Package ‘ArchaeoPhases’

June 21, 2022

Type Package

Title Post-Processing of the Markov Chain Simulated by 'ChronoModel', 'Oxcal' or 'BCal'

Version 1.8

Date 2022-06-10

Author Anne Philippe [aut, cre], Marie-Anne Vibet [aut], Thomas S. Dye [ctb]

Maintainer Anne Philippe <anne.philippe@univ-nantes.fr>

Description Provides a list of functions for the statistical analysis of archaeological dates and groups of dates. It is based on the post-processing of the Markov Chains whose stationary distribution is the posterior distribution of a series of dates. Such output can be simulated by different applications as for instance 'ChronoModel' (see <https://chronomodel.com/>), 'Oxcal' (see <https://c14.arch.ox.ac.uk/oxcal.html>) or 'BCal' (see <https://bcal.shef.ac.uk/>). The only requirement is to have a csv file containing a sample from the posterior distribution. Note that this package interacts with data available through the ‘ArchaeoPhases.dataset’ package which is available in a separate repository. The size of the 'ArchaeoPhases.dataset' package is approximately 4 MB.

License GPL-3

Depends R (>= 3.5.0), coda, hdrcde

Imports stats, utils, graphics, grDevices, shiny, readr, toOrdinal, ggplot2, ggalt, reshape2, dplyr, digest, gplots, magrittr, ggraph, gtools

Suggests knitr, rmarkdown, testthat (>= 2.1.0), ArchaeoPhases.dataset (>= 0.1.0)

VignetteBuilder knitr

BugReports https://github.com/ArchaeoStat/ArchaeoPhases/issues

RoxygenNote 7.2.0

NeedsCompilation no

Encoding UTF-8
LazyData true
Repository CRAN
Date/Publication 2022-06-21 18:20:02 UTC

R topics documented:

AgeDepth ................................................................. 4
allen_analyze ......................................................... 5
allen_basic_relation_set ......................................... 6
allen_basic_relation_strings ................................. 7
allen_composition ................................................... 7
allen_concurrent_relations ................................. 8
allen_create_concurrent_vector ............................ 8
allen_create_distinct_endpoint_vector .......................... 9
allen_create_result_vector ..................................... 9
allen_ensure_set_vector ........................................... 10
allen_illustrate ......................................................... 10
allen_is_result_vector ............................................. 12
allen_is_set_string .................................................. 12
allen_is_set_vector .................................................. 13
allen_joint_concurrency ......................................... 13
allen_lattice_x ......................................................... 14
allen_lattice_y ......................................................... 14
allen_observe .......................................................... 15
allen_observe_frequency ......................................... 16
allen_plot .............................................................. 17
allen_proportion_result .......................................... 18
allen_relate_intervals ............................................. 19
allen_relation .......................................................... 20
allen_set_to_vector ................................................ 20
allen六_value_set .................................................. 21
allen_string_to_set .................................................. 21
allen_string_to_vector .............................................. 22
allen_union .............................................................. 22
allen_update_result ................................................. 23
analyze_allen_relations ......................................... 23
app_ArchaeoPhases .................................................. 24
ArchaeoPhases ......................................................... 24
coda.mcmc .............................................................. 24
composition_lookup_table ....................................... 25
CreateMinMaxGroup .................................................. 26
CredibleInterval ....................................................... 27
credible_interval ...................................................... 27
DatesHiatus ............................................................. 28
dates_hiatus ........................................................... 29
estimate_range ......................................................... 30
Events ................................................................. 32
R topics documented:

- illustrate_allen_relations ........................................ 32
- ImportCSV .......................................................... 33
- ImportCSV.BCal ...................................................... 35
- is.url ............................................................... 36
- MarginalPlot .......................................................... 37
- MarginalProba ......................................................... 38
- MarginalStatistics .................................................... 39
- marginal_plot .......................................................... 40
- marginal_statistics .................................................... 43
- MultiCredibleInterval ............................................. 44
- MultiDatesPlot ....................................................... 45
- MultiHPD .............................................................. 47
- MultiMarginalPlot .................................................... 48
- MultiPhasePlot ....................................................... 50
- MultiPhasesGap ...................................................... 51
- MultiPhasesTransition ........................................... 52
- MultiPhaseTimeRange ............................................. 54
- MultiSuccessionPlot ............................................. 55
- multi_credible_interval ............................................ 56
- multi_dates_plot ..................................................... 57
- multi_hpd ............................................................ 60
- multi_marginal_plot ................................................ 61
- multi_marginal_statistics ........................................ 63
- new_archaeophases_mcmc .......................................... 64
- new_archaeophases_plot .......................................... 65
- OccurrencePlot ...................................................... 66
- occurrence_plot ..................................................... 68
- original_file ........................................................ 71
- original_file.archaeophases_mcmc ................................ 71
- original_file.archaeophases_plot ................................ 72
- oxc ................................................................. 73
- PhaseDurationPlot ................................................ 73
- PhasePlot ........................................................... 74
- Phases ............................................................... 76
- PhasesGap ............................................................ 76
- PhaseStatistics ...................................................... 77
- PhasesTransition ................................................... 78
- phases_gap ........................................................... 79
- PhaseTimeRange .................................................... 80
- phase_statistics ...................................................... 81
- plot.archaeophases_plot .......................................... 82
- read_bcal ............................................................ 82
- read_chronomodel .................................................. 83
- read_oxcal ........................................................... 84
- reproduce ............................................................ 85
- reproduce.archaeophases_mcmc .................................. 87
- reproduce.archaeophases_plot ................................... 88
- SuccessionPlot ....................................................... 89
**AgeDepth**

age depth curve Compute the age-depth curve from the output of mcmc algorithm of ages and the known depth of each dated samples.

**Description**

age depth curve Compute the age-depth curve from the output of mcmc algorithm of ages and the known depth of each dated samples.

**Usage**

```
AgeDepth(
  data, 
  depth, 
  new.depth = NULL, 
  max.iter = nrow(data), 
  sampling = FALSE 
)
```

**Arguments**

- `data` : Data frame containing the output of the MCMC algorithm.
- `depth` : Vector of the depths of the dated samples.
- `new.depth` : Vector of the undated depths whose the age will be predict. By default `new.depth=NULL`.
- `max.iter` : a non-negative integer giving the limit number of MCMC iterations. By default `max.iter=nrow(data)`.
- `sampling` : should sampling be random. By default `sampling = FALSE`.

**Details**

Estimate the Age-Depth relationship from the MCMC output of a Bayesian chronological model and the depth of each dated sample. We assume it exists a function \( f \) relating the age and the depth \( \text{age} = f(\text{depth}) \). We estimate the function using local regression (also called local polynomial regression): \( f = \text{loess(} \text{age depth}) \). This estimated function \( f \) depends on the unknown dates. However, from the posterior distribution of the age/date sequence, we can evaluate the posterior distribution of the age function for each desired depth.
Value

A list containing two matrix. data.stat summarises the MCMC output from the L1 Bayes estimate and credible interval at significance level 68% and 95%. age.depth provide the L1 Bayes estimate, credible interval at significance level 68% and 95% on the age at depth depth and new.depth

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr>

References


Examples

data = matrix(rnorm(6000,(1:6)^2), ncol=6, byrow = TRUE )
depth= 1:6
AgeDepth(data,depth)
AgeDepth(data,depth, 1.5:6)

allen_analyze

Analyze composite relations

Description

Visualize composite Allen relations with a Nokel lattice.

Usage

allen_analyze(relation_1, relation_2, title, ...)

Arguments

relation_1 A string denoting an Allen set.
relation_2 A string denoting an Allen set.
title A string displayed as the title of the Nokel lattice.
... Named arguments to be passed on to allen_plot().
Allen basic relation set

Value

A layout_tbl_graph object.

Author(s)

Thomas S. Dye

Examples

# Plot to the R graphics device
# allen_analyze("mDfO", "MdfO", "Composite reticulation relation")

Description

A vector of one-letter codes for the thirteen basic Allen relations. The codes were proposed by Thomas Alspaugh.

Usage

allen_basic_relation_set()

Value

A vector of thirteen one-letter codes

Author(s)

Thomas S. Dye
**Allen basic relation strings**

*Allen basic relation set as strings*

**Description**

String descriptors of the Allen basic relations.

**Usage**

```r
callen_basic_relation_strings()
```

**Value**

A vector of thirteen strings

**Author(s)**

Thomas S. Dye

---

**Allen composition**

*Composition of two Allen relations*

**Description**

Composition of two Allen relations

**Usage**

```r
callen_composition(first, second)
```

**Arguments**

- `first`: the first Allen relation
- `second`: the second Allen relation

**Value**

A dataframe for input to `allen_plot`

**Author(s)**

Thomas S. Dye
Description

A vector of nine one-letter codes for the Allen concurrent relations. The codes were proposed by Thomas Alspaugh.

Usage

allen_concurrent_relations()

Value

A vector of nine one-letter codes.

Description

Create a result vector where concurrent relations are set to 1 and non-concurrent relations are set to 0.

Usage

allen_create_concurrent_vector()

Value

A result vector

Author(s)

Thomas S. Dye
**Description**

The six relations with distinct endpoints are commonly observed when comparing indefinite intervals, such as those returned by a Bayesian calibration.

**Usage**

```r
allen_create_distinct_endpoint_vector()
```

**Value**

A named vector with distinct endpoint relations set to 1 and all others set to 0.

**Author(s)**

Thomas S. Dye

---

**Description**

Create a named result vector initialized to zero by default or to some other value.

**Usage**

```r
allen_create_result_vector(initial_value = 0)
```

**Arguments**

- `initial_value` A value used to initialize the vector. Typically 0 (default) or 1.

**Value**

An initialized result vector.

**Author(s)**

Thomas S. Dye
**Allen_set_vector**

*Ensure an Allen set is represented as a vector of single character strings*

**Description**

Expects a string, set vector, or result vector and will stop with an error if something else is encountered.

**Usage**

```r
allen_ensure_set_vector(obj)
```

**Arguments**

- **obj**
  
  An Allen set represented as a string, a set vector, or a result vector.

**Value**

An Allen set represented as a set vector.

**Author(s)**

Thomas S. Dye

**Allen_illustrate**

*Illustrate basic and composite relations*

**Description**

Illustrate basic and composite Allen relations for several chronological model domains with a Nokel lattice. Chronological model domains include stratigraphy and branching, transformative, and reticulate processes of artifact change.

**Usage**

```r
allen_illustrate(relations = "basic", ...)
```
Arguments

relations One of:

basic the 13 basic Allen relations (default);
concurrent concurrent relations;
distinct relations with distinct endpoints;
stratigraphic basic relations established by an observation of superposition;
branching basic branching relations;
transformation basic relations of transformation;
reticulation basic relations of reticulation;
sequence composite relations in a stratigraphic sequence;
branch composite relations of branching;
transform composite relations of transformation; or
reticulate composite relations of reticulation.

... Named arguments to be passed on to `allen_plot()`.

Value

A layout_tbl_graph object.

Author(s)

Thomas S. Dye

References


Examples

```r
# Plot the basic Allen relations to the R graphics device
# allen_illustrate()
```
allen_is_result_vector

Test whether an object is a result vector

Description
Checks for vector, names, and class

Usage
allen_is_result_vector(obj)

Arguments
obj An object to test

Value
Boolean, TRUE if obj is a result vector, FALSE otherwise.

allen_is_set_string Test if an object is a set string

Description
Checks for mode 'character', length of 1, and nchar <= 13

Usage
allen_is_set_string(obj)

Arguments
obj An object to test

Value
Boolean, TRUE if obj is a set string, FALSE otherwise.

Author(s)
Thomas S. Dye
Allen is Set Vector

Test if an object is a set vector

Description

Checks for mode 'character', length less than 13. Note: this predicate is a (very) partial implementation.

Usage

allen_is_set_vector(obj)

Arguments

obj An object to test

Value

Boolean, TRUE if obj is a set vector, FALSE otherwise.

Author(s)

Thomas S. Dye

Joint Concurrency

Joint concurrence of two or more observed intervals

Description

Estimate the age of an undated context based on the known depositional history of associated artifacts.

Usage

allen_joint_concurrency(mcmc, chains, ...)

Arguments

mcmc Dataframe or archaeophases_mcmc object with the MCMC output from a Bayesian calibration.
chains a list of vectors of names or indexes of columns in mcmc.
... Arguments to multi_marginal_statistics.
**Value**

foo bar

**Author(s)**

Thomas S. Dye

---

**Description**

A vector of arbitrary coordinates for lattice node placement

**Usage**

`allen_lattice_x()`

**Value**

A vector of integers

**Author(s)**

Thomas S. Dye

---

**Description**

A vector of arbitrary coordinates for lattice node placement

**Usage**

`allen_lattice_y()`

**Value**

A vector of integers

**Author(s)**

Thomas S. Dye
Observe the relation between two phases, each representing an interval of time

Description

Plots an empirical Nökel lattice.

Usage

allen_observe(data, chains, ...)

Arguments

data  Data frame or archaeophasess_mcmc object containing the output of the MCMC algorithm.
chains a list of lists, each with two elements, each of which is a vector of chains.
...  Named arguments to be passed on to allen_plot().

Value

An object of class archaeophasess_plot.

