# Package 'dendroTools'

**Type** Package

**Title** Linear and Nonlinear Methods for Analyzing Daily and Monthly Dendroclimatological Data

**Version** 1.2.14

**Description** Provides novel dendroclimatological methods, primarily used by the Tree-ring research community. There are four core functions. The first one is daily_response(), which finds the optimal sequence of days that are related to one or more tree-ring proxy records. Similar function is daily_response_seascorr(), which implements partial correlations in the analysis of daily response functions. For the enthusiast of monthly data, there is monthly_response() function. The last core function is compare_methods(), which effectively compares several linear and nonlinear regression algorithms on the task of climate reconstruction.

**License** GPL-3

**URL** https://github.com/jernejjevsenak/dendroTools

**BugReports** https://github.com/jernejjevsenak/dendroTools/issues

**Encoding** UTF-8

**LazyData** true

**Suggests** testthat, rmarkdown

**RoxygenNote** 7.3.2

**Imports** ggplot2 (>= 2.2.0), brnn (>= 0.6), reshape2 (>= 1.4.2), scales (>= 0.4.1), stats, oce (>= 1.2-0), MLmetrics (>= 1.1.1), dplyr (>= 0.7.0), knitr (>= 1.19), magrittr (>= 1.5), plotly (>= 4.7.1), randomForest (>= 4.6-14), Cubist (>= 0.2.2), lubridate (>= 1.7.4), psych (>= 1.8.3.3), boot (>= 1.3-22), viridis (>= 0.5.1), dplR (>= 1.7.2)

**Depends** R (>= 3.4)

**NeedsCompilation** no

**Repository** CRAN

**VignetteBuilder** knitr

**Author** Jernej Jevsenak [aut, cre]

**Maintainer** Jernej Jevsenak <jernej.jevsenak@gmail.com>

**Date/Publication** 2025-06-03 15:10:05 UTC

# Contents

---

calculate_metrics            *calculate_metrics*

---

## Description

Calculates performance metrics for train and test data. Calculated performance metrics are correlation coefficient (r), root mean squared error (RMSE), root relative squared error (RRSE), index of agreement (d), reduction of error (RE), coefficient of efficiency (CE), detrended efficiency (DE) and bias.

**Usage**

```
calculate_metrics(
  train_predicted,
  test_predicted,
  train_observed,
  test_observed,
  digits = 4,
  formula,
  test
)
```

**Arguments**

train_predicted

  a vector indicating predicted data for training set

test_predicted    a vector indicating predicted data for testing set

train_observed    a vector indicating observed data for training set

test_observed     a vector indicating observed data for training set

digits            integer of number of digits to be displayed

formula           an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. This additional argument is needed to calculate DE metrics.

test              data frame with test data.

**Value**

a data frame of calculated test and train metrics

**References**

Briffa, K.R., Jones, P.D., Pilcher, J.R., Hughes, M.K., 1988. Reconstructing summer temperatures in northern Fennoscandinavia back to A.D.1700 using tree ring data from Scots Pine. Arct. Alp. Res. 20, 385-394.

Fritts, H.C., 1976. Tree Rings and Climate. Academic Press, London 567 pp.

Lorenz, E.N., 1956. Empirical Orthogonal Functions and Statistical Weather Prediction. Massachusetts Institute of Technology, Department of Meteorology.

Willmott, C.J., 1981. On the validation of models. Phys. Geogr. 2, 184-194.

Witten, I.H., Frank, E., Hall, M.A., 2011. Data Mining: Practical Machine Learning Tools and Techniques, 3rd ed. Morgan Kaufmann Publishers, Burlington 629 pp.

**Examples**

```
data(example_dataset_1)
test_data <- example_dataset_1[1:30, ]
train_data <- example_dataset_1[31:55, ]
lin_mod <- lm(MVA ~., data = train_data)
```

```
train_predicted <- predict(lin_mod, train_data)
test_predicted <- predict(lin_mod, test_data)
train_observed <- train_data[, 1]
test_observed <- test_data[, 1]
calculate_metrics(train_predicted, test_predicted, train_observed,
test_observed, test = test_data, formula = MVA ~.)

test_data <- example_dataset_1[1:20, ]
train_data <- example_dataset_1[21:55, ]
library(brnn)
lin_mod <- brnn(MVA ~., data = train_data)
train_predicted <- predict(lin_mod, train_data)
test_predicted <- predict(lin_mod, test_data)
train_observed <- train_data[, 1]
test_observed <- test_data[, 1]
calculate_metrics(train_predicted, test_predicted, train_observed,
test_observed, test = test_data, formula = MVA ~.)
```

---

compare_methods                *compare_methods*

---

### Description

Calculates performance metrics for calibration (train) and validation (test) data of different regression methods: multiple linear regression (MLR), artificial neural networks with Bayesian regularization training algorithm (BRNN), (ensemble of) model trees (MT) and random forest of regression trees (RF). With the subset argument, specific methods of interest could be specified. Calculated performance metrics are the correlation coefficient (r), the root mean squared error (RMSE), the root relative squared error (RRSE), the index of agreement (d), the reduction of error (RE), the coefficient of efficiency (CE), the detrended efficiency (DE) and mean bias. For each of the considered methods, there are also residual diagnostic plots available, separately for calibration, holdout and edge data, if applicable.

### Usage

```
compare_methods(
  formula,
  dataset,
  k = 10,
  repeats = 2,
  optimize = TRUE,
  dataset_complete = NULL,
  BRNN_neurons = 1,
  MT_committees = 1,
  MT_neighbors = 5,
  MT_rules = 200,
  MT_unbiased = TRUE,
```

```
      MT_extrapolation = 100,
      MT_sample = 0,
      RF_ntree = 500,
      RF_maxnodes = 5,
      RF_mtry = 1,
      RF_nodesize = 1,
      seed_factor = 5,
      digits = 3,
      blocked_CV = FALSE,
      PCA_transformation = FALSE,
      log_preprocess = TRUE,
      components_selection = "automatic",
      eigenvalues_threshold = 1,
      N_components = 2,
      round_bias_cal = 15,
      round_bias_val = 4,
      n_bins = 30,
      edge_share = 0.1,
      MLR_stepwise = FALSE,
      stepwise_direction = "backward",
      methods = c("MLR", "BRNN", "MT", "RF"),
      tuning_metric = "RMSE",
      BRNN_neurons_vector = c(1, 2, 3),
      MT_committees_vector = c(1, 5, 10),
      MT_neighbors_vector = c(0, 5),
      MT_rules_vector = c(100, 200),
      MT_unbiased_vector = c(TRUE, FALSE),
      MT_extrapolation_vector = c(100),
      MT_sample_vector = c(0),
      RF_ntree_vector = c(100, 250, 500),
      RF_maxnodes_vector = c(5, 10, 20, 25),
      RF_mtry_vector = c(1),
      RF_nodesize_vector = c(1, 5, 10),
      holdout = NULL,
      holdout_share = 0.1,
      holdout_manual = NULL,
      total_reproducibility = FALSE
    )
```

## Arguments

| | |
|---|---|
| formula | an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted. |
| dataset | a data frame with dependent and independent variables as columns and (optional) years as row names. |
| k | number of folds for cross-validation |
| repeats | number of cross-validation repeats. Should be equal or more than 1 |

| | |
|---|---|
| optimize | if set to TRUE (default), the optimal values for the tuning parameters will be selected in a preliminary cross-validation procedure |
| dataset_complete | |
| | optional, a data frame with the full length of tree-ring parameter, which will be used to reconstruct the climate variable specified with the formula argument |
| BRNN_neurons | number of neurons to be used for the brnn method |
| MT_committees | an integer: how many committee models (e.g. boosting iterations) should be used? |
| MT_neighbors | how many, if any, neighbors should be used to correct the model predictions |
| MT_rules | an integer (or NA): define an explicit limit to the number of rules used (NA let's Cubist decide). |
| MT_unbiased | a logical: should unbiased rules be used? |
| MT_extrapolation | |
| | a number between 0 and 100: since Cubist uses linear models, predictions can be outside of the outside of the range seen the training set. This parameter controls how much rule predictions are adjusted to be consistent with the training set. |
| MT_sample | a number between 0 and 99.9: this is the percentage of the dataset to be randomly selected for model building (not for out-of-bag type evaluation) |
| RF_ntree | number of trees to grow. This should not be set to too small a number, to ensure that every input row gets predicted at least a few times |
| RF_maxnodes | maximum number of terminal nodes trees in the forest can have |
| RF_mtry | number of variables randomly sampled as candidates at each split |
| RF_nodesize | minimum size of terminal nodes. Setting this number larger causes smaller trees to be grown (and thus take less time). |
| seed_factor | an integer that will be used to change the seed options for different repeats. |
| digits | integer of number of digits to be displayed in the final result tables |
| blocked_CV | default is FALSE, if changed to TRUE, blocked cross-validation will be used to compare regression methods. |
| PCA_transformation | |
| | if set to TRUE, all independent variables will be transformed using PCA transformation. |
| log_preprocess | if set to TRUE, variables will be transformed with logarithmic transformation before used in PCA |
| components_selection | |
| | character string specifying how to select the Principal Components used as predictors. There are three options: "automatic", "manual" and "plot_selection". If parameter is set to automatic, all scores with eigenvalues above 1 will be selected. This threshold could be changed by changing the eigenvalues_threshold argument. If parameter is set to "manual", user should set the number of components with N_components argument. If component selection is se to "plot_selection", Scree plot will be shown and user must manually enter the number of components used as predictors. |
| eigenvalues_threshold | |
| | threshold for automatic selection of Principal Components |

| | |
|---|---|
| N_components | number of Principal Components used as predictors |
| round_bias_cal | number of digits for bias in calibration period. Effects the outlook of the final ggplot of mean bias for calibration data (element 3 of the output list) |
| round_bias_val | number of digits for bias in validation period. Effects the outlook of the final ggplot of mean bias for validation data (element 4 of the output list) |
| n_bins | number of bins used for the histograms of mean bias |
| edge_share | the share of the data to be considered as the edge (extreme) data. This argument could be between 0.10 and 0.50. If the argument is set to 0.10, then the 5 considered to be the edge data. |
| MLR_stepwise | if set to TRUE, stepwise selection of predictors will be used for the MLR method |
| stepwise_direction | |
| | the mode of stepwise search, can be one of "both", "backward", or "forward", with a default of "backward". |
| methods | a vector of strings related to methods that will be compared. A full method vector is methods = c("MLR", "BRNN", "MT", "RF"). To use only a subset of methods, pass a vector of methods that you would like to compare. |
| tuning_metric | a string that specifies what summary metric will be used to select the optimal value of tuning parameters. By default, the argument is set to "RMSE". It is also possible to use "RSquared". |
| BRNN_neurons_vector | |
| | a vector of possible values for BRNN_neurons argument optimization |
| MT_committees_vector | |
| | a vector of possible values for MT_committees argument optimization |
| MT_neighbors_vector | |
| | a vector of possible values for MT_neighbors argument optimization |
| MT_rules_vector | |
| | a vector of possible values for MT_rules argument optimization |
| MT_unbiased_vector | |
| | a vector of possible values for MT_unbiased argument optimization |
| MT_extrapolation_vector | |
| | a vector of possible values for MT_extrapolation argument optimization |
| MT_sample_vector | |
| | a vector of possible values for MT_sample argument optimization |
| RF_ntree_vector | |
| | a vector of possible values for RF_ntree argument optimization |
| RF_maxnodes_vector | |
| | a vector of possible values for RF_maxnodes argument optimization |
| RF_mtry_vector | a vector of possible values for RF_mtry argument optimization |
| RF_nodesize_vector | |
| | a vector of possible values for RF_nodesize argument optimization |
| holdout | this argument is used to define observations, which are excluded from the cross-validation and hyperparameters optimization. The holdout argument must be a character with one of the following inputs: "early", "late" or "manual". If |

"early" or "late" characters are specified, then the early or late years will be used as a holdout data. How many of the "early" or "late" years are used as a holdout is specified with the argument holdout_share. If the argument holdout is set to "manual", then supply a vector of years (or row names) to the argument holdout_manual. Defined years will be used as a holdout. For the holdout data, the same statistical measures are calculated as for the cross-validation. The results for holdout metrics are given in the output element $holdout_results.

holdout_share    the share of the whole dataset to be used as a holdout. Default is 0.10.

holdout_manual  a vector of years (or row names) which will be used as a holdout. calculated as for the cross-validation.

total_reproducibility

logical, default is FALSE. This argument ensures total reproducibility despite the inclusion/exclusion of different methods. By default, the optimization is done only for the methods, that are included in the methods vector. If one method is absent or added, the optimization phase is different, and this affects all the final cross-validation results. By setting the total_reproducibility = TRUE, all methods will be optimized, even though they are not included in the methods vector and the final results will be subset based on the methods vector. Setting the total_reproducibility to TRUE will result in longer optimization phase as well.

**Value**

a list with 19 elements:

1. $mean_std - data frame with calculated metrics for the selected \ regression methods. For each regression method and each calculated metric, mean and standard deviation are given

2. $ranks - data frame with ranks of calculated metrics: mean rank and share of rank_1 are given

3. $edge_results - data frame with calculated performance metrics for the central-edge test. The central part of the data represents the calibration data, while the edge data, i.e. extreme values, represent the test/validation data. Different regression models are calibrated using the central data and validated for the edge (extreme) data. This test is particularly important to assess the performance of models for the predictions of the extreme data. The share of the edge (extreme) data is defined with the edge_share argument

4. $holdout_results - calculated metrics for the holdout data

5. $bias_cal - ggplot object of mean bias for calibration data

6. $bias_val - ggplot object of mean bias for validation data

7. $transfer_functions - ggplot or plotly object with transfer functions of methods

8. $transfer_functions_together - ggplot or plotly object with transfer functions of methods plotted together

9. $parameter_values - a data frame with specifications of parameters used for different regression methods

10. $PCA_output - princomp object: the result output of the PCA analysis

11. $reconstructions - ggplot object: reconstructed dependent variable based on the dataset_complete argument, facet is used to split plots by methods

12. $reconstructions_together - ggplot object: reconstructed dependent variable based on the dataset_complete argument, all reconstructions are on the same plot

13. $normal_QQ_cal - normal q-q plot for calibration data

14. $normal_QQ_holdout - normal q-q plot for holdout data

15. $normal_QQ_edge- normal q-q plot for edge data

16. $residuals_vs_fitted_cal - residuals vs fitted values plot for calibration data

17. $residuals_vs_fitted_holdout - residuals vs fitted values plot for holdout data

18. $residuals_vs_fitted_edge - residuals vs fitted values plot for edge data

19. $reconstructions_data - raw data that is used for creating reconstruction plots

### References

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford University Press, Inc. 482 pp.

Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123-140.

Breiman, L., 2001. Random forests. Machine Learning 45, 5-32.

Burden, F., Winkler, D., 2008. Bayesian Regularization of Neural Networks, in: Livingstone, D.J. (ed.), Artificial Neural Networks: Methods and Applications, vol. 458. Humana Press, Totowa, NJ, pp. 23-42.

Hastie, T., Tibshirani, R., Friedman, J.H., 2009. The Elements of Statistical Learning : Data Mining, Inference, and Prediction, 2nd ed. Springer, New York xxii, 745 p. pp.

Ho, T.K., 1995. Random decision forests, Proceedings of the Third International Conference on Document Analysis and Recognition Volume 1. IEEE Computer Society, pp. 278-282.

Hornik, K., Buchta, C., Zeileis, A., 2009. Open-source machine learning: R meets Weka. Comput. Stat. 24, 225-232.

Perez-Rodriguez, P., Gianola, D., 2016. Brnn: Brnn (Bayesian Regularization for Feed-forward Neural Networks). R package version 0.6.

Quinlan, J.R., 1992. Learning with Continuous Classes, Proceedings of the 5th Australian Joint Conference on Artificial Intelligence (AI '92). World Scientific, Hobart, pp. 343-348.

### Examples

```
# The examples below are enclosed within donttest{} to minimize the execution
# time during R package checks. #'

# An example with default settings of machine learning algorithms
library(dendroTools)
library(ggplot2)

data(example_dataset_1)
data(dataset_TRW)

example_1 <- compare_methods(formula = MVA ~  T_APR,
dataset = example_dataset_1, k = 5, repeats = 1, BRNN_neurons = 1,
```

```
          RF_ntree = 100, RF_mtry = 2, RF_maxnodes = 35, seed_factor = 5)

          # example_1$mean_std
          # example_1$ranks
          # example_1$bias_cal
          # example_1$transfer_functions
          # example_1$transfer_functions_together
          # example_1$PCA_output
          # example_1$parameter_values

          example_2 <- compare_methods(formula = MVA ~ .,
          dataset = example_dataset_1, k = 2, repeats = 2,
          methods = c("MLR", "BRNN", "MT"),
          optimize = TRUE, MLR_stepwise = TRUE)
          # example_2$mean_std
          # example_2$ranks
          # example_2$bias_val
          # example_2$transfer_functions
          # example_2$transfer_functions_together
          # example_2$parameter_values

          comparison_TRW <- compare_methods(formula = T_Jun_Jul ~ TRW, dataset = dataset_TRW,
          k = 3, repeats = 5, optimize = FALSE, methods = c("MLR", "BRNN", "RF", "MT"),
          seed_factor = 5, dataset_complete = dataset_TRW_complete, MLR_stepwise = TRUE,
          stepwise_direction = "backward")

          # comparison_TRW$mean_std
          # comparison_TRW$bias_val
          # comparison_TRW$transfer_functions
          # comparison_TRW$reconstructions
          # comparison_TRW$reconstructions_together
          # comparison_TRW$edge_results
          # comparison_TRW$reconstructions_data
```

---

| critical_r | *critical_r* |
|---|---|

---

### Description

Calculates critical value of Pearson correlation coefficient for a selected alpha.

### Usage

```
critical_r(n, alpha = 0.05)
```

### Arguments

| | |
|---|---|
| n | number of observations |
| alpha | significance level |

**Value**

calculated critical value of Pearson correlation coefficient

**Examples**

```
threshold_1 <- critical_r(n = 55, alpha = 0.01)
threshold_2 <- critical_r(n = 55, alpha = 0.05)
```

---

daily_response                  *daily_response*

---

**Description**

Function calculates all possible values of a selected statistical metric between one or more response variables and daily sequences of environmental data. Calculations are based on moving window which is defined with two arguments: window width and a location in a matrix of daily sequences of environmental data. Window width could be fixed (use fixed_width) or variable width (use lower_limit and upper_limit arguments). In this case, all window widths between lower and upper limit will be used. All calculated metrics are stored in a matrix. The location of stored calculated metric in the matrix is indicating a window width (row names) and a location in a matrix of daily sequences of environmental data (column names).

**Usage**

```
daily_response(
  response,
  env_data,
  method = "cor",
  metric = "r.squared",
  cor_method = "pearson",
  lower_limit = 30,
  upper_limit = 90,
  fixed_width = 0,
  previous_year = FALSE,
  neurons = 1,
  brnn_smooth = TRUE,
  remove_insignificant = FALSE,
  alpha = 0.05,
  row_names_subset = FALSE,
  aggregate_function = "mean",
  temporal_stability_check = "sequential",
  k = 2,
  k_running_window = 30,
  cross_validation_type = "blocked",
  subset_years = NULL,
  ylimits = NULL,
  seed = NULL,
```

```
    tidy_env_data = FALSE,
    reference_window = "start",
    boot = FALSE,
    boot_n = 1000,
    boot_ci_type = "norm",
    boot_conf_int = 0.95,
  day_interval = ifelse(c(previous_year == TRUE, previous_year == TRUE), c(-1, 366), c(1,
      366)),
    dc_method = NULL,
    cor_na_use = "everything",
    skip_window_length = 1,
    skip_window_position = 1
)
```

## Arguments

| | |
|---|---|
| response | a data frame with tree-ring proxy variables as columns and (optional) years as row names. Row.names should be matched with those from a env_data data frame. If not, set row_names_subset = TRUE. |
| env_data | a data frame of daily sequences of environmental data as columns and years as row names. Each row represents a year and each column represents a day of a year. Row.names should be matched with those from a response data frame. If not, set row_names_subset = TRUE. Alternatively, env_data could be a tidy data with three columns, i.e. Year, DOY and third column representing values of mean temperatures, sum of precipitation etc. If tidy data is passed to the function, set the argument tidy_env_data to TRUE. |
| method | a character string specifying which method to use. Current possibilities are "cor" (default), "lm" and "brnn". |
| metric | a character string specifying which metric to use. Current possibilities are "r.squared" and "adj.r.squared". If method = "cor", metric is not relevant. |
| cor_method | a character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman". |
| lower_limit | lower limit of window width |
| upper_limit | upper limit of window width |
| fixed_width | fixed width used for calculation. If fixed_width is assigned a value, upper_limit and lower_limit will be ignored |
| previous_year | if set to TRUE, env_data and response variables will be rearranged in a way, that also previous year will be used for calculations of selected statistical metric. |
| neurons | positive integer that indicates the number of neurons used for brnn method |
| brnn_smooth | if set to TRUE, a smoothing algorithm is applied that removes unrealistic calculations which are a result of neural net failure. |
| remove_insignificant | |
| | if set to TRUE, removes all correlations bellow the significant threshold level, based on a selected alpha. For "lm" and "brnn" method, squared correlation is used as a threshold |

| | |
|---|---|
| alpha | significance level used to remove insignificant calculations. |
| row_names_subset | if set to TRUE, row.names are used to subset env_data and response data frames. Only years from both data frames are kept. |
| aggregate_function | character string specifying how the daily data should be aggregated. The default is 'mean', the other options are 'median', 'sum', 'min' and 'max' |
| temporal_stability_check | character string, specifying, how temporal stability between the optimal selection and response variable(s) will be analysed. Current possibilities are "sequential", "progressive" and "running_window". Sequential check will split data into k splits and calculate selected metric for each split. Progressive check will split data into k splits, calculate metric for the first split and then progressively add 1 split at a time and calculate selected metric. For running window, select the length of running window with the k_running_window argument. |
| k | integer, number of breaks (splits) for temporal stability and cross validation analysis. |
| k_running_window | the length of running window for temporal stability check. Applicable only if temporal_stability argument is set to running window. |
| cross_validation_type | character string, specifying, how to perform cross validation between the optimal selection and response variables. If the argument is set to "blocked", years will not be shuffled. If the argument is set to "randomized", years will be shuffled. |
| subset_years | a subset of years to be analyzed. Should be given in the form of subset_years = c(1980, 2005) |
| ylimits | limit of the y axes for plot_extreme. It should be given in the form of: ylimits = c(0,1) |
| seed | optional seed argument for reproducible results |
| tidy_env_data | if set to TRUE, env_data should be inserted as a data frame with three columns: "Year", "DOY", "Precipitation/Temperature/etc." |
| reference_window | character string, the reference_window argument describes, how each calculation is referred. There are three different options: 'start' (default), 'end' and 'middle'. If the reference_window argument is set to 'start', then each calculation is related to the starting day of window. If the reference_window argument is set to 'middle', each calculation is related to the middle day of window calculation. If the reference_window argument is set to 'end', then each calculation is related to the ending day of window calculation. For example, if we consider correlations with window from DOY 15 to DOY 35. If reference window is set to 'start', then this calculation will be related to the DOY 15. If the reference window is set to 'end', then this calculation will be related to the DOY 35. If the reference_window is set to 'middle', then this calculation is related to DOY 25. The optimal selection, which describes the optimal consecutive days that returns the highest calculated metric and is obtained by the $plot_extreme output, is the same for all three reference windows. |

| | |
|---|---|
| boot | logical, if TRUE, bootstrap procedure will be used to calculate estimates correlation coefficients, R squared or adjusted R squared metrices |
| boot_n | The number of bootstrap replicates |
| boot_ci_type | A character string representing the type of bootstrap intervals required. The value should be any subset of the values c("norm","basic", "stud", "perc", "bca"). |
| boot_conf_int | A scalar or vector containing the confidence level(s) of the required interval(s) |
| day_interval | a vector of two values: lower and upper time interval of days that will be used to calculate statistical metrics. Negative values indicate previous growing season days. This argument overwrites the calculation limits defined by lower_limit and upper_limit arguments. |
| dc_method | a character string to determine the method to detrend climate data. Possible values are "none" (default) and "SLD" which refers to Simple Linear Detrending |
| cor_na_use | an optional character string giving a method for computing covariances in the presence of missing values for correlation coefficients. This must be (an abbreviation of) one of the strings "everything" (default), "all.obs", "complete.obs", "na.or.complete", or "pairwise.complete.obs". See also the documentation for the base cor() function. |

skip_window_length

an integer specifying the frequency of window selection for the calculations of climate-growth relationships. The default value is 1, indicating that every window is included in the calculations. When set to a value greater than 1, the function selectively processes windows at regular intervals defined by this parameter. For instance, if skip_window_length = 2, the function processes every second window. Similarly, if skip_window_length = 3, every third window is processed, skipping two windows in between each selected one. This parameter allows for controlling the granularity of the analysis and can help in reducing computation time by focusing on a subset of the data.

skip_window_position

an integer specifying the frequency of window positions used in the calculations of climate-growth relationships. The default value is 1, indicating that every window position is included in the calculations. When set to a value greater than 1, the function selectively processes window positions at regular intervals defined by this parameter. For instance, if skip_window_position = 2, the function processes every second window position. Similarly, if skip_window_position = 3, every third window position is processed, skipping two positions in between each selected one. This parameter allows for controlling the granularity of the analysis and can help in reducing computation time by focusing on a subset of the data.

**Value**

a list with 17 elements:

1. $calculations - a matrix with calculated metrics
2. $method - the character string of a method
3. $metric - the character string indicating the metric used for calculations

4. $analysed_period - the character string specifying the analysed period based on the information from row names. If there are no row names, this argument is given as NA

5. $optimized_return - data frame with two columns, response variable and aggregated (averaged) daily data that return the optimal results. This data.frame could be directly used to calibrate a model for climate reconstruction

6. $optimized_return_all - a data frame with aggregated daily data, that returned the optimal result for the entire env_data (and not only subset of analysed years)

7. $transfer_function - a ggplot object: scatter plot of optimized return and a transfer line of the selected method

8. $temporal_stability - a data frame with calculations of selected metric for different temporal subsets

9. $cross_validation - a data frame with cross validation results

10. $plot_heatmap - ggplot2 object: a heatmap of calculated metrics

11. $plot_extreme - ggplot2 object: line plot of a row with the highest value in a matrix of calculated metrics

12. $type - the character string describing type of analysis: daily or monthly

13. $reference_window - character string, which reference window was used for calculations

14. $boot_lower - matrix with lower limit of confidence intervals of bootstrap calculations

15. $boot_upper - matrix with upper limit of confidence intervals of bootstrap calculations

