Package: pheno (via r-universe)

October 31, 2024

Title Auxiliary Functions for Phenological Data Analysis

Version 1.7-0

Date 2022-05-12

Author Joerg Schaber

Description Provides some easy-to-use functions for time series
analyses of (plant-) phenological data sets. These functions
mainly deal with the estimation of combined phenological time
series and are usually wrappers for functions that are already
implemented in other R packages adapted to the special
structure of phenological data and the needs of phenologists.
Some date conversion functions to handle Julian dates are also
provided.

Maintainer Maximilian Lange <maximilian.lange@ufz.de>

Depends R (>= 4.0), nlme, SparseM, quantreg, methods, stats

License GPL (>= 2)

Repository CRAN

NeedsCompilation yes

Date/Publication 2022-05-12 21:40:02 UTC

Contents

---

connectedSets   *Connected sets in a matrix*

Description

Finds connected data sets, i.e. connected rows and columns of a numeric matrix M.

Usage

```
connectedSets(M)
```

Arguments

M   Numeric matrix with missing values assumed to be NA or 0.

Details

In a two-way classification of linear models sometimes independent sets of normal equations are obtained due to missing data in the experiments design, i.e. the complete design matrix is not of full rank and thus no solution can be found. However, solutions of the independent sets of normal equations can still exist. This phenomenon is called 'connectedness' of the data. Especially in phenological analysis experimental designs are almost always unbalanced because of missing data. Thus, when combined time series are to be estimated, it is worth checking for and finding connected data sets for which combined time series can then be estimated. Example (also see example data(Simple) and example in 'maxConnectedSet'): In the following matrix dots represent missing values, X represent observations and the lines join the connected sets:

: X___X . .
: |
: X___X . .
:
: . . X___X

Thus, in this matrix observations in rows 1 and 2 or colums 1 and 2 form one connected set. Likewise row 3 (or columns 3 and 4) form also one connected set.

Value

rowclasses	Vector of set numbers of rows of M (>=1). A value of -1 indicates a row with missing data.
colclasses	Vector of set numbers of columns of M(>=1). A value of -1 indicates a column with missing data.

Author(s)

Joerg Schaber

References

Searle (1997) 'Linear Models'. Wiley. page 318.

See Also

maxConnectedSet getConnectedSets

Examples

```
data(Simple)
connectedSets(Simple)
```

---

date2jul1	*Converts string date to Julian date*

Description

Converts a string date "DD.MM.YYYY" into a Julian day of year (DOY).

Usage

```
date2jul1(d)
```

Arguments

d	Date as charater string 'DD.MM.YYYY'.

Value

doy	Day of year as integer.
year	Year as integer.

Author(s)

Joerg Schaber

Examples

```
date2jul1('31.05.1970')
```

| date2jul2 | *Converts a date (day,month,year) to Julian date* |

Description

Converts an integer date (day,month,year) into a Julian day of year (DOY). If y is missing, 2000 is assumed.

Usage

```
date2jul2(d,m,y)
```

Arguments

d	Day of month, numeric coecerd into an integer.
m	Month of year, numeric coerced into an integer.
y	Year, numeric coerced into an integer, default 2000.

Value

doy	Day of year as integer.
year	Year as integer.

Author(s)

Joerg Schaber

Examples

```
date2jul2(31,5,1970)
```

| daylength | *Daylength at julian day i on latitude l* |

Description

Calculates daylength [h] and declination angle delta [radians] on day i [julian day of year] for latitude l [degrees].

Usage

```
daylength(i,l)
```

Arguments

i	Integer as julian day of year (1-365)
l	Float as latitude [degress]

Value

dl	daylength [h]
delta	declination angle [degrees]

Author(s)

Joerg Schaber

Examples

```
daylength(as.integer(120),63)
```

| daysbetween | *Number of days between two dates* |

Description

Number of days between date1 and date2.

Usage

```
daysbetween(d1,d2)
```

Arguments

d1	Date as a character string 'DD.MM.YYYY'.
d2	Date as s character string 'DD.MM.YYYY'.

Value

ndays   Number of days between d1 and d2.