Author(s)

Thomas S. Dye

Examples

## Not run:
# Dates associated with bead BE3 Amber
be3.amber <- c("UB-4836 (WG27)", "UB-5208 (ApD107)", "UB-4965 (ApD117)", "UB-4735 (Ber022)", "UB-4739 (Ber134/1)", "UB-4728 (MH064)", "UB-4729 (MH068)", "UB-4732 (MH094)", "UB-4733 (MH095)", "UB-4734 (MH105c)", "UB-4984 (Lec018)", "UB-4709 (EH014)", "UB-4707 (EH079)", "UB-4708 (EH083)", "UB-6033 (Whes113)", "UB-4706 (Whes118)", "UB-4705 (Whes123)", "UB-6040 (CasD053)", "UB-6037 (CasD134)", "UB-6472 (BuD222)", "UB-6473 (BuD250)", "UB-6476 (BuD339)", "UB-4963 (SPTip208)", "UB-4890 (MelSG075)", "UB-4887 (MelSG082)", "UB-4888 (MelSG089)", "MaDE1 & E2", "UB-4552 (MaDE3)", "UB-4975 (AstCl112)", "UB-4835 (ApD134)", "SUERC-39108 ERLK G322", "SUERC-39109 ERL G362", "SUERC-39112 ERL G405", "SUERC-51560 ERL G038", "SUERC-39091 (ERL G003)", "SUERC-39092 (ERL G005)", "SUERC-39113 (ERL G417)", "SUERC-51549 (ERL G195)", "SUERC-51552 (ERL G107)", "SUERC-51550 (ERL G254)")

# Dates associated with bead BE1 Dghnt
be1.dghnt <- c("UB-4503 (Lec148)", "UB-4506 (Lec172/2)", "UB-6038 (CasD053)", "UB-6037 (CasD134)", "SUERC-39091 (ERL G003)", "SUERC-39092 (ERL G005)", "SUERC-39113 (ERL G417)", "SUERC-51549 (ERL G195)", "SUERC-51552 (ERL G107)", "SUERC-51550 (ERL G254)")


```r
# Construct a list of lists
chains <- list(list(be3.amber, be1.dghnt), list(be1.dghnt, be3.amber))

# Read the calibration MCMC output
oxc <- read_oxcal("https://tsdye.online/AP/beads-mcmc.csv", quiet = 'yes')

# Plot to the R graphics device
# allen_observe(data = oxc, chains = chains)
```

---

### `allen_observe_frequency`

**Observed frequency of an Allen set**

**Description**

Create a matrix of observed frequencies of a given Allen set among two or more groups of chains from the MCMC output of a Bayesian calibration. The groups are permuted to form the matrix.

**Usage**

```r
allen_observe_frequency(mcmc, chains, allen_set)
```

**Arguments**

- **mcmc**
  - Dataframe or archaeophases_mcmc object with the MCMC output from a Bayesian calibration.
- **chains**
  - A list of vectors of names or indexes of columns in mcmc.
- **allen_set**
  - A string representation of an Allen set.

**Value**

A matrix of observed frequencies.

**Author(s)**

Thomas S. Dye
Examples

```r
## Not run:
# Dates associated with bead BE3 Amber
               "UB-4735 (Ber022)", "UB-4739 (Ber134/1)", "UB-4728 (MH064)",
               "UB-4729 (MH068)", "UB-4732 (MH094)", "UB-4733 (MH095)", "UB-4734 (MH105c)",
               "UB-4984 (Lec018)", "UB-4709 (EH014)", "UB-4707 (EH079)", "UB-4708 (EH083)",
               "UB-6033 (WHes113)", "UB-4706 (WHes118)", "UB-4705 (WHes123)",
               "UB-6040 (CasD053)", "UB-6037 (CasD134)", "UB-6472 (BuD222)",
               "UB-6473 (BuD258)", "UB-6476 (BuD339)", "UB-4963 (SPTip208)",
               "UB-4890 (MelSG975)", "UB-4887 (MelSG082)", "UB-4888 (MelSG089)",
               "MaDE1 & E2", "UB-4552 (MaDE3)", "UB-4975 (AstCli12)", "UB-4835 (ApD134)",
               "SUERC-39108 ERLK G322", "SUERC-39109 ERL G362", "SUERC-39110 ERL G405",
               "SUERC-51560 ERL G038", "SUERC-39091 (ERL G003)", "SUERC-39092 (ERL G005)",
               "SUERC-39113 (ERL G171)", "SUERC-51549 (ERL G195)", "SUERC-51552 (ERL G107)",
               "SUERC-51550 (ERL G254)")

# Dates associated with bead BE1 Dghnt
be1.dghnt <- c("UB-4503 (Lec148)", "UB-4506 (Lec172/2)",
                "UB-6038 (CasD183)", "UB-4512 (EH091)", "UB-4501 (Lec014)",
                "UB-4507 (Lec187)", "UB-4502 (Lec138)", "UB-4042 (But1674)",
                "SUERC-39100 ERL G266")

# Construct a list of vectors
chains <- list("BE3-Amber" = be3.amber, "BE1-Dghnt" = be1.dghnt)

# Read the calibration MCMC output
oxc <- read_oxcal("https://tsdye.online/AP/beads-mcmc.csv", quiet = "yes")

# Observe 2x2 frequency matrix of the relation of trunk to branch
# allen_observe(mcmc = oxc, chains = chains, allen_set = "oFD")

## End(Not run)
```

Description

Plots a Nökel lattice to the display and optionally to a file.

Usage

```
allen_plot(
  allen_set,
  file_name = NULL,
  pad = 0.2,

allen_plot
Make a single plot of a Nökel lattice.
```
### Arguments

- **allen_set**
  - A dataframe with plot information, such as the one produced by `illustrate_allen_relations()`

- **file_name**
  - Optional path to the graphic file output

- **pad**
  - Padding in inches to the margins to keep labels from disappearing off the edge of the graphic (default 0.2)

- **font_size**
  - Font size for the labels in the plot (default 11)

- **height**
  - Height in inches of the graphic file output (default 7)

- **width**
  - Width in inches of the graphic file output (default 7)

- **columns**
  - Number of columns for a plot with more than one lattice (default 1)

- **plot_title**
  - Title for the plot, defaults to the title in `allen_set`

- **dpi**
  - Dots per inch for bitmap files (default 600)

### Value

Typically called for its side effects, returns `allen_set`

### Author(s)

Thomas S. Dye

---

**allen_proportion_result**

*Calculate the proportion of each relation in a result vector*

### Description

Divides through by the sum of observations in the result vector. Assigns the names of the result vector to the optionally sorted return vector.

### Usage

```
allen_proportion_result(result_vector, sort = FALSE)
```

### Arguments

- **result_vector**
  - A result vector

- **sort**
  - If TRUE sort in decreasing order else return unsorted vector
**Value**

A named vector with proportions

**Author(s)**

Thomas S. Dye

---

`allen_relate_intervals`

_Relate two or more observed intervals_

---

**Description**

Reads MCMC output to create a dataframe suitable for plotting the observed Allen relation of two intervals.

**Usage**

`allen_relate_intervals(mcmc, chains)`

**Arguments**

- `mcmc`  
  Dataframe or archaeophases_mcmc object with the MCMC output from a Bayesian calibration.

- `chains`  
  A list of lists, each with two named elements, each element a vector of names or indexes of columns in `mcmc`.

**Value**

A dataframe suitable for plotting with `allen_plot`.

**Author(s)**

Thomas S. Dye
**Description**

Calculates the Allen relation of two definite intervals and reports the one-letter code for the interval proposed by Thomas Alspaugh. Stops with an error if the end of an interval is earlier than its start.

**Usage**

```r
allen_relation(start_1, end_1, start_2, end_2)
```

**Arguments**

- `start_1` - The start date of the first interval
- `end_1` - The end date of the first interval
- `start_2` - The start date of the second interval
- `end_2` - The end date of the second interval

**Value**

A one-letter code indicating the Allen relation

**Author(s)**

Thomas S. Dye

---

**Description**

Set elements that are not Allen relation codes are silently ignored.

**Usage**

```r
allen_set_to_vector(s)
```

**Arguments**

- `s` - An Allen relation set, a vector of single letter codes.

**Value**

A named result vector.
**Description**

Return the six value Allen relation set for intervals with distinct endpoints.

**Usage**

```r
allen_six_value_set()
```

**Value**

An Allen relation set

**Author(s)**

Thomas S. Dye

---

**Description**

Convert a string containing Allen relation codes to a relation set

**Usage**

```r
allen_string_to_set(s)
```

**Arguments**

- `s` A string with Allen relation codes.

**Value**

A vector of single letter Allen relation codes.

**Author(s)**

Thomas S. Dye
\texttt{allen_string_to_vector}

\textit{Convert a string containing Allen relation codes to a result vector}

\textbf{Description}

A result vector is named with Allen relation codes and contains counts of observed relations.

\textbf{Usage}

\texttt{allen_string_to_vector(s)}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{s} A string with Allen relation codes
\end{itemize}

\textbf{Value}

A named result vector

\textbf{Author(s)}

Thomas S. Dye

\texttt{allen_union}

\textit{Union of two Allen relation sets.}

\textbf{Description}

Returns the union of two Allen relation sets, taking care to handle empty sets and the sets represented by result vectors.

\textbf{Usage}

\texttt{allen_union(set\_1, set\_2)}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{set\_1} The first Allen relation set or result vector
  \item \texttt{set\_2} The second Allen relation set or result vector
\end{itemize}

\textbf{Value}

An Allen relation set

\textbf{Author(s)}

Thomas S. Dye
**allen_update_result**

*Update a result vector*

**Description**

Increment the element of the result vector corresponding to the given relation.

**Usage**

```
allen_update_result(relation, result_vector)
```

**Arguments**

- **relation**: The relation to increment
- **result_vector**: The result vector to update

**Value**

The updated result vector

**Author(s)**

Thomas S. Dye

---

**analyze_allen_relations**

*Data for an analytic graphic*

**Description**

Calculates the Allen composition of two relations

**Usage**

```
analyze_allen_relations(relation_1, relation_2, title)
```

**Arguments**

- **relation_1**: a string representation of an Allen relation
- **relation_2**: a string representation of an Allen relation
- **title**: a title for the plot

**Value**

A dataframe for input to allen_plot
**Author(s)**

Thomas S. Dye

---

**app_ArchaeoPhases**

*Run ArchaeoPhases shiny apps*

---

**Description**

Run ArchaeoPhases shiny apps

**Usage**

`app_ArchaeoPhases()`

---

**ArchaeoPhases**

*ArchaeoPhases: Post-Processing of the Markov Chain Simulated by 'Chronomodel', 'OxCal', or 'BCal'.*

---

**Description**

Provides a list of functions for the statistical analysis of archaeological dates and groups of dates. It is based on the post-processing of the Markov Chains whose stationary distribution is the posterior distribution of a series of dates. Such output can be simulated by different applications, as for instance ChronoModel, OxCal, or BCal. The only requirement is to have a csv file containing a sample from the posterior distribution.

---

**coda.mcmc**

*Create an mcmc.list object for coda users*

---

**Description**

This wrapper function extracts parallel chains from a data frame to create an `mcmc.list` object for use with `coda` diagnostic tools

**Usage**

`coda.mcmc(data, numberChains = 1, iterationColumn = NULL)`

**Arguments**

- `data`: Data frame containing the output of the MCMC algorithm.
- `numberChains`: Number of parallel chains, default = 1.
- `iterationColumn`: Column number corresponding to the iteration values, default = NULL.
Value

An `mcmc.list` object.

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

See Also

`mcmc`

`mcmc.list`

Examples

```r
data(Events)
mcmcList = coda.mcmc(data = Events, numberChains = 3, iterationColumn = 1)
plot(mcmcList)
gelman.diag(mcmcList)
# The multivariate criterion can not be evaluated when a phase
# contains only one date. This induces colinearity problems.
gelman.diag(mcmcList, multivariate = FALSE)
```
CreateMinMaxGroup

Construct the minimum and maximum for a group of events (phase)

Description

Constructs a data frame containing the output of the MCMC algorithm corresponding to the minimum and maximum of a group of events

Usage

CreateMinMaxGroup(
  data,
  position,
  name = "Phase",
  add = NULL,
  exportFile = NULL
)

Arguments

data Data frame containing the output of the MCMC algorithm.
position Numeric vector containing the position of the column corresponding to the MCMC chains of all dates included in the phase of interest.
name Name of the current group of dates or phase.
add Name of the data frame in which the current minimum and maximum should be added, default = NULL.
exportFile Name of the final file that will be saved if chosen, default = NULL.

Value

A data frame containing the minimum and maximum of the group of dates included in the phase of interest. These values may be appended to a data frame add if given.

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

data(Events)
Temp = CreateMinMaxGroup(Events, c(2,4), name = "Phase2")
## Not run:
#To do for saving the new variables in csv file
Temp = CreateMinMaxGroup(Events, c(3,5), name = "Phase1", add=Temp, exportFile = "MinMaxPhases.csv")
## End(Not run)
CredibleInterval

Bayesian credible interval

Description
Computes the shortest credible interval of the output of the MCMC algorithm for a single parameter.

Usage
CredibleInterval(a_chain, level = 0.95, roundingOfValue = 0)

Arguments
- a_chain: Numeric vector containing the output of the MCMC algorithm for the parameter.
- level: Probability corresponding to the level of confidence used for the credible interval, default = 0.95.
- roundingOfValue: Integer indicating the number of decimal places to be used, default = 0.

Details
A \((100 \times \text{level})\) elements of the sample outside the interval. The \((100 \times \text{level})\) value.

Value
A named vector of values containing the confidence level and the endpoints of the shortest credible interval in calendar years (BC/AD).

Examples
```r
data(Events); attach(Events)
CredibleInterval(Event.1)
CredibleInterval(Event.12, 0.50)
```

credible_interval

Bayesian credible interval

Description
Computes the shortest credible interval for a single parameter.

Usage
credible_interval(data, level = 0.95, round_to = 0)
**Arguments**

- **data**: Numeric vector containing the output of the MCMC algorithm for the parameter.
- **level**: Probability corresponding to the level of confidence used for the credible interval, default = 0.95.
- **round_to**: Integer indicating the number of decimal places to be used, default = 0.

**Details**

A \((100 \times level)\) that keeps \(N \times (1 - level)\) elements of the sample outside the interval. The \((100 \times level)\) of those intervals.

**Value**

A list with the following components:

- **ci**: Named vector of length 2, with \(\text{inf}\) the lower endpoint of the shortest credible interval as a calendar year; and \(\text{sup}\) the upper endpoint of the shortest credible interval as a calendar year;
- **level**: Confidence level for the credible intervals; and
- **call**: Function call.

**Examples**

```r
data(Events); attach(Events)
credible_interval(Event.1)
credible_interval(Event.12, 0.50)
```

---

**DatesHiatus**  
*Test for the existence of a hiatus between two parameters*

**Description**

Finds if a gap exists between two dates and returns the longest interval that satisfies: \(P(a_{\text{chain}} < \text{IntervalInf} < \text{IntervalSup} < b_{\text{chain}} | M) = level\)

**Usage**

```
DatesHiatus(a_chain, b_chain, level = 0.95)
```

**Arguments**

- **a_chain**: Numeric vector containing the output of the MCMC algorithm for the first parameter.
- **b_chain**: Numeric vector containing the output of the MCMC algorithm for the second parameter.
- **level**: Probability corresponding to the confidence level of the interval.
dates_hiatus

Value

A named vector with the level and the endpoints of the gap in calendar years (AD/BC)

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

data(Events); attach(Events)
DatesHiatus(Event.1, Event.12)
DatesHiatus(Event.1, Event.12, level = 0.5)

dates_hiatus
Test for the existence of a hiatus between two MCMC chains.