16. $aggregated_climate - matrix with all aggregated climate series

## Examples

```
# The examples below are enclosed within donttest{} to minimize the execution
# time during R package checks. Additionally, all examples include the
# parameters `skip_window_length` and `skip_window_position`, which limit the
# number of combinations evaluated in climate-growth correlation calculations.
# To explore all possible combinations, users should set both parameters to 1.

# Load the dendroTools R package
library(dendroTools)

# Load data
data(data_MVA)
data(data_TRW)
data(data_TRW_1)
data(example_proxies_individual)
data(example_proxies_1)
data(LJ_daily_temperatures)

example_basic <- daily_response(response = data_MVA,
                         env_data = LJ_daily_temperatures,
                         row_names_subset = TRUE,
                         fixed_width = 25,
                         lower_limit = 35, upper_limit = 45,
                         remove_insignificant = FALSE,
```

```
                        aggregate_function = 'median',
                        alpha = 0.05, cor_method = "spearman",
                        previous_year = FALSE, boot = TRUE,
                        boot_n = 10,
                        skip_window_length = 50,
                        skip_window_position = 50,
                        reference_window = "end", k = 5,
                        dc_method = "SLD",
                        day_interval = c(-100, 250))

# 1 Example with fixed width. Lower and upper limits are ignored.
example_daily_response <- daily_response(response = data_MVA,
    env_data = LJ_daily_temperatures,
    method = "cor", fixed_width = 40, cor_method = "spearman",
    row_names_subset = TRUE, previous_year = TRUE,
    remove_insignificant = TRUE, boot = TRUE,
    alpha = 0.005, aggregate_function = 'mean',
    day_interval = c(-100, 250), skip_window_length = 100,
    reference_window = "start", skip_window_position = 100)

# summary(example_daily_response)
# plot(example_daily_response, type = 1)
# plot(example_daily_response, type = 2)

# 2 Example for past and present. Use subset_years argument.
example_MVA_early <- daily_response(response = data_MVA,
    env_data = LJ_daily_temperatures, cor_method = "kendall",
    method = "lm", lower_limit = 21, upper_limit = 91,
    row_names_subset = TRUE, previous_year = TRUE,
    remove_insignificant = TRUE, alpha = 0.05,
    subset_years = c(1940, 1980),
    fixed_width = 45,
    aggregate_function = 'sum',
    skip_window_length = 50,
    skip_window_position = 50)

example_MVA_late <- daily_response(response = data_MVA,
    env_data = LJ_daily_temperatures,
    method = "cor", lower_limit = 21, upper_limit = 60,
    row_names_subset = TRUE, previous_year = TRUE,
    remove_insignificant = TRUE, alpha = 0.05,
    subset_years = c(1981, 2010),
    skip_window_length = 50,
    skip_window_position = 50)

# plot(example_MVA_early, type = 1)
# plot(example_MVA_late, type = 1)
# plot(example_MVA_early, type = 2)
# plot(example_MVA_late, type = 2)

# 3 Example with negative correlations
example_neg_cor <- daily_response(response = data_TRW_1,
    env_data = LJ_daily_temperatures, previous_year = TRUE,
```

```
      method = "cor", lower_limit = 21, upper_limit = 90,
      row_names_subset = TRUE, remove_insignificant = TRUE,
      alpha = 0.05, skip_window_length = 50,
      skip_window_position = 50)

# summary(example_neg_cor)
# plot(example_neg_cor, type = 1)
# plot(example_neg_cor, type = 2)

# 4 Example of multiproxy analysis
# summary(example_proxies_1)
# cor(example_proxies_1)

example_multiproxy <- daily_response(response = example_proxies_1,
   env_data = LJ_daily_temperatures,
   method = "lm", metric = "adj.r.squared",
   lower_limit = 21, upper_limit = 180,
   row_names_subset = TRUE, previous_year = FALSE,
   remove_insignificant = TRUE, alpha = 0.05,
   skip_window_length = 50,
   skip_window_position = 50)

# plot(example_multiproxy, type = 1)

# 5 Example to test the temporal stability
example_MVA_ts <- daily_response(response = data_MVA,
   env_data = LJ_daily_temperatures, method = "brnn",
   lower_limit = 100, metric = "adj.r.squared", upper_limit = 180,
   row_names_subset = TRUE, remove_insignificant = TRUE, alpha = 0.05,
   temporal_stability_check = "running_window", k_running_window = 10,
   skip_window_length = 50, skip_window_position = 50)

# Check the results for temporal stability
# example_MVA_ts$temporal_stability

# 6 Example with nonlinear brnn estimation
example_brnn <- daily_response(response = data_MVA,
   env_data = LJ_daily_temperatures, method = "brnn", boot = FALSE,
   lower_limit = 100, metric = "adj.r.squared", upper_limit = 101,
   row_names_subset = TRUE, remove_insignificant = TRUE, boot_n = 10,
   skip_window_length = 50, skip_window_position = 50)

# summary(example_brnn)
```

---

daily_response_seascorr

*daily_response_seascorr*

---

**Description**

Function calculates all possible partial correlation coefficients between tree-ring chronology and daily environmental (usually climate) data. Calculations are based on moving window which is defined with two arguments: lower_limit and upper_limit. All calculated (partial) correlation coefficients are stored in a matrix. The location of stored correlation in the matrix is indicating a window width (row names) and a location in a matrix of daily sequences of environmental data (column names).

**Usage**

```
daily_response_seascorr(
  response,
  env_data_primary,
  env_data_control,
  lower_limit = 30,
  upper_limit = 90,
  fixed_width = 0,
  previous_year = FALSE,
  pcor_method = "pearson",
  remove_insignificant = TRUE,
  alpha = 0.05,
  row_names_subset = FALSE,
  aggregate_function_env_data_primary = "mean",
  aggregate_function_env_data_control = "mean",
  temporal_stability_check = "sequential",
  k = 2,
  k_running_window = 30,
  subset_years = NULL,
  ylimits = NULL,
  seed = NULL,
  tidy_env_data_primary = FALSE,
  tidy_env_data_control = FALSE,
  reference_window = "start",
  boot = FALSE,
  boot_n = 1000,
  boot_ci_type = "norm",
  boot_conf_int = 0.95,
 day_interval = ifelse(c(previous_year == TRUE, previous_year == TRUE), c(-1, 366), c(1,
    366)),
  dc_method = NULL,
  pcor_na_use = "pairwise.complete",
  skip_window_length = 1,
  skip_window_position = 1
)
```

**Arguments**

| | |
|---|---|
| response | a data frame with tree-ring proxy variable and (optional) years as row names. Row.names should be matched with those from env_data_primary and env_data_control data frame. If not, set the row_names_subset argument to TRUE. |
| env_data_primary | |
| | primary data frame of daily sequences of environmental data as columns and years as row names. Each row represents a year and each column represents a day of a year. Row.names should be matched with those from the response data frame. If not, set the argument row_names_subset to TRUE. Alternatively, env_data_primary could be a tidy data with three columns, i.e. Year, DOY and third column representing values of mean temperatures, sum of precipitation etc. If tidy data is passed to the function, set the argument tidy_env_data_primary to TRUE. |
| env_data_control | |
| | a data frame of daily sequences of environmental data as columns and years as row names. This data is used as control for calculations of partial correlation coefficients. Each row represents a year and each column represents a day of a year. Row.names should be matched with those from the response data frame. If not, set the row_names_subset argument to TRUE. Alternatively, env_data_control could be a tidy data with three columns, i.e. Year, DOY and third column representing values of mean temperatures, sum of precipitation etc. If tidy data is passed to the function, set the argument tidy_env_data_control to TRUE. |
| lower_limit | lower limit of window width |
| upper_limit | upper limit of window width |
| fixed_width | fixed width used for calculation. If fixed_width is assigned a value, upper_limit and lower_limit will be ignored |
| previous_year | if set to TRUE, env_data_primary, env_data_control and response variables will be rearranged in a way, that also previous year will be used for calculations of selected statistical metric. |
| pcor_method | a character string indicating which partial correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman", can be abbreviated. |
| remove_insignificant | |
| | if set to TRUE, removes all correlations bellow the significant threshold level, based on a selected alpha. |
| alpha | significance level used to remove insignificant calculations. |
| row_names_subset | |
| | if set to TRUE, row.names are used to subset env_data_primary, env_data_control and response data frames. Only years from all three data frames are kept. |
| aggregate_function_env_data_primary | |
| | character string specifying how the daily data from env_data_primary should be aggregated. The default is 'mean', the other options are 'median', 'sum', 'min' and 'max' |
| aggregate_function_env_data_control | |
| | character string specifying how the daily data from env_data_control should be aggregated. The default is 'mean', the other options are 'median', 'sum', 'min' and 'max' |

temporal_stability_check

               character string, specifying, how temporal stability between the optimal selection and response variable(s) will be analysed. Current possibilities are "sequential", "progressive" and "running_window". Sequential check will split data into k splits and calculate selected metric for each split. Progressive check will split data into k splits, calculate metric for the first split and then progressively add 1 split at a time and calculate selected metric. For running window, select the length of running window with the k_running_window argument.

k                                integer, number of breaks (splits) for temporal stability

k_running_window

               the length of running window for temporal stability check. Applicable only if temporal_stability argument is set to running window.

subset_years          a subset of years to be analyzed. Should be given in the form of subset_years = c(1980, 2005)

ylimits                  limit of the y axes for plot_extreme It should be given in the form of: ylimits = c(0,1)

seed                     optional seed argument for reproducible results

tidy_env_data_primary

               if set to TRUE, env_data_primary should be inserted as a data frame with three columns: "Year", "DOY", "Precipitation/Temperature/etc."

tidy_env_data_control

               if set to TRUE, env_data_control should be inserted as a data frame with three columns: "Year", "DOY", "Precipitation/Temperature/etc."

reference_window

               character string, the reference_window argument describes, how each calculation is referred. There are three different options: 'start' (default), 'end' and 'middle'. If the reference_window argument is set to 'start', then each calculation is related to the starting day of window. If the reference_window argument is set to 'middle', each calculation is related to the middle day of window calculation. If the reference_window argument is set to 'end', then each calculation is related to the ending day of window calculation. For example, if we consider correlations with window from DOY 15 to DOY 35. If reference window is set to 'start', then this calculation will be related to the DOY 15. If the reference window is set to 'end', then this calculation will be related to the DOY 35. If the reference_window is set to 'middle', then this calculation is related to DOY 25. The optimal selection, which describes the optimal consecutive days that returns the highest calculated metric and is obtained by the $plot_extreme output, is the same for all three reference windows.

boot                    logical, if TRUE, bootstrap procedure will be used to calculate partial correlation coefficients

boot_n                  The number of bootstrap replicates

boot_ci_type            A character string representing the type of bootstrap intervals required. The value should be any subset of the values c("norm","basic", "stud", "perc", "bca").

boot_conf_int           A scalar or vector containing the confidence level(s) of the required interval(s)

| | |
|---|---|
| day_interval | a vector of two values: lower and upper time interval of days that will be used to calculate statistical metrics. Negative values indicate previous growing season days. This argument overwrites the calculation limits defined by lower_limit and upper_limit arguments. |
| dc_method | a character string to determine the method to detrend climate data. Possible values are "none" (default) and "SLD" which refers to Simple Linear Detrending |
| pcor_na_use | an optional character string giving a method for computing covariances in the presence of missing values for partial correlation coefficients. This must be (an abbreviation of) one of the strings "all.obs", "everything", "complete.obs", "na.or.complete", or "pairwise.complete.obs" (default). See also the documentation for the base partial.r in psych R package |
| skip_window_length | |
| | an integer specifying the frequency of window selection for the calculations of climate-growth relationships. The default value is 1, indicating that every window is included in the calculations. When set to a value greater than 1, the function selectively processes windows at regular intervals defined by this parameter. For instance, if skip_window_length = 2, the function processes every second window. Similarly, if skip_window_length = 3, every third window is processed, skipping two windows in between each selected one. This parameter allows for controlling the granularity of the analysis and can help in reducing computation time by focusing on a subset of the data. |
| skip_window_position | |
| | an integer specifying the frequency of window positions used in the calculations of climate-growth relationships. The default value is 1, indicating that every window position is included in the calculations. When set to a value greater than 1, the function selectively processes window positions at regular intervals defined by this parameter. For instance, if skip_window_position = 2, the function processes every second window position. Similarly, if skip_window_position = 3, every third window position is processed, skipping two positions in between each selected one. This parameter allows for controlling the granularity of the analysis and can help in reducing computation time by focusing on a subset of the data. |

**Value**

a list with 15 elements:

1. $calculations - a matrix with calculated metrics
2. $method - the character string of a method
3. $metric - the character string indicating the metric used for calculations
4. $analysed_period - the character string specifying the analysed period based on the information from row names. If there are no row names, this argument is given as NA
5. $optimized_return - data frame with two columns, response variable and aggregated (averaged) daily data that return the optimal results. This data.frame could be directly used to calibrate a model for climate reconstruction
6. $optimized_return_all - a data frame with aggregated daily data, that returned the optimal result for the entire env_data_primary (and not only subset of analysed years)

7.  $transfer_function - a ggplot object: scatter plot of optimized return and a transfer line of the selected method

8.  $temporal_stability - a data frame with calculations of selected metric for different temporal subsets

9.  $cross_validation - not available for partial correlations

10. $plot_heatmap - ggplot2 object: a heatmap of calculated metrics

11. $plot_extreme - ggplot2 object: line plot of a row with the highest value in a matrix of calculated metrics

12. $type - the character string describing type of analysis: daily or monthly

13. $reference_window - character string, which reference window was used for calculations

14. $aggregated_climate_primary - matrix with all aggregated climate series of primary data

15. $aggregated_climate_control - matrix with all aggregated climate series of control data

**Examples**

```
# The examples below are enclosed within donttest{} to minimize the execution
# time during R package checks. Additionally, all examples include the
# parameters `skip_window_length` and `skip_window_position`, which limit the
# number of combinations evaluated in climate-growth correlation calculations.
# To explore all possible combinations, users should set both parameters to 1.

# Load the dendroTools R package
library(dendroTools)

# Load data
data(data_MVA)
data(data_TRW)
data(data_TRW_1)
data(example_proxies_individual)
data(example_proxies_1)
data(LJ_daily_temperatures)
data(LJ_daily_precipitation)

# 1 Basic example using the partial correlation coefficient
example_basic <- daily_response_seascorr(response = data_MVA,
                         env_data_primary = LJ_daily_temperatures,
                         env_data_control = LJ_daily_precipitation,
                         row_names_subset = TRUE,
                         fixed_width = 25,
                         lower_limit = 35, upper_limit = 45,
                         remove_insignificant = FALSE,
                         aggregate_function_env_data_primary = 'mean',
                         aggregate_function_env_data_control = 'mean',
                         tidy_env_data_primary = FALSE,
                         tidy_env_data_control = TRUE,
                         alpha = 0.05, pcor_method = "spearman",
                         previous_year = FALSE, boot = TRUE,
                         boot_n = 10,
```

```
                                reference_window = "end", k = 5,
                                dc_method = "SLD",
                                day_interval = c(-100, 250),
                                skip_window_position = 50,
                                skip_window_length= 50
                                )
# summary(example_basic)
# plot(example_basic, type = 1)
# plot(example_basic, type = 2)
# example_basic$optimized_return
# example_basic$optimized_return_all
# example_basic$temporal_stability

# 2 Example with fixed temporal time window
example_fixed_width <- daily_response_seascorr(response = data_MVA,
                           env_data_primary = LJ_daily_temperatures,
                           env_data_control = LJ_daily_precipitation,
                           row_names_subset = TRUE,
                           remove_insignificant = TRUE,
                           aggregate_function_env_data_primary = 'mean',
                           aggregate_function_env_data_control = 'mean',
                           alpha = 0.05,
                           dc_method = "SLD",
                           fixed_width = 45,
                           tidy_env_data_primary = FALSE,
                           tidy_env_data_control = TRUE,
                           reference_window = "end",
                           skip_window_position = 50,
                           skip_window_length= 50)

# summary(example_fixed_width)
# plot(example_fixed_width, type = 1)
# plot(example_fixed_width, type = 2)
# example_fixed_width$optimized_return
# example_fixed_width$optimized_return_all
```

---

dataset_MVA                 *MVA and mean April temperature*

---

### Description

A dataset with a mean vessel area (MVA) chronology of Quercus robur from a lowland oak forest
in Eastern Slovenia and a mean April temperature. This dataset includes years for the period 2012-
1934. For a detailed description about the MVA chronology development, sampling site and the
calculations of mean monthly correlations, see Jevšenak and Levanič (2015).

