Package ‘ArchaeoPhases’

May 29, 2020

Type Package

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

Version 1.4.5

Date 2020-05-27

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 (see <doi:10.18637/jss.v093.c01> for a description). 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 <http://www.chronomodel.fr>), 'Oxcal' (see <https://c14.arch.ox.ac.uk/oxcal.html>) or 'BCal' (see <http://bcal.shef.ac.uk/>). The only requirement is to have a csv file containing a sample from the posterior distribution.

License GPL-3

Depends R (>= 2.10), coda, hdrcde

Imports stats, utils, graphics, grDevices, shiny, shinythemes, DT, readr, ggthemes, toOrdinal, ggplot2, ggalt

Suggests knitr, rmarkdown

VignetteBuilder knitr

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

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2020-05-29 15:10:02 UTC

R topics documented:

`app_ArchaeoPhases` *Run ArchaeoPhases shiny apps*

Description

A wrapper for `runApp` to start interactive shiny apps for the R package ArchaeoPhases.

Usage

```
app_ArchaeoPhases()
```

Arguments

none

Details

The ArchaeoPhases package provides a function from which a shiny app can be started: `app_ArchaeoPhases()`. The `app_ArchaeoPhases()` function is just a wrapper for `runApp`. Via the `...` argument further arguments can be directly passed to `runApp`. See `?shiny::runApp` for further details on valid arguments.

Author(s)

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

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

See Also

`runApp`

Examples

```
## Not run:
app_ArchaeoPhases()


## End(Not run)
```

`ArchaeoPhases` *ArchaeoPhases: Post-Processing of the Markov Chain Simulated by 'ChronoModel', 'Oxcal' or 'BCal'*

Description

This package provides a list of functions for the statistical analysis of archaeological phases. It is based on the post-processing of the Markov Chains whose stationary distribution is the posterior distribution of a series of dates. Such MCMC output can be simulated by different applications as for instance 'ChronoModel' (see <http://www.chronomodel.fr>), 'Oxcal' (see <https://c14.arch.ox.ac.uk/oxcal.html>) or 'BCal' (see <http://bcal.shef.ac.uk/>). The only requirement is to have a CSV file containing a sample from the posterior distribution.

ArchaeoPhases functions

The most important functions are TempoPlot that is "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", PhaseTimeRange or MultiPhaseTimeRange that calculate time intervals to characterize archaeological phases or periods, PhasesTransition or MultiPhasesTransition that calculate time intervals to characterize the transition between two successive phases or periods, PhasesGap or PhasesGap that are Testing procedures to check the presence of a gap between two successive phases or periods. A gap interval is estimated if we accept its existence. See the help files for these functions for examples. See the vignette for more complete documentation

---

coda.mcmc — *Create a* `mcmc.list` *object for CODA users.*

Description

This is a wrapper function that extracts parallel chains from the data given and coerces the output to a single `mcmc.list` object and diagnostic tools from "coda" can be used.

Usage

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

Arguments

`data`	dataframe containing the output of the MCMC algorithm
`numberChains`	number of parallel chains. Default = 1.
`iterationColumn`	Column corresponding to the iteration values. Default = NULL

Value

A `mcmc.list` object.

Author(s)

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

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

Examples

```
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` *Constructing the minimum and the maximum for a group of dates(phase)*

Description

Constructs a dataframe containing the output of the MCMC algorithm corresponding to the minimum and the maximum of a group of dates (phase)

Usage

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

Arguments

`data`	dataframe 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`	the name of the dataframe in which the current minimum and maximum should be added. Null by default.
`exportFile`	the name of the final file that will be saved if chosen. Null by default.

Value

A dataframe containing the minimum and the maximum of the group of dates included in the phase of interest. These values may be added to an already existing file "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")
```

`CredibleInterval` *Bayesian credible interval*

Description

Computes the shortest credible interval at the desired level.

Usage

```
CredibleInterval(a_chain, level = 0.95)
```

Arguments

`a_chain`	numeric vector containing the output of the MCMC algorithm for a one-parameter. The MCMC samples should be in calendar year (BC/AD).
`level`	probability corresponding to the level of confidence used for the credible interval

Details

A (100 * level) % credible intervalgives the shortest interval, whose posterior probability is equal to the desired level. This interval is approximated by constructing the shortest interval such that N*(1-level) elements of the sample are outside the interval.

Value

Returns a vector of values containing the level of confidence and the endpoints of the shortest credible interval. The result is 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

```
data(Events); attach(Events)

CredibleInterval(Event.1)
CredibleInterval(Event.12, 0.50)
```

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

Description

Finds if it exists a gap between two dates that is the longest interval that satisfies : P(a_chain < IntervalInf < IntervalSup < b_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 one-parameter (date) a.The MCMC samples should be in calendar year (BC/AD)
`b_chain`	numeric vector containing the output of the same MCMC algorithm for the second one-parameter (date) b. The MCMC samples should be in calendar year (BC/AD)
`level`	probability corresponding to the level of confidence used for the credible interval and the highest density region

Value

Returns the endpoints of the longest hiatus between two parameters. The result is 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

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

`Events` *Events*

Description

Contains the output of the MCMC algorithm for four events modelled by ChronoModel. The MCMC samples are in calendar year (BC/AD).

Usage

```
data(Events)
```

Format

A data frame with 30000 observations on the following 5 variables.

