upper' and 'lower' limits of the hdi, and the associated probability 'p'.

Examples

```
tmp <- data.frame(n_sapwood = seq(1,30, 1),
                  p = dnorm(seq(1,30, 1), 15, 5))
hdi(tmp, cred_mass = 0.954)
```

Hollstein_1980	<i>Hollstein 1980 sapwood data set.</i>
----------------	---

Description

Sapwood data set for South and Central Germany published by Hollstein in 1980.

Usage

Hollstein_1980

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 490 observations

Source

Hollstein E. 1980. Mitteleuropäische Eichenchronologie: Trierer dendrochronologische Forschungen zur Archäologie und Kunstgeschichte, Trierer Grabungen und Forschungen. Verlag Phillipp von Zabern, Mainz am Rhein.

Examples

```
sw_data_info("Hollstein_1980")
```

```
sw_model("Hollstein_1980", plot = TRUE)
```

Miles_1997_NM	<i>Miles 1997 sapwood data set.</i>
---------------	-------------------------------------

Description

Sapwood data set for the Northern Midland counties (U.K.) - Cheshire, Staffordshire, West Midlands, Northamptonshire, Cambridgeshire, and everything to the north - published by Miles in 1997.

Usage

Miles_1997_NM

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 295 observations

Source

Miles D. 1997. The interpretation, presentation and use of tree-ring dates. Vernacular architecture 28, 40–56. <https://doi.org/10.1179/030554797786050563>

Examples

```
sw_data_info("Miles_1997_NM")
```

```
sw_model("Miles_1997_NM", plot = TRUE)
```

Miles_1997_SC

Miles 1997 sapwood data set.

Description

Sapwood data set for the Southern counties (U.K.), up to and including Gloucestershire, Warwickshire, Bedfordshire, Suffolk and Norfolk, published by Miles in 1997.

Usage

```
Miles_1997_SC
```

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 406 observations

Source

Miles D. 1997. The interpretation, presentation and use of tree-ring dates. Vernacular architecture 28, 40–56. <https://doi.org/10.1179/030554797786050563>

Examples

```
sw_data_info("Miles_1997_SC")

sw_model("Miles_1997_SC", plot = TRUE)
```

Miles_1997_WBC	<i>Miles 1997 sapwood data set.</i>
----------------	-------------------------------------

Description

Sapwood data set for Wales and border counties (U.K.), Shropshire, Hereford and Worcesterthe, published by Miles in 1997.

Usage

```
Miles_1997_WBC
```

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings
count number of times n_sapwood was observed

Details

sample size = 219 observations

Source

Miles D. 1997. The interpretation, presentation and use of tree-ring dates. Vernacular architecture 28, 40–56. <https://doi.org/10.1179/030554797786050563>

Examples

```
sw_data_info("Miles_1997_WBC")

sw_model("Miles_1997_WBC", plot = TRUE)
```

 mov_av

Compute a running mean on a time series

Description

This vectorized function computes a running mean/moving average over a continuous (time) series with a specified window width.

Usage

```
mov_av(x, w = 11, align = "center", edges = "fill")
```

Arguments

- | | |
|-------|---|
| x | A numeric vector of length(x), with the time series data. |
| w | The width of the moving average window. When w is even, one more value from the future is included. |
| align | The alignment of the window relative to the current data point. It can be one of: <ul style="list-style-type: none"> • align = "center": The average is assigned to the center of the window (default). • align = "left": The average includes the current value and the next (w-1) values. • align = "right": The average includes the current value and the previous (w-1) values. |
| edges | Defines how values are computed at the start and end of the series. Options are: <ul style="list-style-type: none"> • edges = "fill": The average is calculated with a decreasing number of values near the edges of the vector (default). • edges = "nofill": The result includes NA values when the window does not cover w values. |

Value

A numeric vector of the same length of x with the computed running mean values.

Examples

```
num_vec <- sample(seq(50, 100, 1), 100, replace = TRUE)
filtered <- mov_av(num_vec, w = 5, align = "center", edges = "nofill")
plot(num_vec, type = "l")
lines(filtered, col = "darkblue")
```

Pilcher_1987	<i>Pilcher 1987 sapwood data set.</i>
--------------	---------------------------------------

Description

Sapwood data set for northern France , published by Pilcher in 1987.

Usage

```
Pilcher_1987
```

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 219 observations

Source

Pilcher J.R. 1987. A 700 year dating chronology for northern France. Applications of tree-ring studies. Current research in dendrochronology and related subjects. BAR International Series 333, 127–139.

Examples

```
sw_data_info("Pilcher_1987")

sw_model("Pilcher_1987", plot = TRUE)
```

read_fh	<i>Read a Heidelberg format (.fh) tree-ring file</i>
---------	--

Description

This function reads a Heidelberg format file (.fh) containing ring width data and returns the HEADER fields as attributes of the data.frame with the measurement data.

The Heidelberg file format is described in detail here: https://www.treeringsociety.org/resources/SOM/Brewer_Murphy_Supp

Usage

```
read_fh(fname, BC_correction = FALSE, verbose = TRUE, header = FALSE)
```

Arguments

fname	A character vector specifying the name of the .fh file to read.
BC_correction	A logical. If TRUE the correction moves BC-dates one year forward.
verbose	A logical. If FALSE, messages are suppressed during the reading process.
header	A logical. If TRUE the HEADER fields are returned as a data.frame; if FALSE, the measurement data is returned.

Details

This function reads .fh file with ring width data in either block (decadal format) or column format (e.g., with comment flags) as used by TSAP program. The function is also capable of reading chronologies or half-chronos in decadal format. The read_fh function is case-insensitive. Information found in the HEADER fields is listed as attributes of the data.frame with the measurement data.

The header fields harvested from the .fh file include:

"Project", "FirstMeasurementDate", "Location", "Town", "Street", "Client", "Longitude", "Latitude", "DateOfSampling", "FirstMeasurementDate", "SapWoodRings", "Comment", "MissingRingsAfter", "InvalidRingsAfter", "MissingringsBefore", "DeltaMissingringsBefore", "ChronoMemberKeycodes", "PersId"

Value

If header is TRUE, a data.frame is returned with HEADER fields as attributes. If header is FALSE, a data.frame of class rwl with ring-width measurements in columns is returned, with (calendar) years as row names.

Author(s)

The original read.fh() function is part of the **dplR package** (<https://github.com/opendendro/dplR>) and was developed by Christian Zang, with new features and patches contributed by Mikko Korpela and Ronald Visser. This read_fh() function expands the functionalities of the original `dplR::read.fh()`.

References

This function is an extension of read.fh() from the **dplR package** (<https://github.com/opendendro/dplR>), developed and maintained by Prof. dr. Andy Bunn (Bunn 2008, Bunn 2010, Bunn et al. 2022) on <https://opendendro.org/>.

Examples

```
Doel1 <- system.file("extdata", "DOEL1.fh", package = "fellingdater")
Doel1_trs <- read_fh(Doel1, verbose = FALSE)
head(Doel1_trs, 10)

Doel1_header <- read_fh(Doel1, verbose = FALSE, header = TRUE)
Doel1_header
```

sgc_for_plot	<i>Detect synchronous growth changes between two tree-ring series (helper for trs_plot_dated())</i>
--------------	---

Description

Identifies years when two tree-ring series show synchronous growth changes (SGC)

- both increasing or both decreasing from the previous year. Based on the logic used in trs_pv() for distinguishing between synchronous and semi-synchronous growth.

Usage

```
sgc_for_plot(x, y)
```

Arguments

x	A data.frame with one column of ring-width values and years as row names.
y	A data.frame with one column of ring-width values and years as row names.

Details

The function compares year-to-year growth direction changes between two series using the same logic as trs_pv():

- Synchronous Growth Changes (SGC): Both series change in the same direction (both + or both -)
- Semi-synchronous/Asynchronous: One series changes while other doesn't, or they change in opposite directions

Only years with synchronous growth changes receive TRUE values. Years with missing values in either series are marked as FALSE. The first year cannot be evaluated (no previous year for comparison) and is marked as FALSE.