### Usage

```
dataset_MVA
```

**Format**

A data frame with 79 rows and 2 variables:

**MVA** Mean vessel area measurements from 2012 - 1934

**T_Apr** Mean April temperature for the meteorological station Maribor from 2012 - 1934

**Source**

Jevšenak J., Levanič T. 2015. Dendrochronological and wood-anatomical features of differently vital pedunculate oak (Quercus robur L.) stands and their response to climate. Topola, 195/196: 85-96

---

dataset_MVA_individual

*Example of dataset with individual chronologies of MVA and mean April temperature*

---

**Description**

A dataset of individual tree-ring chronologies from a lowland forest in Slovenia. The first row represents a value of a year in 2015.

**Usage**

dataset_MVA_individual

**Format**

A data frame with 56 rows and 54 columns :

**T_Apr** mean April temperature for Ljubljana

**MVA_1** Mean vessel area chronology for tree 1

**MVA_2** Mean vessel area chronology for tree 2 [mm^2]

**MVA_3** Mean vessel area chronology for tree 3 [mm^2]

**MVA_4** Mean vessel area chronology for tree 4 [mm^2]

**MVA_5** Mean vessel area chronology for tree 5 [mm^2]

**MVA_6** Mean vessel area chronology for tree 6 [mm^2]

**MVA_7** Mean vessel area chronology for tree 7 [mm^2]

**MVA_8** Mean vessel area chronology for tree 8 [mm^2]

**MVA_9** Mean vessel area chronology for tree 9 [mm^2]

**MVA_10** Mean vessel area chronology for tree 10 [mm^2]

**Source**

Slovenian Forestry Institute, Večna pot 2, Ljubljana, Slovenia

---

dataset_TRW            *TRW and mean June - July temperature from Albania*

---

## Description

A dataset with a tree-ring width (TRW) chronology of Pinus nigra from Albania and mean June-July temperature. This TRW chronology has a span of 59 years (period 2009 - 1951) and was already used to reconstruct summer temperatures by Levanič et al. (2015). In this paper, all the details about sample replication, site description and correlation statistics are described.

## Usage

```
dataset_TRW
```

## Format

A data frame with 59 rows and 2 variables:

**TRW** Standardised tree-ring width chronology of Pinus nigra from Albania

**T_Jun_Jul** Mean June - July temperature for Albania downloaded from KNMI Climate Explorer

## Source

Levanič, T., Poljanšek, S., Toromani, E., 2015. Early summer temperatures reconstructed from black pine (Pinus nigra Arnold) tree-ring widths from Albania. The Holocene 25, 469-481.

---

dataset_TRW_complete    *The complete dataset of standardized tree-ring chronology from Albania*

---

## Description

A dataset with a tree-ring width (TRW) chronology of Pinus nigra from Albania This TRW chronology has a span of 551 years (period 2009 - 1459) and was already used to reconstruct summer temperatures by Levanič et al. (2015). In this paper, all the details about sample replication, site description and correlation statistics are described.

## Usage

```
dataset_TRW_complete
```

## Format

A data frame with 551 rows and 1 variable:

**TRW** Standardised tree-ring width chronology of Pinus nigra from Albania

**Source**

Levanič, T., Poljanšek, S., Toromani, E., 2015. Early summer temperatures reconstructed from black pine (Pinus nigra Arnold) tree-ring widths from Albania. The Holocene 25, 469-481.

---

data_MVA                                    *Mean vessel area example proxy from 2012 - 1940*

---

**Description**

A dataset with MVA proxy records from a lowland forest Mlače in Slovenia. The first row represents a value of a year in 2012. Row names represent years.

**Usage**

```
data_MVA
```

**Format**

A data frame with 73 rows and 1 variable:

**MVA** Mean vessel area [mm^2] indices from 2012 - 1940

**Source**

Jernej Jevšenak, Slovenian Forestry Institute, Večna pot 2, Ljubljana, Slovenia

---

data_transform                             *data_transform*

---

**Description**

Transforms daily data with two columns (date and variable) into data frame suitable for daily or monthly analysis with dendroTools.

**Usage**

```
data_transform(
  input,
  format = "daily",
  monthly_aggregate_function = "auto",
  date_format = "ymd"
)
```

## Arguments

| | |
|---|---|
| `input` | typical daily data format: Data frame with two columns, first column represents date, second column represents variable, such as mean temperature, precipitation, etc. Date should be in format Year-Month-Day (e.g. "2019-05-15") |
| `format` | character string indicating the desired output format. Should be "daily" or "monthly". Daily format returns a data frame with 366 columns (days), while monthly format returns data frame with 12 columns (months). Years are indicated as row names. |
| `monthly_aggregate_function` | |
| | character string indicating, how to aggregate daily into monthly data. It can be "mean" or "sum". Third option is "auto" (default). In this case function will try to guess whether input is temperature or precipitation data. For temperature, it will use "mean", for precipitation "sum". |
| `date_format` | Describe the format of date. It should be one of "ymd", "ydm", "myd", "mdy", "dmy", "dym". |

## Value

env_data suitable for daily or monthly analysis with dendroTools.

## Examples

```
data(swit272_daily_temperatures)
proper_daily_data <- data_transform(swit272_daily_temperatures, format = "daily",
  date_format = "ymd")

proper_monthly_data <- data_transform(swit272_daily_temperatures, format = "monthly",
  date_format = "ymd")

data(swit272_daily_precipitation)
proper_daily_data <- data_transform(swit272_daily_precipitation, format = "daily",
  date_format = "ymd")

proper_monthly_data <- data_transform(swit272_daily_precipitation, format = "monthly",
  date_format = "ymd")
```

---

data_TRW                    *Tree-ring width (TRW) example proxy from 1981 - 1757*

---

## Description

A dataset with TRW proxy records from a site in Slovenian Alps - Vrsic. The first row represents a TRW value in a year 1757. Row names represent years.

## Usage

```
data_TRW
```

## Format

A data frame with 225 rows and 1 variable:

**TRW** residual TRW indices from 1981 - 1757

## Source

- Schweingruber, F.H., 1981. Vrsic Krajnska Gora - PCAB - ITRDB YUGO001.

- https://www.ncei.noaa.gov/access/paleo-search/study/4728

---

data_TRW_1 *Tree-ring width (TRW) data from 2012 - 1961*

---

## Description

A dataset of tree-ring widths (TRW) from a site in Krakovo forest (Slovenia). The first row represents a value of a year in 1961.

## Usage

```
data_TRW_1
```

## Format

A data frame with 52 rows and 1 variable:

**TRW** Standardized tree-ring width indices from 2012 - 1961

## Source

Tom Levanič, Slovenian Forestry Institute, Večna pot 2, Ljubljana, Slovenia

---

example_dataset_1 *Example of dataset as required for compare_methods()*

---

## Description

A dataset of Mean Vessel Area (MVA) tree-ring parameter from a lowland forest in Slovenia. The first row represents a value of a year in 2012.

## Usage

```
example_dataset_1
```

## Format

A data frame with 58 rows and 3 columns :

**MVA** Mean Vessel Area measurements from 2012 - 1955

**T_APR** Mean April temperatures from 2012 - 1955

**T_aug_sep** Mean August-September temperatures from preceding growing season from 2012 - 1955

## Source

Jernej Jevšenak, Slovenian Forestry Institute, Večna pot 2, Ljubljana, Slovenia

---

example_proxies_1       *Tree-ring example proxies 1 from 2015 - 1961*

---

## Description

A dataset with three tree-ring proxy records from a site near Ljubljana (Slovenia). The first row represents a value of a year in 1961. The three proxy records are MVA (Mean vessel area [mm ^2]), O (stable oxygen isotope ratios) and TRW (Tree-ring widths)

## Usage

```
example_proxies_1
```

## Format

A data frame with 55 rows and 3 variables:

**MVA** Mean vessel area [mm^2] indices from 2015 - 1961

**O18** Scaled Stable oxygen isotope ratios from 2015 - 1961

**TRW** Tree-ring widths from 2015 - 1961

## Source

Jernej Jevšenak, Slovenian Forestry Institute, Večna pot 2, Ljubljana, Slovenia

---

example_proxies_individual

*Example of dataset with individual chronologies of MVA.*

---

### Description

A dataset of individual tree-ring chronologies from a lowland forest in Slovenia. The first row represents a value of a year in 2015.

### Usage

```
example_proxies_individual
```

### Format

A data frame with 56 rows and 54 columns :

**MVA_1** Mean vessel area chronology for tree 1

**MVA_2** Mean vessel area chronology for tree 2

**MVA_3** Mean vessel area chronology for tree 3

**MVA_4** Mean vessel area chronology for tree 4

**MVA_5** Mean vessel area chronology for tree 5

**MVA_6** Mean vessel area chronology for tree 6

**MVA_7** Mean vessel area chronology for tree 7

**MVA_8** Mean vessel area chronology for tree 8

**MVA_9** Mean vessel area chronology for tree 9

**MVA_10** Mean vessel area chronology for tree 10

### Source

Jernej Jevšenak, Slovenian Forestry Institute, Večna pot 2, Ljubljana, Slovenia

---

glimpse_daily_data                *glimpse_daily_data*

---

### Description

Visual presentation of daily data to spot missing values.

## Usage

```
glimpse_daily_data(
  env_data,
  na.color = "red",
  low_color = "blue",
  high_color = "green",
  tidy_env_data = FALSE
)
```

## Arguments

env_data        a data frame of daily sequences of environmental data as columns and years
                as row names. Each row represents a year and each column represents a day
                of a year. Alternatively, env_data could be a tidy data with three columns, i.e.
                Year, DOY and third column representing values of mean temperatures, sum
                of precipitation etc. If tidy data is passed to the function, set the argument
                tidy_env_data to TRUE.

na.color        color to use for missing values

low_color       colours for low end of the gradient

high_color      colours for high end of the gradient

tidy_env_data   if set to TRUE, env_data should be inserted as a data frame with three columns:
                "Year", "DOY", "Precipitation/Temperature/etc."

## Examples

```
library(dendroTools)
data("LJ_daily_temperatures")
# glimpse_daily_data(env_data = LJ_daily_temperatures,
#tidy_env_data = FALSE, na.color = "white")

data("LJ_daily_precipitation")
# glimpse_daily_data(env_data = LJ_daily_precipitation,
#   tidy_env_data = TRUE, na.color = "white")
```

---

KRE_daily_temperatures

*Daily mean temperatures for Kredarica (Alps in Slovenia) from 2017*
*- 1955*

---

## Description

A dataset of daily mean temperatures in Kredarica (Slovenia). The first row represents temperatures
in 1955. The first column represents the first day of a year, the second column represents the second
day of a year, etc. Row names represent years.

**Usage**

```
KRE_daily_temperatures
```

**Format**

A data frame with 63 rows and 366 variables:

**X1** Temperatures on the day 1 of a year

**X2** Temperatures on the day 2 of a year

**X3** Temperatures on the day 3 of a year

**X4** Temperatures on the day 4 of a year

**X5** Temperatures on the day 5 of a year

**X6** Temperatures on the day 6 of a year

**X7** Temperatures on the day 7 of a year

**X8** Temperatures on the day 8 of a year

**X9** Temperatures on the day 9 of a year

**X10** Temperatures on the day 10 of a year

**X11** Temperatures on the day 11 of a year

**X12** Temperatures on the day 12 of a year

**X13** Temperatures on the day 13 of a year

**X14** Temperatures on the day 14 of a year

**X15** Temperatures on the day 15 of a year

**X16** Temperatures on the day 16 of a year

**X17** Temperatures on the day 17 of a year

**X18** Temperatures on the day 18 of a year

**X19** Temperatures on the day 19 of a year

**X20** Temperatures on the day 20 of a year

**X21** Temperatures on the day 21 of a year

**X22** Temperatures on the day 22 of a year

**X23** Temperatures on the day 23 of a year

**X24** Temperatures on the day 24 of a year

**X25** Temperatures on the day 25 of a year

**X26** Temperatures on the day 26 of a year

**X27** Temperatures on the day 27 of a year

**X28** Temperatures on the day 28 of a year

**X29** Temperatures on the day 29 of a year

**X30** Temperatures on the day 30 of a year

**X31** Temperatures on the day 31 of a year

**X32** Temperatures on the day 32 of a year

**X33** Temperatures on the day 33 of a year

**X34** Temperatures on the day 34 of a year

**X35** Temperatures on the day 35 of a year

**X36** Temperatures on the day 36 of a year

**X37** Temperatures on the day 37 of a year

**X38** Temperatures on the day 38 of a year

**X39** Temperatures on the day 39 of a year

**X40** Temperatures on the day 40 of a year

**X41** Temperatures on the day 41 of a year

**X42** Temperatures on the day 42 of a year

**X43** Temperatures on the day 43 of a year

**X44** Temperatures on the day 44 of a year

**X45** Temperatures on the day 45 of a year

**X46** Temperatures on the day 46 of a year

**X47** Temperatures on the day 47 of a year

**X48** Temperatures on the day 48 of a year

**X49** Temperatures on the day 49 of a year

**X50** Temperatures on the day 50 of a year

**X51** Temperatures on the day 51 of a year

**X52** Temperatures on the day 52 of a year

**X53** Temperatures on the day 53 of a year

**X54** Temperatures on the day 54 of a year

**X55** Temperatures on the day 55 of a year

**X56** Temperatures on the day 56 of a year

**X57** Temperatures on the day 57 of a year

**X58** Temperatures on the day 58 of a year

**X59** Temperatures on the day 59 of a year

**X60** Temperatures on the day 60 of a year

**X61** Temperatures on the day 61 of a year

**X62** Temperatures on the day 62 of a year

**X63** Temperatures on the day 63 of a year

**X64** Temperatures on the day 64 of a year

**X65** Temperatures on the day 65 of a year

**X66** Temperatures on the day 66 of a year

**X67** Temperatures on the day 67 of a year

**X68** Temperatures on the day 68 of a year

**X69** Temperatures on the day 69 of a year

**X70**  Temperatures on the day 70 of a year

**X71**  Temperatures on the day 71 of a year

**X72**  Temperatures on the day 72 of a year

**X73**  Temperatures on the day 73 of a year

**X74**  Temperatures on the day 74 of a year

**X75**  Temperatures on the day 75 of a year

**X76**  Temperatures on the day 76 of a year

**X77**  Temperatures on the day 77 of a year

**X78**  Temperatures on the day 78 of a year

**X79**  Temperatures on the day 79 of a year

**X80**  Temperatures on the day 80 of a year

**X81**  Temperatures on the day 81 of a year

**X82**  Temperatures on the day 82 of a year

**X83**  Temperatures on the day 83 of a year

**X84**  Temperatures on the day 84 of a year

**X85**  Temperatures on the day 85 of a year

**X86**  Temperatures on the day 86 of a year

**X87**  Temperatures on the day 87 of a year

**X88**  Temperatures on the day 88 of a year

**X89**  Temperatures on the day 89 of a year

**X90**  Temperatures on the day 90 of a year

**X91**  Temperatures on the day 91 of a year

**X92**  Temperatures on the day 92 of a year

**X93**  Temperatures on the day 93 of a year

**X94**  Temperatures on the day 94 of a year

**X95**  Temperatures on the day 95 of a year

**X96**  Temperatures on the day 96 of a year

**X97**  Temperatures on the day 97 of a year

**X98**  Temperatures on the day 98 of a year

**X99**  Temperatures on the day 99 of a year

**X100**  Temperatures on the day 100 of a year

**X101**  Temperatures on the day 101 of a year

**X102**  Temperatures on the day 102 of a year

**X103**  Temperatures on the day 103 of a year

**X104**  Temperatures on the day 104 of a year

**X105**  Temperatures on the day 105 of a year

**X106**  Temperatures on the day 106 of a year

**X107** Temperatures on the day 107 of a year

**X108** Temperatures on the day 108 of a year

**X109** Temperatures on the day 109 of a year

**X110** Temperatures on the day 110 of a year

**X111** Temperatures on the day 111 of a year

**X112** Temperatures on the day 112 of a year

**X113** Temperatures on the day 113 of a year

**X114** Temperatures on the day 114 of a year

**X115** Temperatures on the day 115 of a year

**X116** Temperatures on the day 116 of a year

**X117** Temperatures on the day 117 of a year

**X118** Temperatures on the day 118 of a year

**X119** Temperatures on the day 119 of a year

**X120** Temperatures on the day 120 of a year

**X121** Temperatures on the day 121 of a year

**X122** Temperatures on the day 122 of a year

**X123** Temperatures on the day 123 of a year

**X124** Temperatures on the day 124 of a year

**X125** Temperatures on the day 125 of a year

**X126** Temperatures on the day 126 of a year

**X127** Temperatures on the day 127 of a year

**X128** Temperatures on the day 128 of a year

**X129** Temperatures on the day 129 of a year

**X130** Temperatures on the day 130 of a year

**X131** Temperatures on the day 131 of a year

**X132** Temperatures on the day 132 of a year

**X133** Temperatures on the day 133 of a year

**X134** Temperatures on the day 134 of a year

**X135** Temperatures on the day 135 of a year

**X136** Temperatures on the day 136 of a year

**X137** Temperatures on the day 137 of a year

**X138** Temperatures on the day 138 of a year

**X139** Temperatures on the day 139 of a year

**X140** Temperatures on the day 140 of a year

**X141** Temperatures on the day 141 of a year

**X142** Temperatures on the day 142 of a year

**X143** Temperatures on the day 143 of a year

**X144** Temperatures on the day 144 of a year

**X145** Temperatures on the day 145 of a year

**X146** Temperatures on the day 146 of a year

**X147** Temperatures on the day 147 of a year

**X148** Temperatures on the day 148 of a year

**X149** Temperatures on the day 149 of a year

**X150** Temperatures on the day 150 of a year

**X151** Temperatures on the day 151 of a year

**X152** Temperatures on the day 152 of a year

**X153** Temperatures on the day 153 of a year

**X154** Temperatures on the day 154 of a year

**X155** Temperatures on the day 155 of a year

**X156** Temperatures on the day 156 of a year

**X157** Temperatures on the day 157 of a year

**X158** Temperatures on the day 158 of a year

**X159** Temperatures on the day 159 of a year

**X160** Temperatures on the day 160 of a year

**X161** Temperatures on the day 161 of a year

**X162** Temperatures on the day 162 of a year

**X163** Temperatures on the day 163 of a year

**X164** Temperatures on the day 164 of a year

**X165** Temperatures on the day 165 of a year

**X166** Temperatures on the day 166 of a year

**X167** Temperatures on the day 167 of a year

**X168** Temperatures on the day 168 of a year

**X169** Temperatures on the day 169 of a year

**X170** Temperatures on the day 170 of a year

**X171** Temperatures on the day 171 of a year

**X172** Temperatures on the day 172 of a year

**X173** Temperatures on the day 173 of a year

**X174** Temperatures on the day 174 of a year

**X175** Temperatures on the day 175 of a year

**X176** Temperatures on the day 176 of a year

**X177** Temperatures on the day 177 of a year

**X178** Temperatures on the day 178 of a year

**X179** Temperatures on the day 179 of a year

**X180** Temperatures on the day 180 of a year

**X181** Temperatures on the day 181 of a year

**X182** Temperatures on the day 182 of a year

**X183** Temperatures on the day 183 of a year

**X184** Temperatures on the day 184 of a year

**X185** Temperatures on the day 185 of a year

**X186** Temperatures on the day 186 of a year

**X187** Temperatures on the day 187 of a year

**X188** Temperatures on the day 188 of a year

**X189** Temperatures on the day 189 of a year

**X190** Temperatures on the day 190 of a year

**X191** Temperatures on the day 191 of a year

**X192** Temperatures on the day 192 of a year

**X193** Temperatures on the day 193 of a year

**X194** Temperatures on the day 194 of a year

**X195** Temperatures on the day 195 of a year

**X196** Temperatures on the day 196 of a year

**X197** Temperatures on the day 197 of a year

**X198** Temperatures on the day 198 of a year

**X199** Temperatures on the day 199 of a year

**X200** Temperatures on the day 200 of a year

**X201** Temperatures on the day 201 of a year

**X202** Temperatures on the day 202 of a year

**X203** Temperatures on the day 203 of a year

**X204** Temperatures on the day 204 of a year

**X205** Temperatures on the day 205 of a year

**X206** Temperatures on the day 206 of a year

**X207** Temperatures on the day 207 of a year

**X208** Temperatures on the day 208 of a year

**X209** Temperatures on the day 209 of a year

**X210** Temperatures on the day 210 of a year

**X211** Temperatures on the day 211 of a year

**X212** Temperatures on the day 212 of a year

**X213** Temperatures on the day 213 of a year

**X214** Temperatures on the day 214 of a year

**X215** Temperatures on the day 215 of a year

**X216** Temperatures on the day 216 of a year

**X217** Temperatures on the day 217 of a year

**X218** Temperatures on the day 218 of a year

**X219** Temperatures on the day 219 of a year

**X220** Temperatures on the day 220 of a year

**X221** Temperatures on the day 221 of a year

**X222** Temperatures on the day 222 of a year

**X223** Temperatures on the day 223 of a year

**X224** Temperatures on the day 224 of a year

**X225** Temperatures on the day 225 of a year

**X226** Temperatures on the day 226 of a year

**X227** Temperatures on the day 227 of a year

**X228** Temperatures on the day 228 of a year

**X229** Temperatures on the day 229 of a year

**X230** Temperatures on the day 230 of a year

**X231** Temperatures on the day 231 of a year

**X232** Temperatures on the day 232 of a year

**X233** Temperatures on the day 233 of a year

**X234** Temperatures on the day 234 of a year

**X235** Temperatures on the day 235 of a year

**X236** Temperatures on the day 236 of a year

**X237** Temperatures on the day 237 of a year

**X238** Temperatures on the day 238 of a year

**X239** Temperatures on the day 239 of a year

**X240** Temperatures on the day 240 of a year

**X241** Temperatures on the day 241 of a year

**X242** Temperatures on the day 242 of a year

**X243** Temperatures on the day 243 of a year

**X244** Temperatures on the day 244 of a year

**X245** Temperatures on the day 245 of a year

**X246** Temperatures on the day 246 of a year

**X247** Temperatures on the day 247 of a year

**X248** Temperatures on the day 248 of a year

**X249** Temperatures on the day 249 of a year

**X250** Temperatures on the day 250 of a year

**X251** Temperatures on the day 251 of a year

**X252** Temperatures on the day 252 of a year

**X253** Temperatures on the day 253 of a year

**X254** Temperatures on the day 254 of a year

**X255** Temperatures on the day 255 of a year

**X256** Temperatures on the day 256 of a year

**X257** Temperatures on the day 257 of a year

**X258** Temperatures on the day 258 of a year

**X259** Temperatures on the day 259 of a year

**X260** Temperatures on the day 260 of a year

**X261** Temperatures on the day 261 of a year

**X262** Temperatures on the day 262 of a year

**X263** Temperatures on the day 263 of a year

**X264** Temperatures on the day 264 of a year

**X265** Temperatures on the day 265 of a year

**X266** Temperatures on the day 266 of a year

**X267** Temperatures on the day 267 of a year

**X268** Temperatures on the day 268 of a year

**X269** Temperatures on the day 269 of a year

**X270** Temperatures on the day 270 of a year

**X271** Temperatures on the day 271 of a year

**X272** Temperatures on the day 272 of a year

**X273** Temperatures on the day 273 of a year

**X274** Temperatures on the day 274 of a year

**X275** Temperatures on the day 275 of a year

**X276** Temperatures on the day 276 of a year

**X277** Temperatures on the day 277 of a year

**X278** Temperatures on the day 278 of a year

**X279** Temperatures on the day 279 of a year

**X280** Temperatures on the day 280 of a year

**X281** Temperatures on the day 281 of a year

**X282** Temperatures on the day 282 of a year

**X283** Temperatures on the day 283 of a year

**X284** Temperatures on the day 284 of a year

**X285** Temperatures on the day 285 of a year

**X286** Temperatures on the day 286 of a year

**X287** Temperatures on the day 287 of a year

**X288** Temperatures on the day 288 of a year

**X289** Temperatures on the day 289 of a year

**X290** Temperatures on the day 290 of a year

**X291** Temperatures on the day 291 of a year

**X292** Temperatures on the day 292 of a year

**X293** Temperatures on the day 293 of a year

**X294** Temperatures on the day 294 of a year

**X295** Temperatures on the day 295 of a year

**X296** Temperatures on the day 296 of a year

**X297** Temperatures on the day 297 of a year

**X298** Temperatures on the day 298 of a year

**X299** Temperatures on the day 299 of a year

**X300** Temperatures on the day 300 of a year

**X301** Temperatures on the day 301 of a year

**X302** Temperatures on the day 302 of a year

**X303** Temperatures on the day 303 of a year

**X304** Temperatures on the day 304 of a year

**X305** Temperatures on the day 305 of a year

**X306** Temperatures on the day 306 of a year

**X307** Temperatures on the day 307 of a year

**X308** Temperatures on the day 308 of a year

**X309** Temperatures on the day 309 of a year

**X310** Temperatures on the day 310 of a year

**X311** Temperatures on the day 311 of a year

**X312** Temperatures on the day 312 of a year

**X313** Temperatures on the day 313 of a year

**X314** Temperatures on the day 314 of a year

**X315** Temperatures on the day 315 of a year

**X316** Temperatures on the day 316 of a year

**X317** Temperatures on the day 317 of a year

**X318** Temperatures on the day 318 of a year

**X319** Temperatures on the day 319 of a year

**X320** Temperatures on the day 320 of a year

**X321** Temperatures on the day 321 of a year

**X322** Temperatures on the day 322 of a year

**X323** Temperatures on the day 323 of a year

**X324** Temperatures on the day 324 of a year

**X325** Temperatures on the day 325 of a year

**X326** Temperatures on the day 326 of a year

**X327** Temperatures on the day 327 of a year

**X328** Temperatures on the day 328 of a year

**X329** Temperatures on the day 329 of a year

**X330** Temperatures on the day 330 of a year

**X331** Temperatures on the day 331 of a year

**X332** Temperatures on the day 332 of a year

**X333** Temperatures on the day 333 of a year

**X334** Temperatures on the day 334 of a year

**X335** Temperatures on the day 335 of a year

**X336** Temperatures on the day 336 of a year

**X337** Temperatures on the day 337 of a year

**X338** Temperatures on the day 338 of a year

**X339** Temperatures on the day 339 of a year

**X340** Temperatures on the day 340 of a year

**X341** Temperatures on the day 341 of a year

**X342** Temperatures on the day 342 of a year

**X343** Temperatures on the day 343 of a year

**X344** Temperatures on the day 344 of a year

**X345** Temperatures on the day 345 of a year

**X346** Temperatures on the day 346 of a year

**X347** Temperatures on the day 347 of a year

**X348** Temperatures on the day 348 of a year

**X349** Temperatures on the day 349 of a year

**X350** Temperatures on the day 350 of a year

**X351** Temperatures on the day 351 of a year

**X352** Temperatures on the day 352 of a year

**X353** Temperatures on the day 353 of a year

**X354** Temperatures on the day 354 of a year

**X355** Temperatures on the day 355 of a year

**X356** Temperatures on the day 356 of a year

**X357** Temperatures on the day 357 of a year

**X358** Temperatures on the day 358 of a year

**X359** Temperatures on the day 359 of a year

**X360** Temperatures on the day 360 of a year

**X361** Temperatures on the day 361 of a year

**X362** Temperatures on the day 362 of a year

**X363** Temperatures on the day 363 of a year

**X364** Temperatures on the day 364 of a year

**X365** Temperatures on the day 365 of a year

**X366** Temperatures on the day 366 of a year

## Source

<https://meteo.arso.gov.si/met/sl/archive/>

---

`LJ_daily_precipitation`

*Daily precipitation for Ljubljana from 2017 - 1900*

---

#### Description

A dataset of daily sum of precipitation [mm] in Ljubljana (Slovenia). The first row represents precipitation in 1900 on DOY 1.