`iter` a numeric vector corresponding to iteration number

`Event.1` a numeric vector containing the output of the MCMC algorithm for the parameter Event 1

`Event.12` a numeric vector containing the output of the MCMC algorithm for the parameter Event 12

`Event.2` a numeric vector containing the output of the MCMC algorithm for the parameter Event 2

`Event.22` a numeric vector containing the output of the MCMC algorithm for the parameter Event 22

Author(s)

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

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

Examples

```
data(Events)
```

`Fishpond` *Polynesian fishpond site on the island of Oahu, Hawaii.*

Description

MCMC samples from the posterior distribution of dates estimated by BCal software according to a Polynesian fishpond site on the island of Oahu, Hawaii. The site contains two main layers and 5 radiocarbon dates. See the tutorial of BCal for the details of the modeling.

Usage

```
data("Fishpond")
```

Format

A data frame with 55965 observations on the following 12 variables contained in the outputs generated by BCal. The MCMC samples are in format cal BP (before 1950).

`Iteration` a vector corresponding to iteration number. Here it is a non numeric vector as the last value is "EOF".

`beta.2..Layer.II.` a numeric vector containing the output of the MCMC algorithm for the date of the end of Layer II.

`theta.5..Layer.II.` a numeric vector containing the output of the MCMC algorithm for the date theta.5.

`theta.4..Layer.II.` a numeric vector containing the output of the MCMC algorithm for the date theta.4.

`theta.3..Layer.II.` a numeric vector containing the output of the MCMC algorithm for the date theta.3.

`theta.2..Layer.II.` a numeric vector containing the output of the MCMC algorithm for the date theta.2.

`alpha.2..Layer.II.` a numeric vector containing the output of the MCMC algorithm for the date of the start of Layer II.

`beta.1..Layer.III.` a numeric vector containing the output of the MCMC algorithm for the date of the end of Layer III.

`theta.1..Layer.III.` a numeric vector containing the output of the MCMC algorithm for the date theta.1.

`alpha.1..Layer.III.` a numeric vector containing the output of the MCMC algorithm for the date of the start of Layer III.

`phi.1` a numeric vector containing the output of the MCMC algorithm for the floating parameter phi 1.

`X` vector of NA

Author(s)

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

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

Source

http://bcal.shef.ac.uk/tutorial/tutorial.html

References

BCal is an easy to use, on-line Bayesian radiocarbon calibration tool for interpreting radiocarbon data along with expert chronological information. It is hosted by the School of Mathematics and Statistics at the University of Sheffield. See http://bcal.shef.ac.uk/.

Examples

```
##load data
data(Fishpond)
# deleting the last row containing "NAs" only
Fishpond = Fishpond[-55965,]
# Note that these MCMC samples are in format cal BP (that is in year before 1950).
# In order to use the functions from ArhcaheoPhases, the MCMC samples have to be
# converted in date format BC/AD, for example, using the following code
Fishpond2 = Fishpond
  L = length(Fishpond)
  conv <- function(value, T0){
    T0 - value
  }
  for (i in 1:L){
    if( is.numeric(Fishpond[,i]) == TRUE){
      Fishpond2[,i] = sapply(Fishpond[,i], conv, 1950)
    }
  }
```

ImportCSV *Importing a CSV file containing the output of the MCMC algorithm*

Description

Use of the read.csv with the default values for CSV files extracted from ChronoModel software. For MCMC in a date format different from BC/AD, use the parameter referenceYear to convert the MCMC in BC/AD otherwise, the remaining functions of this package 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 the package 'ArchaeoPhases' will not work properly.

Usage

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

Arguments

`file`	the name of the CSV file containing the output of the MCMC algorithm
`dec`	the character used in the file for decimal points for the use of read.csv()
`sep`	the field separator character for the use of read.csv()
`comment.char`	a character vector of length one containing a single character or an empty string for the use of read.csv()
`header`	a logical value indicating whether the file contains the names of the variables as its first line.
`iterationColumn`	number of the column corresponding to the iteration values. Default = 1.
`referenceYear`	the year of reference for MCMC in date format other than BC/AD. Default value = NULL as ChronoModel and Oxacl export the MCMC samples in BC/AD format.
`rowToWithdraw`	the number of the row to be withdrawn, or "last" for the last row of the data frame. Default = NULL.
`bin.width`	the 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.

Value

Returns a dataframe containing a representation of the data in the file. MCMC should now be in date format BC/AD.

Author(s)

Anne Philippe <Anne.Philippe@univ-nantes.fr>, Thomas S. Dye <TSD@tsdye.com> and

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

Examples

```
 data(Events)
# write.csv(Events, "data.csv", row.names=FALSE)
#  data = ImportCSV("data.csv", dec = '.', sep=',', comment.char='#', header = TRUE,
#  iterationColumn = 1)

 # 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("fishpond_MCMC.csv", dec = '.', sep=',', header = TRUE,
#  iterationColum= 1, referenceYear = 1950, rowToWithdraw = "last")
```

ImportCSV.BCal *Importing a BCal csv file*

Description

Importing a csv file containing the output of the MCMC algorithm from the BCal software. MCMC generated by BCal software are in date format cal BP.

Usage

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

Arguments

`file`	the name of the CSV file containing the output of the MCMC algorithm
`bin.width`	the 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.

Value

Returns a data frame (data.frame) containing a representation of the data in the file. MCMC should now be in date format BC/AD.