Description

Determines whether there is a hiatus between two MCMC chains and returns the longest interval that satisfies: $P(a_{chain} < IntervalInf < IntervalSup < b_{chain}|M) = level$

Usage

dates_hiatus(a_chain, b_chain, level = 0.95)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a_chain</td>
<td>Numeric vector containing the output of the MCMC algorithm for the first parameter.</td>
</tr>
<tr>
<td>b_chain</td>
<td>Numeric vector containing the output of the MCMC algorithm for the second parameter.</td>
</tr>
<tr>
<td>level</td>
<td>Probability corresponding to the confidence level of the interval.</td>
</tr>
</tbody>
</table>

Value

A list with the following components:

- **hiatus** A named vector where inf is the lower endpoint of the hiatus as a calendar year (AD/BC) or NA if there is no hiatus at level, and sup is the upper endpoint of the gap as a calendar year (AD/BC), or NA if there is no hiatus at level.
- **duration** The duration of the hiatus at level.
- **level** Probability corresponding to the confidence level of the interval.
- **call** The function call.
estimate_range

**Description**

Calculates the ranges of summary statistics from the output of two or more runs of the MCMC algorithm. Results are given in calendar years for statistics that estimate them.

**Usage**

```r
estimate_range(
  mcmc,
  position,
  app = "bcal",
  estimates = c("mean", "q1", "median", "q3", "ci.inf", "ci.sup"),
  quiet = "partial",
  bin_width = 1,
  decimal = ".",
  separator = ","
)
```

**Arguments**

- **mcmc**: A vector of path names to the MCMC files.
- **position**: Numeric vector containing the positions of the columns corresponding to the MCMC chains of interest, or a vector of column names.
- **app**: Name of the application that created the MCMC files, one of `bcal`, `oxcal`, `chronomodel`.
- **estimates**: Numeric vector containing the positions of the columns corresponding to the statistics of interest returned by the `multi_marginal_statistics()` function, or a vector of column names.
- **quiet**: One of `no` (default) to allow messages and warnings, `partial` to suppress messages and allow warnings, or `yes` to suppress messages and warnings.

**Examples**

```r
data(Events); attach(Events)
dates_hiatus(Event.1, Event.12)
dates_hiatus(Event.1, Event.12, level = 0.5)
```
estimate_range

bin_width  If app is set to bcal, the bin width specified for the BCal calibration. Defaults to the BCal default of 1.

decimal   If app is set to chronomodel, either . (default) or ,, the two choices offered by ChronoModel.

separator  If app is set to chronomodel, the character used to separate fields in the CSV file. Defaults to ,,.

Details

This function is useful for estimating the sensitivity of calibration results to different model parameters.

Value

A list with the following components:

- **range_table**  A matrix of estimate ranges.
- **mean**  The mean of the ranges in range_table.
- **sd**  The standard deviation of the ranges in range_table.
- **min**  The minimum of the ranges in range_table.
- **median**  The median of the ranges in range_table.
- **max**  The maximum value of the ranges in range_table.

Author(s)

Thomas S. Dye, <tsd@tsdye.online>

Examples

```r
## Not run:
## Generate 0's
res <- estimate_range(mcmc = c("http://tsdye.online/AP/ox.csv", "http://tsdye.online/AP/ox.csv"), position = c(1, 2), app = "oxcal", quiet = "yes")
sum(res$range_table)
## End(Not run)
```
Illustrate Allen Relations

Description

A data set containing information on the ages of four dated events.

Usage

Events

Format

A data frame with 30,000 rows and 5 variables:

- **iter**: iteration of the MCMC algorithm
- **Event.2**: information on event 2
- **Event.1**: information on event 1
- **Event.22**: information on event 22
- **Event.12**: information on event 12

Illustrate Allen Relations

Data for an illustrative graphic

Description

Create a dataframe that can be used as input for an illustrative plot. Useful for describing the Allen operators: illustrate the full set of Allen relations, concurrent Allen relations, and relations with distinct endpoints. Also, useful for describing the chronological domains of stratification, branching, transformation, and reticulation.

Usage

illustrate_allen_relations(relations = "basic")

Arguments

- **relations**: One of 'basic', 'concurrent', 'distinct', 'stratigraphic', 'branching', 'transformation', 'reticulation', 'sequence', 'branch', 'transform', or 'reticulate'.
Details

The illustrative graphics include:

- **basic** the 13 basic Allen relations (default);
- **concurrent** concurrent relations;
- **distinct** relations with distinct endpoints;
- **stratigraphic** basic relations established by an observation of superposition;
- **branching** basic branching relations;
- **transformation** basic relations of transformation;
- **reticulation** basic relations of reticulation;
- **sequence** composite relations in a stratigraphic sequence;
- **branch** composite relations of branching;
- **transform** composite relations of transformation; or
- **reticulate** composite relations of reticulation.

Value

A dataframe for input to `allen_plot`

Author(s)

Thomas S. Dye

References


---

**Importing a CSV file**

**ImportCSV**

**Description**

Import a CSV file containing the output of the MCMC algorithm
Usage

ImportCSV(
  file,
  dec = ".", 
  sep = ",", 
  comment.char = "#", 
  header = TRUE, 
  iterationColumn = NULL, 
  referenceYear = NULL, 
  rowToWithdraw = NULL, 
  bin.width = NULL 
)

Arguments

file  Name of the CSV file containing the output of the MCMC algorithm.
dec   Character used in the file for decimal points for the use of read.csv().
sep   Field separator character for the use of read.csv().
comment.char Character vector of length one containing a single character or an empty string for the use of read.csv().
header Logical value indicating whether the file contains the names of the variables as its first line.
iterationColumn Column number corresponding to the iteration values, default = NULL.
referenceYear Year of reference for MCMC in date format other than BC/AD, default = NULL.
rowToWithdraw Number of the row to be withdrawn or "last" for the last row of the data frame, default = NULL.
bin.width Bin width specified in a BCal project (note that bin.width does not have to be set if the BCal default bin width of 1 is used).

Details

Use of the read.csv() function with default values for CSV files produced by ChronoModel software. For MCMC in a date format different from BC/AD, use the parameter referenceYear to convert the MCMC to BC/AD, otherwise the remaining functions of ArchaeoPhases will not work. MCMC files generated by BCal may contain an empty last row. This row should be withdrawn using the rowToWithdraw parameter. Otherwise, the functions of ArchaeoPhases will not work properly.

Value

A data frame containing a representation of the data in the file.

Author(s)

Anne Philippe, Anne.Philippe@univ-nantes.fr, Thomas S. Dye, tsd@tsdye.online, and Marie-Anne Vibet, Marie-Anne.Vibet@univ-nantes.fr
Importing a BCal csv file

Description
Importing a csv file containing the output of the MCMC algorithm from the BCal software

Usage
ImportCSV.BCal(file, bin.width = NULL)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>Name of the CSV file containing the output of the MCMC algorithm.</td>
</tr>
<tr>
<td>bin.width</td>
<td>Bin width specified in a BCal project (note: bin.width does not have to be set if the BCal default bin width of 1 is used).</td>
</tr>
</tbody>
</table>

Value

A data frame containing a representation of the data in the CSV file
is.url

Author(s)
Anne Philippe, <Anne.Philippe@univ-nantes.fr>,
Thomas S. Dye, <tsd@tsdye.online>, and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

## Not run:
# Import of MCMC generated by BCal and extracted in cal BP (the year of reference is 1950)
data(Fishpond)
write.csv(Fishpond, "fishpond_MCMC.csv", row.names = FALSE)
Fishpond = ImportCSV.BCal("fishpond_MCMC.csv", bin.width = 1)
## End(Not run)

## Not run:
# equivalent call
Fishpond2 = ImportCSV("fishpond_MCMC.csv", dec = ".", sep="," , referenceYear = 1950,
                      rowToWithdraw = "last", bin.width = 1)
## End(Not run)

is.url

Check if string is a URL

Description

Uses a regex approach to check if a string is a URL. This approach is faster than url.exists but
does do the actual verification.

Usage

is.url(x)

Arguments

x A character string.

Details


Value

Returns a logical evaluation as to whether a string is a URL.
MarginalPlot

Plot a marginal posterior density

Description

Draws a plot of the estimated marginal posterior density for the one-parameter and adds the mean and the credible interval at the desired level.

Usage

MarginalPlot(
  a_chain,
  level = 0.95,
  GridLength = 1024,
  title = "Characteristics of a date",
  subtitle = NULL,
  caption = "ArchaeoPhases",
  x.label = "Calendar year",
  y.label = NULL,
  y.grid = TRUE,
  x.scale = "calendar",
  elapsed.origin.position = NULL,
  x.min = NULL,
  x.max = NULL,
  height = 7,
  width = 7,
  units = "in",
  file = NULL,
  newWindow = TRUE
)

Arguments

  a_chain       Numeric vector containing the output of the MCMC algorithm for the parameter.
  level         Probability corresponding to the level of confidence.
  GridLength    Length of the grid used to estimate the density.
  title         Title of the graph.
  subtitle      Subtitle of the graph.
  caption       Caption of the graph.
  x.label       Label of the x-axis.
  y.label       Label of the y-axis.
  y.grid        Switch for horizontal grid lines.
  x.scale       One of "calendar" for calendar years, "BP" for years before present, or "elapsed" for time elapsed from a specified origin.
MarginalProba

elapsed.origin.position
Position of the column to use as the origin for elapsed time calculations.
x.min
Minimum x axis value.
x.max
Maximum x axis value.
height
Plot height in units.
width
Plot width in units.
units
String recognized by the ggsave() function, one of "in", "cm", "mm".
file
Name of the file that will be saved if chosen, default = NULL.
newWindow
Whether or not the plot is drawn within a new window.

Details
The density is estimated using density() function with n = GridLength.

Value
NULL, called for its side effects

Author(s)
Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

data(Events);
MarginalPlot(a_chain = Events$Event.1, level = 0.95)

MarginalProba  Bayesian test for anteriority / posteriority between two parameters

Description
This function estimates the posterior probability that event 'a' is older than event 'b' using the output of the MCMC algorithm. This provides a Bayesian test for checking the following assumption: "Event a is older than event b".

Usage
MarginalProba(a_chain, b_chain)

Arguments
a_chain : Numeric vector containing the output of the MCMC algorithm for the first parameter.
b_chain : Numeric vector containing the output of the MCMC algorithm for the second parameter.
MarginalStatistics

Details
For a given output of MCMC algorithm, this function estimates the posterior probability of the event 'a' < 'b' by the relative frequency of the event "the value of event 'a' is less than the value of event 'b'" in the simulated Markov chain.

Value
An unnamed vector with the posterior probability of the assumption: "event a is older than event b"

Author(s)
Anne Philippe, <Anne.Philippe@univ-nantes.fr> and Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

```r
data(Events); attach(Events)
# Probability that Event.1 is older than Event.12
MarginalProba(Event.1, Event.12)
# Probability that Event.1 is older than Event.2
MarginalProba(Event.1, Event.2)
# Probability that the beginning of the phase 1 is older than the end of the phase 1
# Should always be 1 for every phase
data(Phases); attach(Phases)
MarginalProba(Phase.1.alpha, Phase.1.beta)
```

MarginalStatistics

Marginal summary statistics

Description
Calculates summary statistics of the output of the MCMC algorithm for a one-parameter. Results are given in calendar years (BC/AD).

Usage

```r
MarginalStatistics(a_chain, level = 0.95, roundingOfValue = 0)
```

Arguments

- **a_chain**: Numeric vector containing the output of the MCMC algorithm for the parameter.
- **level**: Probability corresponding to the level of confidence used for the credible interval and the highest posterior density region.
- **roundingOfValue**: Integer indicating the number of decimal places.
Details

The \((100 \times level)\)\.

Value

A named matrix of values corresponding to all the following statistics:

- **title**  The title of the summary statistics.
- **mean** The mean of the MCMC chain. Use of `mean()` function.
- **map** The maximum a posteriori of the MCMC chain. Use of `hdr()` function.
- **sd** The standard deviation of the MCMC chain. Use of `sd()` function.
- **Q1, median, Q3** The quantiles of the MCMC chain corresponding to 0.25, 0.50 and 0.75. Use of `quantile` function.
- **CI** The credible interval corresponding to the desired level. Use of `CredibleInterval()` function.
- **HPDR** The highest posterior density regions corresponding to the desired level. Use of `hdr()` function.

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

References


Examples

```r
data(Events); attach(Events)
MarginalStatistics(Event.1)
MarginalStatistics(Event.2, level = 0.90)
```

---

**marginal_plot**

*Plot a marginal posterior density*

**Description**

Draws a plot of the marginal posterior density for a single parameter, with an option to add the mean and the credible interval at the desired level
Usage

```r
marginal_plot(
  data,
  position = 1,
  level = 0.95,
  grid_length = 1024,
  title = if (is.numeric(position)) names(data)[position] else position,
  subtitle = "Marginal posterior density",
  caption = paste(level * 100, "% credible interval", sep = " "),
  x_label = "Calendar year",
  y_label = "Density",
  y_grid = TRUE,
  x_scale = "calendar",
  elapsed_origin_position = NULL,
  x_min = NULL,
  x_max = NULL,
  height = 7,
  width = 7,
  units = "in",
  file = NULL,
  plot_result = TRUE,
  mean_linetype = "dashed",
  mean_color = "white",
  mean_size = 0.5,
  ci_linetype = "dotted",
  ci_color = mean_color,
  ci_size = mean_size,
  line_linetype = "solid",
  line_color = "black",
  line_size = 1,
  density_color = "gray30",
  fill_palette = NULL
)
```

Arguments

data: Data frame containing the output of the MCMC algorithm.
position: Index of the column corresponding to the MCMC chain of interest, or a column name.
level: Probability corresponding to the level of confidence.
grid_length: Length of the grid used to estimate the density.
title: Title of the graph. The default uses the data column name.
subtitle: Subtitle of the graph. The default is "Marginal posterior density".
caption: Caption of the graph. The default describes the confidence of the credible interval.
x_label: Label of the x-axis.
y_label  
Label of the y-axis.

y_grid  
Switch for horizontal grid lines.

x_scale  
One of "calendar" for calendar years, "BP" for years before present, or "elapsed" for time elapsed from a specified origin.

elapsed_origin_position  
Position of the column to use as the origin for elapsed time calculations.

x_min  
Minimum x axis value.

x_max  
Maximum x axis value.

height  
Plot height in units.

width  
Plot width in units.

units  
String recognized by the ggsave() function, one of "in", "cm", "mm". This parameter has no effect on the display plot.

file  
Name of the file that will be saved if chosen, default = NULL.

plot_result  
If TRUE, then draw a plot on the display, else suppress drawing.

mean_linetype  
The linetype used to indicate the mean density.

mean_color  
The color of the line used to indicate mean density.

mean_size  
The width of the line used to indicate the mean density.

ci_linetype  
The linetype used to indicate the credible intervals.

ci_color  
The color of the lines used to indicate the credible intervals.

ci_size  
The width of the lines used to indicate the credible intervals.

line_linetype  
The linetype used to indicate the density.

line_color  
The color of the line used to indicate the density.

line_size  
The width of the line used to indicate the density.

density_color  
Color to use if fill_palette is not specified.

fill_palette  
Palette to use for fills.