Author(s)

Joerg Schaber

Examples

```
daysbetween('31.05.1970','10.03.2004')
```

DWD   *Phenological observations*

Description

Phenological observations of nine stations from 1951 to 1998. Data from the German Weather Service.

Usage

```
data(DWD)
```

Format

Data frame containing three columns (day of year of observations,year,station-id)

Source

German Weather Service

References

Schaber J, Badeck F-W (2002) 'Evaluation of methods for the combination of phenological time series and outlier detection'. Tree Physiology 22:973-982

---

getConnectedSets *Finds connected sets in a matrix or data frame*

Description

Finds a list of connected data sets in a matrix or data frame and returns them accordingly.

Usage

```
getConnectedSets(M)
```

Arguments

M   Numeric matrix with missing values considered as 0, or a data frame. The data
                     frame is internally converted to a matrix and should have three columns (x, factor
                     1, factor 2) where x are considered the entries of the matrix, rows correspond to
                     levels of factor 2 and columns correspond to levels of factor 1.

Details

getConnctedSets returns a list of connected data sets as numeric data frames D with three columns
(x, factor 1, factor 2) or a n*m matrix M, where the n rows correspond to n levels of factor 2 and
m columns correspond to m levels of factor the respective factors. Output as data frame or matrix,
depending on input.

Value

cs_i   List of connected sets as matrix or data frame, corresponding to the input. named
                     as cs_i with i being the number of the connected sets.

Author(s)

Joerg Schaber

References

Searle (1997) 'Linear Models'. Wiley. page 318.

See Also

[connectedSets](#) [maxConnectedSet](#)

Examples

```
data(Searle)
getConnectedSets(Searle)
```

---

jul2date1   *Converts Julian date into string date*

Description

Converts Julian day of year (DOY) into a string date 'DD.MM.YYYY'. If y is missing a non-leap year is assumed.

Usage

```
jul2date1(d,y)
```

Arguments

d	DOY, numeric coerced into an integer.
y	Year, numeric coerced into an integer, default 2000.

Value

date	Date, as character string 'DD.MM.YYYY'

Author(s)

Joerg Schaber

Examples

```
jul2date1(151,1970)
```

---

jul2date2   *Converts Julian date to integers day,month,year*

Description

Converts Julian day of year (DOY) into an integer date (day,month,year). If y is missing a non-leap year is assumed.

Usage

```
jul2date2(d,y)
```

Arguments

d	DOY, numeric coerced into an integer.
y	Year, numeric coerced into an integer, default 2000.

Value

day	Day of month as integer.
month	Month of year as integer.
year	Year as integer.

Author(s)

Joerg Schaber

Examples

```
jul2date2(151,1970)
```

---

leapyear	*Boolean test for leap year*

Description

Tests whether a given year is a leap year or not.

Usage

```
leapyear(y)
```

Arguments

y	Year, numeric coerced into integer.

Value

TRUE	leap year
FALSE	non leap year

Author(s)

Joerg Schaber

Examples

```
leapyear(2000)
leapyear(2004)
```

matrix2raw   *Converts numeric matrix to data frame*

Description

Converts a numeric matrix M into a dataframe D with three columns (x, factor 1, factor 2) where rows of M are ranks of factor 1 levels and columns of M are ranks of factor 2 levels, missing values are assumed to be 0 or NA. The resulting dataframe D has no missing values.

Usage

```
matrix2raw(M,l1,l2)
```

Arguments

M	Numeric matrix, with missing values assumed to be NA or 0.
l1	**Optional** numeric vector of level names of column 2 (factor 1) of returned data frame. If missing it is assigned row numbers of M.
l2	**Optional** numeric vector of level names of column 3 (factor 2) of returned data frame. If missing it is assigned column numbers of M.

Value

D	Data frame with three columns: (y,f1,f1). y: observations, i.e. non-zero entries, in matrix. f1: factor 1, i.e. row number of M or l1. f2: factor 2, i.e. column number of M or l2. D is ordered first by factor 2 and then factor 1. D has no missing values.

Author(s)

Joerg Schaber

Examples

```
data(DWD)
M <- raw2matrix(DWD)   # conversion to matrix
D1 <- matrix2raw(M)    # back conversion, but with different level names
D2 <- matrix2raw(M,c(1951:1998),c(1:9)) # with original level names
```

---

maxConnectedSet   *Maximal connected set in a matrix*

Description

Finds connected data set, i.e. connected rows and columns of a numeric matrix M, that has the largest number of data entries.