Author(s)

Anne Philippe <Anne.Philippe@univ-nantes.fr>, Thomas S. Dye <TSD@tsdye.com> and

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

Examples

```
# 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)
# equivalent to
#Fishpond2 = ImportCSV("fishpond_MCMC.csv", dec = '.', sep=',',referenceYear = 1950,
#rowToWithdraw = "last", bin.width=1)
```

`KADatesChronoModel` *Ksar Akil chronology build with ChronoModel software. Dates part.*

Description

MCMC samples from the posterior distribution of dates included in the Ksar Akil chronology build with ChronoModel software.

Usage

```
data("KADatesChronoModel")
```

Format

A data frame with 30000 observations on the following 17 variables, one variable per AMS date and the MCMC iteration number. These observations come from 3 parallel chains. The MCMC samples are in calendar year (BC/AD).

- `iter` a numeric vector corresponding to iteration number
- `Layer.V` a numeric vector containing the output of the MCMC algorithm for the date of the shell extracted from layer V.
- `Layer.VI` a numeric vector containing the output of the MCMC algorithm for the date of the shell extracted from layer VI.
- `Layer.XI` a numeric vector containing the output of the MCMC algorithm for the date of the shell extracted from layer XI.
- `Layer.XII` a numeric vector containing the output of the MCMC algorithm for the date of the shell extracted from layer XII.
- `Layer.XVI.1` a numeric vector containing the output of the MCMC algorithm for the date of a shell extracted from layer XVI.
- `Layer.XVI.2` a numeric vector containing the output of the MCMC algorithm for the date of a shell extracted from layer XVI.
- `Layer.XVI.3` a numeric vector containing the output of the MCMC algorithm for the date of a shell extracted from layer XVI.
- `Layer.XVI.4` a numeric vector containing the output of the MCMC algorithm for the date of a shell extracted from layer XVI.
- `Layer.XVII.1` a numeric vector containing the output of the MCMC algorithm for the date of a shell extracted from layer XVII.
- `Layer.XVII.2` a numeric vector containing the output of the MCMC algorithm for the date of a shell extracted from layer XVII.
- `Layer.XVII.3` a numeric vector containing the output of the MCMC algorithm for the date of a shell extracted from layer XVII.
- `Layer.XVII.4` a numeric vector containing the output of the MCMC algorithm for the date of a shell extracted from layer XVII.

`Layer.XVIII` a numeric vector containing the output of the MCMC algorithm for the date of the shell extracted from layer XVIII.

`Layer.XIX` a numeric vector containing the output of the MCMC algorithm for the date of the shell extracted from layer XIX.

`Layer.XX` a numeric vector containing the output of the MCMC algorithm for the date of the shell extracted from layer XX.

`Layer.XXII` a numeric vector containing the output of the MCMC algorithm for the date of the shell extracted from layer XXII.

Author(s)

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

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

Source

Bosch, M. et al. (2015) New chronology for Ksar Akil (Lebanon) supports Levantine route of modern human dispersal into Europe. Proceedings of the National Academy of Sciences, 112, 7683–6.

References

Lanos, P. et al. (2016) Chronomodel : Chronological Modelling of Archaeological Data using Bayesian Statistics (Version 1.5). www.chronomodel.fr

Vibet. M.-A. et al. (2016) ChronoModel V1.5 User's manual.

Examples

```
##load data
data("KADatesChronoModel")
```

`KADatesOxcal` *Ksar Akil chronology build with Oxcal software*

Description

MCMC samples from the posterior distribution of dates included in the Ksar Akil chronology build with Oxcal software. See Bosch, M. et al. (2015) for more details. The MCMC samples are in calendar year (BC/AD).

Usage

```
data(KADatesOxcal)
```

Format

A data frame with 10000 observations on the following 28 variables contained in the outputs generated by Oxcal.

`Pass` a numeric vector corresponding to iteration number

`Ethelruda` a numeric vector containing the output of the MCMC algorithm for .

`start.dated.IUP` a numeric vector containing the output of the MCMC algorithm for the date of the start of phase IUP.

`GrA.53000` a numeric vector containing the output of the MCMC algorithm for the date GrA.53000.

`end.dated.IUP` a numeric vector containing the output of the MCMC algorithm for the date of the end of phase IUP.

`start.Ahmarian` a numeric vector containing the output of the MCMC algorithm for the date of the start of phase Ahmarian.

`GrA.57597` a numeric vector containing the output of the MCMC algorithm for the date GrA.57597.

`GrA.53004` a numeric vector containing the output of the MCMC algorithm for the date GrA.53004.

`GrA.57542` a numeric vector containing the output of the MCMC algorithm for the date GrA.57542.

`GrA.54846` a numeric vector containing the output of the MCMC algorithm for the date GrA.54846.

`GrA.57603` a numeric vector containing the output of the MCMC algorithm for the date GrA.57603.

`GrA.57602` a numeric vector containing the output of the MCMC algorithm for the date GrA.57602.

`GrA.53001` a numeric vector containing the output of the MCMC algorithm for the date GrA.53001.

`Egbert` a numeric vector containing the output of the MCMC algorithm for the date Egbert.

`GrA.54847` a numeric vector containing the output of the MCMC algorithm for the date GrA.54847.

`GrA.57599` a numeric vector containing the output of the MCMC algorithm for the date GrA.57599.

`GrA.57598` a numeric vector containing the output of the MCMC algorithm for the date GrA.57598.

`GrA.57544` a numeric vector containing the output of the MCMC algorithm for the date GrA.57544.