Value

A data.frame with two columns:

pv_logi	Logical vector indicating synchronous growth changes (TRUE) or not (FALSE)
year	Numeric vector of years corresponding to the logical values

Examples

```
# Create sample data
years <- 1950:1980
x <- data.frame(series1 = runif(length(years), 0.5, 2.0))
rownames(x) <- years
y <- data.frame(series2 = runif(length(years), 0.8, 1.8))
rownames(y) <- years
```

```
# Detect synchronous growth changes
sync_growth <- sgc_for_plot(x, y)
head(sync_growth)
```

Sohar_2012_ELL_c

Sohar et al. 2012 sapwood data set.

Description

Sapwood data set for Eastern Estonia, Latvia, Lithuania, published by Sohar et al. in 2012.

Usage

Sohar_2012_ELL_c

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 562 observations
!!! sapwood determined by `_color_`

Source

Sohar K., Vitas A. & Läänelaid A. 2012. Sapwood estimates of pedunculate oak (*Quercus robur* L.) in eastern Baltic, Dendrochronologia 30.1, 49–56. <https://doi.org/10.1016/j.dendro.2011.08.001>

Examples

```
sw_data_info("Sohar_2012_ELL_c")

sw_model("Sohar_2012_ELL_c", plot = TRUE)
```

Sohar_2012_ELL_t	<i>Sohar et al. 2012 sapwood data set.</i>
------------------	--

Description

Sapwood data set for Eastern Estonia, Latvia, Lithuania, published by Sohar et al. in 2012.

Usage

Sohar_2012_ELL_t

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 533 observations

!!! sapwood determined by presence of _tyloses_

Source

Sohar K., Vitas A. & Läänelaid A. 2012. Sapwood estimates of pedunculate oak (*Quercus robur* L.) in eastern Baltic, Dendrochronologia 30.1, 49–56. <https://doi.org/10.1016/j.dendro.2011.08.001>

Examples

```
sw_data_info("Sohar_2012_ELL_t")
```

```
sw_model("Sohar_2012_ELL_t", plot = TRUE)
```

Sohar_2012_FWE_c	<i>Sohar et al. 2012 sapwood data set.</i>
------------------	--

Description

Sapwood data set for Southern Finland and western Estonia, published by Sohar et al. in 2012.

Usage

Sohar_2012_FWE_c

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 98 observations

!!! sapwood determined by _color_

Source

Sohar K., Vitas A. & Läänelaid A. 2012. Sapwood estimates of pedunculate oak (*Quercus robur* L.) in eastern Baltic, *Dendrochronologia* 30.1, 49–56. <https://doi.org/10.1016/j.dendro.2011.08.001>

Examples

```
sw_data_info("Sohar_2012_FWE_c")
```

```
sw_model("Sohar_2012_FWE_c", plot = TRUE)
```

Sohar_2012_FWE_t

Sohar et al. 2012 sapwood data set.

Description

Sapwood data set for Southern Finland and western Estonia, published by Sohar et al. in 2012.

Usage

Sohar_2012_FWE_t

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 97 observations

!!! sapwood determined by the presence of _tyloses_

Source

Sohar K., Vitas A. & Läänelaid A. 2012. Sapwood estimates of pedunculate oak (*Quercus robur* L.) in eastern Baltic, *Dendrochronologia* 30.1, 49–56. <https://doi.org/10.1016/j.dendro.2011.08.001>

Examples

```
sw_data_info("Sohar_2012_FWE_t")

sw_model("Sohar_2012_FWE_t", plot = TRUE)
```

sw_combine

Combine multiple sapwood estimates into a single felling date range

Description

This function evaluates whether multiple sapwood estimates can be combined into a single felling date range and computes the resulting interval if possible.

Usage

```
sw_combine(
  x,
  series = "series",
  last = "last",
  n_sapwood = "n_sapwood",
  waneyedge = "waneyedge",
  sw_data = "Hollstein_1980",
  densfun = "lognormal",
  cred_mass = 0.954,
  hdi = TRUE,
  plot = FALSE
)
```

Arguments

x	A data.frame with at least four columns, providing information on the ID's of the tree-ring series, the number of sapwood rings observed, the presence of waney edge and the calendar date assigned to the last measured ring. A column describing the sapwood data set to be used for modelling and the computation of the hdi can be provided as well.
series	Name of the column in x where ID's of the tree-ring series are listed as character values.
last	Name of the column in x where calendar years assigned to the last measured ring are listed (should be numeric).

n_sapwood	Name of the column in x where the number of observed sapwood rings are listed (should be numeric).
waneyedge	Name of the column in x indicating the presence (TRUE)/absence (FALSE) of waney edge (should be a logical vector).
sw_data	The name of the sapwood data set to use for modelling. It should be one of the data sets listed in sw_data_overview() , or the name of a data.frame with sapwood data in columns n_sapwood and count.
densfun	Name of the density function fitted to the sapwood data set. Should be one of: • <i>lognormal</i> (the default value), • <i>normal</i>, • <i>weibull</i>, • <i>gamma</i>.
cred_mass	A scalar [0, 1] specifying the mass within the credible interval (default = .954).
hdi	A logical parameter. If TRUE, the lower and upper limit of the highest density interval (credible interval) is given for the combined felling date. When FALSE, a matrix is returned with scaled p values for calendar years covering the combined estimate of the felling date range.
plot	A logical parameter. If TRUE a ggplot-plot style graph is returned of the individual and combined estimate of the felling date. If FALSE, a list with numeric output of the modelling process is returned.

Value

@return Depends on plot:

- If TRUE: a ggplot-style plot showing individual and combined felling date estimates.
- If FALSE: a list with: - \$raw_data: matrix of combined and individual densities. - \$sapwood_data: sapwood data set used. - \$sapwood_model: density function used. - \$cred_mass: credibility mass for the interval. - \$hdi_combine: HDI for the combined felling date range. - \$individual_series: summary with agreement indices. - \$A_model: overall model agreement index. - \$A_c: critical threshold (default 60%). - \$model_summary: text summary of the result. Depends on the value of plot. The rationale of the agreement index is outlined by Bronk Ramsey (1995; 2009)

References

- Bronk Ramsey, C. (1995) Radiocarbon calibration and analysis of stratigraphy: the OxCal program. *Radiocarbon* **37**, 425–430. <https://doi.org/10.1017/S0033822200030903>
- Bronk Ramsey, C. (2009) Bayesian analysis of radiocarbon dates. *Radiocarbon* **51**, 337–360. <https://doi.org/10.1017/S0033822200033865>

See Also

[sw_combine_plot\(\)](#)

Examples

```
# Example with all series preserving sapwood

sw_combine(sw_example1, plot = TRUE)

# Example with one exact felling date (= waney edge preserved)

sw_example2
sw_combine(sw_example2, plot = TRUE)

# Example with multiple exact felling dates

sw_example3
sw_combine(sw_example3, plot = FALSE)
```

sw_combine_plot	<i>Plot the output of <code>sw_combine()</code></i>
-----------------	---

Description

This function plots the result of `sw_combine()` and the interval for, or the exact felling date of, a group of tree-ring series, with agreement indices for the global model.

Usage

```
sw_combine_plot(x)
```

Arguments

x	Output of <code>sw_combine()</code> .
---	---------------------------------------

Value

A ggplot-style graph, with calendar years on the X-axis and the probability (p) on the Y-axis. Dots represent exact felling dates.

Examples

```
sw_example2
tmp <- sw_combine(sw_example2)
sw_combine_plot(tmp)
```

sw_data_info	<i>Detailed information on a sapwood data set</i>
--------------	---

Description

This function returns more information on a sapwood data set. It lists the correct citation for the data set, the area the data set is supposed to represent, the number of observations and summary stats of the raw data.

Usage

```
sw_data_info(x = NULL)
```

Arguments

x	The name of a sapwood data set. you can use sw_data_overview() to get an overview of available data sets.
---	---

Value

A list with the following components:

- data: the name of the data set,
- citation: the correct citation for the data set,
- area: the area represented by the data set,
- n_observations: the number of observations in the data set and,
- summary_raw_data: summary stats of the raw data.

Examples

```
# Get detailed information on a sapwood data set
sw_data_info("Hollstein_1980")

# Retrieve the citation for a specific data set
sw_data_info("Sohar_2012_FWE_c")$citation
```

sw_data_overview	<i>Overview of available sapwood data sets in the fellingdater package</i>
------------------	--

Description

This function provides an overview of the available sapwood data sets and models within the fellingdater package.