Details

The plot is drawn with the current theme and color scales; the function does not alter or override theme elements.

Value

An archaeophases_plot object with the data and metadata needed to reproduce the plot.

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr>;
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>; and
Thomas S. Dye, <tsd@tsdye.online>
Examples

```r
data(Events)
mp <- marginal_plot(data = Events, position = 2, level = 0.95)
## View data and metadata
str(mp)
```

---

**marginal_statistics**  
*Marginal summary statistics*

**Description**

Calculates summary statistics of the output of the MCMC algorithm for a single parameter. Results are given in calendar years (BC/AD).

**Usage**

```r
marginal_statistics(a_chain, level = 0.95, round_to = 0)
```

**Arguments**

- `a_chain`  
  Numeric vector containing the output of the MCMC algorithm for the parameter.

- `level`  
  Probability corresponding to the level of confidence used for the credible interval and the highest posterior density region.

- `round_to`  
  Integer indicating the number of decimal places.

**Details**

The \((100 \times \text{level})\) using \texttt{hdr()} function from \texttt{hdrcde} package.

**Value**

A list with the following components:

- `mean`  
  The mean of the MCMC chain.

- `map`  
  The maximum a posteriori of the MCMC chain.

- `sd`  
  The standard deviation of the MCMC chain.

- `quantiles`  
  A vector with the following elements: `min` = minimum value of the MCMC chain; `q1` = first quantile of the MCMC chain; `median` = median of the MCMC chain; `q2` = second quantile of the MCMC chain; and `max` = maximum value of the MCMC chain.

- `level`  
  Confidence level for the credible interval and highest posterior density.

- `ci`  
  A vector with the following elements: `inf` = lower credible interval of the MCMC chain at `level`; and `sup` = upper credible interval of the MCMC chain at `level`.

- `hpdr`  
  A variable length vector with the lower and upper highest posterior density regions of the MCMC chain at `level`. List components are named `inf_n` and `sup_n` for `n = 1` to the number of highest posterior density regions.
MultiCredibleInterval

Author(s)
Anne Philippe, <Anne.Philippe@univ-nantes.fr>,
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>, and
Thomas S. Dye, <tsd@tsdye.online>

References

Examples
```r
data(Events); attach(Events)
marginal_statistics(Event.1)
marginal_statistics(Event.2, level = 0.90)
## convenient vector
foo <- marginal_statistics(Event.1)
unlist(foo)
```

MultiCredibleInterval Bayesian credible interval for a series of dates

Description
Estimation of the shortest credible interval for each variable of a simulated Markov chain

Usage
```r
MultiCredibleInterval(data, position, level = 0.95, roundingOfValue = 0)
```

Arguments
data data frame containing the output of the MCMC algorithm.
position Numeric vector containing the position of the column corresponding to the MCMC chains of interest.
level Probability corresponding to the level of confidence used for the credible interval.
roundingOfValue Integer indicating the number of decimal places.

Details
A \( (100 \times level) \) The \( (100 \times level) \)
Value

Returns a matrix of values containing the level of confidence and the endpoints of the shortest credible interval for each variable of the MCMC chain. The name of the resulting rows are the positions of the corresponding columns in the CSV file. The result is given in calendar years (BC/AD).

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

data(Events)
MultiCredibleInterval(Events, c(2, 4, 3), 0.95)

Description

Plot of credible intervals or HPD regions of a series of events

Usage

MultiDatesPlot(
  data,
  position,
  level = 0.95,
  roundingOfValue = 0,
  intervals = "CI",
  order = "default",
  title = "Plot of intervals",
  subtitle = NULL,
  caption = "ArchaeoPhases",
  labelXaxis = "Calendar Year",
  labelYaxis = NULL,
  height = 7,
  width = 7,
  units = "in",
  x.min = NULL,
  x.max = NULL,
  x.scale = "calendar",
  elapsed.origin.position = NULL,
  dumbbell.size = 3,
  dot.guide = FALSE,
  dot.guide.size = 0.25,
MultiDatesPlot

y.grid = FALSE,
file = NULL,
newWindow = TRUE,
print.data.result = FALSE
)

Arguments

data Data frame containing the output of the MCMC algorithm.
position Numeric vector containing the position of the column corresponding to the MCMC chains of interest.
level Probability corresponding to the level of confidence.
roundingOfValue Integer indicating the number of decimal places to be used.
intervals One of "CI" for credible intervals, or "HPD" for highest posterior density intervals.
order Order of the events. If "default" then the order of the csv file is followed, if "increasing" events are ordered by the HPDInf of the first region or the CIInf
title Title of the plot.
subtitle Subtitle of the plot.
caption Caption of the plot.
labelXaxis X axis label of the plot.
labelYaxis Y axis label of the plot.
height Height of the plot in units.
width Width of the plot in units.
units A string recognized by ggsave() function, one of "in", "cm", "mm".
x.min Minimum x axis value.
x.max Maximum x axis value.
x.scale One of "calendar" for calendar years, "BP" for years before present, or "elapsed" for years after a specified origin.
elapsed.origin.position Position of the column corresponding to the origin for elapsed time calculations.
dumbbell.size Size of the symbols used to plot events.
dot.guide Switch for guides from y-axis to plot symbols.
dot.guide.size Size of the dot guides.
y.grid Switch for horizontal grids.
file Name of the file to be saved. If NULL then no plot is saved.
newWindow Whether the plot is drawn within a new window or not.
print.data.result If TRUE, the list containing the data to plot will be returned.
MultiHPD

Bayesian HPD regions for a series of MCMC chains

Description

Estimation of the highest posterior density regions for each variable of a simulated Markov chain. This function uses thehdr() function included in thehdrcde package. An HPD region may be a union of several intervals.

Usage

MultiHPD(data, position, level = 0.95, roundingOfValue = 0)

Arguments

data

Data frame containing the output of the MCMC algorithm.

position

Numeric vector containing the position of the column corresponding to the MCMC chains of interest.

level

Probability corresponding to the level of confidence.

roundingOfValue

Integer indicating the number of decimal places.

Details

Highest posterior density function region using the functionhdr() from thehdrcde package.

Value

NULL, called for its side effects. Ifprint.data.result = TRUE then a list containing the data to plot will be returned.

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr>, Thomas S. Dye, <tsd@tsdye.online>, and Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

data(Events)
MultiDatesPlot(Events, c(2, 4, 3), level = 0.95, intervals = "CI",
        title = "Plot of CI intervals")
MultiDatesPlot(Events, c(2, 4, 3), level = 0.95, intervals = "HPD",
        title = "Plot of HPD intervals")
MultiDatesPlot(Events, c(2, 4, 3), level = 0.95, intervals = "HPD",
        order = "increasing")

MultiHPD

Bayesian HPD regions for a series of MCMC chains
Value

Returns a matrix of values containing the level of confidence and the endpoints of each interval for each variable of the MCMC chain. The name of the resulting rows are the positions of the corresponding columns in the CSV file. The result is given in calendar years (BC/AD).

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

References


Examples

data(Events)
MultiHPD(Events, c(2, 4, 3), 0.95)

---

### MultiMarginalPlot

#### Marginal posterior densities of several events

Description

Draws a plot of the estimated marginal posterior density for a parameter and adds the mean and the credible interval at the desired level

Usage

```r
MultiMarginalPlot(
  data,
  position,
  level = 0.95,
  GridLength = 1024,
  x.scale = rep("calendar", length(position)),
  elapsed.origin = NULL,
  title = "Characteristics of several dates",
  subtitle = NULL,
  caption = "ArchaeoPhases",
  x.label = "Calendar year",
  y.label = NULL,
  y.grid = TRUE,
  x.min = NULL,
  x.max = NULL,
  legend.title = "Legend",
)```
Arguments

data          Data frame containing the output of the MCMC algorithm.
position      Numeric vector containing the position of the column corresponding to the MCMC chains of interest.
level         Probability corresponding to the level of confidence.
GridLength    Number of equally spaced points at which the density is to be estimated (for density() function).
x.scale       One of "calendar" for calendar years, "BP" for years before present, or "elapsed" for time elapsed from a specified origin.
elapsed.origin Position of the column to use as the origin for elapsed time calculations.
title         Title of the plot.
subtitle      Subtitle of the plot.
caption       Caption of the plot.
x.label       Label of the x-axis.
y.label       Label of the y-axis.
y.grid        Switch for horizontal grid lines.
x.min         Minimum x-axis value.
x.max         Maximum x-axis value.
legend.title  Title for the legend.
height        Plot height in units.
width         Plot width in units.
units         String recognized by the ggsave() function, one of "in", "cm", "mm".
file          Name of the file that will be saved if specified, default = NULL.
newWindow     Whether or not the plot is drawn within a new window.

Details

The density is estimated using density() function with n = GridLength. The input MCMC chains should either be in calendar years or converted to calendar years using x.scale vector or elapsed.origin.

Value

NULL, called for its side effects.
Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

```r
data(Events);
MultiMarginalPlot(Events, position = c(2, 3, 4), level = 0.95)
```

---

**MultiPhasePlot**  
*Several phase density plots*

Description

Plot of the marginal posterior densities of several groups

Usage

```r
MultiPhasePlot(
  data,
  position_minimum,
  position_maximum = position_minimum + 1,
  level = 0.95,
  title = "Characterisation of several groups",
  colors = NULL,
  exportFile = NULL,
  exportFormat = "PNG"
)
```

Arguments

- `data` Data frame containing the output of the MCMC algorithm.
- `position_minimum` Numeric vector containing the column number corresponding to the minimum of the events included in each group.
- `position_maximum` Numeric vector containing the column number corresponding to the end of the groups set in the same order as in `position_minimum`.
- `level` Probability corresponding to the level of confidence.
- `title` Title of the plot.
- `colors` Numeric vector of colors for each group of dates.
- `exportFile` Name of the file to be saved. If NULL then no plot is saved.
- `exportFormat` Format of the export file, one of "PNG" or "SVG".
Details

Draws a plot with the marginal posterior densities of the minimum and the maximum of the dates included in each group. No temporal order between phases is required. The result is given in calendar years (BC/AD).

Value

NULL, called for its side effects

Author(s)
Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

```r
# Data extracted from ChronoModel software
data(Phases)
# List of the name of the groups
names(Phases)
# Stipulating position_maximum
MultiPhasePlot(Phases, c(4, 2), c(5, 3), title = "Succession of phase 1 and phase 2")
# In this case, equivalent to
MultiPhasePlot(Phases, c(4, 2), title = "Succession of phase 1 and phase 2", colors = c(3, 4))
```

MultiPhasesGap

Gap or hiatus between a succession of groups (for groups in temporal order constraint)

Description

Finds, if it exists, a gap or hiatus between two successive groups. This gap or hiatus is the longest interval that satisfies \( P(\text{Phase1Max} < \text{IntervalInf} < \text{IntervalSup} < \text{Phase2Min}|M) = \text{level} \)

Usage

```r
MultiPhasesGap(
  data,
  position_minimum,
  position_maximum = position_minimum + 1,
  level = 0.95
)
```
MultiPhasesTransition

Transition range for a succession of groups (for groups in temporal order constraint)

Description

Finds, if it exists, the shortest interval that satisfies \( P(\text{TransitionRangeInf} < \text{Phase1Max} < \text{Phase2Min} < \text{TransitionRangeSup}|M) = level \)
Usage

```r
MultiPhasesTransition(
  data,
  position_minimum,
  position_maximum = position_minimum + 1,
  level = 0.95
)
```

Arguments

data Data frame containing the output of the MCMC algorithm.
position_minimum Numeric vector containing the column number corresponding to the minimum of the events included in each group.
position_maximum Numeric vector containing the column number corresponding to the end of the groups set in the same order as in codeposition_minimum.
level Probability corresponding to the level of confidence.

Details

For each i, MultiPhasesTransition() computes the transition interval for the phase defined by its minimum position_minimum[i] and its maximum position_maximum[i]. The default value of position_maximum corresponds to CSV files exported from ChronoModel software.

Value

A matrix of values containing the level of confidence and the endpoints of the transition interval for each pair of successive groups. The result is given in calendar years (BC/AD).

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

```r
# Data extracted from ChronoModel software
data(Phases)
# List of the name of the groups
names(Phases)
# Stipulating position_maximum
MultiPhasesTransition(Phases, position_minimum = c(4, 2), position_maximum = c(5, 3))
# In this case, equivalent to
MultiPhasesTransition(Phases, position_minimum = c(4, 2))
```
MultiPhaseTimeRange

Phase time range for multiple groups

Description

Computes the shortest interval that satisfies \( P(PhaseMin < IntervalInf < IntervalSup < PhaseMax|M) = level \) for each phase.

Usage