`end.Ahmarian` a numeric vector containing the output of the MCMC algorithm for the date of the end of phase Ahmarian.

`start.UP` a numeric vector containing the output of the MCMC algorithm for the date of the start of phase UP.

`GrA.57545` a numeric vector containing the output of the MCMC algorithm for the date GrA.57545.

`GrA.53006` a numeric vector containing the output of the MCMC algorithm for the date GrA.53006.

`GrA.54848` a numeric vector containing the output of the MCMC algorithm for the date GrA.54848.

`end.UP` a numeric vector containing the output of the MCMC algorithm for the date of the end of phase UP.

`start.EPI` a numeric vector containing the output of the MCMC algorithm for the date of the start of phase EPI.

`GrA.53005` a numeric vector containing the output of the MCMC algorithm for the date GrA.53005.

`end.EPI` a numeric vector containing the output of the MCMC algorithm for the date of the end of phase EPI.

Author(s)

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

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

Source

Bosch, M. et al. (2015) New chronology for Ksar Akil (Lebanon) supports Levantine route of modern human dispersal into Europe. Proceedings of the National Academy of Sciences, 112, 7683–6.

References

Bronk Ramsey, C. (2009) Bayesian analysis of radiocarbon dates. Radiocarbon, 51(1), 337-360.

Bronk Ramsey, C. (2016) Oxcal 4.2.

Examples

```
##load data
data(KADatesOxcal)
```

KAPhasesChronoModel *Ksar Akil chronology build with ChronoModel software. Phases part.*

Description

MCMC samples from the posterior distribution of dates included in the Ksar Akil chronology build with ChronoModel software. The MCMC samples are in calendar year (BC/AD).

Usage

```
data("KAPhasesChronoModel")
```

Format

A data frame with 30000 observations on the following 9 variables, two variables for each group of dates and the MCMC iteration number. These four phases are in stratigraphic order, the order from the oldest to the youngest phase being : IUP, Ahmarian, UP and EPI.These observations come from 3 parallel chains.

`iter` a numeric vector corresponding to iteration number

`EPI.alpha` a numeric vector containing the output of the MCMC algorithm for the minimum of the phase "EPI".

`EPI.beta` a numeric vector containing the output of the MCMC algorithm for the maximum of the phase "EPI".

`UP.alpha` a numeric vector containing the output of the MCMC algorithm for the the minimum of the phase "UP".

`UP.beta` a numeric vector containing the output of the MCMC algorithm for the maximum of the phase "UP".

`Ahmarian.alpha` a numeric vector containing the output of the MCMC algorithm for the the minimum of the phase "Ahmarian".

`Ahmarian.beta` a numeric vector containing the output of the MCMC algorithm for the maximum of the phase "Ahmarian".

`IUP.alpha` a numeric vector containing the output of the MCMC algorithm for the minimum of the phase "IUP".

`IUP.beta` a numeric vector containing the output of the MCMC algorithm for the maximum of the phase "IUP".

Author(s)

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

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

Source

Bosch, M. et al. (2015) New chronology for Ksar Akil (Lebanon) supports Levantine route of modern human dispersal into Europe. Proceedings of the National Academy of Sciences, 112, 7683–6.

References

Lanos, P. et al. (2016) Chronomodel : Chronological Modelling of Archaeological Data using Bayesian Statistics (Version 1.5). www.chronomodel.fr

Vibet. M.-A. et al. (2016) ChronoModel V1.5 User's manual.

Examples

```
##load data
data(KAPhasesChronoModel)
```

`MarginalPlot` *Plot of a marginal posterior density*

Description

This function draws the density of a one-parameter and adds summary statistics.

Usage

```
MarginalPlot(a_chain, level = 0.95, title = "Characteristics of a date",
  colors = TRUE, exportFile = NULL, exportFormat = "PNG", GridLength = 1024)
```

Arguments

`a_chain`	numeric vector containing the output of the MCMC algorithm for a one-parameter. The MCMC samples should be in calendar year (BC/AD).
`level`	probability corresponding to the level of confidence
`title`	label of the title
`colors`	if TRUE -> use of colors in the graph
`exportFile`	the name of the file to be saved. If NULL then no graph is saved.
`exportFormat`	the format of the export file : SVG or PNG.
`GridLength`	length of the grid used to estimate the density

Details

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

Value

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

Author(s)

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

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

Examples

```
data(Events); attach(Events)

MarginalPlot(Event.1, level = 0.95)
MarginalPlot(Event.1, level = 0.50)

MarginalPlot(Event.2, level = 0.95, title="Characteristics of Event 2")
MarginalPlot(Event.2, level = 0.95, colors = FALSE)
```

`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 one-parameter (date) a. The MCMC samples should be in calendar year (BC/AD).

`b_chain` numeric vector containing the output of the same MCMC algorithm for the second one-parameter (date) b. The MCMC samples should be in calendar year (BC/AD).

