Package: SCEM (via r-universe)

October 27, 2026

Type Package

Title Splitting-Coalescence-Estimation Method

Version 1.2.0

Maintainer Hannah Chazin <h.chazin@columbia.edu>

Description We introduce improved methods for statistically assessing birth seasonality and intra-annual variation. The first method we propose is a new idea that uses a nonparametric clustering procedure to group individuals with similar time series data and estimate birth seasonality based on the clusters. One can use the function SCEM() to implement this method. The second method estimates input parameters for use with a previously-developed parametric approach (Tornero et al., 2013). The relevant code for this approach is makeFits_OLS(), while makeFits_initial() is the code to implement the same method but with given initial conditions for two parameters. The latter can be used to show the disadvantage of the existing approach. One can use the function makeFits() to generate parametric birth seasonality estimates using either initialization. Detailed description can be found here: Chazin Hannah, Soudeep Deb, Joshua Falk, and Arun Srinivasan (2019) <doi:10.1111/arcm.12432> ``New Statistical Approaches to Intra-Individual Isotopic Analysis and Modeling Birth Seasonality in Studies of Herd Animals".

License GPL-3

URL https://github.com/hchazin/SCEM, https://hchazin.github.io/SCEM/

BugReports https://github.com/hchazin/SCEM/issues

Encoding UTF-8

LazyData true

RoxygenNote 8.0.0

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

VignetteBuilder knitr

Imports stats, mathjaxr

RdMacros mathjaxr

Depends R (>= 2.10)

Config/testthat/edition 3

Repository https://hchazin.r-universe.dev

Date/Publication 2026-06-25 23:21:06 UTC

RemoteUrl https://github.com/hchazin/scem

RemoteRef HEAD

RemoteSha c4a60886e226c933092a4c927cf0f8de0c769c32

Contents

armenia 2
calculateRSS 3
convertParameters 4
EBIC 5
EstTrend 6
iteration 7
kernel 8
makeFits 8
makeFits_initial 10
makeFits_OLS 11
SCalgo 12
SCEM 12
sine_initial 14
sine_OLS 15
sineFit 16

Index **18**

`armenia` *Armenia dataset*

Description

Archaeological faunal remains (24 sheep second molars) from Late Bronze Age (1500–1100 BCE) sites in the Tsaghkahovit Plain, Armenia

Usage

```
data("armenia")
```

Format

A data frame with 223 observations on the following 4 variables.

`ID` a numeric vector

`Subsample` a factor with levels `A B C D E F G H I J`

`distance` a numeric vector

`oxygen` a numeric vector

Source

H. Chazin, S. Deb, J Falk and A. SRINIVASAN

References

Chazin, Hannah, Soudeep Deb, Joshua Falk, and Arun Srinivasan. 2019. "New Statistical Approaches to Intra-Individual Isotopic Analysis and Modeling Birth Seasonality in Studies of Herd Animals." Archaeometry 61 (2): 478–93

Examples

```
data(armenia)
```

`calculateRSS` *Residual sum of squares (RSS) for all time series in a group.*

Description

SCEM uses the residual sum of squares for each group to give a sense of the error in estimation. It is defined by:

$$RSS(S_q) = \sum_{k \in S_q} \sum_{i=1}^{n_k} ||y_{k,i} - \hat{\mu}_{S_q}\left(\frac{i}{n_k}\right) - \hat{c}_k||^2$$

(See Chazin et al. 2019, Supplemental Materials 1).

The trend function for each individual time series is estimated non-parametrically by the local linear estimate (as discussed in Fan and Gijbels (1996)). Then, the common trend function for the group is estimated by taking the average over the group. Next, the shift functions are estimated as the differences from the individual trend functions and finally, the residual sum of squares are calculated using the original values, the common trend functions and the shifts.

Usage

```
calculateRSS(paths, S, bandwidth)
```

Arguments

paths	A list of data frames, where each frame contains the data for one individual. Every data frame should have two columns with names 'distance' and 'oxygen'.
S	A vector of integers showing which individuals are considered in the group.
bandwidth	Denotes the order of the bandwidth that should be used in the estimation process. bandwidth = k will mean that the bandwidth is n^k.

Value

A vector of length equal to the group-size, so that each element is the RSS for the corresponding individual in the group.