```r
MultiPhaseTimeRange(
  data,
  position_minimum,
  position_maximum = position_minimum + 1,
  level = 0.95
)
```

Arguments

- **data**: Data frame containing the output of the MCMC algorithm.
- **position_minimum**: Numeric vector containing the column number corresponding to the minimum of the events included in each phase.
- **position_maximum**: Numeric vector containing the column number corresponding to the maximum of the phases set in the same order as in `position_minimum`.
- **level**: Probability corresponding to the desired level of confidence.

Details

For each \( i \), \( MultiPhaseTimeRange() \) computes the time range interval for the phase defined by its minimum \( position_{minimum[i]} \) and its maximum \( position_{maximum[i]} \). The default value of `position_maximum` corresponds to CSV files exported from ChronoModel software.

Value

A matrix of values containing the level of confidence and the endpoints of the shortest time range associated with the desired `level`. The result is given in calendar years (BC/AD).

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>
Examples

```r
# Data extracted from ChronoModel software
data(Phases)
# List of the name of the groups
names(Phases)
# Stipulating position_maximum
MultiPhaseTimeRange(Phases, position_minimum = c(4, 2), position_maximum = c(5, 3))
# In this case, equivalent to
MultiPhaseTimeRange(Phases, position_minimum = c(4, 2))
```

---

**MultiSuccessionPlot**  
**Successive Phases Density Plots (for phases in temporal order constraint)**

**Description**

This function draws a plot of the densities of several successive phases and adds several statistics (mean, CI, HPDR). The result is given in calendar years (BC/AD).

**Usage**

```r
MultiSuccessionPlot(
  data, 
  position_minimum, 
  position_maximum = position_minimum + 1, 
  level = 0.95, 
  title = "Characterisation of a succession of groups", 
  colors = NULL, 
  exportFile = NULL, 
  exportFormat = "PNG"
)
```

**Arguments**

- `data`: Data frame containing the output of the MCMC algorithm.
- `position_minimum`: Numeric vector containing the column number corresponding to the minimum of the events included in each group.
- `position_maximum`: Numeric vector containing the column number corresponding to the end of the groups set in the same order as in `position_minimum`.
- `level`: Probability corresponding to the level of confidence.
- `title`: Title of the plot.
- `colors`: Vector of colors corresponding to each group of dates.
- `exportFile`: Name of the file to be saved. If NULL then no plot is saved.
- `exportFormat`: Format of the export file, either "PNG" or "SVG" (default).
multi_credible_interval

Bayesian credible interval for a series of dates

Description

Estimate the shortest credible interval for each of several MCMC chains.

Usage

multi_credible_interval(data, position, level = 0.95, round_to = 0)
Arguments

**data**
data frame containing the output of the MCMC algorithm.

**position**
Numeric vector containing the position of the column corresponding to the MCMC chains of interest, or a list of column names.

**level**
Probability corresponding to the level of confidence used for the credible interval.

**round_to**
Integer indicating the number of decimal places.

Details

A \((100 \times \text{level})\) that keeps \(N \times (1 - \text{level})\) elements of the sample outside the interval. The \((100 \times \text{level})\).

Value

Returns a list with the following components:

**ci** A data frame with a row for each column in `data` and two columns: `inf`, the lower credible interval in calendar years (BC/AD); and `sup`, the upper credible interval in calendar years (BC/AD).

**level** Probability corresponding to the level of confidence used for the credible interval.

**call** The function call.

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr>, Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>, and Thomas S. Dye, <tsd@tsdye.online>.

Examples

```r
data(Events)
multi_credible_interval(Events, c(2, 4, 3), 0.95)
# round to decade
multi_credible_interval(Events, c(2, 4, 3), 0.95, -1)
```

---

**multi_dates_plot**  
Plot of credible intervals or HPD regions of a series of events

Description

Plot of credible intervals or HPD regions of a series of events
Usage

```r
multi_dates_plot(
  data,
  position = 1:ncol(data),
  level = 0.95,
  plot_result = TRUE,
  round = 0,
  intervals = "CI",
  order = "default",
  title = "Plot of intervals",
  subtitle = NULL,
  caption = "ArchaeoPhases",
  x_label = "Calendar Year",
  y_label = NULL,
  height = 7,
  width = 7,
  units = "in",
  x_min = NULL,
  x_max = NULL,
  x_scale = "calendar",
  elapsed_origin_position = NULL,
  dumbbell_size = 1,
  dot_guide = FALSE,
  dot_guide_size = 0.25,
  y_grid = FALSE,
  file = NULL,
  new_window = TRUE
)
```

Arguments

data | Data frame containing the output of the MCMC algorithm.
position | Numeric vector containing the positions of the columns corresponding to the MCMC chains of interest, or a vector of column names.
level | Probability corresponding to the level of confidence.
plot_result | If TRUE, then draw a plot on the display, else suppress drawing.
round | Integer indicating the number of decimal places to be used.
intervals | One of "CI" for credible intervals, or "HPD" for highest posterior density intervals.
order | Order of the events. If "default" then the order of the csv file is followed, if "increasing" events are ordered by the HPDInf of the first region or the CIInf

title | Title of the plot.
subtitle | Subtitle of the plot.
caption | Caption of the plot.
x_label | X axis label of the plot.
y_label Y axis label of the plot.
height Height of the plot in units.
width Width of the plot in units.
units A string recognized by gg saves() function, one of "in", "cm", "mm".
x_min Minimum x axis value.
x_max Maximum x axis value.
x_scale One of "calendar" for calendar years, "BP" for years before present, or "elapsed" for years after a specified origin.
elapsed_origin_position Position of the column corresponding to the origin for elapsed time calculations.
dumbbell_size Size of the symbols used to plot events.
dot_guide Switch for guides from y-axis to plot symbols.
dot_guide_size Size of the dot guides.
y_grid Switch for horizontal grids.
file Name of the file to be saved. If NULL then no plot is saved.
new_window Whether the plot is drawn within a new window or not.

Value
An archaeophases_plot object with the data and metadata needed to reproduce the plot.

Author(s)
Anne Philippe, <Anne.Philippe@univ-nantes.fr>,
Thomas S. Dye, <tsd@tsdye.online>, and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples
data(Events)
multi_dates_plot(Events, c(2, 4, 3), level = 0.95, intervals = "CI",
        title = "Plot of CI intervals")
multi_dates_plot(Events, c(2, 4, 3), level = 0.95, intervals = "HPD",
        title = "Plot of HPD intervals")
multi_dates_plot(Events, c(2, 4, 3), level = 0.95, intervals = "HPD",
        order = "increasing")
multi_hpd

Bayesian HPD regions for a series of MCMC chains

Description
Estimation of the highest posterior density regions for each variable of a simulated Markov chain. This function uses the \texttt{hdr()} function included in the \texttt{hdrcde} package. An HPD region may be a union of several intervals.

Usage
\begin{verbatim}
multi_hpd(data, position, level = 0.95, round_to = 0)
\end{verbatim}

Arguments
- \texttt{data} Data frame containing the output of the MCMC algorithm.
- \texttt{position} Numeric vector containing the position of the column corresponding to the MCMC chains of interest.
- \texttt{level} Probability corresponding to the level of confidence.
- \texttt{round_to} Integer indicating the number of decimal places.

Details
Highest posterior density function region using the function \texttt{hdr()} from the \texttt{hdrcde} package

Value
Returns a list with the following components:

- \texttt{results} A data frame where the rows correspond to the columns in the selected data set and the columns labeled \texttt{inf} and \texttt{sup} correspond to the lower and upper endpoints of each highest posterior density interval, respectively.
- \texttt{level} Probability corresponding to the level of confidence.
- \texttt{call} The function call.

matrix of values containing the level of confidence and for each variable of the MCMC chain. The name of the resulting rows are the positions of the corresponding columns in the CSV file. The result is given in calendar years (BC/AD).

Author(s)
Anne Philippe, \texttt{<Anne.Philippe@univ-nantes.fr>} and Marie-Anne Vibet, \texttt{<Marie-Anne.Vibet@univ-nantes.fr>}

References
Examples

```r
data(Events)
multi_hpd(Events, c(2, 4, 3), 0.95)
```

---

**multi_marginal_plot**  
*Marginal posterior densities of several events*

Description

Draws a plot of the estimated marginal posterior density for a parameter and adds the mean and the credible interval at the desired level

Usage

```r
multi_marginal_plot(
data,
position = 1:ncol(data),
level = 0.95,
grid_length = 1024,
x_scale = "calendar",
elapsed_origin_position = NULL,
title = "Characteristics of several dates",
subtitle = "Marginal densities",
caption = paste(level * 100, "% credible interval", sep = ""),
x_label = "Calendar year",
y_label = NULL,
density_fill = "gray30",
density_color = "black",
density_alpha = 1,
mean_color = "white",
mean_linetype = "dashed",
mean_size = 0.5,
ci_color = mean_color,
ci_linetype = "dotted",
ci_size = mean_size,
y_grid = TRUE,
x_min = NULL,
x_max = NULL,
height = 7,
width = 7,
units = "in",
file = NULL,
new_window = TRUE,
plot_result = TRUE,
fill_palette = NULL,
colors = NULL,
```
color_legend_name = "Legend"

Arguments

data Data frame containing the output of the MCMC algorithm.
position Numeric vector containing the position of the column corresponding to the MCMC chains of interest, or a vector of column names.
level Probability corresponding to the level of confidence.
grid_length Number of equally spaced points at which the density is to be estimated (for density() function).
x_scale One of "calendar" for calendar years, "BP" for years before present, or "elapsed" for time elapsed from a specified origin.
elapsed_origin_position Position of the column to use as the origin for elapsed time calculations.
title Title of the plot.
subtitle Subtitle of the plot.
caption Caption of the plot.
x_label Label of the x-axis.
y_label Label of the y-axis.
density_fill A color specification for the fill under the density line.
density_color A color specification for the density line.
density_alpha A number between 0 for transparent and 1 for opaque.
mean_color A color specification for the mean line.
mean_linetype A line type specification for the mean line.
mean_size A size specification for the mean line.
ci_color A color specification for the credible interval lines.
ci_linetype A line type specification for the credible interval lines.
ci_size A size specification of the credible interval lines.
y_grid Switch for horizontal grid lines.
x_min Minimum x-axis value.
x_max Maximum x-axis value.
height Plot height in units.
width Plot width in units.
units String recognized by the ggsave() function, one of "in", "cm", "mm".
file Name of the file that will be saved if specified, default = NULL.
new_window Whether or not the plot is drawn within a new window.
plot_result If TRUE, then draw a plot on the display, else suppress drawing.
fill_palette A vector of colors for qualitative data.
colors A vector of indices into palette keyed by position.
color_legend_name A label for the legend.
multi_marginal_statistics

Details

The density is estimated using density() function with n = grid_length. The input MCMC chains should either be in calendar years or converted to calendar years using x_scale vector or elapsed_origin_position.

Value

An archaeophases_plot object with the data and metadata needed to reproduce the plot.

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr>; Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>; and Thomas S. Dye, <tsd@tsdye.online>

Examples

data(Events);
multi_marginal_plot(Events, position = c(2, 3, 4), level = 0.95)

multi_marginal_statistics

Marginal summary statistics for multiple MCMC chains

Description

Calculates summary statistics of the output of the MCMC algorithm for multiple parameters. Results are given in calendar years (BC/AD).

Usage

multi_marginal_statistics(
  data,
  position = 1:ncol(data),
  level = 0.95,
  round_to = 0
)

Arguments

data
  Data frame containing the output of the MCMC algorithm.
position
  Numeric vector containing the positions of the columns corresponding to the MCMC chains of interest, or a vector of column names.
level
  Probability corresponding to the level of confidence used for the credible interval and the highest posterior density region.
round_to
  Integer indicating the number of decimal places.
Value

A data frame where the rows correspond to the chains of interest and columns to the following statistics:

- **mean**: The mean of the MCMC chain.
- **sd**: The standard deviation of the MCMC chain.
- **min**: Minimum value of the MCMC chain;
- **q1**: First quantile of the MCMC chain;
- **median**: Median of the MCMC chain;
- **q3**: Third quantile of the MCMC chain; and
- **max**: Maximum value of the MCMC chain.
- **ci.inf**: Lower credible interval of the MCMC chain at level.
- **ci.sup**: Upper credible interval of the MCMC chain at level.

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr>, Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>, and Thomas S. Dye, <tsd@tsdye.online>

Examples

```r
data(Events)
multi_marginal_statistics(Events, 2:5)
multi_marginal_statistics(Events, 2:5, level = 0.90)
## round to decades
multi_marginal_statistics(Events, 2:5, round_to = -1)
```

---

new_archaeophases_mcmc

*Constructor for archaeophases_mcmc object*

Description

Object to be returned by functions that read MCMC data from csv files.

Usage