Details

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

Value

Returns the posterior probability of the following 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

```
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

Gives a list of summary statistics resulting from the output of the MCMC algorithm for a one-parameter. Results are given in calendar year (BC/AD).

Usage

```
MarginalStatistics(a_chain, level = 0.95)
```

Arguments

a_chain	numeric vector containing the output of the MCMC algorithm for a one-parameter. The MCMC samples should be in calendar year (BC/AD).
level	probability corresponding to the level of confidence used for the credible interval and the highest density region

Details

The 100*level % HPD (highest posterior density) region is estimated using HDR function from Package 'hdrcde'.

Value

A matrix of values corresponding to the following summary 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.

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>

References

Hyndman, R.J. (1996) Computing and graphing highest density regions. American Statistician, 50, 120-126.

Examples

```
data(Events); attach(Events)

MarginalStatistics(Event.1)
MarginalStatistics(Event.2, level = 0.90)
```

`MultiCredibleInterval` *Bayesian credible intervals for a series of dates*

Description

Estimation of the shorest credible interval for each variables of simulated Markov chain.

Usage

```
MultiCredibleInterval(data, position, level = 0.95)
```

Arguments

`data`	dataframe containing the output of the MCMC algorithm. The MCMC samples should be in calendar year (BC/AD).
`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 to estimate the credible interval

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 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

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

`MultiDatesPlot` *Plot of the endpoints of credible intervals or HPD intervals of a series of dates*

Description

Draws a plot of segments corresponding to the endpoints of the intervals (CI or HPD) of each selected date. The result is given in calendar year (in format BC/AD).

Usage

```
MultiDatesPlot(data, position, level = 0.95, 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, y.grid = FALSE,
                           file = NULL, newWindow=TRUE, print.data.result = FALSE)
```

Arguments

`data`	dataframe containing the output of the MCMC algorithm. The MCMC samples should be in calendar year (BC/AD).
`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 to estimate the credible interval
`intervals`	"CI" corresponds to the credible intervals, "HPD" to the highest density regions
`order`	"default" the y axis follows the order of the csv file. "increasing" the y axis is ordered according to the values of the interval (CI or the first HPD interval)
`title`	title of the graph
`subtitle`	subtitle of the graph
`caption`	caption of the graph
`labelXaxis`	x axis label of the graph
`labelYaxis`	y axis label of the graph
`height`	height of the graph in units
`width`	width of the graph in units

`units`	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`	the position of the column corresponding to the origin for elapsed time calculations
`dumbbell.size`	size of the symbols used to plot dates
`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`	the name of the file to be saved. If NULL then no graph 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 given

Author(s)

Anne Philippe <Anne.Philippe@univ-nantes.fr>, Thomas S. Dye <TSD@tsdye.com> 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 highest posterior density regions for a series of MCMC chains*

Description

Estimation of the highest posterior density regions for each variables of simulated Markov chain. This function uses the "hdr" function oincluded in the package "hdrcde. A HPD region may be a union of several intervals.

Usage

```
MultiHPD(data, position, level=0.95)
```

Arguments

- `data` — dataframe containing the output of the MCMC algorithm. The MCMC samples should be in calendar year (BC/AD).
- `position` — numeric vector containing the position of the column corresponding to the MCMC chains of interest
- `level` — probability corresponding to the level of confidence

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 year (in format BC/AD).

Author(s)

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

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

References

Hyndman, R.J. (1996) Computing and graphing highest density regions. American Statistician, 50, 120-126.

Examples

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

`MultiPhasePlot` *Plot of the marginal posterior densities of several groups*

Description

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 year (in format BC/AD).

Usage

```
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`	dataframe containing the output of the MCMC algorithm. The MCMC samples should be in calendar year (BC/AD).
`position_minimum`	numeric vector containing the column number corresponding to the minimum of the dates included in each group
`position_maximum`	numeric vector containing the column number corresponding to the maximum of the dates included in each group. By default, position_maximum = position_minimum + 1.
`level`	probability corresponding to the level of confidence
`title`	title of the graph
`colors`	numeric vector of colors for each group of dates
`exportFile`	the name of the file to be saved. If NULL then no graph is saved
`exportFormat`	the format of the export file : SVG or PNG

Value

Draws a plot with the marginal posterior densities of the minimum and the maximum of the dates included in each group and adds the time range of each group.

Author(s)

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

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

Examples

```
# 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))

# Export
# MultiPhasePlot(Phases, c(4,2), exportFile = "MultiPhasePlot", exportFormat = "PNG")
```

MultiPhasesGap *Gap/Hiatus between a succession of groups (for groups in temporal order constraint)*

Description

This function finds, if it exists, the gap between two successive groups. This gap or hiatus is the longest interval [IntervalInf, IntervalSup] that satisfies : P(Phase1Max < IntervalInf < IntervalSup < Phase2Min | M) = level for each successive group.

Usage

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

Arguments

`data`	dataframe containing the output of the MCMC algorithm. The MCMC samples should be in calendar year (BC/AD).
`position_minimum`	numeric vector containing the column number corresponding to the minimum of the dates included in each group
`position_maximum`	numeric vector containing the column number corresponding to the maximum of the dates included in each group. By default, position_maximum = position_minimum + 1.
`level`	probability corresponding to the level of confidence

Details

For each i, MultiPhasesGap computes the gap 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

Returns a matrix of values containing the level of confidence and the endpoints of the gap for each pair of successive groups. The result is 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

```
# Data extracted from ChronoModel software
data(Phases)

# List of the name of the groups
names(Phases)

# Stipulating position_maximum
MultiPhasesGap(Phases, position_minimum = c(4,2), position_maximum = c(5,3))

# In this case, equivalent to
MultiPhasesGap(Phases, position_minimum = c(4,2))
```

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

Description

Finds if it exists the shortest interval [TransitionRangeInf, TransitionRangeSup] that satisfies : P(TransitionRangeInf < Phase1Max < Phase2Min < TransitionRangeSup | M) = level for each phase