Usage

```
sw_data_overview()
```

Value

A character vector with the names of all available data sets with sapwood counts in the fellingdater package.

Examples

```
# Get an overview of available sapwood data sets
sw_data_overview()
```

sw_example0	<i>Example dataset 0</i>
-------------	--------------------------

Description

A dataset in which all series have preserved sapwood. Unconventional variable names.

Usage

```
sw_example0
```

Format

A tibble of 4 variables:

trs unique ID of the tree-ring series
end calendar year assigned to the last measured ring
swr number of observed sapwood rings
bark waney edge present TRUE/FALSE

Examples

```
sw_example0
```

sw_example1

Example dataset 1

Description

A dataset in which all series have preserved sapwood rings.

Usage

```
sw_example1
```

Format

A tibble of 4 variables:

series unique ID of the tree-ring series

last calendar year assigned to the last measured ring

n_sapwood number of observed sapwood rings

waneyedge waney edge present TRUE/FALSE

Examples

```
sw_example1
```

sw_example2

Example dataset 2

Description

A dataset in which one series has an exact felling date (= waney edge present).

Usage

```
sw_example2
```

Format

A tibble of 4 variables:

series unique ID of the tree-ring series

last calendar year assigned to the last measured ring

n_sapwood number of observed sapwood rings

waneyedge waney edge present TRUE/FALSE

Examples

```
sw_example2
```

```
sw_example3
```

Example dataset 3

Description

A dataset with multiple exact felling dates.

Usage

```
sw_example3
```

Format

A tibble of 4 variables:

series unique ID of the tree-ring series

last calendar year assigned to the last measured ring

n_sapwood number of observed sapwood rings

waneyedge waney edge present TRUE/FALSE

Examples

```
sw_example3
```

```
sw_example4
```

Example dataset 4

Description

A combination of series with and without sapwood rings.

Usage

```
sw_example4
```

Format

A tibble of 4 variables:

series unique ID of the tree-ring series
last calendar year assigned to the last measured ring
n_sapwood number of observed sapwood rings
waneyedge waney edge present TRUE/FALSE

Examples

sw_example4

sw_example5

Example dataset 5

Description

None of the series in this dataset have preserved sapwood rings.

Usage

sw_example5

Format

A tibble of 4 variables:

series unique ID of the tree-ring series
last calendar year assigned to the last measured ring
n_sapwood number of observed sapwood rings
waneyedge waney edge present TRUE/FALSE

Examples

sw_example5

`sw_example6`*Example dataset 6*

Description

A test dataset for `sw_sum()`.

Usage

```
sw_example6
```

Format

A tibble of 4 variables:

series unique ID of the tree-ring series

last calendar year assigned to the last measured ring

n_sapwood number of observed sapwood rings

waneyedge waney edge present TRUE/FALSE

Examples

```
sw_example6
```

`sw_example7`*Example dataset 7*

Description

A test dataset for `sw_sum()`.

Usage

```
sw_example7
```

Format

A tibble of 4 variables:

series unique ID of the tree-ring series

last calendar year assigned to the last measured ring

n_sapwood number of observed sapwood rings

waneyedge waney edge present TRUE/FALSE

Examples

```
sw_example7
```

sw_interval	<i>Computes the limits of the felling date range</i>
-------------	--

Description

This function computes the probability density function (PDF) and the highest posterior density interval (HDI) of the felling date range, based on the observed number of sapwood rings, their dating, and a selected sapwood dataset and model.

Usage

```
sw_interval(
  n_sapwood = NA,
  last = 1,
  hdi = FALSE,
  cred_mass = 0.954,
  sw_data = "Hollstein_1980",
  densfun = "lognormal",
  plot = FALSE
)
```

Arguments

n_sapwood	A numeric. The number of observed sapwood rings.
last	A numeric. The calendar year assigned to the outermost sapwood ring (default = 1).
hdi	Logical. If TRUE, the function returns the lower and upper bounds of the highest density interval (HDI). If FALSE, it returns a data frame with scaled probability values for each possible number of sapwood rings.
cred_mass	A numeric scalar in the range of $[0, 1]$, specifying the mass of the credible interval (default = .954)
sw_data	The name of the sapwood data set to use for modelling. It should be one of the data sets listed in sw_data_overview() , or the name of a data.frame with sapwood data in columns n_sapwood and count.
densfun	A character string naming the distribution to fit. One of: • <i>lognormal</i> (default), • <i>normal</i>, • <i>weibull</i>, • <i>gamma</i>.
plot	A logical.

- If TRUE, a ggplot-style graph is returned of the individual sapwood model and estimate of the felling date range.
- If FALSE, a list with the numeric output of the modelling process is returned.

Value

The type of output depends on the value of hdi:

- If hdi = TRUE, a numeric vector with the lower and upper limits of the highest density interval (HDI). Attributes include cred_mass, sw_data, and the fitted model.
- If hdi = FALSE, a data frame with scaled probability values (p) for each year after the last dated sapwood ring.

Examples

```
# 10 sapwood rings observed and the Wazny 1990 sapwood model:
sw_interval(
  n_sapwood = 10,
  last = 1234,
  hdi = TRUE,
  cred_mass = .95,
  sw_data = "Wazny_1990",
  densfun = "lognormal",
  plot = FALSE
)
# same example as above, but with numerical output (hdi = FALSE):
sw_interval(
  n_sapwood = 10,
  last = 1234,
  hdi = FALSE,
  cred_mass = .95,
  sw_data = "Wazny_1990",
  densfun = "lognormal",
  plot = FALSE
)
```

sw_interval_plot

Plot function for the output of `sw_interval()`

Description

Returns a ggplot-style graph of the probability density function for the felling date range, as computed by `sw_interval()`.

Usage

```
sw_interval_plot(x, area_fill = "tomato3", line_color = "tomato3")
```

Arguments

x	Output of <code>sw_interval()</code> with <code>hdi = FALSE</code> .
area_fill	Fill color for the area under fitted distribution.
line_color	Line color for the fitted distribution.

Value

A ggplot-style graph, with calendar years on the X-axis and the probability (p) on the Y-axis.

Examples

```
tmp <- sw_interval(
  n_sapwood = 10,
  last = 1000,
  hdi = FALSE,
  cred_mass = .95,
  sw_data = "Hollstein_1980",
  densfun = "lognormal",
  plot = FALSE
)
sw_interval_plot(tmp, area_fill = "forestgreen", line_color = "forestgreen")
```

sw_model	<i>Model sapwood data and compute the highest posterior density interval</i>
----------	--

Description

This function fits a distribution to a data set of observed sapwood numbers and computes the highest posterior density interval (hdi) for a given credibility mass.

Usage

```
sw_model(
  sw_data = "Hollstein_1980",
  densfun = "lognormal",
  cred_mass = 0.954,
  source = NA,
  plot = FALSE
)
```

Arguments

sw_data	The name of the sapwood data set to use for modelling. It should be one of the data sets listed in <code>sw_data_overview()</code> , or the name of a data.frame with sapwood data in columns <code>n_sapwood</code> and <code>count</code> .
---------	---

densfun	Name of the density function to fit to the sapwood data set. Should be one of: • <i>lognormal</i> (the default value), • <i>normal</i>, • <i>weibull</i>, • <i>gamma</i>.
cred_mass	A scalar in the range of $[0, 1]$ specifying the mass within the credible interval (default = .954).
source	A character string with info on the data source. This info is incorporated in the title of the plot when a sapwood data set is provided from a user-defined data.frame.
plot	A logical. If TRUE a plot of the fitted density function is returned. When FALSE, a list with numeric output of the modelling process is returned.

Value

The return value depends on the plot parameter.

- if plot is TRUE, a ggplot-style graph is returned.
- if plot is FALSE, a list with the numeric output of the modelling process is returned.

Examples

```
sw_model(
  sw_data = "Sohar_2012_ELL_c",
  densfun = "lognormal",
  cred_mass = .095,
  plot = FALSE
)

sw_model("Hollstein_1980")
```

sw_model_plot	<i>Plot function for the output of sw_model()</i>
---------------	---

Description

Returns a ggplot-style graph of the probability density function modelled to a data set of observed sapwood numbers, as computed by [sw_model\(\)](#).