```r
new_archaeophases_mcmc(x = list(), call = match.call(), hash = character())
```

Arguments

- **x**: A data frame with the data from the csv file.
- **call**: How the function was called.
- **hash**: A SHA256 hash of the csv file.
Details

The SHA256 hash should be secure against intentional and unintentional alterations of the MCMC csv file.

Value

An archaeophases_mcmc object that inherits from tbl_df.

Author(s)

Thomas S. Dye, <tsd@tsdye.online>

See Also

read_chronomodel
read_bcal
read_oxcal

---

new_archaeophases_plot

Constructor for archaeophases_plot object

Description

Objects returned by ArchaeoPhases plot functions.

Usage

new_archaeophases_plot(x = list(), mcmc = list(), call = match.call())

Arguments

x A data frame with the plot data.
mcmc An archaeophases_mcmc object.
call How the function was called.

Value

An archaeophases_plot object that inherits from archaeophases_mcmc.

Author(s)

Thomas S. Dye, <tsd@tsdye.online>
See Also

read_chronomodel
read_bcal
read_oxcal

OccurrencePlot

Plot occurrences

Description

A statistical graphic designed for the archaeological study of when events of a specified kind occurred

Usage

OccurrencePlot(
  data,
  position,
  plot.result = NULL,
  level = 0.95,
  intervals = "CI",
  title = "Occurrence plot",
  subtitle = NULL,
  caption = "ArchaeoPhases",
  labelXaxis = "Calendar year",
  labelYaxis = NULL,
  language = "English",
  occurrence = "occurrence",
  height = 7,
  width = 7,
  units = "in",
  x.min = NULL,
  x.max = NULL,
  x.scale = "calendar",
  elapsed.origin.position = NULL,
  dumbbell.size = 1,
  dot.guide = FALSE,
  dot.guide.size = 0.25,
  y.grid = FALSE,
  file = NULL,
  newWindow = TRUE,
  print.data.result = FALSE
)
Arguments

- **data**: Data frame containing the output of the MCMC algorithm.
- **position**: Numeric vector containing the position of the column corresponding to the MCMC chains of interest.
- **plot.result**: If TRUE, then draw a plot on the display, else suppress drawing.
- **level**: Probability corresponding to the level of confidence.
- **intervals**: One of "CI" for credible intervals or "HPD" for highest posterior density intervals.
- **title**: Title of the plot.
- **subtitle**: Subtitle of the plot.
- **caption**: Caption of the plot.
- **labelXaxis**: Label of the x-axis.
- **labelYaxis**: Label of the y-axis.
- **language**: String indicating a language recognized by the `toOrdinal` package.
- **occurrence**: String to append to each y-axis tic label.
- **height**: Plot height in units.
- **width**: Plot width in units.
- **units**: String recognized by the `ggsave()` function, one of "in", "cm", "mm".
- **x.min**: Minimum x-axis value.
- **x.max**: Maximum x-axis value.
- **x.scale**: One of "calendar" for calendar years, "BP" for years before present, or "elapsed" for time elapsed from a specified origin.
- **elapsed.origin.position**: Position of the column to use as the origin for elapsed time calculations.
- **dumbbell.size**: Size of the plot symbol.
- **dot.guide**: Switch for a horizontal guide from the y axis.
- **dot.guide.size**: Size of the dot guide.
- **y.grid**: Switch for horizontal grid lines.
- **file**: Name of the file that will be saved if specified. If NULL no plot will be saved.
- **newWindow**: Whether or not the plot is drawn within a new window.
- **print.data.result**: If TRUE, the list containing the data to plot will be returned.

Details

If we have k events, then we can estimate the calendar date t corresponding to the smallest date such that the number of events observed before t is equal to k. The `OccurrencePlot()` estimates these occurrences and gives the credible interval or the highest posterior density (HPD) region with a desired level of confidence.
Value

NULL, called for its side effects. It may also return a list containing the data to plot (if print.data.result = TRUE).

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr>,
Thomas S. Dye, <tsd@tsdye.online>, and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

data(Events);
OccurrencePlot(Events[1:1000, ], c(2:5), print.data.result = FALSE)

---

occurrence_plot

Plot occurrences

Description

A statistical graphic designed for the archaeological study of when events of a specified kind occurred.

Usage

occurrence_plot(
  data,
  position = 1:ncol(data),
  name = list("All"),
  level = 0.95,
  plot_result = TRUE,
  intervals = "CI",
  title = "Occurrence plot",
  subtitle = NULL,
  caption = "ArcheoPhases",
  x_label = "Calendar year",
  y_label = NULL,
  language = "English",
  occurrence = "occurrence",
  height = 7,
  width = 7,
  unit = "in",
  x_min = NULL,
  x_max = NULL,
  x_scale = "calendar",
)
elapsed_origin_position = NULL,
  dumbbell_size = 1,
  dot_guide = FALSE,
  dot_guide_size = 0.25,
  y_grid = FALSE,
  columns = 1,
  file = NULL,
  new_window = TRUE
)

**Arguments**

- **data**: Data frame containing the output of the MCMC algorithm.
- **position**: A list, each member of which is either a numeric vector containing the positions of the columns corresponding to the MCMC chains of interest, or a vector of column names. For convenience, a vector can be substituted for the singleton list.
- **name**: A list, each member of which is a string that names the kind of event in the corresponding element of position. For convenience, a string can be substituted for the singleton list.
- **level**: Probability corresponding to the level of confidence.
- **plot_result**: If TRUE, then draw a plot on the display, else suppress drawing.
- **intervals**: One of "CI" for credible intervals or "HPD" for highest posterior density intervals.
- **title**: Title of the plot.
- **subtitle**: Subtitle of the plot.
- **caption**: Caption of the plot.
- **x_label**: Label of the x-axis.
- **y_label**: Label of the y-axis.
- **language**: String indicating a language recognized by the **toOrdinal** package.
- **occurrence**: String to append to each y-axis tic label.
- **height**: Plot height in unit.
- **width**: Plot width in unit.
- **unit**: String recognized by the ggsave() function, one of "in", "cm", "mm".
- **x_min**: Minimum x-axis value.
- **x_max**: Maximum x-axis value.
- **x_scale**: One of "calendar" for calendar years, "BP" for years before present, or "elapsed" for time elapsed from a specified origin.
- **elapsed_origin_position**: Position of the column to use as the origin for elapsed time calculations.
- **dumbbell_size**: Size of the plot symbol.
- **dot_guide**: Switch for a horizontal guide from the y axis.
dot_guide_size  Size of the dot guide.
y_grid            Switch for horizontal grid lines.
columns           Number of columns for facet.
file              Name of the file that will be saved if specified. If NULL no plot will be saved.
new_window        Whether or not the plot is drawn within a new window.

Details

If we have k events, then we can estimate the calendar date t corresponding to the smallest date such that the number of events observed before t is equal to k. The `OccurrencePlot()` estimates these occurrences and gives the credible interval or the highest posterior density (HPD) region with a desired level of confidence.

Value

An `archaeophases_plot` object with the data and metadata needed to reproduce the plot.

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr>,
Thomas S. Dye, <tsd@tsdye.online>, and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

data(Events);
OccurrencePlot(Events[1:1000, ], c(2:5), print.data.result = FALSE)

## Not run:
# Read from connection
ox <- read_oxcal("http://tsdye.online/AP/ox.csv")
# Plot all the columns
op <- occurrence_plot(ox, position = 1:ncol(ox))
# Plot again
plot(op)
# View metadata
str(op)

## End(Not run)
original_file

Check for an original mcmc file

Description

Checks whether or not a file is identical to the one used to create an archaeophases_mcmc object.

Usage

original_file(x, ...)

Arguments

x

An archaeophases_mcmc object.

...  

Either a path to a CSV file, a connection, or the value clipboard() to read from the system clipboard. The CSV file can be compressed or plain.

Value

A boolean, TRUE if the files match, FALSE otherwise.

Author(s)

Thomas S. Dye, <tsd@tsdye.online>

Examples

## Not run:
rem <- read_chronomodel("http://tsdye.online/AP/cm/Chain_all_Events.csv")
original_file(rem, "http://tsdye.online/AP/cm/Chain_all_Events.csv")

## End(Not run)
Arguments

x     An archaeophases_mcmc object.
file  Either a path to a CSV file, a connection, or the value clipboard() to read from
      the system clipboard. The CSV file can be compressed or plain.
...  Other parameters.

Details

If called with a single argument, checks the file indicated by the file_path attribute.

Value

A boolean, TRUE if the files match, FALSE otherwise.

Author(s)

Thomas S. Dye, <tsd@tsdye.online>

original_file.archeophases_plot

Check for an original archaeophases_plot file

Description

Checks whether or not a file is identical to the one used to create an archaeophases_plot object.

Usage

## S3 method for class 'archeophases_plot'
original_file(x, file = NULL, ...)

Arguments

x     An archaeophases_plot object.
file  Either a path to a plot file, a connection, or the value clipboard() to read from
      the system clipboard.
...  Other parameters.

Details

If called with a single argument, checks the file indicated by the file_path attribute.

Value

A boolean, TRUE if the files match, FALSE otherwise.
Author(s)

Thomas S. Dye, <tsd@tsdye.online>

Description

A data set containing information on the ages of two events. See the vignette Reproductibility for more details.

Usage

oxc

Format

A data frame with 1000 rows and 2 variables:

- **foo-early** date oo-early
- **foo-late** date foo-late

PhaseDurationPlot

Plot the duration of a group

Description

This function draws the marginal posterior densities of the time elapsed between the minimum and the maximum of the dates included in a phase, and adds summary statistics (mean, CI).

Usage

```r
PhaseDurationPlot(
  PhaseMin_chain,
  PhaseMax_chain,
  level = 0.95,
  title = "Duration of a group of dates",
  colors = TRUE,
  exportFile = NULL,
  exportFormat = "PNG",
  GridLength = 1024
)
```
PhasePlot

Arguments

PhaseMin_chain Numeric vector containing the output of the MCMC algorithm for the minimum of the events included in the phase.

PhaseMax_chain Numeric vector containing the output of the MCMC algorithm for the maximum of the events included in the phase.

level Probability corresponding to the level of confidence used for the credible interval and the time range.

title Title of the plot.

colors If TRUE, use colors in the plot, otherwise produce a black and white plot.

exportFile Name of the file to be saved. If NULL, then no plot is saved.

exportFormat Format of the export file, either "PNG" or "SVG".

GridLength Length of the grid used to estimate the density.

Details

Plot of the density of the time elapsed between the minimum and the maximum calendar years of the events included in a phase, along with mean and credible interval.

Value

NULL, called for its side effects

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

data(Phases); attach(Phases)
PhaseDurationPlot(Phase.1.alpha, Phase.1.beta, 0.95, "Duration of Phase 1")
PhaseDurationPlot(Phase.2.alpha, Phase.2.beta, 0.95, "Duration of Phase 2", colors = FALSE)

Description

This function draws the marginal posterior densities of the minimum and the maximum of the events included in the phase and summary statistics including mean, credible interval, and time range. The result is given in calendar years (BC/AD).
Usage

PhasePlot(
  PhaseMin_chain,
  PhaseMax_chain,
  level = 0.95,
  title = "Characterisation of a group of dates",
  colors = TRUE,
  exportFile = NULL,
  exportFormat = "PNG",
  GridLength = 1024
)

Arguments

PhaseMin_chain  Numeric vector containing the output of the MCMC algorithm for the minimum of the events included in the phase.
PhaseMax_chain  Numeric vector containing the output of the MCMC algorithm for the maximum of the events included in the phase.
level           Probability corresponding to the level of confidence used for the credible interval and the time range.
title           The title of the plot
colors          If TRUE, then use of colors in the plot, otherwise draw the plot in black and white.
exportFile      Name of the file to be saved. If NULL, then no plot is saved.
exportFormat    Format of the export file, either "PNG" or "SVG".
GridLength      Length of the grid used to estimate the density.

Value

NULL, called for its side effects

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

data(Phases); attach(Phases)
PhasePlot(Phase.1.alpha, Phase.1.beta, level = 0.95, title = "Densities of Phase 1")
Phases

Description

A data set containing information on the start and end dates of two phases.

Usage

Phases

Format

A data frame with 30,000 rows and 5 variables:

- **iter**: iteration of the MCMC algorithm
- **Phase.2.alpha**: start date of Phase 2
- **Phase.2.beta**: end date of Phase 2
- **Phase.1.alpha**: start date of Phase 1
- **Phase.1.beta**: end date of Phase 1

PhasesGap

Gap or hiatus between two successive phases (for phases in temporal order constraint)

Description

This function finds, if it exists, a gap or hiatus between two successive phases. This gap or hiatus is the longest interval that satisfies:

\[ P(\text{Phase1Max}_{\text{chain}} < \text{IntervalInf} < \text{IntervalSup} < \text{Phase2Min}_{\text{chain}} | M) = \text{level} \]

Usage

PhasesGap(Phase1Max_chain, Phase2Min_chain, level = 0.95)

Arguments

- **Phase1Max_chain**: Numeric vector containing the output of the MCMC algorithm for the maximum of the events included in the oldest phase.
- **Phase2Min_chain**: Numeric vector containing the output of the MCMC algorithm for the minimum of the events included in the following phase.
- **level**: Probability corresponding to the level of confidence.
PhaseStatistics

Value

Returns a vector of values containing the level of confidence and the endpoints of the gap between the successive phases. The result is given in calendar years (BC/AD).

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

data(Phases); attach(Phases)
PhasesGap(Phase.1.beta, Phase.2.alpha, 0.95)
PhasesGap(Phase.1.beta, Phase.2.alpha, 0.50)

PhaseStatistics

Summary statistics of a phase

Description

Estimation of summary statistics, including the beginning and end of a phase, and the duration of the phase

Usage

PhaseStatistics(
  PhaseMin_chain,
  PhaseMax_chain,
  level = 0.95,
  roundingOfValue = 0
)

Arguments

PhaseMin_chain Numeric vector containing the output of the MCMC algorithm for the minimum of the dates included in the phase.
PhaseMax_chain Numeric vector containing the output of the MCMC algorithm for the maximum of the dates included in the phase.
level Probability corresponding to the level of confidence used for the credible interval and the highest density region.
roundingOfValue Integer indicating the number of decimal places.
Details

The summary statistics are those given by the `MarginalStatistics()` function. The time range is given by `PhaseTimeRange()` function. The duration is computed as follows: $duration = \text{maximum} - \text{minimum}$ at each iteration of the MCMC output.

Value

A matrix of values corresponding to the summary statistics:

1. Statistics of the minimum of the dates included in the phase
2. Statistics of the maximum of the dates included in the phase
3. Statistics of the duration of the dates included in the phase

The results are given in calendar year (in format BC/AD).

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples

```r
data(Phases); attach(Phases)
PhaseStatistics(Phase.1.alpha, Phase.1.beta, 0.95)
PhaseStatistics(Phase.2.alpha, Phase.2.beta, 0.95)
```

---

### PhasesTransition

**Transition range between two successive phases (for phases in temporal order constraint)**

**Description**

Finds, if it exists, the shortest interval that satisfies $P(\text{TransitionRangeInf} < \text{Phase1Max\_chain} < \text{Phase2Min\_chain} < \text{TransitionRangeSup}|M) = level$

**Usage**