Usage

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

Arguments

- `data` — dataframe containing the output of the MCMC algorithm. The MCMC samples should be in calendar year (BC/AD).
- `position_minimum` — numeric vector containing the column number corresponding to the minimum of the dates included in each group
- `position_maximum` — numeric vector containing the column number corresponding to the maximum of the dates included in each group. By default, position_maximum = position_minimum + 1.
- `level` — probability corresponding to the level of confidence

Details

For each i, MultiPhasesTransition computes the transition interval for the group 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

Returns 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 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

```
# Data extracted from ChronoModel software
data(Phases)

# List of the name of the groups
names(Phases)

# Stipulating position_maximum
MultiPhasesTransition(Phases, c(4,2), c(5,3))

# In this case, equivalent to
MultiPhasesTransition(Phases, c(4,2))
```

MultiPhaseTimeRange *Phase Time Range for multiple groups*

Description

Computes the shortest interval that satisfies : P(PhaseMin < IntervalInf < IntervalSup < PhaseMax | M) = level

Usage

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

Arguments

`data` dataframe containing the output of the MCMC algorithm. The MCMC samples should be in calendar year (BC/AD).

`position_minimum` numeric vector containing the column number corresponding to the minimum of the dates included in each group

`position_maximum` numeric vector containing the column number corresponding to the maximum of the dates included in each group. By default, position_maximum = position_minimum + 1.

`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

Returns 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 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

```
# 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 groups in temporal order constraint)*

Description

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

Usage

```
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`	dataframe containing the output of the MCMC algorithm. The MCMC samples should be in calendar year (BC/AD).
`position_minimum`	numeric vector containing the column number corresponding to the minimum of the dates included in each group
`position_maximum`	numeric vector containing the column number corresponding to the maximum of the dates included in each group. By default, position_maximum = position_minimum + 1.
`level`	probability corresponding to the level of confidence
`title`	title of the graph
`colors`	numeric vector of colors for each group of dates
`exportFile`	the name of the file to be saved. If NULL then no graph is saved
`exportFormat`	the format of the export file : SVG or PNG.

Details

Curves represent the density of the minimum (oldest dates) and the maximum (youngest dates) of the dates included in each group. Curves of the same color refer to the same phase. When there is only one curve of one color, it means that there is only one event in the corresponding group and then the minimum equals the maximum. Time range intervals are symbolised by segments above the curves drawn using the same color as the one of the curves of the associated group. Transition and gap range intervals are represented by two-coloured segments using the colors of successive phases. If the gap between the successive groups does not exist, a cross is drawn instead of a segment.

Value

Returns a plot of all densities and adds several summary statistics. The result is 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>

References

Anne Philippe, Marie-Anne Vibet. (2020). Analysis of Archaeological Phases using the CRAN Package 'ArchaeoPhases', <doi:10.18637/jss.v093.c01>.

Examples

```
# Data extracted from ChronoModel software
data(Phases)

# List of the name of the groups
names(Phases)

# Stipulating position_end
MultiSuccessionPlot(Phases, c(4,2), c(5,3), title = "Succession of phase 1 and phase 2")

# In this case, equivalent to
MultiSuccessionPlot(Phases, c(4,2), title = "Succession of phase 1 and phase 2", colors = c(3,4))

# export
# MultiSuccessionPlot(Phases, c(4,2), exportFile = "MultiSuccessionPlot", exportFormat = "SVG")
```

OccurrencePlot *Plot of the occurence of events*

Description

A statistical graphic designed for the archaeological study of the timing of the occurrence of events.

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`	dataframe containing the output of the MCMC algorithm. The MCMC samples should be in calendar year (BC/AD).
`position`	numeric vector containing the position of the column corresponding to the MCMC chains of interest

`plot.result`	a list containing the data to plot, typically the result of a previous run of OccurrencePlot()
`level`	probability corresponding to the level of confidence used for the credible interval
`intervals`	"CI" corresponds to the credible intervals, "HPD" to the highest density regions
`title`	title of the graph
`subtitle`	subtitle of the graph
`caption`	caption of the graph
`labelXaxis`	x axis label of the graph
`labelYaxis`	y axis label of the graph
`language`	English by default
`occurrence`	Text for the y-values
`height`	height of the graph in units
`width`	width of the graph in units
`units`	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`	the position of the column corresponding to the origin for elapsed time calculations
`dumbbell.size`	size of the symbols used to plot dates
`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`	the name of the file to be saved. If NULL then no graph 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 given

Details

If we have k events. 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 theses dates and give the credible interval or the highest posterior density (HPD) region of theses dates associated to a desired level of confidence.

Value

It calculates the calendar date t corresponding to the smallest date such that the number of events observed before t is equal to k. The result is given in calendar year (in format BC/AD). 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.com> and Marie-Anne Vibet <Marie-Anne.Vibet@univ-nantes.fr>

References

Anne Philippe, Marie-Anne Vibet. (2020). Analysis of Archaeological Phases using the CRAN Package 'ArchaeoPhases', <doi:10.18637/jss.v093.c01>.