Usage

```
sw_model_plot(
  x,
  bar_fill = "steelblue3",
  bar_color = "grey60",
  line_color = "red3"
)
```

Arguments

x	Output of <code>sw_model()</code> .
bar_fill	Fill color for the bars (original data).
bar_color	Line color for the bars (original data).
line_color	Line color for the fitted distribution.

Value

A ggplot-style graph.

Examples

```
tmp <- sw_model(
  sw_data = "Hollstein_1980",
  densfun = "lognormal",
  cred_mass = .95,
  plot = FALSE
)
sw_model_plot(tmp,
  bar_fill = "forestgreen",
  bar_color = "darkgreen",
  line_color = "tomato3"
)
```

sw_sum

Compute the summed probability of multiple felling date ranges

Description

Computes the summed probability density (SPD) for a set of felling date ranges.

Usage

```
sw_sum(
  x,
  series = "series",
  last = "last",
  n_sapwood = "n_sapwood",
  waneyedge = "waneyedge",
  sw_data = "Hollstein_1980",
  densfun = "lognormal",
  cred_mass = 0.954,
  plot = FALSE,
  scale_p = FALSE
)
```

Arguments

x	A data.frame with columns c("series", "swr", "waneyedge", "last"). Could be the output of <code>read_fh()</code> . Column "sw_model" is optional in order to specify the sapwood model to use for each series.
series	Name of the column in x where ID's of the tree-ring series are listed as character values.
last	Name of the column in x where calendar years assigned to the last measured ring are listed (should be numeric).
n_sapwood	Name of the column in x where the number of observed sapwood rings are listed (should be numeric)
waneyedge	Name of the column in x indicating the presence (TRUE)/absence (FALSE) of waney edge (should be a logical).
sw_data	The name of the sapwood data set to use for modelling. It should be one of the data sets listed in <code>sw_data_overview()</code> , or the name of a data.frame with sapwood data in columns n_sapwood and count.
densfun	Name of the density function fitted to the sapwood data set. Should be one of: • <i>lognormal</i> (the default value), • <i>normal</i>, • <i>weibull</i>, • <i>gamma</i>.
cred_mass	A scalar [0, 1] specifying the mass within the credible interval (default = .954).
plot	A logical. • If TRUE, <code>sw_sum_plot()</code> is triggered and a ggplot-style graph is returned with the summed probability density (SPD). • If FALSE, a list with the numeric output of the modelling process is returned.
scale_p	A logical. If TRUE the summed probability density is scaled to 1 (default).

Value

Depends on the value of plot.

See Also

`sw_sum_plot()`

Examples

```
sw_example7
```

```
sw_sum(sw_example7, densfun = "lognormal", cred_mass = 0.63, plot = FALSE)
```

sw_sum_plot

*Plot the output of `sw_sum()`***Description**

This function creates a visualization of the summed probability density (SPD) output from `sw_sum()`. It displays both the SPD as a bar plot and exact felling dates (waney edge series) as stacked points. A smoothing spline is also added to reveal long-term trends in felling activity.

Usage

```
sw_sum_plot(
  x,
  bar_col = "steelblue",
  trend_col = "red3",
  dot_col = "steelblue4",
  dot_size = 2,
  dot_shape = 21,
  window_smooth = 11
)
```

Arguments

<code>x</code>	A <code>data.frame</code> , typically the output of <code>sw_sum()</code> with <code>plot = FALSE</code> . Must contain columns "year" and "spd". If available, "spd_wk" (waney edge counts) will be used to add symbols representing exact felling years.
<code>bar_col</code>	Fill color for the SPD bars. Default is "steelblue".
<code>trend_col</code>	Color of the smoothing spline line.
<code>dot_col</code>	Fill color of the symbols representing exact felling dates (waney edge). Default is "steelblue4".
<code>dot_size</code>	Size of the felling date symbols. Default is 2.
<code>dot_shape</code>	Shape code for the felling date symbols. See <code>?points</code> for options. Default is 21 (circle).
<code>window_smooth</code>	Numeric value specifying the smoothing window width (in years) for calculating the moving average trend line. Default is 11.

Value

A `ggplot` object showing:

- The SPD as a bar plot.
- A smoothing spline through the SPD.
- Stacked symbols for exact felling years (waney edge), if available.

See Also

[sw_sum\(\)](#) to generate the SPD data.

Examples

```
sw_example6
tmp <- sw_sum(sw_example6, plot = FALSE)
sw_sum_plot(tmp,
  bar_col = "burlywood1",
  trend_col = "brown",
  dot_col = "orange",
  dot_shape = 23, dot_size = 5
)
```

 trs_crossdate

Sliding window crossdating analysis using multiple statistical methods

Description

This function performs crossdating analysis using a sliding window with multiple statistical methods: Parallel Variation (GLK/SGC), Baillie & Pilcher t-values, Hollstein t-values, and standard Pearson correlation t-values. It tests series at different temporal lags to find a potential date positions.

Usage

```
trs_crossdate(
  x,
  y = NULL,
  min_overlap = 30,
  sliding = TRUE,
  top_n = NULL,
  rank_by = NULL,
  pb = TRUE
)
```

Arguments

- | | |
|-------------|---|
| x | A data frame of test tree-ring series in rwl format (years as rownames, series as columns). All columns must be numeric. |
| y | A data frame of reference tree-ring series in rwl format. If NULL, uses x for self-comparison. Default is NULL. |
| min_overlap | Integer. Minimum number of overlapping years required between series pairs to compute statistics. Must be ≥ 3 . Default is 30. |
| sliding | Logical. If TRUE, performs sliding window analysis testing multiple temporal lags. If FALSE, tests only at lag 0 (contemporary positioning). Default is TRUE. |

top_n	Integer or NULL. If specified, returns only the top n matches per test series, ranked by the statistic given in rank_by. Requires rank_by to be specified. Default is NULL.
rank_by	Character or NULL. Statistic to rank by when sorting results per test series. If NULL, no additional sorting is applied beyond default ordering. Options are "r_pearson", "t_St", "t_BP", "t_Ho", "sgc", "ssgc", "glk". Default is NULL.
pb	Logical. If TRUE (default), displays a progress bar during computation. Set to FALSE to suppress progress bar output, useful when calling the function programmatically or in non-interactive sessions.

Details

This function implements sliding window crossdating, a fundamental technique in dendrochronology for dating wood samples of unknown age. The process involves:

1. Determining the range of possible lags based on series overlap
2. For each lag, shifting the test series temporally
3. Computing all crossdating statistics
4. Returning results for each tested position

The sliding window approach allows identification of the temporal position where test and reference series show maximum similarity, which indicates the most likely dating for the test series.

Statistical Methods:

- t_BP: t-values according to the Baillie & Pilcher (1973) algorithm
- t_Ho: t-values according to the Hollstein (1980) algorithm
- sgc, sgc_p: synchronous growth change and associated significance level (Visser 2020)
- ssgc: semi-synchronous growth change (Visser 2020)
- glk, glk_p: 'Gleichläufigkeit' or 'percentage of parallel variation' (Buras & Wilmking 2015; Eckstein & Bauch 1969; Huber 1942), and significance level (p-value) associated with the glk-value (Jansma 1995)
- r_pearson: the Pearson's correlation coefficient
- t_St: Student's t-value

Sorting and Filtering:

- If rank_by = NULL: Results are returned with default ordering (series, reference, first)
- If rank_by is specified: Results are sorted per test series by the chosen statistic (high to low)
- If top_n is specified: Only the top n matches per test series are returned (after optional sorting)

Value

A data frame with crossdating results containing the following columns:

series Name of the test series from x

length Length of the test series

first Date of the first ring at the particular lag position
last Date of the last ring at the particular lag position
reference Name of the reference series
ref_first Date of the first ring in the reference series
ref_last Date of the last ring in the reference series
overlap Number of overlapping years between series and reference
r_pearson Pearson correlation coefficient
t_St Student's t-statistic
t_BP Baillie & Pilcher t-statistic
t_Ho Hollstein t-statistic
sgc Synchronous Growth Change (SGC) statistic
ssgc Semi-SGC statistic
sgc_p SGC p-value
glk Gleichläufigkeit (GLK) statistic
glk_p GLK p-value