Examples

```
armenia_split = split(armenia,f = armenia$ID)
band = -0.33
p = length(armenia_split)
calculateRSS(armenia_split,1:p,band)
```

---

convertParameters *Parameter estimates from a nonlinear lest squares (NLS) fit.*

Description

This function converts the estimated parameters from the non-linear least squares (NLS) model fit to the appropriate parameter space corresponding to the cosine model proposed by Balasse et al (2012).

Usage

```
convertParameters(curve)
```

Arguments

curve	A fitted model object from nls function. The fitted model should have the following parameter estimates - amplitude, intercept, frequency, phase.

Value

A list containing the following components:

amplitude	estimated amplitude
intercept	estimated intercept
x0	delay of the data
X	period of the data
birth	birth seasonality estimate

Examples

```
armenia_split = split(armenia,f = armenia$ID)
curve = sineFit(armenia_split[[1]],method = "OLS")
convertParameters(curve)
```

EBIC *Bayesian Information Criterion (BIC) for a partition.*

Description

This function calculates an extended version of BIC, which is computed using a particular weighted average of the total residual sum of squares and the number of clusters.

SCEM uses the following equation for the BIC of each partition:

$$BIC(P) = (np)\log\left\{\frac{RSS(P)}{np}\right\} + |P|(B_n^{-1} - 1)\log(nB_n),$$

where $RSS(P) = \sum_{q=1}^{Q} RSS(S_q)$.

The sample size of each individual time series (i.e. the number of observations) is denoted by n, but in dealing with archaeological data, not all the time series in a data set will have the same number of observations.

In order to have a reasonable representative value for the sample size, we have chosen to use the natural arithmetic mean $n = (n_1 + \cdots + n_p)/p$.

$(B_n^{-1} - 1)\log(nB_n)$ is the tuning parameter that places the penalty on the number of clusters (also note that the term nB_n). Using a different tuning parameter γ_n in place of $(B_n^{-1} - 1)\log(nB_n)$ allows stronger or weaker penalties on the number of clusters.

Usage

```
EBIC(paths, partition, bandwidth)
```

Arguments

paths	A list of data frames, where each frame contains the data for one individual. Every data frame should have two columns with names 'distance' and 'oxygen'.
partition	A list of vectors. Each element in the list is a vector of integers, corresponding to individuals considered in one group.
bandwidth	Denotes the order of the bandwidth that should be used in the estimation process. bandwidth = k will mean that the bandwidth is n^k.

Value

Value of the extended BIC function for the partition.

Examples

```
armenia_split = split(armenia,f = armenia$ID)
band = -0.33
p = length(armenia_split)
EBIC(armenia_split,1:p,band)
```

`EstTrend` *Estimates the trend function for a time series.*

Description

The trend function for each individual time series is estimated non-parametrically by the local linear estimate (as discussed in Fan and Gijbels (1996)). Detailed description can be found in Chazin et al. 2019, Supplemental Materials 1.

Usage

```
EstTrend(y, time, bandwidth)
```

Arguments

`y`	A vector of time series observations.
`time`	A vector of time points where the value of the trend needs to be estimated.
`bandwidth`	Denotes the order of the bandwidth that should be used in the estimation process. bandwidth = k will mean that the bandwidth is n^k.

Value

A vector of estimated values for the trend function at the given time-points.

Examples

```
armenia_split = split(armenia,f = armenia$ID)
band = -0.33
z = armenia_split[[1]]$oxygen
n = length(z)
ndx = (1:n)/n
EstTrend(z,ndx,band)
```

`iteration` *Iteration step for the Splitting-Coalescence-Estimation Method (SCEM).*

Description

This function performs the iteration step. Detailed description can be found in Chazin et al. 2019, Supplemental Materials 1.

Usage

```
iteration(paths, U, bandwidth)
```

Arguments

`paths`	A list of data frames, where each frame contains the data for one individual. Every data frame should have two columns with names 'distance' and 'oxygen'.
`U`	A list of vectors. Each element in the list is a vector of integers, corresponding to individuals considered in one group.
`bandwidth`	Denotes the order of the bandwidth that should be used in the estimation process. bandwidth = k will mean that the bandwidth is n^k.

Value

A list containing the following components:

`S1`	A set of individuals who are in the cluster
`U`	A set of individuals to be used in the next iteration.

Examples

```
armenia_split = split(armenia,f = armenia$ID)
band = -0.33
p = length(armenia_split)
iteration(armenia_split,1:p,band)
```

kernel *Epanechnikov kernel*

Description

Calculates the value of the Epanechnikov kernel function for any vector.