Examples

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

PhaseDurationPlot *Plot of 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

```
PhaseDurationPlot(PhaseMin_chain, PhaseMax_chain, level=0.95,
title = "Duration 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 dates included in the phase. The MCMC samples should be in calendar year (BC/AD).
PhaseMax_chain	numeric vector containing the output of the MCMC algorithm for the maximum of the dates included in the phase. The MCMC samples should be in calendar year (BC/AD).
level	probability corresponding to the level of confidence used for the credible interval and the time range
title	title of the graph
colors	if TRUE -> use of colors in the graph
exportFile	the name of the file to be saved. If NULL then no graph is saved.
exportFormat	the format of the export file : SVG or PNG.
GridLength	length of the grid used to estimate the density

Value

A plot with the marginal posterior densities of the duration of a phase and adds several summary statistics (mean, Credible interval, Time range). The result is given in year.

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)
```

PhasePlot — *Plot of the characteristics of a group of dates*

Description

This function draws the marginal posterior densities of the minimum and the maximum of the dates included in the phase

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 dates included in the phase. The MCMC samples should be in calendar year (BC/AD).
`PhaseMax_chain`	numeric vector containing the output of the MCMC algorithm for the maximum of the dates included in the phase. The MCMC samples should be in calendar year (BC/AD).
`level`	probability corresponding to the level of confidence used for the credible interval and the time range
`title`	title of the graph
`colors`	if TRUE -> use of colors in the graph
`exportFile`	the name of the file to be saved. If NULL then no graph is saved.
`exportFormat`	the format of the export file : SVG or PNG.
`GridLength`	length of the grid used to estimate the density

Value

A plot with the marginal posterior densities of the minimum and the maximum of the dates included in the phase and adds several summary statistics (mean, Credible interval, Time range). The result is 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>

References

Anne Philippe, Marie-Anne Vibet. (2020). Analysis of Archaeological Phases using the CRAN Package 'ArchaeoPhases', <doi:10.18637/jss.v093.c01>.

Examples

```
data(Phases); attach(Phases)

PhasePlot(Phase.1.alpha, Phase.1.beta, level = 0.95, title = "Densities of Phase 1")
#PhasePlot(Phase.2.alpha, Phase.2.beta, level = 0.95, title = "Densities of Phase 2",
#colors = FALSE, exportFile = "CharacteristicsOfPhase", exportFormat = "SVG")
```

Phases *Phases*

Description

Contains the output of the MCMC algorithm for all the phases (beginning and end) of two successive phases modelled in ChronoModel. Phase 1 is assued to be older than Phase 2. The MCMC samples are in calendar year (BC/AD).

Usage

```
data(Phases)
```

Format

A data frame with 30000 observations on the following 5 variables.

`iter` a numeric vector corresponding to iteration number

`Phase.1.alpha` a numeric vector containing the output of the MCMC algorithm for the beginning of the phase "Phase 1"

`Phase.1.beta` a numeric vector containing the output of the MCMC algorithm for the end of the phase "Phase 1"

`Phase.2.alpha` a numeric vector containing the output of the MCMC algorithm for the the beginning of the phase "Phase 2"

`Phase.2.beta` a numeric vector containing the output of the MCMC algorithm for the end of the phase "Phase 2"

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)
PhaseTimeRange(Phase.1.alpha, Phase.1.beta)

PhasesGap(Phase.1.beta, Phase.2.alpha)
PhasesTransition(Phase.1.beta, Phase.2.alpha)
```

PhasesGap — *Gap or Hiatus between two successive phases (for phases in temporal order constraint)*

Description

This function finds, if it exists, the gap between two successive phases. This gap or hiatus is the longest interval [IntervalInf ; IntervalSup] that satisfies : P(Phase1Max_chain < IntervalInf < IntervalSup < Phase2Min_chain | M) = 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 dates included in the oldest phase. The MCMC samples should be in calendar year (BC/AD).
`Phase2Min_chain`	numeric vector containing the output of the MCMC algorithm for the minimum of the dates included in the youngest phase. The MCMC samples should be in calendar year (BC/AD).
`level`	probability corresponding to the level of confidence

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 year (in format BC/AD).

Author(s)

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

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

References

Anne Philippe, Marie-Anne Vibet. (2020). Analysis of Archaeological Phases using the CRAN Package 'ArchaeoPhases', <doi:10.18637/jss.v093.c01>.

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 several summary statistics of the minimum, the maximum and the duration of the dates included in the phase.

Usage

```
PhaseStatistics(PhaseMin_chain, PhaseMax_chain, level = 0.95)
```

Arguments

PhaseMin_chain	numeric vector containing the output of the MCMC algorithm for the minimum of the dates included in the phase. The MCMC samples should be in calendar year (BC/AD).
PhaseMax_chain	numeric vector containing the output of the MCMC algorithm for the maximum of the dates included in the phase. The MCMC samples should be in calendar year (BC/AD).
level	probability corresponding to the level of confidence used for the credible interval and the highest density region

Details

The summary statistics are those given by MarginalStatistics function. The time range is given by PhaseTimeRange function. The duration is computed as follow duration = maximum - minimum at each iteration of the MCMC output.

Value

Returns a list 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

```
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 [TransitionRangeInf , TransitionRangeSup] that satisfies : P(TransitionRangeInf < Phase1Max_chain < Phase2Min_chain < TransitionRangeSup | M) = level