#### Usage

`LJ_daily_precipitation`

#### Format

A data frame with 43067 rows and 3 variables:

**Year** year

**DOY** day of year

**Precipitation** Sum of precipitation in mm

#### Source

<http://climexp.knmi.nl/start.cgi>

---

`LJ_daily_temperatures`   *Daily mean temperatures for Ljubljana from 2016 - 1930*

---

#### Description

A dataset of daily mean temperatures in Ljubljana (Slovenia). The first row represents temperatures in 1930. The first column represents the first day of a year, the second column represents the second day of a year, etc.

#### Usage

`LJ_daily_temperatures`

**Format**

A data frame with 87 rows and 366 variables:

**X1** Temperatures on the day 1 of a year

**X2** Temperatures on the day 2 of a year

**X3** Temperatures on the day 3 of a year

**X4** Temperatures on the day 4 of a year

**X5** Temperatures on the day 5 of a year

**X6** Temperatures on the day 6 of a year

**X7** Temperatures on the day 7 of a year

**X8** Temperatures on the day 8 of a year

**X9** Temperatures on the day 9 of a year

**X10** Temperatures on the day 10 of a year

**X11** Temperatures on the day 11 of a year

**X12** Temperatures on the day 12 of a year

**X13** Temperatures on the day 13 of a year

**X14** Temperatures on the day 14 of a year

**X15** Temperatures on the day 15 of a year

**X16** Temperatures on the day 16 of a year

**X17** Temperatures on the day 17 of a year

**X18** Temperatures on the day 18 of a year

**X19** Temperatures on the day 19 of a year

**X20** Temperatures on the day 20 of a year

**X21** Temperatures on the day 21 of a year

**X22** Temperatures on the day 22 of a year

**X23** Temperatures on the day 23 of a year

**X24** Temperatures on the day 24 of a year

**X25** Temperatures on the day 25 of a year

**X26** Temperatures on the day 26 of a year

**X27** Temperatures on the day 27 of a year

**X28** Temperatures on the day 28 of a year

**X29** Temperatures on the day 29 of a year

**X30** Temperatures on the day 30 of a year

**X31** Temperatures on the day 31 of a year

**X32** Temperatures on the day 32 of a year

**X33** Temperatures on the day 33 of a year

**X34** Temperatures on the day 34 of a year

**X35** Temperatures on the day 35 of a year

**X36** Temperatures on the day 36 of a year

**X37** Temperatures on the day 37 of a year

**X38** Temperatures on the day 38 of a year

**X39** Temperatures on the day 39 of a year

**X40** Temperatures on the day 40 of a year

**X41** Temperatures on the day 41 of a year

**X42** Temperatures on the day 42 of a year

**X43** Temperatures on the day 43 of a year

**X44** Temperatures on the day 44 of a year

**X45** Temperatures on the day 45 of a year

**X46** Temperatures on the day 46 of a year

**X47** Temperatures on the day 47 of a year

**X48** Temperatures on the day 48 of a year

**X49** Temperatures on the day 49 of a year

**X50** Temperatures on the day 50 of a year

**X51** Temperatures on the day 51 of a year

**X52** Temperatures on the day 52 of a year

**X53** Temperatures on the day 53 of a year

**X54** Temperatures on the day 54 of a year

**X55** Temperatures on the day 55 of a year

**X56** Temperatures on the day 56 of a year

**X57** Temperatures on the day 57 of a year

**X58** Temperatures on the day 58 of a year

**X59** Temperatures on the day 59 of a year

**X60** Temperatures on the day 60 of a year

**X61** Temperatures on the day 61 of a year

**X62** Temperatures on the day 62 of a year

**X63** Temperatures on the day 63 of a year

**X64** Temperatures on the day 64 of a year

**X65** Temperatures on the day 65 of a year

**X66** Temperatures on the day 66 of a year

**X67** Temperatures on the day 67 of a year

**X68** Temperatures on the day 68 of a year

**X69** Temperatures on the day 69 of a year

**X70** Temperatures on the day 70 of a year

**X71** Temperatures on the day 71 of a year

**X72** Temperatures on the day 72 of a year

**X73** Temperatures on the day 73 of a year

**X74** Temperatures on the day 74 of a year

**X75** Temperatures on the day 75 of a year

**X76** Temperatures on the day 76 of a year

**X77** Temperatures on the day 77 of a year

**X78** Temperatures on the day 78 of a year

**X79** Temperatures on the day 79 of a year

**X80** Temperatures on the day 80 of a year

**X81** Temperatures on the day 81 of a year

**X82** Temperatures on the day 82 of a year

**X83** Temperatures on the day 83 of a year

**X84** Temperatures on the day 84 of a year

**X85** Temperatures on the day 85 of a year

**X86** Temperatures on the day 86 of a year

**X87** Temperatures on the day 87 of a year

**X88** Temperatures on the day 88 of a year

**X89** Temperatures on the day 89 of a year

**X90** Temperatures on the day 90 of a year

**X91** Temperatures on the day 91 of a year

**X92** Temperatures on the day 92 of a year

**X93** Temperatures on the day 93 of a year

**X94** Temperatures on the day 94 of a year

**X95** Temperatures on the day 95 of a year

**X96** Temperatures on the day 96 of a year

**X97** Temperatures on the day 97 of a year

**X98** Temperatures on the day 98 of a year

**X99** Temperatures on the day 99 of a year

**X100** Temperatures on the day 100 of a year

**X101** Temperatures on the day 101 of a year

**X102** Temperatures on the day 102 of a year

**X103** Temperatures on the day 103 of a year

**X104** Temperatures on the day 104 of a year

**X105** Temperatures on the day 105 of a year

**X106** Temperatures on the day 106 of a year

**X107** Temperatures on the day 107 of a year

**X108** Temperatures on the day 108 of a year

**X109** Temperatures on the day 109 of a year

**X110** Temperatures on the day 110 of a year

**X111** Temperatures on the day 111 of a year

**X112** Temperatures on the day 112 of a year

**X113** Temperatures on the day 113 of a year

**X114** Temperatures on the day 114 of a year

**X115** Temperatures on the day 115 of a year

**X116** Temperatures on the day 116 of a year

**X117** Temperatures on the day 117 of a year

**X118** Temperatures on the day 118 of a year

**X119** Temperatures on the day 119 of a year

**X120** Temperatures on the day 120 of a year

**X121** Temperatures on the day 121 of a year

**X122** Temperatures on the day 122 of a year

**X123** Temperatures on the day 123 of a year

**X124** Temperatures on the day 124 of a year

**X125** Temperatures on the day 125 of a year

**X126** Temperatures on the day 126 of a year

**X127** Temperatures on the day 127 of a year

**X128** Temperatures on the day 128 of a year

**X129** Temperatures on the day 129 of a year

**X130** Temperatures on the day 130 of a year

**X131** Temperatures on the day 131 of a year

**X132** Temperatures on the day 132 of a year

**X133** Temperatures on the day 133 of a year

**X134** Temperatures on the day 134 of a year

**X135** Temperatures on the day 135 of a year

**X136** Temperatures on the day 136 of a year

**X137** Temperatures on the day 137 of a year

**X138** Temperatures on the day 138 of a year

**X139** Temperatures on the day 139 of a year

**X140** Temperatures on the day 140 of a year

**X141** Temperatures on the day 141 of a year

**X142** Temperatures on the day 142 of a year

**X143** Temperatures on the day 143 of a year

**X144** Temperatures on the day 144 of a year

**X145** Temperatures on the day 145 of a year

**X146** Temperatures on the day 146 of a year

**X147** Temperatures on the day 147 of a year

**X148** Temperatures on the day 148 of a year

**X149** Temperatures on the day 149 of a year

**X150** Temperatures on the day 150 of a year

**X151** Temperatures on the day 151 of a year

**X152** Temperatures on the day 152 of a year

**X153** Temperatures on the day 153 of a year

**X154** Temperatures on the day 154 of a year

**X155** Temperatures on the day 155 of a year

**X156** Temperatures on the day 156 of a year

**X157** Temperatures on the day 157 of a year

**X158** Temperatures on the day 158 of a year

**X159** Temperatures on the day 159 of a year

**X160** Temperatures on the day 160 of a year

**X161** Temperatures on the day 161 of a year

**X162** Temperatures on the day 162 of a year

**X163** Temperatures on the day 163 of a year

**X164** Temperatures on the day 164 of a year

**X165** Temperatures on the day 165 of a year

**X166** Temperatures on the day 166 of a year

**X167** Temperatures on the day 167 of a year

**X168** Temperatures on the day 168 of a year

**X169** Temperatures on the day 169 of a year

**X170** Temperatures on the day 170 of a year

**X171** Temperatures on the day 171 of a year

**X172** Temperatures on the day 172 of a year

**X173** Temperatures on the day 173 of a year

**X174** Temperatures on the day 174 of a year

**X175** Temperatures on the day 175 of a year

**X176** Temperatures on the day 176 of a year

**X177** Temperatures on the day 177 of a year

**X178** Temperatures on the day 178 of a year

**X179** Temperatures on the day 179 of a year

**X180** Temperatures on the day 180 of a year

**X181** Temperatures on the day 181 of a year

**X182** Temperatures on the day 182 of a year

**X183** Temperatures on the day 183 of a year

**X184** Temperatures on the day 184 of a year

**X185** Temperatures on the day 185 of a year

**X186** Temperatures on the day 186 of a year

**X187** Temperatures on the day 187 of a year

**X188** Temperatures on the day 188 of a year

**X189** Temperatures on the day 189 of a year

**X190** Temperatures on the day 190 of a year

**X191** Temperatures on the day 191 of a year

**X192** Temperatures on the day 192 of a year

**X193** Temperatures on the day 193 of a year

**X194** Temperatures on the day 194 of a year

**X195** Temperatures on the day 195 of a year

**X196** Temperatures on the day 196 of a year

**X197** Temperatures on the day 197 of a year

**X198** Temperatures on the day 198 of a year

**X199** Temperatures on the day 199 of a year

**X200** Temperatures on the day 200 of a year

**X201** Temperatures on the day 201 of a year

**X202** Temperatures on the day 202 of a year

**X203** Temperatures on the day 203 of a year

**X204** Temperatures on the day 204 of a year

**X205** Temperatures on the day 205 of a year

**X206** Temperatures on the day 206 of a year

**X207** Temperatures on the day 207 of a year

**X208** Temperatures on the day 208 of a year

**X209** Temperatures on the day 209 of a year

**X210** Temperatures on the day 210 of a year

**X211** Temperatures on the day 211 of a year

**X212** Temperatures on the day 212 of a year

**X213** Temperatures on the day 213 of a year

**X214** Temperatures on the day 214 of a year

**X215** Temperatures on the day 215 of a year

**X216** Temperatures on the day 216 of a year

**X217** Temperatures on the day 217 of a year

**X218** Temperatures on the day 218 of a year

**X219** Temperatures on the day 219 of a year

**X220** Temperatures on the day 220 of a year

**X221** Temperatures on the day 221 of a year

**X222** Temperatures on the day 222 of a year

**X223** Temperatures on the day 223 of a year

**X224** Temperatures on the day 224 of a year

**X225** Temperatures on the day 225 of a year

**X226** Temperatures on the day 226 of a year

**X227** Temperatures on the day 227 of a year

**X228** Temperatures on the day 228 of a year

**X229** Temperatures on the day 229 of a year

**X230** Temperatures on the day 230 of a year

**X231** Temperatures on the day 231 of a year

**X232** Temperatures on the day 232 of a year

**X233** Temperatures on the day 233 of a year

**X234** Temperatures on the day 234 of a year

**X235** Temperatures on the day 235 of a year

**X236** Temperatures on the day 236 of a year

**X237** Temperatures on the day 237 of a year

**X238** Temperatures on the day 238 of a year

**X239** Temperatures on the day 239 of a year

**X240** Temperatures on the day 240 of a year

**X241** Temperatures on the day 241 of a year

**X242** Temperatures on the day 242 of a year

**X243** Temperatures on the day 243 of a year

**X244** Temperatures on the day 244 of a year

**X245** Temperatures on the day 245 of a year

**X246** Temperatures on the day 246 of a year

**X247** Temperatures on the day 247 of a year

**X248** Temperatures on the day 248 of a year

**X249** Temperatures on the day 249 of a year

**X250** Temperatures on the day 250 of a year

**X251** Temperatures on the day 251 of a year

**X252** Temperatures on the day 252 of a year

**X253** Temperatures on the day 253 of a year

**X254** Temperatures on the day 254 of a year

**X255** Temperatures on the day 255 of a year

**X256** Temperatures on the day 256 of a year

**X257** Temperatures on the day 257 of a year

**X258** Temperatures on the day 258 of a year

**X259** Temperatures on the day 259 of a year

**X260** Temperatures on the day 260 of a year

**X261** Temperatures on the day 261 of a year

**X262** Temperatures on the day 262 of a year

**X263** Temperatures on the day 263 of a year

**X264** Temperatures on the day 264 of a year

**X265** Temperatures on the day 265 of a year

**X266** Temperatures on the day 266 of a year

**X267** Temperatures on the day 267 of a year

**X268** Temperatures on the day 268 of a year

**X269** Temperatures on the day 269 of a year

**X270** Temperatures on the day 270 of a year

**X271** Temperatures on the day 271 of a year

**X272** Temperatures on the day 272 of a year

**X273** Temperatures on the day 273 of a year

**X274** Temperatures on the day 274 of a year

**X275** Temperatures on the day 275 of a year

**X276** Temperatures on the day 276 of a year

**X277** Temperatures on the day 277 of a year

**X278** Temperatures on the day 278 of a year

**X279** Temperatures on the day 279 of a year

**X280** Temperatures on the day 280 of a year

**X281** Temperatures on the day 281 of a year

**X282** Temperatures on the day 282 of a year

**X283** Temperatures on the day 283 of a year

**X284** Temperatures on the day 284 of a year

**X285** Temperatures on the day 285 of a year

**X286** Temperatures on the day 286 of a year

**X287** Temperatures on the day 287 of a year

**X288** Temperatures on the day 288 of a year

**X289** Temperatures on the day 289 of a year

**X290** Temperatures on the day 290 of a year

**X291** Temperatures on the day 291 of a year

**X292** Temperatures on the day 292 of a year