```r
PhasesTransition(Phase1Max_chain, Phase2Min_chain, level = 0.95)
```

**Arguments**

- **Phase1Max_chain**
  Numeric vector containing the output of the MCMC algorithm for the maximum of the events included in the oldest phase.

- **Phase2Min_chain**
  Numeric vector containing the output of the MCMC algorithm for the minimum of the events included in the following phase.

- **level**
  Probability corresponding to the level of confidence.
**Value**

a vector of values containing the level of confidence and the endpoints of the transition interval between the successive phases. The result is given in calendar years (BC/AD).

**Author(s)**

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and

Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

**Examples**

data(Phases); attach(Phases)

PhasesTransition(Phase.1.beta, Phase.2.alpha, 0.95)

PhasesTransition(Phase.1.beta, Phase.2.alpha, 0.50)

**phases_gap**

*Gap or hiatus between two successive phases (for phases in temporal order constraint)*

**Description**

This function finds, if it exists, a gap or hiatus between two successive phases. This gap or hiatus is the longest interval that satisfies

\[
P(\text{Phase1Max\_chain} < \text{IntervalInf} < \text{IntervalSup} < \text{Phase2Min\_chain} | M) = \text{level}
\]

**Usage**

phases_gap(a_chain, b_chain, level = 0.95)

**Arguments**

- **a_chain**: Numeric vector containing the output of the MCMC algorithm for the upper boundary of the older phase.
- **b_chain**: Numeric vector containing the output of the MCMC algorithm for the lower boundary of the younger phase.
- **level**: Probability corresponding to the level of confidence.

**Value**

A list with the following components:

- **hiatus**: A named vector where inf is the lower endpoint of the hiatus as a calendar year (AD/BC) or NA if there is no hiatus at level, and sup is the upper endpoint of the gap as a calendar year (AD/BC), or NA if there is no hiatus at level.
- **level**: Probability corresponding to the confidence level of the interval.
- **call**: The function call.
Author(s)
Anne Philippe, <Anne.Philippe@univ-nantes.fr>,
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>, and
Thomas S. Dye, <tsd@tsdye.online>

Examples
data(Phases); attach(Phases)
phases_gap(Phase.1.beta, Phase.2.alpha, 0.95)
phases_gap(Phase.1.beta, Phase.2.alpha, 0.50)

---

PhaseTimeRange | Phase time range

Description
Computes the shortest interval that satisfies \( P(\text{PhaseMin\_chain} =< \text{IntervalInf} < \text{IntervalSup} =< \text{PhaseMax\_chain}|M) = \text{level} \)

Usage
PhaseTimeRange(PhaseMin\_chain, PhaseMax\_chain, level = 0.95)

Arguments
- **PhaseMin\_chain**: Numeric vector containing the output of the MCMC algorithm for the minimum of the events included in the phase.
- **PhaseMax\_chain**: Numeric vector containing the output of the MCMC algorithm for the maximum of the events included in the phase.
- **level**: Probability corresponding to the desired level of confidence.

Value
A vector of values containing the desired level of confidence and the endpoints of the shortest time range associated with this desired level. The result is given in calendar years (BC/AD).

Author(s)
Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples
data(Phases); attach(Phases)
PhaseTimeRange(Phase.1.alpha, Phase.1.beta, 0.95)
PhaseTimeRange(Phase.2.alpha, Phase.2.beta, 0.90)
**phase_statistics**

*Summary statistics of a phase*

**Description**

Estimation of summary statistics for the beginning, end, and duration of a phase.

**Usage**

```r
phase_statistics(min_chain, max_chain, level = 0.95, round_to = 0)
```

**Arguments**

- `min_chain` Numeric vector containing the output of the MCMC algorithm for the start of the phase.
- `max_chain` Numeric vector containing the output of the MCMC algorithm for the end of the phase.
- `level` Probability corresponding to the level of confidence used for the credible interval and the highest density region.
- `round_to` Integer indicating the number of decimal places.

**Details**

The summary statistics are those given by the `MarginalStatistics()` function. The time range is given by `PhaseTimeRange()` function. The duration is computed as follows: 

\[
\text{duration} = \text{maximum} - \text{minimum}
\]

at each iteration of the MCMC output.

**Value**

A list with the following components:

- `statistics` A data frame where the rows correspond to the summary statistics and the columns include: `start`, the start of the phase in calendar years (BC/AD); `end` the end of the phase in calendar years (BC/AD); and `duration` the duration of the phase in years.
- `level` Probability corresponding to the level of confidence used for the credible interval and the highest density region.
- `call` The function call.

**Author(s)**

Anne Philippe, <Anne.Philippe@univ-nantes.fr>, Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>, and Thomas S. Dye, <tsd@tsdye.online>
Examples

```r
data(Phases); attach(Phases)
phase_statistics(Phase.1.alpha, Phase.1.beta, 0.95)
phase_statistics(Phase.2.alpha, Phase.2.beta, 0.95)
## round to decade
phase_statistics(Phase.2.alpha, Phase.2.beta, 0.95, -1)
```

---

`plot.archaeophases_plot`

*Recreate a graphical plot*

Description

Recreates a graphic from data and metadata held in a `archaeophases_plot` object.

Usage

```r
## S3 method for class 'archaeophases_plot'
plot(x, ...)
```

Arguments

- `x` An `archaeophases_plot` object.
- `...` Other parameters.

Details

Uses data stored in the `archaeophases_plot` object, along with metadata from the call of the plotting function, to recreate the original graphic on the display.

Author(s)

Thomas S. Dye, <tsd@tsdye.online>

See Also

- `tempo_plot`
- `occurrence_plot`
- `marginal_plot`
- `multi_marginal_plot`
- `tempo_activity_plot`
- `multi_dates_plot`
Examples

```r
## Not run:
# Read from connection
ox <- read_oxcal("http://tsdye.online/AP/ox.csv")
tp_1 <- tempo_plot(ox, position = 1:ncol(ox))
# Recreate the tempo_plot with the original arguments
plot(tp_1)

## End(Not run)
```

---

**read_bcal**

*Read MCMC output from BCal*

**Description**

Import a CSV file containing the output of the MCMC algorithm produced by BCal.

**Usage**

```r
read_bcal(file, bin_width = 1, quiet = "no")
```

**Arguments**

- `file` Either a path to a CSV file, a connection, or the value `clipboard()` to read from the system clipboard. The CSV file can be compressed or plain. See `read_csv` for details.
- `bin_width` The bin width specified for the BCal calibration. Defaults to the BCal default of 1.
- `quiet` One of "no" (default) to allow messages and warnings, "partial" to suppress messages and allow warnings, or "yes" to suppress messages and warnings.

**Details**

The `read_bcal` function is built on `read_csv`. It aims to be fast and simple, and to return the marginal posteriors free of extraneous artifacts. The iteration column in the CSV file is discarded, as are an empty last column and an empty last row.

**Value**

An `archaeophases_mcmc` object containing the marginal posterior(s) as a data frame, or NULL if file is not found.

**Author(s)**

Thomas S. Dye, <tsd@tsdye.online>
### Description

Import a CSV file containing the output of the MCMC algorithm produced by `ChronoModel`.

### Usage

```r
read_chronomodel(file, decimal = ".", separator = ",", quiet = "no")
```

### Arguments

- **file**: Either a path to a CSV file, a connection, or the value `clipboard()` to read from the system clipboard. The CSV file can be compressed or plain. See `read_delim` for details.
- **decimal**: Either "." (default) or ",", the two choices offered by `ChronoModel`.
- **separator**: The character used to separate fields in the CSV file. Defaults to ",,.
- **quiet**: One of "no" (default) to allow messages and warnings, "partial" to suppress messages and allow warnings, or "yes" to suppress messages and warnings.

### Details

The `read_chronomodel` function is built on `read_delim`. It aims to be fast and simple, and to return the marginal posteriors free of extraneous artifacts. The iteration column in the CSV file is discarded.
Value
An **archaeophases_mcmc** object containing the marginal posterior(s) from file, or NULL if file is not found.

Author(s)
Thomas S. Dye, <tsd@tsdye.online>

See Also
- **read_delim**
- **ImportCSV**
- **new_archaeophases_mcmc**

Examples

```r
data(Events)
## Not run:
write.csv(Events, "events.csv", row.names=FALSE)
events = read_chronomodel("events.csv", decimal = ".", separator = ",")
# equivalent
events = read_chronomodel("events.csv")
rem <- read_chronomodel("http://tsdye.online/AP/cm/Chain_all_Events.csv")
## End(Not run)
```

---

**read_oxcal**

*Read MCMC output from OxCal*

Description
Import a CSV file containing the output of the MCMC algorithm produced by **OxCal**.

Usage
```
read_oxcal(file, quiet = "no")
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>file</td>
<td>Either a path to a CSV file, a connection, or the value <code>clipboard()</code> to read from the system clipboard. The CSV file can be compressed or plain. See <code>read_csv</code> for details.</td>
</tr>
<tr>
<td>quiet</td>
<td>One of &quot;no&quot; (default) to allow messages and warnings, &quot;partial&quot; to suppress messages and allow warnings, or &quot;yes&quot; to suppress messages and warnings.</td>
</tr>
</tbody>
</table>
Details

The `read_oxcal` function is built on `read_csv`. It aims to be fast and simple, and to return the marginal posteriors free of extraneous artifacts. The iteration column in the CSV file is discarded, as is an empty last column.

Value

An `archaeophases_mcmc` object containing the marginal posterior(s) as a data frame, or NULL if file is not found.

Author(s)

Thomas S. Dye, <tsd@tsdye.online>

See Also

`read_csv`

ImportCSV

Examples

```r
## Not run:
# Import of MCMC output from OxCal
data(Events)
#To do for saving in csv file
# write.csv(Events, "events.csv", row.names = FALSE)
fishpond <- read_oxcal("events.csv")

# Read from connection
oxc <- read_oxcal("http://tsdye.online/AP/ox.csv")

## End(Not run)
```

---

**reproduce**

Reproduce an MCMC data frame

Description

Reproduces a data frame from metadata held in an `archaeophases_mcmc` object.

Usage

`reproduce(x, ...)`
Arguments

x An archaeophases_mcmc object.
... Other parameters.

Author(s)

Thomas S. Dye, <tsd@tsdye.online>

reproduce.archaeophases_mcmc

*Reproduce an MCMC data frame*

Description

Reproduces a data frame from metadata held in an archaeophases_mcmc object. Returns NULL if file is not the original file.

Usage

```r
## S3 method for class 'archaeophases_mcmc'
reproduce(x, file = NULL, ...)
```

Arguments

x An archaeophases_mcmc object.
file A path to the original MCMC csv file, or a copy of the file.
... Other parameters.

Author(s)

Thomas S. Dye, <tsd@tsdye.online>

See Also

`original_file`

Examples

```r
## Not run:
x <- read_bcal("http://tsdye.online/AP/bc-1.csv")
y <- reproduce(x)
# TRUE
identical(x, y)

## End(Not run)
```
reproduce.archaeophases_plot

Reproduce an ArchaeoPhases plot

Description

Reproduces a plot from metadata held in an archaeophases_plot object. Returns NULL if file is not the original file.

Usage

## S3 method for class 'archaeophases_plot'
reproduce(x, file = NULL, ...)

Arguments

x
An archaeophases_plot object.

file
Path to the original MCMC csv file, or a copy of the file.

... Other parameters.

Author(s)

Thomas S. Dye, <tsd@tsdye.online>

See Also

original_file

Examples

## Not run:
ox <- read_bcal("http://tsdye.online/AP/bc-1.csv")
y <- multi_dates_plot(x)
z <- reproduce(y)
# TRUE
identical(y, z)

#ERROR, Not the original file.
z <- reproduce(y, file = "foo.csv")

## End(Not run)
SuccessionPlot

Description

Plot of the densities of the minimum and the maximum of the events included in each group, with summary statistics including the mean, credible interval, and highest posterior density. The result is given in calendar years (BC/AD).

Usage

SuccessionPlot(
  Phase1Min_chain,
  Phase1Max_chain,
  Phase2Min_chain,
  Phase2Max_chain,
  level = 0.95,
  title = "Characterisation of a succession of groups",
  exportFile = NULL,
  exportFormat = "PNG",
  GridLength = 1024
)

Arguments

Phase1Min_chain
  Numeric vector containing the output of the MCMC algorithm for the minimum of the events included in the oldest phase.

Phase1Max_chain
  Numeric vector containing the output of the MCMC algorithm for the maximum of the events included in the oldest phase.

Phase2Min_chain
  Numeric vector containing the output of the MCMC algorithm for the minimum of the events included in the youngest phase.

Phase2Max_chain
  Numeric vector containing the output of the MCMC algorithm for the maximum of the events included in the youngest phase.

level
  Probability corresponding to the level of confidence.

title
  Title of the plot.

exportFile
  Name of the file to be saved. If NULL then no plot is saved.

exportFormat
  Format of the export file, either "PNG" or "SVG".

GridLength
  Length of the grid used to estimate the density.
Details
Curves represent the density of the minimum (oldest event) and the maximum (youngest event) of the events included in each group. Curves of the same color refer to the same group. Time range intervals are symbolised by segments above the curves drawn using the same color as curves of the associated group. Transition and gap range intervals are represented by two-coloured segments using the colors of the both groups in succession. If the gap between the successive groups does not exist, a cross is drawn instead of a segment.

Value
NULL, called for its side effects

Author(s)
Anne Philippe, <Anne.Philippe@univ-nantes.fr> and Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

Examples
data(Phases); attach(Phases)
SuccessionPlot(Phase.1.alpha, Phase.1.beta, Phase.2.alpha, Phase.2.beta, level = 0.95)

Usage
TempoActivityPlot(
data,
position,
plot.result = NULL,
level = 0.95,
title = "Activity plot",
subtitle = NULL,
caption = "ArcheoPhases",
x.label = "Calendar year",
y.label = "Activity",
line.types = c("solid"),
width = 7,
height = 7,
)

Description
A statistical graphic designed for the archaeological study of rhythms of the long term that embodies a theory of archaeological evidence for the occurrence of events

Usage
TempoActivityPlot(
data,
position,
plot.result = NULL,
level = 0.95,
title = "Activity plot",
subtitle = NULL,
caption = "ArcheoPhases",
x.label = "Calendar year",
y.label = "Activity",
line.types = c("solid"),
width = 7,
height = 7,
)
TempoActivityPlot

units = "in",
x.min = NULL,
x.max = NULL,
file = NULL,
x.scale = "calendar",
elapsed.origin.position = NULL,
newWindow = TRUE,
print.data.result = FALSE
}

Arguments

data Data frame containing the output of the MCMC algorithm.
position Numeric vector containing the position of the column corresponding to the MCMC chains of interest.
plot.result List containing the data to plot, typically the result of a previous run of TempoActivityPlot().
level Probability corresponding to the level of confidence.
title Title of the plot.
subtitle Subtitle of the plot.
caption Caption of the plot.
x.label Label of the x-axis.
y.label Label of the y-axis.
line.types Type of the lines drawn on the plot.
width Width of the plot in units.
height Height of the plot in units.
units Units used to specify width and height, one of "in" (default), "cm", or "mm".
x.min Minimum value for x-axis.
x.max Maximum value for x-axis.
file Name of the file to be saved if specified. If NULL, then no file is saved.
x.scale One of "calendar", "bp", or "elapsed".
elapsed.origin.position If x.scale is "elapsed", the position of the column corresponding to the event from which elapsed time is calculated.
newWindow Whether or not the plot is drawn within a new window.
print.data.result If TRUE, the list containing the data to plot is returned.