Usage

```
kernel(v)
```

Arguments

v	A vector of real numbers.

Value

A vector of the calculated kernel values for the input vector.

References

Epanechnikov, V. A. (1969). Non-parametric estimation of a multivariate probability density. Theory of Probability and its Applications, 14(1), 153-6.

Examples

```
x = runif(10)
kernel(x)
```

makeFits *Prepare results for cosine model fit.*

Description

This function performs the nonlinear least squares (NLS) regression method for the cosine model. It fits the NLS method as required, and then computes different quantities for the birth seasonality estimates corresponding to different individuals.

Usage

```
makeFits(
  paths,
  amplitude = NULL,
  intercept = NULL,
  method = c("OLS", "initial")
)
```

Arguments

paths	A list of data frames, where each frame contains the data for one individual. Every data frame should have two columns with names 'distance' and 'oxygen'.
amplitude	Initial value for the amplitude parameter for the method="initial" method.
intercept	Initial value for the intercept parameter for the method="initial" method.
method	A character string giving the initialization for the nonlinear least squares regression. This must be either method="initial" or method="OLS". Default is method="OLS" method. method="initial" performs the nonlinear least squares (NLS) regression method for the cosine model without initializing parameter selections. It begins with the given initial values for amplitude and intercept. method="OLS" uses the least squares estimates (see Chazin et al. 2019) as the initial parameter selection.

Value

A data frame containing the following components:

amplitude	estimated amplitude
intercept	estimated intercept
x0	delay of the data
X	period of the data
birth	birth seasonality estimate
predictedMin	predicted minimum for the oxygen isotope variable
predictedMax	predicted maximum for the oxygen isotope variable
observedMin	observed minimum for the oxygen isotope variable
observedMax	observed minimum for the oxygen isotope variable
MSE	mean squared error corresponding to the model fit for every individual
Pearson	Pearson's R^2 corresponding to the model fit for every individual

Examples

```
armenia_split = split(armenia,f = armenia$ID)
amp = seq(1,10,by=0.5)
int = seq(-25,0,by=0.5)
makeFits(armenia_split,amp[1],int[1],method = "initial")
makeFits(armenia_split, method = "OLS")
```

`makeFits_initial` *Prepare results for cosine model fit with given initialization for two parameters.*

Description

Performs the nonlinear least squares (NLS) regression method for the cosine model, with the given initial values for amplitude and intercept. It fits the NLS method as required, and then computes different quantities for the birth seasonality estimates corresponding to different individuals.

Usage

```
makeFits_initial(paths, amplitude, intercept)
```

Arguments

`paths`	A list of data frames, where each frame contains the data for one individual. Every data frame should have two columns with names 'distance' and 'oxygen'.
`amplitude`	Initial value for the amplitude parameter.
`intercept`	Initial value for the intercept parameter.

Value

A data frame containing the following components:

`amplitude`	estimated amplitude
`intercept`	estimated intercept
`x0`	delay of the data
`X`	period of the data
`birth`	birth seasonality estimate
`predictedMin`	predicted minimum for the oxygen isotope variable
`predictedMax`	predicted maximum for the oxygen isotope variable
`observedMin`	observed minimum for the oxygen isotope variable
`observedMax`	observed minimum for the oxygen isotope variable
`MSE`	mean squared error corresponding to the model fit for every individual
`Pearson`	Pearson's R^2 corresponding to the model fit for every individual

Examples

```
armenia_split = split(armenia,f = armenia$ID)
amp = seq(1,10,by=0.5)
int = seq(-25,0,by=0.5)
makeFits_initial(armenia_split,amp[1],int[1])
```

makeFits_OLS *Prepare results for cosine model fit with proposed initialization.*

Description

Performs the nonlinear least squares (NLS) regression method for the cosine model, with the proposed initialization for all the parameters. It fits the NLS method as required, and then computes different quantities for the birth seasonality estimates corresponding to different individuals.

Usage

```
makeFits_OLS(paths)
```

Arguments

paths	A list of data frames, where each frame contains the data for one individual. Every data frame should have two columns with names 'distance' and 'oxygen'.