References

- Baillie, M.G.L., Pilcher, J.R. (1973) A simple crossdating program for tree-ring research. *Tree-Ring Bulletin* **33**, 7–14. <http://hdl.handle.net/10150/260029>
- Buras, A. and Wilmking, M. (2015) Correcting the calculation of Gleichläufigkeit, *Dendrochronologia* **34**, 29-30. <https://doi.org/10.1016/j.dendro.2015.03.003>
- Eckstein, D. and Bauch, J. (1969) Beitrag zur Rationalisierung eines dendrochronologischen Verfahrens und zur Analyse seiner Aussagesicherheit. *Forstwissenschaftliches Centralblatt*, **88**(1), 230-250.
- Huber, B. (1943) Über die Sicherheit jahrringchronologischer Datierung. *Holz als Roh- und Werkstoff* **6**, 263-268. <https://doi.org/10.1007/BF02603303>
- Hollstein E. (1980) Mitteleuropäische Eichenchronologie. Trierer dendrochronologische Forschungen zur Archäologie und Kunstgeschichte, Trierer Grabungen und Forschungen **11**, Mainz am Rhein.
- Jansma, E. (1995) RememberRINGS; The development and application of local and regional tree-ring chronologies of oak for the purposes of archaeological and historical research in the Netherlands, *Nederlandse Archeologische Rapporten* **19**, Rijksdienst voor het Oudheidkundig Bodemonderzoek, Amersfoort. <https://dspace.library.uu.nl/handle/1874/45149>
- Schweingruber, F. H. (1988) Tree rings: basics and applications of dendrochronology, Kluwer Academic Publishers, Dordrecht, Netherlands, 276 p.
- Visser, R.M. (2020) On the similarity of tree-ring patterns: Assessing the influence of semi-synchronous growth changes on the Gleichläufigkeit for big tree-ring data sets, *Archaeometry* **63**, 204-215. <https://doi.org/10.1111/arcm.12600>

See Also

[trs_pv](#), [trs_tbp](#), [trs_tho](#)

Examples

```
## Not run:
# Create sample data
rwl_test <- trs_pseudo_rwl(n_series = 2, series_length = 80, end_date = 1950)
rwl_ref <- trs_pseudo_rwl(n_series = 3, series_length = 150, end_date = 2000)

# Full sliding window analysis with default ordering
result <- trs_crossdate(rwl_test, rwl_ref, min_overlap = 30)

# Sort by Hollstein t-values
result_sorted <- trs_crossdate(rwl_test, rwl_ref, min_overlap = 30, rank_by = "t_Ho")

# Contemporary positioning only
contemp <- trs_crossdate(rwl_test, rwl_ref, sliding = FALSE)

# Self-comparison for quality control
self_check <- trs_crossdate(rwl_ref, sliding = FALSE)

# Get top 3 matches per series ranked by Hollstein t-values
top_matches <- trs_crossdate(rwl_test, rwl_ref, top_n = 3, rank_by = "t_Ho")

## End(Not run)
```

| trs_end_date |

| *Assign calendar years to tree-ring series in an rwl-style data frame* |

Description

Sets the row names (years) of an rwl-style data frame based on a specified end year. Optionally trims trailing rows with only NA values.

Usage

```
trs_end_date(x, end_year = 2025, trim = FALSE)
```

Arguments

x	A data.frame of class "rwl", with each column representing a tree-ring series.
end_year	An integer. The calendar year assigned to the most recent tree ring (last row).
trim	Logical. If TRUE, the data frame is trimmed to remove leading and trailing rows with no data (see trs_trim).

Value

A data.frame with row names set to calendar years.

Examples

```
rw1 <- trs_pseudo_rwl(n_series = 2, series_length = 50)
trs_end_date(rw1, end_year = 2000)
trs_end_date(rw1, end_year = 2000, trim = TRUE)
```

trs_plot_dated	<i>Plot a dated tree-ring series with a reference series or chronology</i>
----------------	--

Description

Creates a line plot of a dated tree-ring series and a reference chronology, with optional z-score standardization, statistical annotations, and visual highlighting of synchronous growth changes (SCG).

Usage

```
trs_plot_dated(
  x,
  y,
  end_year = NULL,
  zscore = TRUE,
  pv_highlight = TRUE,
  pv_alpha = 0.2,
  show_stats = TRUE,
  labels = TRUE,
  label_size = 3,
  x_breaks = NULL
)
```

Arguments

x	A data.frame (typically of class 'rw1') with one column of ring-width values. Row names must be numeric and represent calendar years.
y	A data.frame (typically of class 'rw1') with one column of ring-width values. Row names must be numeric and represent calendar years.
end_year	Optional numeric. If provided, the rownames of x will be set to match this year for the last ring (using trs_end_date()).
zscore	Logical. If TRUE, the ring-width series are standardized to z-scores before plotting. Defaults to TRUE.
pv_highlight	Logical. If TRUE, highlights regions of parallel variation (synchronous growth change - sgc) using shaded bars. Defaults to TRUE.
pv_alpha	Numeric between 0 and 1. Controls the transparency of the parallel variation highlight. Defaults to 0.2.
show_stats	Logical. If TRUE, displays crossdating statistics (e.g., overlap, correlation, sgc, t-values) as a plot subtitle. Defaults to TRUE.

labels	Logical. If TRUE, displays year labels and points at the start and end positions. If FALSE, neither labels nor points are shown. Defaults to TRUE.
label_size	Numeric. Controls the size of the year labels. Defaults to 3.
x_breaks	Numeric. The interval between x-axis tick marks (in years). If NULL (default), the interval is chosen automatically based on the length of the series. Must be a positive number if provided.

Details

The function assumes that x and y are already calendar-dated and aligned by row names. It performs internal trimming and optional z-scoring before plotting. Crossdating statistics are computed using `trs_crossdate()`, and regions of synchronous growth change (SGC) are highlighted. Required helper functions include: `trs_trim`, `trs_zscore`, `trs_end_date`, and `sgc_for_plot`. Statistics are derived from `trs_crossdate`.

Required packages: **ggplot2**, **ggtext**, **plyr**

Value

A ggplot object showing the plotted time series.

Examples

```
x <- trs_pseudo_rwl(n_series = 1, series_length = 80, end_date = 1500, prefix = "trs_")
y <- trs_pseudo_rwl(n_series = 1, series_length = 400, end_date = 1700, prefix = "ref_")
trs_plot_dated(x, y)
```

trs_plot_rwl	<i>Plot all series in a tree-ring width dataframe</i>
--------------	---

Description

Creates a line plot of all tree-ring series in an rwl dataframe, with optional z-score standardization and faceting options.

Usage

```
trs_plot_rwl(
  x,
  zscore = TRUE,
  facet = FALSE,
  ncol = 1,
  color = NULL,
  scale_y = "fixed"
)
```

Arguments

<code>x</code>	A data.frame (typically of class 'rwl') with multiple columns of ring-width values. Row names must be numeric and represent calendar years.
<code>zscore</code>	Logical. If TRUE, the ring-width series are standardized to z-scores before plotting. Defaults to TRUE.
<code>facet</code>	Logical. If TRUE, creates a faceted plot with each series in a separate panel. If FALSE, plots all series overlaid on the same plot. Defaults to FALSE.
<code>ncol</code>	Numeric. Number of columns for faceted plots (only used when facet = TRUE). Defaults to 1.
<code>color</code>	Character string or vector specifying line color(s). When NULL and facet = FALSE, each series gets a different color. When a single color is specified, all lines use that color. When a vector of colors is provided, each series gets the corresponding color. Defaults to NULL.
<code>scale_y</code>	Character string specifying y-axis scaling for faceted plots. Options are "fixed", "free", "free_x", or "free_y". Defaults to "fixed".

Details

The function assumes that the rwl dataframe is already calendar-dated with years as row names. When facet = FALSE, all series are plotted on the same axes with different colors. When facet = TRUE, each series gets its own panel for easier individual inspection. When providing a vector of colors, the length must match the number of series in the data. Required helper functions include: [trs_trim](#) and [trs_zscore](#).

Required packages: **ggplot2**, **plyr**, **tidyr**, **dplyr**

Value

A ggplot object showing the plotted time series.