Value

NULL, called for its side effects. It may also return a list containing the data to plot (if print.data.result = TRUE). The result is given in calendar years (BC/AD).
Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

References


Examples

```r
data(Events);
TempoActivityPlot(Events[, c(2:5), print.data.result = FALSE);
```

```
```

```
```

Description

A statistical graphic designed for the archaeological study of rhythms of the long term that embodies a theory of archaeological evidence for the occurrence of events

Usage

```r
TempoPlot(
  data,
  position,
  plot.result = NULL,
  level = 0.95,
  count = TRUE,
  Gauss = FALSE,
  title = "Tempo plot",
  subtitle = NULL,
  caption = "ArcheoPhases",
  legend.title = "Legend",
  legend.labels = c("Bayes estimate", "Credible interval, low", "Credible interval, high", "Gaussian approx., high", "Gaussian approx., low"),
  x.label = "Calendar year",
  y.label = "Cumulative events",
  line.types = c("solid", "12", "11", "28", "28"),
  width = 7,
  height = 7,
  units = "in",
  x.min = NULL,
  x.max = NULL,
```
colors = TRUE,
file = NULL,
x.scale = "calendar",
elapsed.origin.position = NULL,
newWindow = TRUE,
print.data.result = FALSE
)

Arguments

data Data frame containing the output of the MCMC algorithm.
position Numeric vector containing the position of the column corresponding to the MCMC chains of interest.
plot.result List containing the data to plot, typically the result of a previous run of TempoPlot() .
level Probability corresponding to the level of confidence.
count If TRUE the counting process is a number, otherwise it is a probability.
Gauss If TRUE, the Gaussian approximation of the credible interval is used.
title Title of the plot.
subtitle Subtitle of the plot.
caption Caption of the plot.
legend.title Title of the plot legend.
legend.labels Vector of strings to label legend entries.
x.label Label of the x-axis.
y.label Label of the y-axis.
line.types Type of the lines drawn on the plot in the order of legend.labels.
width Width of the plot in units.
height Height of the plot in units.
units Units used to specify width and height, one of "in" (default), "cm", or "mm".
x.min Minimum value for x-axis.
x.max Maximum value for x-axis.
colors If TRUE, the plot is drawn with colors, otherwise it is drawn in black and white.
file Name of the file that will be saved if specified. If NULL no file is saved.
x.scale One of "calendar", "bp", or "elapsed".
elapsed.origin.position If x.scale is "elapsed", the position of the column corresponding to the event from which elapsed time is calculated.
newWindow Whether or not the plot is drawn within a new window.
print.data.result If TRUE, a list containing the data to plot will be returned.
Details

The tempo plot is one way to measure change over time: it estimates the cumulative occurrence of archaeological events in a Bayesian calibration. The tempo plot yields a graphic where the slope of the plot directly reflects the pace of change: a period of rapid change yields a steep slope and a period of slow change yields a gentle slope. When there is no change, the plot is horizontal. When change is instantaneous, the plot is vertical.

Value

NULL, called for its side effects. It may also return a list containing the data to plot (if `print.data.result = TRUE`).

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr>, Thomas S. Dye, <tsd@tsdye.online>, and Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

References


See Also

tempo_plot

Examples

data(Events);
TempoPlot(Events[1:1000, ], c(2:5), print.data.result = FALSE)
TempoPlot(Events[1:1000, ], c(2:5), count = TRUE, print.data.result = FALSE)

tempo_activity_plot

Plot the derivative of the tempo plot Bayesian estimate

Description

A statistical graphic designed for the archaeological study of rhythms of the long term that embodies a theory of archaeological evidence for the occurrence of events
tempo_activity_plot

Usage

tempo_activity_plot(
  data,
  position = 1:ncol(data),
  title = "Tempo Activity Plot",
  subtitle = NULL,
  caption = "ArcheoPhases",
  x_label = "Calendar year",
  y_label = "Activity",
  line_types = c("solid"),
  width = 7,
  height = 7,
  units = "in",
  x_min = NULL,
  x_max = NULL,
  file = NULL,
  x_scale = "calendar",
  elapsed_origin_position = NULL,
  new_window = TRUE,
  plot_result = TRUE
)

Arguments

data          Data frame containing the output of the MCMC algorithm.
position       Numeric vector containing the position of the column corresponding to the
               MCMC chains of interest, or a vector of column names.
title          Title of the plot.
subtitle       Subtitle of the plot.
caption        Caption of the plot.
x_label        Label of the x-axis.
y_label        Label of the y-axis.
line_types     Type of the lines drawn on the plot.
width          Width of the plot in units.
height         Height of the plot in units.
units          Units used to specify width and height, one of "in" (default), "cm", or "mm".
x_min          Minimum value for x-axis.
x_max          Maximum value for x-axis.
file           Name of the file to be saved if specified. If NULL, then no file is saved.
x_scale        One of "calendar", "bp", or "elapsed".
elapsed_origin_position
               If x_scale is "elapsed", the position of the column corresponding to the event
               from which elapsed time is calculated.
new_window     Whether or not the plot is drawn within a new window.
plot_result    If TRUE, then draw a plot on the display, else suppress drawing.
Value
An archaeophases_plot object with the data and metadata needed to reproduce the plot.

Author(s)
Anne Philippe, <Anne.Philippe@univ-nantes.fr> and
Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>
Thomas S. Dye, <tsd@tsdye.online>

References

Examples
data(Events);
tempo_activity_plot(Events[1:1000, ], c(2:5))

tempo_plot  Tempo plot

Description
A statistical graphic designed for the archaeological study of rhythms of the long term that embodies a theory of archaeological evidence for the occurrence of events.

Usage
tempo_plot(
  data,
  position = 1:ncol(data),
  name = list("All"),
  level = 0.95,
  count = TRUE,
  Gauss = FALSE,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  legend_title = NULL,
  legend_position = "bottom",
  legend_labels = c("Bayes estimate", "Credible interval high", "Credible interval low"),
  x_label = "Calendar year",
  y_label = "Cumulative events",
  line_types = c("solid", "dotted", "dotted"),
)
```r
line_sizes = c(1.2, 0.8, 0.8),
line_colors = c("black", "grey50", "grey50"),
width = 7,
height = 7,
unit = "in",
x_min = NULL,
x_max = NULL,
color_palette = NULL,
file = NULL,
x_scale = "calendar",
elapsed_origin_position = NULL,
columns = 1,
new_window = TRUE,
plot_result = TRUE)
```

**Arguments**

- `data` Data frame or archaeophases_mcmc object containing the output of the MCMC algorithm.
- `position` A list, each member of which is either a numeric vector containing the positions of the columns corresponding to the MCMC chains of interest, or a vector of column names. For convenience, a vector can be substituted for the singleton list.
- `name` A list, each member of which is a string that names the kind of event in the corresponding element of `position`. For convenience, a string can be substituted for the singleton list.
- `level` Probability corresponding to the level of confidence.
- `count` If TRUE the counting process is a number, otherwise it is a probability.
- `Gauss` If TRUE, the Gaussian approximation of the credible interval is used.
- `title` Title of the plot.
- `subtitle` Subtitle of the plot.
- `caption` Caption of the plot.
- `legend_title` Title of the plot legend.
- `legend_position` One of "top", "bottom" (default), "left", "right".
- `legend_labels` Vector of three strings to label legend entries. The strings must be unique. The first string labels the central tendency and the second and third strings label the high and low spreads.
- `x_label` Label of the x-axis.
- `y_label` Label of the y-axis.
- `line_types` Type of the lines drawn on the plot in the order of `legend_labels`.
- `line_sizes` Width of the lines drawn on the plot in the order of `legend_labels`.
line_colors  Color names for the lines drawn on the plot in the order of legend_labels. If color_palette is NULL, then standard color names are expected, otherwise the color names are from the supplied color_palette.

width  Width of the plot in unit.

height  Height of the plot in unit.

unit  String recognized by the ggsave() function, one of "in" (default), "cm", or "mm".

x_min  Minimum value for x-axis.

x_max  Maximum value for x-axis.

color_palette  A palette that supplies the colors used in the plot.

file  Name of the file that will be saved if specified. If NULL no file is saved.

x_scale  One of "calendar" for calendar years, "BP" for years before present, or "elapsed" for time elapsed from a specified origin.

elapsed_origin_position  If x.scale is "elapsed", the position of the column corresponding to the event from which elapsed time is calculated.

columns  Number of columns for facet.

new_window  Whether or not the plot is drawn within a new window.

plot_result  If TRUE, then draw a plot on the display, else suppress drawing.

Details

The tempo plot is one way to measure change over time: it estimates the cumulative occurrence of archaeological events in a Bayesian calibration. The tempo plot yields a graphic where the slope of the plot directly reflects the pace of change: a period of rapid change yields a steep slope and a period of slow change yields a gentle slope. When there is no change, the plot is horizontal. When change is instantaneous, the plot is vertical.

Value

An archaeophases_plot object with the data and metadata needed to reproduce the plot.

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr>,

Thomas S. Dye, <tsd@tsdye.online>, and

Marie-Anne Vibet, <Marie-Anne.Vibet@univ-nantes.fr>

References

undated_sample

See Also

TempoPlot
new_archaeophases_plot

Examples

data(Events);
tempo_plot(Events[1:1000, ], c(2:5))
tempo_plot(Events[1:1000, ], c(2:5), count = TRUE)

## Not run:
# Read from connection
ox <- read_oxcal("http://tsdye.online/AP/ox.csv")
# Plot all the columns
tp <- tempo_plot(ox)
# Reproduce the tempo plot
plot(tp)
# View metadata
str(tp)
# Check that the MCMC data file hasn't changed
original_file(tp)

# Use a custom palette
library(khroma)
light <- colours("light")
tp <- tempo_plot(ox, color_palette = light(2),
        line_colors = c("light blue", "pale grey", "pale grey"))

## End(Not run)

undated_sample Predictive distribution of date

Description

Predictive distribution of date

Usage

undated_sample(data1, data2, level = 0.95)

Arguments

data1 Numeric vector containing the output of the MCMC algorithm for the begining of interval
data2 Numeric vector containing the output of the MCMC algorithm for the end of interval
level Probability corresponding to the desired level of confidence. @return A list with the following components:
Details

Simulate the sample from the predictive distribution of an undated sample in stratigraphic constraint between two dates. The input is an MCMC sample simulated from the joint posterior distribution of these dates.

Author(s)

Anne Philippe, <Anne.Philippe@univ-nantes.fr> and

Examples

data(Phases);
attach(Phases)
sample = undated_sample(Phase.1.alpha,Phase.1.beta)
# credible interval for the new date.
sample$credible
# time range interval
sample$timerange
# graphics = densities / IC / time range ggplot
sample$gr

valid_url Check if a resource can be located

Description

Function retrieved from https://stackoverflow.com/questions/52911812/check-if-url-exists-in-r

Usage

valid_url(url_in, t = 2)

Arguments

url_in A character string.

t Timeout in seconds.
Index

* datasets

Events, 32
oxc, 73
Phases, 76

AgeDepth, 4
allen_analyze, 5
allen_basic_relation_set, 6
allen_basic_relation_strings, 7
allen_composition, 7
allen_concurrent_relations, 8
allen_create_concurrent_vector, 8
allen_create_distinct_endpoint_vector, 9
allen_create_result_vector, 9
allen_ensure_set_vector, 10
allen_illustrate, 10
allen_is_result_vector, 12
allen_is_set_string, 12
allen_is_set_vector, 13
allen_joint_concurrency, 13
allen_lattice_x, 14
allen_lattice_y, 14
allen_observe, 15
allen_observe_frequency, 16
allen_plot, 17
allen_proportion_result, 18
allen_relate_intervals, 19
allen_relation, 20
allen_set_to_vector, 20
allen_six_value_set, 21
allen_string_to_set, 21
allen_string_to_vector, 22
allen_union, 22
allen_update_result, 23
analyze_allen_relations, 23
app_ArchaeoPhases, 24
ArchaeoPhases, 24
coda.mcmc, 24

composition_lookup_table, 25
CreateMinMaxGroup, 26
credible_interval, 27
CredibleInterval, 27
dates_hiatus, 29
DatesHiatus, 28
estimate_range, 30
Events, 32
illustrate_allen_relations, 32
ImportCSV, 33, 84–86
ImportCSV.BCal, 35, 35
is.url, 36
marginal_plot, 40, 82
marginal_statistics, 43
MarginalPlot, 37
MarginalProba, 38
MarginalStatistics, 39
mcmc, 25
mcmc.list, 25
multi_credible_interval, 56
multi_dates_plot, 57, 82
multi_hpd, 60
multi_marginal_plot, 61, 82
multi_marginal_statistics, 63
MultiCredibleInterval, 44
MultiDatesPlot, 45
MultiHPD, 47
MultiMarginalPlot, 48
MultiPhasePlot, 50
MultiPhasesGap, 51
MultiPhasesTransition, 52
MultiPhaseTimeRange, 54
MultiSuccessionPlot, 55

new_archaeophases_mcmc, 64, 84, 85
new_archaeophases_plot, 65, 99
occurrence_plot, 68, 82
OccurrencePlot, 66
original_file, 71, 87, 88
original_file.archaeophases_mcmc, 71
original_file.archaeophases_plot, 72
oxc, 73

phase_statistics, 81
PhaseDurationPlot, 73
PhasePlot, 74
Phases, 76
phases_gap, 79
PhasesGap, 76
PhaseStatistics, 77
PhasesTransition, 78
PhaseTimeRange, 80
plot.archaeophases_plot, 82

read_bcal, 65, 66, 83
read_chronomodel, 35, 65, 66, 84
read_csv, 83–86
read_delim, 84, 85
read_oxcal, 35, 65, 66, 85
reproduce, 86
reproduce.archaeophases_mcmc, 87
reproduce.archaeophases_plot, 88

SuccessionPlot, 89

tempo_activity_plot, 82, 94
tempo_plot, 82, 94, 96
TempoActivityPlot, 90
TempoPlot, 92, 99

undated_sample, 99
url.exists, 36

valid_url, 100