**X293** Temperatures on the day 293 of a year

**X294** Temperatures on the day 294 of a year

**X295** Temperatures on the day 295 of a year

**X296** Temperatures on the day 296 of a year

**X297** Temperatures on the day 297 of a year

**X298** Temperatures on the day 298 of a year

**X299** Temperatures on the day 299 of a year

**X300** Temperatures on the day 300 of a year

**X301** Temperatures on the day 301 of a year

**X302** Temperatures on the day 302 of a year

**X303** Temperatures on the day 303 of a year

**X304** Temperatures on the day 304 of a year

**X305** Temperatures on the day 305 of a year

**X306** Temperatures on the day 306 of a year

**X307** Temperatures on the day 307 of a year

**X308** Temperatures on the day 308 of a year

**X309** Temperatures on the day 309 of a year

**X310** Temperatures on the day 310 of a year

**X311** Temperatures on the day 311 of a year

**X312** Temperatures on the day 312 of a year

**X313** Temperatures on the day 313 of a year

**X314** Temperatures on the day 314 of a year

**X315** Temperatures on the day 315 of a year

**X316** Temperatures on the day 316 of a year

**X317** Temperatures on the day 317 of a year

**X318** Temperatures on the day 318 of a year

**X319** Temperatures on the day 319 of a year

**X320** Temperatures on the day 320 of a year

**X321** Temperatures on the day 321 of a year

**X322** Temperatures on the day 322 of a year

**X323** Temperatures on the day 323 of a year

**X324** Temperatures on the day 324 of a year

**X325** Temperatures on the day 325 of a year

**X326** Temperatures on the day 326 of a year

**X327** Temperatures on the day 327 of a year

**X328** Temperatures on the day 328 of a year

**X329** Temperatures on the day 329 of a year

**X330** Temperatures on the day 330 of a year

**X331** Temperatures on the day 331 of a year

**X332** Temperatures on the day 332 of a year

**X333** Temperatures on the day 333 of a year

**X334** Temperatures on the day 334 of a year

**X335** Temperatures on the day 335 of a year

**X336** Temperatures on the day 336 of a year

**X337** Temperatures on the day 337 of a year

**X338** Temperatures on the day 338 of a year

**X339** Temperatures on the day 339 of a year

**X340** Temperatures on the day 340 of a year

**X341** Temperatures on the day 341 of a year

**X342** Temperatures on the day 342 of a year

**X343** Temperatures on the day 343 of a year

**X344** Temperatures on the day 344 of a year

**X345** Temperatures on the day 345 of a year

**X346** Temperatures on the day 346 of a year

**X347** Temperatures on the day 347 of a year

**X348** Temperatures on the day 348 of a year

**X349** Temperatures on the day 349 of a year

**X350** Temperatures on the day 350 of a year

**X351** Temperatures on the day 351 of a year

**X352** Temperatures on the day 352 of a year

**X353** Temperatures on the day 353 of a year

**X354** Temperatures on the day 354 of a year

**X355** Temperatures on the day 355 of a year

**X356** Temperatures on the day 356 of a year

**X357** Temperatures on the day 357 of a year

**X358** Temperatures on the day 358 of a year

**X359** Temperatures on the day 359 of a year

**X360** Temperatures on the day 360 of a year

**X361** Temperatures on the day 361 of a year

**X362** Temperatures on the day 362 of a year

**X363** Temperatures on the day 363 of a year

**X364** Temperatures on the day 364 of a year

**X365** Temperatures on the day 365 of a year

**X366** Temperatures on the day 366 of a year

**Source**

[http://climexp.knmi.nl/start.cgi](http://climexp.knmi.nl/start.cgi)

---

```
LJ_monthly_precipitation
```
*Monthly sums of precipitation for Ljubljana from 2018 - 1900. Tidy format.*

---

## Description

A dataset of monthly sums of precipitations in Ljubljana (Slovenia). The first row represents precipitation sum for January 1900.

## Usage

```
LJ_monthly_precipitation
```

## Format

A data frame with 1417 rows and 3 variables:

**Year** year

**Month** Month

**Precipitation** Sum of precipitation

## Source

<http://climexp.knmi.nl/start.cgi>

---

```
LJ_monthly_temperatures
```
*Monthly mean air temperatures for Ljubljana from 2015 - 1900*

---

## Description

A dataset of monthly mean air temperatures in Ljubljana (Slovenia). The first row represents temperatures in 2015. The first column represents mean January temperature, the second column represents mean February temperature. etc. Row names represent year.

## Usage

```
LJ_monthly_temperatures
```

**Format**

A data frame with 116 rows and 12 variables:

**Jan** Mean monthly air temperature for January from 1900 to 2015

**Feb** Mean monthly air temperature for February from 1900 to 2015

**Mar** Mean monthly air temperature for March from 1900 to 2015

**Apr** Mean monthly air temperature for April from 1900 to 2015

**May** Mean monthly air temperature for May from 1900 to 2015

**Jun** Mean monthly air temperature for June from 1900 to 2015

**Jul** Mean monthly air temperature for July from 1900 to 2015

**Aug** Mean monthly air temperature for August from 1900 to 2015

**Sep** Mean monthly air temperature for September from 1900 to 2015

**Oct** Mean monthly air temperature for October from 1900 to 2015

**Nov** Mean monthly air temperature for November from 1900 to 2015

**Dec** Mean monthly air temperature for December from 1900 to 2015

**Source**

<https://meteo.arso.gov.si/met/sl/archive/>

---

monthly_response                        *monthly_response*

---

**Description**

Function calculates all possible values of a selected statistical metric between one or more response variables and monthly sequences of environmental data. Calculations are based on moving window which slides through monthly environmental data. All calculated metrics are stored in a matrix. The location of stored calculated metric in the matrix is indicating a window width (row names) and a location in a matrix of monthly sequences of environmental data (column names).

**Usage**

```
monthly_response(
  response,
  env_data,
  method = "cor",
  metric = "r.squared",
  cor_method = "pearson",
  previous_year = FALSE,
  neurons = 1,
  lower_limit = 1,
  upper_limit = 12,
  fixed_width = 0,
```

```
      brnn_smooth = TRUE,
      remove_insignificant = TRUE,
      alpha = 0.05,
      row_names_subset = FALSE,
      reference_window = "start",
      aggregate_function = "mean",
      temporal_stability_check = "sequential",
      k = 2,
      k_running_window = 30,
      cross_validation_type = "blocked",
      subset_years = NULL,
      ylimits = NULL,
      seed = NULL,
      tidy_env_data = FALSE,
      boot = FALSE,
      boot_n = 1000,
      boot_ci_type = "norm",
      boot_conf_int = 0.95,
     month_interval = ifelse(c(previous_year == TRUE, previous_year == TRUE), c(-1, 12),
        c(1, 12)),
      dc_method = NULL,
      cor_na_use = "everything"
    )
```

## Arguments

| | |
|---|---|
| response | a data frame with tree-ring proxy variables as columns and (optional) years as row names. Row.names should be matched with those from a env_data data frame. If not, set row_names_subset = TRUE. |
| env_data | a data frame of monthly sequences of environmental data as columns and years as row names. Each row represents a year and each column represents a day of a year (or month). Row.names should be matched with those from a response data frame. If not, set row_names_subset = TRUE. Alternatively, env_data could be a tidy data with three columns, i.e. Year, DOY (Month) and third column representing values of mean temperatures, sum of precipitation etc. If tidy data is passed to the function, set the argument tidy_env_data to TRUE. |
| method | a character string specifying which method to use. Current possibilities are "cor" (default), "lm" and "brnn". |
| metric | a character string specifying which metric to use. Current possibilities are "r.squared" and "adj.r.squared". If method = "cor", metric is not relevant. |
| cor_method | a character string indicating which correlation coefficient is to be computed. One of "pearson" (default), "kendall", or "spearman". |
| previous_year | if set to TRUE, env_data and response variables will be rearranged in a way, that also previous year will be used for calculations of selected statistical metric. |
| neurons | positive integer that indicates the number of neurons used for brnn method |
| lower_limit | lower limit of window width (i.e. number of consecutive months to be used for calculations) |

| upper_limit | upper limit of window width (i.e. number of consecutive months to be used for calculations) |
|---|---|
| fixed_width | fixed width used for calculations (i.e. number of consecutive months to be used for calculations) |
| brnn_smooth | if set to TRUE, a smoothing algorithm is applied that removes unrealistic calculations which are a result of neural net failure. |
| remove_insignificant | |
| | if set to TRUE, removes all correlations bellow the significant threshold level, based on a selected alpha. For "lm" and "brnn" method, squared threshold is used, which corresponds to R squared statistics. |
| alpha | significance level used to remove insignificant calculations. |
| row_names_subset | |
| | if set to TRUE, row.names are used to subset env_data and response data frames. Only years from both data frames are kept. |
| reference_window | |
| | character string, the reference_window argument describes, how each calculation is referred. There are two different options: 'start' (default) and 'end'. If the reference_window argument is set to 'start', then each calculation is related to the starting month of window. If the reference_window argument is set to 'end', then each calculation is related to the ending day of window calculation. |
| aggregate_function | |
| | character string specifying how the monthly data should be aggregated. The default is 'mean', the two other options are 'median' and 'sum' |
| temporal_stability_check | |
| | character string, specifying, how temporal stability between the optimal selection and response variable(s) will be analysed. Current possibilities are "sequential", "progressive" and "running_window". Sequential check will split data into k splits and calculate selected metric for each split. Progressive check will split data into k splits, calculate metric for the first split and then progressively add 1 split at a time and calculate selected metric. For running window, select the length of running window with the k_running_window argument. |
| k | integer, number of breaks (splits) for temporal stability and cross validation analysis. |
| k_running_window | |
| | the length of running window for temporal stability check. Applicable only if temporal_stability argument is set to running window. |
| cross_validation_type | |
| | character string, specifying, how to perform cross validation between the optimal selection and response variables. If the argument is set to "blocked", years will not be shuffled. If the argument is set to "randomized", years will be shuffled. |
| subset_years | a subset of years to be analyzed. Should be given in the form of subset_years = c(1980, 2005) |
| ylimits | limit of the y axes for plot_extreme. It should be given in the form of: ylimits = c(0,1) |

| | |
|---|---|
| seed | optional seed argument for reproducible results |
| tidy_env_data | if set to TRUE, env_data should be inserted as a data frame with three columns: "Year", "Month", "Precipitation/Temperature/etc." |
| boot | logical, if TRUE, bootstrap procedure will be used to calculate estimates correlation coefficients, R squared or adjusted R squared metrices |
| boot_n | The number of bootstrap replicates |
| boot_ci_type | A character string representing the type of bootstrap intervals required. The value should be any subset of the values c("norm","basic", "stud", "perc", "bca"). |
| boot_conf_int | A scalar or vector containing the confidence level(s) of the required interval(s) |
| month_interval | a vector of two values: lower and upper time interval of months that will be used to calculate statistical metrics. Negative values indicate previous growing season months. This argument overwrites the calculation limits defined by lower_limit and upper_limit arguments. |
| dc_method | a character string to determine the method to detrend climate data. Possible values are "none" (default) and "SLD" which refers to Simple Linear Detrending |
| cor_na_use | an optional character string giving a method for computing covariances in the presence of missing values for correlation coefficients. This must be (an abbreviation of) one of the strings "everything" (default), "all.obs", "complete.obs", "na.or.complete", or "pairwise.complete.obs". See also the documentation for the base cor() function. |

**Value**

a list with 17 elements:

1. $calculations - a matrix with calculated metrics
2. $method - the character string of a method
3. $metric - the character string indicating the metric used for calculations
4. $analysed_period - the character string specifying the analysed period based on the information from row names. If there are no row names, this argument is given as NA
5. $optimized_return - data frame with two columns, response variable and aggregated (averaged) monthly data that return the optimal results. This data.frame could be directly used to calibrate a model for climate reconstruction
6. $optimized_return_all - a data frame with aggregated monthly data, that returned the optimal result for the entire env_data (and not only subset of analysed years)
7. $transfer_function - a ggplot object: scatter plot of optimized return and a transfer line of the selected method
8. $temporal_stability - a data frame with calculations of selected metric for different temporal subsets
9. $cross_validation - a data frame with cross validation results
10. $plot_heatmap - ggplot2 object: a heatmap of calculated metrics
11. $plot_extreme - ggplot2 object: line or bar plot of a row with the highest value in a matrix of calculated metrics

12. $type - the character string describing type of analysis: daily or monthly

13. $reference_window - character string, which reference window was used for calculations

14. $boot_lower - matrix with lower limit of confidence intervals of bootstrap calculations

15. $boot_upper - matrix with upper limit of confidence intervals of bootstrap calculations