Examples

```
rwl_data <- trs_pseudo_rwl(n_series = 5, series_length = c(75, 100), end_date = c(2000, 2025))

# Overlaid plot with z-scores
trs_plot_rwl(rwl_data, zscore = TRUE, facet = FALSE)

# Faceted plot with raw values
trs_plot_rwl(rwl_data, zscore = FALSE, facet = TRUE, ncol = 1)

# Custom colors for each series
col_vec <- c("#0fa3b1", "#07beb8", "#3dccc7", "#68d8d6", "#9ceaf")
trs_plot_rwl(rwl_data, color = c("tomato3", "navyblue", "forestgreen", "purple", "orange"))
trs_plot_rwl(rwl_data, color = col_vec, zscore = FALSE)
```

trs_pseudo_rwl	Create a pseudo-population of tree-ring series
----------------	--

Description

Generates a synthetic tree-ring width dataset in rwl format, suitable for testing crossdating and chronology construction methods.

Usage

```
trs_pseudo_rwl(
  n_series = 10,
  series_length = 100,
  end_date = 2024,
  related = TRUE,
  trend = NULL,
  neg_lin_param = -0.1,
  neg_exp_param = 0.05,
  ar = TRUE,
  ar_params = 0.7,
  mean_rw = 100,
  noise_sd = 8,
  common_signal_strength = 0.75,
  prefix = "trs_",
  seed = NULL
)
```

Arguments

n_series	Integer. Number of tree-ring series to generate.
series_length	Numeric or numeric vector of length 2. If a single number, all series have this fixed length. If a vector of two values, series lengths are randomly sampled between the minimum and maximum values (inclusive).
end_date	Numeric or numeric vector of length 2. If a single number, all series end in this year. If a vector of two values, series end years are randomly sampled within this range.
related	Logical. If TRUE, all series share a common growth pattern (i.e., internally correlated). If FALSE, each series is generated independently.
trend	Character. Type of trend to apply to series. Options: "neg_lin" for a negative linear trend, "neg_exp" for a negative exponential trend, or NULL for no trend.
neg_lin_param	Numeric. Slope for the linear decline (typically between -0.1 and -1.0).
neg_exp_param	Numeric. Rate parameter for the exponential decline (typically between 0.01 and 0.1).
ar	Logical. If TRUE, adds autocorrelation using an AR model. Note: when related = TRUE, AR is applied to the common base signal; when related = FALSE, AR is applied to each series individually.

ar_params	Numeric vector. Autoregressive coefficients (e.g., <code>c(0.8)</code> , or <code>c(0.6, 0.2)</code>).
mean_rw	Numeric. Mean ring width around which the data is centered (default is 100).
noise_sd	Numeric. Standard deviation of noise added to each series (default is 8).
common_signal_strength	Numeric. Strength of common signal when <code>related = TRUE</code> (0-1). Higher values produce higher inter-series correlations (<code>rbar</code>). Default 0.75 typically produces <code>rbar</code> around 0.5-0.7, which is realistic for tree-ring chronologies.
prefix	Character. Prefix used when naming each tree-ring series (default is <code>"trs_"</code>).
seed	Numeric. Random seed for reproducible results. If <code>NULL</code> , no seed is set.

Value

A data frame of class `"rwl"`, with years as row names and each column representing one synthetic tree-ring series.

Examples

```
# Basic usage
pseudo_data <- trs_pseudo_rwl(n_series = 5, series_length = 50)

# Variable lengths and end dates with reproducible results
pseudo_data <- trs_pseudo_rwl(
  n_series = 10,
  series_length = c(80, 120),
  end_date = c(2020, 2024),
  seed = 123
)

# Related series with strong common signal (high rbar)
pseudo_data <- trs_pseudo_rwl(
  n_series = 8,
  series_length = 100,
  related = TRUE,
  common_signal_strength = 0.85, # Should produce rbar ~ 0.7-0.8
  trend = "neg_exp",
  ar = TRUE,
  ar_params = c(0.7, 0.2),
  seed = 456
)
```

trs_pv	<i>Computes statistics to describe parallel variations between two rwl-style data frames</i>
--------	--

Description

Compares tree-ring series in two data frames using Gleichläufigkeit (GLK) and Synchronous Growth Changes (SGC), adapted from the `dplR` package.

Usage

```
trs_pv(x, y = NULL, min_overlap = 50, prob = TRUE, as_df = FALSE)
```

Arguments

x	A data frame of test tree-ring series (rwl format: years in rows, series in columns).
y	A data frame of reference series (rwl format).
min_overlap	Integer. Minimum number of overlapping years required for computing GLK/SGC statistics. Default is 50.
prob	Logical. If TRUE (default), computes two-tailed p-values for GLK and SGC.
as_df	Logical. If TRUE, returns a long-format data.frame with all combinations and results. Default is FALSE.

Value

If `as_df = FALSE`, a list with components:

glk Matrix of Gleichläufigkeit values.

glk_p Matrix of GLK p-values (if `prob = TRUE`).

sgc Matrix of synchronous growth change values.

ssgc Matrix of semi-synchronous growth change values.

sgc_p Matrix of SGC p-values (if `prob = TRUE`).

overlap Matrix of the number of overlapping years used in each comparison.

If `as_df = TRUE`, returns a data frame with columns: `series`, `reference`, `glk`, `glk_p`, `sgc`, `ssgc`, `sgc_p`, and `overlap`.

Examples

```
# Create sample data,
trs <- trs_pseudo_rwl(n_series = 5, series_length = c(80, 100), end_date = c(2010, 2020))
trs_pv(trs)$sgc
```

| trs_remove |

| *Removes one or more series from an rwl-style data frame* |
Description

Removes one or more tree-ring series (columns) from a data frame of class "rwl". Optionally adds the rownames (assumed to be years) as a column.

Usage

```
trs_remove(x, series, trim = FALSE)
```

Arguments

x	A data.frame of class "rwl", where each column is a tree-ring width series and rownames represent years.
series	A character vector specifying the names of the series (columns) to remove.
trim	Logical. If TRUE, leading and trailing all-NA rows are removed using trs_trim().

Value

A data.frame with the specified series removed.

Examples

```

rwl <- trs_pseudo_rwl(n_series = 3, series_length = 60, end_date = 2000)
trs_remove(rwl, series = "trs_1", trim = TRUE)
trs_remove(rwl, series = c("trs_1", "trs_3"))

```

trs_select	<i>Select series from an rwl-style data frame</i>
------------	---

Description

Extracts one or more tree-ring series (columns) from a data frame of class "rwl". Optionally adds the rownames (assumed to be years) as a column.

Usage

```
trs_select(x, series, trim = FALSE)
```

Arguments

x	A data.frame of class "rwl", where each column is a tree-ring width series and rownames represent years.
series	A character string or vector specifying the name(s) of the series (columns) to extract.
trim	Logical. If TRUE, leading and trailing all-NA rows are removed using trs_trim().

Value

A data.frame with the selected series. If trim = TRUE, leading and trailing all-NA rows are removed. If a single series is selected, returns a single-column data frame.

Examples

```
rw1 <- trs_pseudo_rw1(n_series = 5, series_length = 60, end_date = 2000)

# Select single series
trs_select(rw1, series = "trs_1", trim = TRUE)

# Select multiple series
trs_select(rw1, series = c("trs_1", "trs_2"), trim = TRUE)
trs_select(rw1, series = c("trs_1", "trs_3", "trs_5"))
```

trs_tbp	<i>Compute Baillie & Pilcher-style t-values between all series in two rw1-style data frames</i>
---------	---

Description

This function computes crossdating t-statistics between all series in two datasets using the Baillie & Pilcher (1973) method. Series are standardized using a 5-year centered moving average and log-transformed before computing correlations and t-values.

Usage

```
trs_tbp(x, y = NULL, min_overlap = 50, as_df = FALSE, transform = TRUE)
```

Arguments

x	A data frame of test tree-ring series in rw1 format (years as rownames, series as columns). All columns must be numeric.
y	A data frame of reference tree-ring series in rw1 format. If NULL, uses x for self-comparison. Default is NULL.
min_overlap	Integer. Minimum number of overlapping years required between series pairs to compute t-values. Must be ≥ 3 . Default is 50.
as_df	Logical. If TRUE, returns results in long-format data frame. If FALSE, returns list of matrices. Default is FALSE.
transform	Logical. If TRUE, applies Baillie & Pilcher transformation via <code>trs_tbp_transform()</code> . If FALSE, assumes input data is already transformed. Default is TRUE.