Value

A data frame containing the following components:

amplitude	estimated amplitude
intercept	estimated intercept
x0	delay of the data
X	period of the data
birth	birth seasonality estimate
predictedMin	predicted minimum for the oxygen isotope variable
predictedMax	predicted maximum for the oxygen isotope variable
observedMin	observed minimum for the oxygen isotope variable
observedMax	observed minimum for the oxygen isotope variable
MSE	mean squared error corresponding to the model fit for every individual
Pearson	Pearson's R^2 corresponding to the model fit for every individual

Examples

```
armenia_split = split(armenia,f = armenia$ID)
makeFits_OLS(armenia_split)
```

SCalgo *Splitting-Coalescence (SC) algorithm.*

Description

This function performs the iterative clustering algorithm on the archaeological time series data. Detailed description can be found in Chazin et al. 2019, Supplemental Materials 1.

Usage

```
SCalgo(paths, bandwidth)
```

Arguments

paths	A list of data frames, where each frame contains the data for one individual. There should be two columns with names 'distance' and 'oxygen'.
bandwidth	Denotes the order of the bandwidth that should be used in the splitting-coalescence (SC) clustering algorithm. A value k will mean that the bandwidth used in the algorithm is n^k.

Value

A list of vectors where each vector gives the indexes of the individuals to be assigned in the same cluster.

Examples

```
armenia_split = split(armenia,f = armenia$ID)
band = -0.33
results = SCalgo(armenia_split,bandwidth = band)
```

SCEM *Splitting-Coalescence-Estimation Method (SCEM) for archaeological time series.*

Description

This function performs the clustering algorithm SCEM on the bivariate time series data – where one series is the distance from the cementum-enamel junction, and the other series is the value of the oxygen-18 isotope at that distance. It returns the class assignments and birth seasonality estimates for all the individuals.

The SCEM assumes that the oxygen isotope values $(z(t))$ can be expressed as a function of $x(t)$, the natural logarithm of the distance from the CEJ for each of the incremental samples, scaled down by the period (X, the length of the tooth crown). In other words,

$$z(t) = f(x(t)/X) + e(t)$$

where the form of f is unknown and $e(t)$ is an error process. Also, following our definition, it assumes that the value of $x(t)/X$ that maximizes $z(t)$ is the estimated birth seasonality.

Birth seasonality is estimated using the combined data from all individuals in a single cluster, but birth seasonality estimates for individuals in a cluster are based on individual estimates of the length of the tooth crown (X_k).

For a detailed description of the algorithm, please see Chazin et al. 2019, Supplemental Materials 1.

Usage

```
SCEM(paths, bandwidth)
```

Arguments

`paths`	A list of data frames, where each frame contains the data for one individual. There should be two columns with names 'distance' and 'oxygen'.
`bandwidth`	Denotes the order of the bandwidth that should be used in the splitting-coalescence (SC) clustering algorithm. A value k will mean that the bandwidth used in the algorithm is n^k.

Value

A list containing the following components:

`results`	A data frame that has the individual information (ID, species, number of observations in the time series), cluster assignment, estimated period, delay and the birth seasonality estimate for every individual.
`groups`	The groups formed by the clustering algorithm

References

Chazin, Hannah, Soudeep Deb, Joshua Falk, and Arun Srinivasan. 2019. "New Statistical Approaches to Intra-Individual Isotopic Analysis and Modeling Birth Seasonality in Studies of Herd Animals." Archaeometry 61 (2): 478–93.

Examples

```
armenia_split = split(armenia,f = armenia$ID)
results = SCEM(armenia_split,bandwidth = -0.33)
```

`sine_initial` *Cosine model fitting with given initialization for two parameters.*

Description

Performs the updated nonlinear least squares (NLS) regression method for the cosine model proposed by Balasse et al. The method calculates with the proposed initial values for ampliitude and intercept, and then fits the NLS method as required.

Usage

```
sine_initial(data, amplitude, intercept)
```

Arguments

`data`	A data frame that contains the data for one individual. There should be two columns with names 'distance' and 'oxygen'.
`amplitude`	Initial value for the amplitude parameter.
`intercept`	Initial value for the intercept parameter.