16. $aggregated_climate - matrix with all aggregated climate series

**Examples**

```
# The examples below are enclosed within donttest{} to minimize the execution
# time during R package checks.

# Load the dendroTools R package
library(dendroTools)

# Load data used for examples
data(data_MVA)
data(data_TRW)
data(data_TRW_1)
data(example_proxies_individual)
data(example_proxies_1)
data(LJ_monthly_temperatures)
data(LJ_monthly_precipitation)

# 1 Example with tidy precipitation data
example_tidy_data <- monthly_response(response = data_MVA,
    lower_limit = 1, upper = 24,
    env_data = LJ_monthly_precipitation, fixed_width = 0,
    method = "cor", row_names_subset = TRUE,
    remove_insignificant = TRUE, previous_year = FALSE,
    reference_window = "end",
    alpha = 0.05, aggregate_function = 'sum', boot = FALSE,
    tidy_env_data = TRUE, boot_n = 100, month_interval = c(-5, 10))

# summary(example_tidy_data)
# plot(example_tidy_data, type = 1)
# plot(example_tidy_data, type = 2)

# 2 Example with split data for early and late
example_MVA_early <- monthly_response(response = data_MVA,
    env_data = LJ_monthly_temperatures,
    method = "cor", row_names_subset = TRUE, previous_year = TRUE,
    remove_insignificant = TRUE, alpha = 0.05,
    subset_years = c(1940, 1980), aggregate_function = 'mean')

example_MVA_late <- monthly_response(response = data_MVA,
    env_data = LJ_monthly_temperatures,
    method = "cor", row_names_subset = TRUE, alpha = 0.05,
    previous_year = TRUE, remove_insignificant = TRUE,
    subset_years = c(1981, 2010), aggregate_function = 'mean')
```

```
# summary(example_MVA_late)
# plot(example_MVA_early, type = 1)
# plot(example_MVA_late, type = 1)
# plot(example_MVA_early, type = 2)
# plot(example_MVA_late, type = 2)

# 3 Example negative correlations
example_neg_cor <- monthly_response(response = data_TRW_1, alpha = 0.05,
   env_data = LJ_monthly_temperatures,
   method = "cor", row_names_subset = TRUE,
   remove_insignificant = TRUE, boot = FALSE)

# summary(example_neg_cor)
# plot(example_neg_cor, type = 1)
# plot(example_neg_cor, type = 2)
# example_neg_cor$temporal_stability

# 4 Example of multiproxy analysis
# summary(example_proxies_1)
# cor(example_proxies_1)

example_multiproxy <- monthly_response(response = example_proxies_1,
   env_data = LJ_monthly_temperatures,
   method = "lm", metric = "adj.r.squared",
   row_names_subset = TRUE, previous_year = FALSE,
   remove_insignificant = TRUE, alpha = 0.05)

# summary(example_multiproxy)
# plot(example_multiproxy, type = 1)

# 5 Example to test the temporal stability
example_MVA_ts <- monthly_response(response = data_MVA,
   env_data = LJ_monthly_temperatures,
   method = "lm", metric = "adj.r.squared", row_names_subset = TRUE,
   remove_insignificant = TRUE, alpha = 0.05,
   temporal_stability_check = "running_window", k_running_window = 10)

# summary(example_MVA_ts)
# example_MVA_ts$temporal_stability
```

---

monthly_response_seascorr

*monthly_response_seascorr*

---

### Description

Function calculates all possible partial correlation coefficients between tree-ring chronology and monthly environmental (usually climate) data. All calculated (partial) correlation coefficients are

stored in a matrix. The location of stored correlation in the matrix is indicating a window width (row
names) and a location in a matrix of monthly sequences of environmental data (column names).

**Usage**

```
monthly_response_seascorr(
  response,
  env_data_primary,
  env_data_control,
  previous_year = FALSE,
  pcor_method = "pearson",
  remove_insignificant = TRUE,
  lower_limit = 1,
  upper_limit = 12,
  fixed_width = 0,
  alpha = 0.05,
  row_names_subset = FALSE,
  reference_window = "start",
  aggregate_function_env_data_primary = "mean",
  aggregate_function_env_data_control = "mean",
  temporal_stability_check = "sequential",
  k = 2,
  k_running_window = 30,
  subset_years = NULL,
  ylimits = NULL,
  seed = NULL,
  tidy_env_data_primary = FALSE,
  tidy_env_data_control = FALSE,
  boot = FALSE,
  boot_n = 1000,
  boot_ci_type = "norm",
  boot_conf_int = 0.95,
 month_interval = ifelse(c(previous_year == TRUE, previous_year == TRUE), c(-1, 12),
    c(1, 12)),
  dc_method = NULL,
  pcor_na_use = "pairwise.complete"
)
```

**Arguments**

response         a data frame with tree-ring proxy variable and (optional) years as row names.
                 Row.names should be matched with those from env_data_primary and env_data_control
                 data frame. If not, set the row_names_subset argument to TRUE.

env_data_primary

                 primary data frame of monthly sequences of environmental data as columns and
                 years as row names. Each row represents a year and each column represents
                 a day of a year. Row.names should be matched with those from the response
                 data frame. If not, set the argument row_names_subset to TRUE. Alternatively,
                 env_data_primary could be a tidy data with three columns, i.e. Year, Month and

third column representing values of mean temperatures, sum of precipitation etc.
If tidy data is passed to the function, set the argument tidy_env_data_primary to
TRUE.

env_data_control

a data frame of monthly sequences of environmental data as columns and years
as row names. This data is used as control for calculations of partial corre-
lation coefficients. Each row represents a year and each column represents a
day of a year. Row.names should be matched with those from the response
data frame. If not, set the row_names_subset argument to TRUE. Alternatively,
env_data_control could be a tidy data with three columns, i.e. Year, Month and
third column representing values of mean temperatures, sum of precipitation etc.
If tidy data is passed to the function, set the argument tidy_env_data_control to
TRUE.

previous_year      if set to TRUE, env_data_primary, env_data_control and response variables will
be rearranged in a way, that also previous year will be used for calculations of
selected statistical metric.

pcor_method        a character string indicating which partial correlation coefficient is to be com-
puted. One of "pearson" (default), "kendall", or "spearman", can be abbreviated.

remove_insignificant

if set to TRUE, removes all correlations bellow the significant threshold level,
based on a selected alpha.

lower_limit        lower limit of window width (i.e. number of consecutive months to be used for
calculations)

upper_limit        upper limit of window width (i.e. number of consecutive months to be used for
calculations)

fixed_width        fixed width used for calculations (i.e. number of consecutive months to be used
for calculations)

alpha              significance level used to remove insignificant calculations.

row_names_subset

if set to TRUE, row.names are used to subset env_data_primary, env_data_control
and response data frames. Only years from all three data frames are kept.

reference_window

character string, the reference_window argument describes, how each calcula-
tion is referred. There are two different options: 'start' (default) and 'end'. If the
reference_window argument is set to 'start', then each calculation is related to
the starting month of window. If the reference_window argument is set to 'end',
then each calculation is related to the ending day of window calculation.

aggregate_function_env_data_primary

character string specifying how the monthly data from env_data_primary should
be aggregated. The default is 'mean', the two other options are 'median' and
'sum'

aggregate_function_env_data_control

character string specifying how the monthly data from env_data_control should
be aggregated. The default is 'mean', the two other options are 'median' and
'sum'

temporal_stability_check

        character string, specifying, how temporal stability between the optimal selection and response variable(s) will be analysed. Current possibilities are "sequential", "progressive" and "running_window". Sequential check will split data into k splits and calculate selected metric for each split. Progressive check will split data into k splits, calculate metric for the first split and then progressively add 1 split at a time and calculate selected metric. For running window, select the length of running window with the k_running_window argument.

k                            integer, number of breaks (splits) for temporal stability

k_running_window

        the length of running window for temporal stability check. Applicable only if temporal_stability argument is set to running window.

subset_years                 a subset of years to be analyzed. Should be given in the form of subset_years = c(1980, 2005)

ylimits                      limit of the y axes for plot_extreme. It should be given in the form of: ylimits = c(0,1)

seed                         optional seed argument for reproducible results

tidy_env_data_primary

        if set to TRUE, env_data_primary should be inserted as a data frame with three columns: "Year", "Month", "Precipitation/Temperature/etc."

tidy_env_data_control

        if set to TRUE, env_data_control should be inserted as a data frame with three columns: "Year", "Month", "Precipitation/Temperature/etc."

boot                         logical, if TRUE, bootstrap procedure will be used to calculate partial correlation coefficients

boot_n                       The number of bootstrap replicates

boot_ci_type                 A character string representing the type of bootstrap intervals required. The value should be any subset of the values c("norm","basic", "stud", "perc", "bca").

boot_conf_int                A scalar or vector containing the confidence level(s) of the required interval(s)

month_interval               a vector of two values: lower and upper time interval of months that will be used to calculate statistical metrics. Negative values indicate previous growing season months. This argument overwrites the calculation limits defined by lower_limit and upper_limit arguments.

dc_method                    a character string to determine the method to detrend climate data. Possible values are "none" (default) and "SLD" which refers to Simple Linear Detrending

pcor_na_use                  an optional character string giving a method for computing covariances in the presence of missing values for partial correlation coefficients. This must be (an abbreviation of) one of the strings "all.obs", "everything", "complete.obs", "na.or.complete", or "pairwise.complete.obs" (default). See also the documentation for the base partial.r in psych R package

**Value**

a list with 15 elements:

    1. $calculations - a matrix with calculated metrics

2. $method - the character string of a method

3. $metric - the character string indicating the metric used for calculations

4. $analysed_period - the character string specifying the analysed period based on the information from row names. If there are no row names, this argument is given as NA

5. $optimized_return - data frame with two columns, response variable and aggregated (averaged) monthly data that return the optimal results. This data.frame could be directly used to calibrate a model for climate reconstruction

6. $optimized_return_all - a data frame with aggregated monthly data, that returned the optimal result for the entire env_data_primary (and not only subset of analysed years)

7. $transfer_function - a ggplot object: scatter plot of optimized return and a transfer line of the selected method

8. $temporal_stability - a data frame with calculations of selected metric for different temporal subsets

9. $cross_validation - not available for partial correlation method

10. $plot_heatmap - ggplot2 object: a heatmap of calculated metrics

11. $plot_extreme - ggplot2 object: line plot of a row with the highest value in a matrix of calculated metrics

12. $type - the character string describing type of analysis: monthly or monthly

13. $reference_window - character string, which reference window was used for calculations

14. $aggregated_climate_primary - matrix with all aggregated climate series of primary data

15. $aggregated_climate_control - matrix with all aggregated climate series of control data

## Examples

```
# Load the dendroTools R package
library(dendroTools)

# Load data
data(data_MVA)
data(data_TRW)
data(data_TRW_1)
data(example_proxies_individual)
data(example_proxies_1)
data(LJ_monthly_temperatures)
data(LJ_monthly_precipitation)

# 1 Basic example
example_basic <- monthly_response_seascorr(response = data_MVA,
  fixed_width = 11,
  env_data_primary = LJ_monthly_temperatures,
  env_data_control = LJ_monthly_precipitation,
  row_names_subset = TRUE,
  remove_insignificant = TRUE,
  reference_window = "start",
  aggregate_function_env_data_primary = 'median',
  aggregate_function_env_data_control = 'median',
  alpha = 0.05, pcor_method = "spearman",
```

```
    tidy_env_data_primary = FALSE,
    tidy_env_data_control = TRUE,
    previous_year = TRUE)

# summary(example_basic)
# plot(example_basic, type = 1)
# plot(example_basic, type = 2)
# example_basic$optimized_return
# example_basic$optimized_return_all
# example_basic$temporal_stability

# 2 Extended example
example_extended <- monthly_response_seascorr(response = data_MVA,
    env_data_primary = LJ_monthly_temperatures,
    env_data_control = LJ_monthly_precipitation,
    row_names_subset = TRUE,
    remove_insignificant = TRUE,
    aggregate_function_env_data_primary = 'mean',
    aggregate_function_env_data_control = 'mean',
    alpha = 0.05,
    reference_window = "end",
    tidy_env_data_primary = FALSE,
    tidy_env_data_control = TRUE)

# summary(example_extended)
# plot(example_extended, type = 1)
# plot(example_extended, type = 2)
# example_extended$optimized_return
# example_extended$optimized_return_all
```

---

swit272                        *Standardised tree-ring width chronology swit272, Larix decidua Mill.*

---

#### Description

A TRW chronology swit272 Investigators: Bigler, C.; Claluna, A. Site_Name: Sils-Maria GR Blais dal Fo Location: Switzerland Northernmost_Latitude: 46.4333 Southernmost_Latitude: 46.4333 Easternmost_Longitude: 9.7833 Westernmost_Longitude: 9.7833 Elevation: 2100

#### Usage

```
swit272
```

#### Format

A data frame with 273 rows and 1 variable:

**TRWi** Standardised tree-ring width chronology

**Source**

<https://www.ncei.noaa.gov/access/paleo-search/study/14108>

---

swit272_daily_precipitation

*Daily precipitation for swit272 chronology*

---

**Description**

Sum of daily precipitation in millimeters for the period 1950 - 2019. This gridded E-OBS data on 0.1° regular grid, version 20e. Extracted data is for the grid point with lon = 9.75 and lat = 46.45.

**Usage**

```
swit272_daily_precipitation
```

**Format**

A data frame with 25414 rows and 2 variables:

**date** character string describing date

**p_sum** mean temperature

**Details**

We acknowledge the E-OBS dataset from the EU-FP6 project UERRA (http://www.uerra.eu) and the Copernicus Climate Change Service, and the data providers in the ECA&D project (https://www.ecad.eu). Cornes, R., G. van der Schrier, E.J.M. van den Besselaar, and P.D. Jones. 2018: An Ensemble Version of the E-OBS Temperature and Precipitation Datasets, J. Geophys. Res. Atmos., 123. doi:10.1029/2017JD028200

**Source**

<https://www.ecad.eu/download/ensembles/download.php>

---

`swit272_daily_temperatures`

*Daily temperatures for swit272 chronology*

---

### Description

Mean daily temperature in Celsius for the period 1950 - 2019. This gridded E-OBS data on 0.1°
regular grid, version 20e. Extracted data is for the grid point with lon = 9.75 and lat = 46.45.

### Usage

```
swit272_daily_temperatures
```

### Format

A data frame with 25414 rows and 2 variables:

**date**  character string describing date

**t_avg**  mean temperature

### Details

We acknowledge the E-OBS dataset from the EU-FP6 project UERRA (http://www.uerra.eu) and
the Copernicus Climate Change Service, and the data providers in the ECA&D project (https://www.ecad.eu).
Cornes, R., G. van der Schrier, E.J.M. van den Besselaar, and P.D. Jones. 2018: An Ensemble
Version of the E-OBS Temperature and Precipitation Datasets, J. Geophys. Res. Atmos., 123.
doi:10.1029/2017JD028200

### Source

https://www.ecad.eu/download/ensembles/download.php

---

`years_to_rownames`      *Function returns a data frame with row names as years*

---

### Description

Function returns a data frame with row names as years

### Usage

```
years_to_rownames(data, column_year)
```

### Arguments

| | |
|---|---|
| `data` | a data frame to be manipulated |
| `column_year` | string specifying a column with years |

## Value

a data frame with years as row names

## Examples

```
data <- data.frame(years = seq(1950, 2015), observations = rnorm(66))
new_data <- years_to_rownames(data = data, column_year = "years")

data <- data.frame(observations1 = rnorm(66), years = seq(1950, 2015),
observations2 = rnorm(66), observations3 = rnorm(66))
new_data <- years_to_rownames(data = data, column_year = "years")
```

# Index