Details

The Baillie & Pilcher method involves:

1. Alignment of series by common years
2. Application of 5-year centered moving average to each series
3. Log transformation: $\log(100 \times \text{value}/\text{moving average})$
4. Computation of Pearson correlation between transformed series

5. Calculation of t-statistic: $t = r\sqrt{(n-4-2)}/\sqrt{1-r^2}$

Where n is the number of overlapping observations and is adjusted by subtracting 4 to account for degrees of freedom lost in the moving average.

Negative correlations are set to 0, and perfect correlations ($|r| = 1$) result in infinite t-values.

Value

Depending on as_df parameter:

- If as_df = FALSE (default): A list containing:
 - t_BP: Matrix of t-values with test series as rows, reference series as columns
 - overlap: Matrix of overlap counts (number of common years) with same dimensions
- If as_df = TRUE: A data frame with columns:
 - series: Name of test series
 - reference: Name of reference series
 - t_BP: Baillie & Pilcher t-value
 - overlap: Number of overlapping years

References

Baillie, M.G.L. & Pilcher, J.R. (1973). A simple crossdating program for tree-ring research. *Tree-Ring Bulletin*, 33, 7–14.

See Also

[trs_tbp_transform](#) for the transformation function

Examples

```
## Not run:
# Create sample data
rwl_test <- trs_pseudo_rwl(n_series = 3, series_length = 100, end_date = 2020)
rwl_ref <- trs_pseudo_rwl(n_series = 2, series_length = 80, end_date = 2015)

# Compute t-values (matrix format)
result <- trs_tbp(rwl_test, rwl_ref, min_overlap = 30)
print(result$t_BP)
print(result$overlap)

# Compute t-values (data frame format)
result_df <- trs_tbp(rwl_test, rwl_ref, min_overlap = 30, as_df = TRUE)
print(result_df)

# Self-comparison within single dataset
self_comparison <- trs_tbp(rwl_test, min_overlap = 40)

## End(Not run)
```

trs_tbp_transform	<i>Pre-transform tree-ring series using Baillie & Pilcher method</i>
-------------------	--

Description

Applies a 5-year centered moving average and computes the log index: $\log(100 * \text{value} / \text{MA5})$. Uses edge mirroring to maintain constant window size and avoid NA values.

Usage

```
trs_tbp_transform(x)
```

Arguments

x A rwl-style data frame.

Value

A matrix of log-transformed index values.

Examples

```
# Create sample data,
trs <- trs_pseudo_rwl(n_series = 5, series_length = 100, end_date = c(2020, 2025))
transformed <- trs_tbp_transform(trs)
head(transformed, 10)
```

trs_tho	<i>Compute Hollstein-style t-values between all series in two rwl-style data frames</i>
---------	---

Description

This function calculates Hollstein (1980) t-values for all pairwise comparisons between tree-ring series in two data frames, using log-transformed growth ratios. The method computes year-to-year growth ratios and applies correlation-based t-statistics for crossdating analysis.

Usage

```
trs_tho(x, y = NULL, min_overlap = 50, as_df = FALSE, transform = TRUE)
```

Arguments

x	A data frame of test tree-ring series in rwl format (years as rownames, series as columns). All columns must be numeric.
y	A data frame of reference tree-ring series in rwl format. If NULL, uses x for self-comparison. Default is NULL.
min_overlap	Integer. Minimum number of overlapping years required between series pairs to compute t-values. Must be ≥ 3 . Default is 50.
as_df	Logical. If TRUE, returns results in long-format data frame. If FALSE, returns list of matrices. Default is FALSE.
transform	Logical. If TRUE, applies Hollstein transformation (log growth ratios). If FALSE, assumes input data is already transformed. Default is TRUE.

Details

The Hollstein method involves:

1. Alignment of series by common years
2. Computation of log-transformed growth ratios: $100 \times \log_{10}(x_t/x_{t-1})$
3. Calculation of Pearson correlation between transformed series
4. Computation of t-statistic: $t = r\sqrt{n-3}/\sqrt{1-r^2}$

Where n is the number of overlapping observations (original years, not growth ratios). The degrees of freedom are adjusted by subtracting 3 to account for the growth ratio transformation.

Negative correlations are set to 0, and perfect correlations ($|r| \geq 1$) result in infinite t-values to handle floating-point precision issues.

Value

Depending on as_df parameter:

- If as_df = FALSE (default): A list containing:
 - t_Ho: Matrix of Hollstein t-values with test series as rows, reference series as columns
 - overlap: Matrix of overlap counts (number of common years) with same dimensions
- If as_df = TRUE: A data frame with columns:
 - series: Name of test series
 - reference: Name of reference series
 - t_Ho: Hollstein t-value
 - overlap: Number of overlapping years

References

Hollstein, E. (1980). *Mitteleuropäische Eichenchronologie*. Verlag Philipp von Zabern, Mainz.

See Also

[trs_tbp](#) for Baillie & Pilcher t-values

Examples

```
## Not run:
# Create sample data
rwl_test <- trs_pseudo_rwl(n_series = 3, series_length = 100, end_date = 2020)
rwl_ref <- trs_pseudo_rwl(n_series = 2, series_length = 80, end_date = 2015)

# Compute Hollstein t-values (matrix format)
result <- trs_tho(rwl_test, rwl_ref, min_overlap = 30)
print(result$t_Ho)
print(result$overlap)

# Compute t-values (data frame format)
result_df <- trs_tho(rwl_test, rwl_ref, min_overlap = 30, as_df = TRUE)
print(result_df)

# Self-comparison within single dataset
self_comparison <- trs_tho(rwl_test, min_overlap = 40)

## End(Not run)
```

trs_tho_transform	<i>Pre-transform tree-ring series using Hollstein method</i>
-------------------	--

Description

Computes log-transformed growth ratios for tree-ring series following the Hollstein (1980) method: $100 * \log_{10}(\text{value}_t / \text{value}_{(t-1)})$. This transformation emphasizes year-to-year growth variations.

Usage

```
trs_tho_transform(x)
```

Arguments

x	A rwl-style data frame with years as rownames and series as columns. All columns must be numeric.
---	---

Details

The Hollstein transformation computes year-to-year growth ratios:

$$G_t = 100 \times \log_{10}(x_t / x_{t-1})$$

where x_t is the ring width at year t . This transformation:

- Emphasizes relative changes between consecutive years

- Reduces the influence of long-term trends
- Makes series more suitable for crossdating analysis

Values that would result in undefined logarithms (zero or negative ratios) are set to NA. The first year of each series is always NA since no previous year exists for ratio calculation.

Value

A matrix of log-transformed growth ratios with the same dimensions as the input. The first row (first year) is set to NA since no previous year exists for ratio calculation. Row names match the input data.

References

Hollstein, E. (1980). *Mitteleuropäische Eichenchronologie*. Verlag Philipp von Zabern, Mainz.

See Also

[trs_tbp_transform](#) for Baillie & Pilcher transformation, [trs_tho](#) for computing Hollstein t-values

Examples

```
## Not run:
# Create sample rwl data
rwl_data <- trs_pseudo_rwl(n_series = 3, series_length = 50, end_date = 2020)

# Apply Hollstein transformation
transformed <- trs_tho_transform(rwl_data)

# Check dimensions (should be one row fewer)
nrow(rwl_data) # Original rows
nrow(transformed) # One fewer row

# View first few transformed values
head(transformed)

## End(Not run)
```

trs_trim	<i>Trim leading and trailing rows with only NA values from a tree-ring data frame</i>
----------	---

Description

Removes rows at the beginning and end of a data.frame of class "rwl" that contain only missing values across all series.

Usage

```
trs_trim(x, rownames_to_years = FALSE)
```

Arguments

x A data.frame or matrix where each column represents a tree-ring series and rows are years. Rownames must be numeric years.

rownames_to_years Logical. If TRUE, the rownames (years) will be transferred to a new column named year.

Value

A trimmed data.frame with all leading and trailing all-NA rows removed. If rownames_to_years = TRUE, a column year is added.