Usage

```
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 dates included in the oldest phase
- `Phase2Min_chain` numeric vector containing the output of the MCMC algorithm for the minimum of the dates included in the youngest phase
- `level` probability corresponding to the level of confidence

Value

Returns 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 year (in format BC/AD).

Author(s)

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

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

References

Anne Philippe, Marie-Anne Vibet. (2020). Analysis of Archaeological Phases using the CRAN Package 'ArchaeoPhases', <doi:10.18637/jss.v093.c01>.

Examples

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

PhaseTimeRange *Phase Time Range*

Description

Computes the shortest interval [IntervalInf ; IntervalSup] that satisfies : P(PhaseMin_chain =< IntervalInf < IntervalSup =< PhaseMax_chain | M) = 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 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 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 year (in format BC/AD).

Author(s)

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

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

References

Anne Philippe, Marie-Anne Vibet. (2020). Analysis of Archaeological Phases using the CRAN Package 'ArchaeoPhases', <doi:10.18637/jss.v093.c01>.

Examples

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

SuccessionPlot	*Density Plots of two successive groups (for groups in temporal order constraint)*

Description

Plot of the densities of the minimum and the maximum of the dates included in each phase and adds several summary statistics (mean, CI, HPDR). The result is given in calendar year (in format 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 dates included in the oldest phase. The MCMC samples should be in calendar year (BC/AD).

`Phase1Max_chain`
: numeric vector containing the output of the MCMC algorithm for the maximum of the dates included in the oldest phase. The MCMC samples should be in calendar year (BC/AD).

`Phase2Min_chain`
: numeric vector containing the output of the MCMC algorithm for the minimum of the dates included in the youngest phase. The MCMC samples should be in calendar year (BC/AD).

`Phase2Max_chain`
: numeric vector containing the output of the MCMC algorithm for the maximum of the dates included in the youngest phase. The MCMC samples should be in calendar year (BC/AD).

`level`	probability corresponding to the level of confidence
`title`	title of the graph
`exportFile`	the name of the file to be saved. If NULL then no graph is saved.
`exportFormat`	the format of the export file : SVG or PNG.
`GridLength`	length of the grid used to estimate the density

Details

Curves represent the density of the minimum (oldest dates) and the maximum (youngest dates) of the dates included in each phase. Curves of the same color refer to the same phase. Time range intervals are symbolised by segments above the curves drawn using the same color as the one of the curves of the associated phase. Transition and gap range intervals are represented by two-coloured segments using the colors of the both phases in succession. If the gap between the successive phases does not exist, a cross is drawn instead of a segment.

Value

Plot of the densities of the minimum and the maximum of the dates included in each phase. The result is 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

```
data(Phases); attach(Phases)
SuccessionPlot(Phase.1.alpha, Phase.1.beta, Phase.2.alpha, Phase.2.beta, level = 0.95)
# SuccessionPlot(Phase.1.alpha, Phase.1.beta, Phase.2.alpha, Phase.2.beta,
# exportFile = "Succession", exportFormat = "PNG")
```

`TempoActivityPlot` *Plot of the activity of events*

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, 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`	dataframe containing the output of the MCMC algorithm. The MCMC samples should be in calendar year (BC/AD).
`position`	numeric vector containing the position of the column corresponding to the MCMC chains of interest
`plot.result`	a list containing the data to plot, typically the result of a previous run of TempoActivityPlot()
`level`	probability corresponding to the level of confidence used for the credible interval
`title`	title of the graph
`subtitle`	subtitle of the graph
`caption`	caption of the graph
`x.label`	x axis label of the graph
`y.label`	y axis label of the graph
`line.types`	type of the lines drawn of the graph in the order of legend.labels
`height`	height of the graph in units
`width`	width of the graph in units
`units`	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`	the position of the column corresponding to the origin for elapsed time calculations
`file`	the name of the file to be saved. If NULL then no graph 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 given

Value

It is the derivative of the TempoPlot bayesian estimate. It may also return a list containing the data to plot (if print.data.result = TRUE). The result is given in calendar year (in format BC/AD).

Author(s)

Anne Philippe <Anne.Philippe@univ-nantes.fr>, Thomas S. Dye <TSD@tsdye.com> and

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

References

Dye, T.S. (2016) Long-term rhythms in the development of Hawaiian social stratification. Journal of Archaeological Science, 71, 1–9.

Examples

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

TempoPlot *Plot of the occurence of events*

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

```
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`	dataframe containing the output of the MCMC algorithm. The MCMC samples should be in calendar year (BC/AD).
`position`	numeric vector containing the position of the column corresponding to the MCMC chains of interest
`plot.result`	a list containing the data to plot, typically the result of a previous run of TempoPlot()
`level`	probability corresponding to the level of confidence used for the credible interval
`count`	if TRUE the counting process is given as a number, otherwise it is a probability
`Gauss`	if TRUE, the Gaussian approximation of the CI is used
`title`	title of the graph
`subtitle`	subtitle of the graph
`caption`	caption of the graph
`legend.title`	the title of the legend
`legend.labels`	a 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 of the graph 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", "cm", or "mm". Default = "in".
`x.min`	minimum value for x axis
`x.max`	maximum value for x axis
`colors`	if TRUE, the graph is drawn with colors, otherwise it is drawn in black and white
`file`	the name of the graph (+ extension) that will be saved if chosen. Null by default.
`x.scale`	one of "calendar", "bp", or "elapsed"
`elapsed.origin.position`	if x.scale is "elapsed", the position of the column corresponding to the occurrence from which elapsed time is calculated
`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 given. Default = TRUE.

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

It calculates the cumulative frequency of specified events by calculating how many events took place before each date in a specified range of dates. The result is given in calendar year (in format BC/AD). 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.com> and

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

References

Dye, T.S. (2016) Long-term rhythms in the development of Hawaiian social stratification. Journal of Archaeological Science, 71, 1–9.

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)
```

Index