Value

A fitted model object from the nls function in R:

`m`	an 'nlsModel' object incorporating the model.
`convInfo`	a list with convergence information
`data`	the expression that was passed to 'nls' as the data argument. The actual data values are present in the environment of the 'm' component.
`call`	the matched call with several components, notably 'algorithm'
`dataClasses`	the '"dataClasses"' attribute (if any) of the '"terms"' attribute of the model frame.
`control`	the control 'list' used

References

Florent Baty, Christian Ritz, Sandrine Charles, Martin Brutsche, Jean-Pierre Flandrois, Marie-Laure Delignette-Muller (2015). A Toolbox for Nonlinear Regression in R: The Package nlstools. Journal of Statistical Software, 66(5), 1-21. URL http://www.jstatsoft.org/v66/i05/.

Examples

```
armenia_split = split(armenia,f = armenia$ID)
amp = seq(1,10,by=0.5)
int = seq(-25,0,by=0.5)
sine_initial(armenia_split[[2]],amp[3],int[4])
```

`sine_OLS` *Cosine model fitting with proposed initialization.*

Description

Performs the updated nonlinear least squares (NLS) regression method for the cosine model (see Chazin et al. 2019).

Usage

```
sine_OLS(data)
```

Arguments

`data`	A data frame that contains the data for one individual. There should be two columns with names 'distance' and 'oxygen'.

Value

A fitted model object from the nls function in R:

`m`	an 'nlsModel' object incorporating the model.
`convInfo`	a list with convergence information
`data`	the expression that was passed to 'nls' as the data argument. The actual data values are present in the environment of the 'm' component.
`call`	the matched call with several components, notably 'algorithm'
`dataClasses`	the '"dataClasses"' attribute (if any) of the '"terms"' attribute of the model frame.
`control`	the control 'list' used

References

Florent Baty, Christian Ritz, Sandrine Charles, Martin Brutsche, Jean-Pierre Flandrois, Marie-Laure Delignette-Muller (2015). A Toolbox for Nonlinear Regression in R: The Package nlstools. Journal of Statistical Software, 66(5), 1-21. URL http://www.jstatsoft.org/v66/i05/.

Examples

```
armenia_split = split(armenia,f = armenia$ID)
sine_OLS(armenia_split[[1]])
```

`sineFit` *Cosine model fitting*

Description

This function performs the updated nonlinear least squares (NLS) regression method for the cosine model (see Chazin et al. 2019).

Usage

```
sineFit(data, amplitude = NULL, intercept = NULL, method = c("OLS", "initial"))
```

Arguments

`data`	A data frame that contains the data for one individual. There should be two columns with names 'distance' and 'oxygen'.
`amplitude`	Initial value for the amplitude parameter for the `method="initial"` method.
`intercept`	Initial value for the intercept parameter for the `method="initial"` method.
`method`	A character string giving the initialization for the nonlinear least squares regression. This must be either `method="initial"` or `method="OLS"`. Default is `method="OLS"` method. `method="initial"` performs the nonlinear least squares (NLS) regression method for the cosine model without initializing parameter selections. It begins with the given initial values for amplitude and intercept. `method="OLS"` uses the least squares estimates (see Chazin et al. 2019) as the initial parameter selection.

Value

A fitted model object from the nls function in R:

`m`	an 'nlsModel' object incorporating the model.
`convInfo`	a list with convergence information
`data`	the expression that was passed to 'nls' as the data argument. The actual data values are present in the environment of the 'm' component.
`call`	the matched call with several components, notably 'algorithm'
`dataClasses`	the '"dataClasses"' attribute (if any) of the '"terms"' attribute of the model frame.
`control`	the control 'list' used

References

Florent Baty, Christian Ritz, Sandrine Charles, Martin Brutsche, Jean-Pierre Flandrois, Marie-Laure Delignette-Muller (2015). A Toolbox for Nonlinear Regression in R: The Package nlstools. Journal of Statistical Software, 66(5), 1-21. URL http://www.jstatsoft.org/v66/i05/.

Examples

```
armenia_split = split(armenia,f = armenia$ID)
amp = seq(1,10,by=0.5)
int = seq(-25,0,by=0.5)
sineFit(armenia_split[[2]],amp[3],int[4],method = "initial")
sineFit(armenia_split[[1]],method = "OLS")
```

Index

∗ **datasets**
armenia, 2

armenia, 2

calculateRSS, 3
convertParameters, 4

EBIC, 5
EstTrend, 6

iteration, 7

kernel, 8

makeFits, 8
makeFits_initial, 10
makeFits_OLS, 11

SCalgo, 12
SCEM, 12
sine_initial, 14
sine_OLS, 15
sineFit, 16