Examples

```
x <- trs_pseudo_rwl(n_series = 3, series_length = 80, end_date = 1990)
x[1:5, ] <- NA # Add leading NA rows
x[81:85, ] <- NA # Add trailing NA rows
dim(x)
x_trim <- trs_trim(x)
dim(x_trim)

# With year column added
head(trs_trim(x, rownames_to_years = TRUE))
```

| trs_tSt |

| *Compute Student's t-statistics from correlation coefficients* |

Description

This function computes correlation coefficients and their corresponding Student's t-statistics between all pairs of series from two tree-ring datasets.

Usage

```
trs_tSt(x, y = NULL, min_overlap = 30, as_df = FALSE)
```

Arguments

x A data frame of test tree-ring series in rwl format (years as rownames, series as columns). All columns must be numeric.

y A data frame of reference tree-ring series in rwl format.

min_overlap Integer. Minimum number of overlapping years required between series pairs to compute statistics. Must be ≥ 3 . Default is 30.

as_df Logical. If TRUE, returns results as data frames. If FALSE, returns as matrices. Default is TRUE.

Details

The function computes Pearson correlation coefficients between all pairs of series from the two input datasets, then converts these to Student's t-statistics using the formula: $t = r * \sqrt{n-2} / \sqrt{1-r^2}$, where n is the number of overlapping observations.

Value

- A list containing:
- r** Correlation coefficients between series pairs
 - t** Student's t-statistics
 - overlap** Number of overlapping years between series pairs

Examples

```
# Create sample data,
trs <- trs_pseudo_rwl(n_series = 5, series_length = c(80, 100), end_date = c(2010, 2020))
trs_tSt(trs)$t_St
```

trs_zscore	<i>Standardize tree-ring series to z-scores</i>
------------	---

Description

Converts each series (column) in an rwl-style data.frame to z-scores by subtracting the mean and dividing by the standard deviation.

Usage

```
trs_zscore(x)
```

Arguments

x A data.frame of class "rwl", where each column is a tree-ring series.

Value

A data.frame of the same dimensions as x, with each column transformed to z-scores.

Examples

```

rwl <- trs_pseudo_rwl(n_series = 3, series_length = 50, end_date = 1990)
z_rwl <- trs_zscore(rwl)
apply(z_rwl, 2, mean, na.rm = TRUE) # should be ~0
apply(z_rwl, 2, sd, na.rm = TRUE) # should be ~1

```

vanDaalen_NLBE

van Daalen (unpublished) sapwood data set.

Description

Sapwood data set for historical timbers found in the Netherlands en Belgium, with a local provenance. Unpublished data by S. van Daalen (version 19 Dec 2022, Van Daalen Dendrochronologie - <www.dendro.nl>).

Usage

```
vanDaalen_NLBE
```

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 644 observations

Source

S. van Daalen, unpublished dataset (version: 19 Dec 2022).

Examples

```

sw_data_info("vanDaalen_NLBE")

sw_model("vanDaalen_NLBE", plot = TRUE)

```

vanDaalen_Norway	<i>van Daalen (unpublished) sapwood data set.</i>
------------------	---

Description

Sapwood data set for historical timbers found in the Netherlands that were imported from Norway. Unpublished data by S. van Daalen (version 19 Dec 2022, Van Daalen Dendrochronologie - <www.dendro.nl>).

Usage

vanDaalen_Norway

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 104 observations

Source

S. van Daalen, unpublished dataset (version: 19 Dec 2022).

Examples

```
sw_data_info("vanDaalen_Norway")
```

```
sw_model("vanDaalen_Norway", plot = TRUE)
```

Wazny_1990	<i>Wazny 1990 sapwood data set.</i>
------------	-------------------------------------

Description

Sapwood data set for Poland, published by Wazny in 1990.

Usage

Wazny_1990

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 206 observations

Source

Wazny T. 1990. Aufbau und Anwendung der Dendrochronologie für Eichenholz in Polen (PhD dissertation). Hamburg University, Hamburg.

Examples

```
sw_data_info("Wazny_1990")
```

```
sw_model("Wazny_1990", plot = TRUE)
```

Weitz_2025

Weitz 2025 sapwood data set.

Description

Sapwood data set for the Brussels region in Belgium, published by Weitz et al. in 2025.

Usage

```
Weitz_2025
```

Format

A tibble of 2 variables:

n_sapwood number of sapwood rings

count number of times n_sapwood was observed

Details

sample size = 443 observations

Source

Weitz A., Haneca K. & Hoffsummer P. (in press): Estimating sapwood counts for historical oak timbers from Brussels, Belgium (13th-18th C), International Journal of Wood Culture.

Examples

```
sw_data_info("Weitz_2025")
```

```
sw_model("Weitz_2025", plot = TRUE)
```

Index

- * **Hollstein_1980**
 - Hollstein_1980, [9](#)
- * **datasets**
 - Brathen_1982, [3](#)
 - Miles_1997_NM, [9](#)
 - Miles_1997_SC, [10](#)
 - Miles_1997_WBC, [11](#)
 - Pilcher_1987, [13](#)
 - Sohar_2012_ELL_c, [16](#)
 - Sohar_2012_ELL_t, [17](#)
 - Sohar_2012_FWE_c, [17](#)
 - Sohar_2012_FWE_t, [18](#)
 - sw_example0, [23](#)
 - sw_example1, [24](#)
 - sw_example2, [24](#)
 - sw_example3, [25](#)
 - sw_example4, [25](#)
 - sw_example5, [26](#)
 - sw_example6, [27](#)
 - sw_example7, [27](#)
 - vanDaalen_NLBE, [54](#)
 - vanDaalen_Norway, [55](#)
 - Wazny_1990, [55](#)
 - Weitz_2025, [56](#)
- * **package**
 - fellingdater, [6](#)
- Brathen_1982, [3](#)
- dplR::read.fh(), [14](#)
- fd_report, [4](#)
- fellingdater, [6](#)
- fellingdater-package (fellingdater), [6](#)
- fh_header, [7](#)
- hdi, [8](#)
- Hollstein_1980, [9](#)
- Miles_1997_NM, [9](#)
- Miles_1997_SC, [10](#)
- Miles_1997_WBC, [11](#)
- mov_av, [12](#)
- Pilcher_1987, [13](#)
- read_fh, [13](#)
- read_fh(), [33](#)
- sgc_for_plot, [15](#), [40](#)
- Sohar_2012_ELL_c, [16](#)
- Sohar_2012_ELL_t, [17](#)
- Sohar_2012_FWE_c, [17](#)
- Sohar_2012_FWE_t, [18](#)
- sw_combine, [8](#), [19](#)
- sw_combine(), [21](#)
- sw_combine_plot, [21](#)
- sw_combine_plot(), [20](#)
- sw_data_info, [22](#)
- sw_data_overview, [23](#)
- sw_data_overview(), [5](#), [20](#), [22](#), [28](#), [30](#), [33](#)
- sw_example0, [23](#)
- sw_example1, [24](#)
- sw_example2, [24](#)
- sw_example3, [25](#)
- sw_example4, [25](#)
- sw_example5, [26](#)
- sw_example6, [27](#)
- sw_example7, [27](#)
- sw_interval, [8](#), [28](#)
- sw_interval(), [5](#), [29](#), [30](#)
- sw_interval_plot, [29](#)
- sw_interval_plot(), [5](#)
- sw_model, [8](#), [30](#)
- sw_model(), [31](#), [32](#)
- sw_model_plot, [31](#)
- sw_sum, [32](#)
- sw_sum(), [34](#), [35](#)
- sw_sum_plot, [34](#)
- sw_sum_plot(), [33](#)
- trs_crossdate, [35](#), [40](#)

trs_end_date, [38](#), [40](#)
trs_plot_dated, [39](#)
trs_plot_rwl, [40](#)
trs_pseudo_rwl, [42](#)
trs_pv, [37](#), [43](#)
trs_remove, [44](#)
trs_select, [45](#)
trs_tbp, [37](#), [46](#), [49](#)
trs_tbp_transform, [47](#), [48](#), [51](#)
trs_tho, [37](#), [48](#), [51](#)
trs_tho_transform, [50](#)
trs_trim, [38](#), [40](#), [41](#), [51](#)
trs_tSt, [52](#)
trs_zscore, [40](#), [41](#), [53](#)

vanDaalen_NLBE, [54](#)
vanDaalen_Norway, [55](#)

Wazny_1990, [55](#)
Weitz_2025, [56](#)