Usage

```
maxConnectedSet(M)
```

Arguments

M   Numeric matrix with missing values considered as 0, or a data frame. The data frame is internally converted to a matrix and should have three columns (x, factor 1, factor 2) where x are considered the entries of the matrix, rows correspond to levels of factor 2 and columns correspond to levels of factor 1.

Details

In a two-way classification of linear models sometimes independent sets of normal equations are obtained due to missing data in the experiments design, i.e. the complete design matrix is not of full rank and thus no solution can be found. However, solutions of the independent sets of normal equations can still exist. This phenomenon is called 'connectedness' of the data. Especially in phenological analysis experimental designs are almost always unbalanced because of missing data. Thus, when combined time series are to be estimated, it is worth checking for and finding connected data sets for which combined time series can then be estimated. This can also be interpreted in the way that a prerequisite to obtain a combined time series is to have overlapping time series. Example (also see example data(Searle) from Searle (1997), page 324 and example in 'connectedSets'): In the following matrix dots represent missing values, X represent observations and the lines join the connected sets:

```
: X\_\_\_.\_\_\_.\_\_\_.\_\_\_X . . .
: |
: . . X\_\_\_.\_\_\_!\_\_\_.\_\_\_.\_\_\_X
: ||
: . X\_\_\_.\_\_\_.\_\_\_!\_\_\_X\_\_\_X !
: |||
: . X\_\_\_.\_\_\_.\_\_\_!\_\_\_X\_\_\_X !
: ||
: . . . . X . . !
: ||
: . . X\_\_\_.\_\_\_!\_\_\_.\_\_\_.\_\_\_X
: |
: . . . X\_\_\_X . . .
```

Thus, in this matrix observations of rows 1, 5 and 7 or colums 1, 4 and 5 form one connected set. Likewise observations of rows 2 and 6 (or columns 3 and 8) and rows 3 and 4 (or columns 2, 6 and 7) form also connected sets, respectively.

Value

ms	maximal connected set as matrix or data frame, corresponding to the input.
maxl	Number of observations in the maximal connected data set.
nsets	Number of connected data sets.
lsets	Vector with number of observations in each connected data sets, i.e. lsets[i] is the number of observations in connected data set i.

Author(s)

Joerg Schaber

References

Searle (1997) 'Linear Models'. Wiley. page 318.

See Also

connectedSets maxConnectedSet

Examples

```
data(Searle)
maxConnectedSet(Searle)
```

---

maxdaylength   *Maximal day length on latitude l*

Description

Calculates maximal daylength maxdl [h] at a certain latitude l [degrees].

Usage

```
maxdaylength(l)
```

Arguments

l	Latitude in degrees.

Value

maxdl	Maximal daylength [h] at a certain latitude l [degrees]

Author(s)

Joerg Schaber

Examples

```
maxdaylength(60)
```

---

pheno.ddm   *Dense design matrix for phenological data*

Description

Creation of dense two-way classification design matrix. The sum of the second factor is constrained to be zero. No general mean.

Usage

```
pheno.ddm(D,na.omit=TRUE)
```

Arguments

D	Data frame with three columns: (observations, factor 1, factor 2).
na.omit	Determined whether missing values should be omitted or not. Default is TRUE.

Details

In phenological applications observations should be the julian day of observation of a certain phase, factor 1 should be the observation year and factor 2 should be a station-id. Usually this is much easier created by: y <- factor(f1),s <- factor(f2),ddm <- as.matrix.csr(model.matrix(~ y + s -1, contrasts=list(s=("contr.sum")))). However, this procedure can be quite memory demanding and might exceed storage capacity for large problems. This procedure here is much less memory comsuming. Moreover, in order to get direct estimates for all coefficients, an additional row is appended to the matrix, where the columns for the second factor are set to 1. Therefore, dimensions of ddm are (nlevels(factor1)+1)x(nlevels(factor2)).

Value

ddm	Dense roworder matrix, matrix.csr format (see matrix.csr in package SparseM)
D	Data frame D sorted first by f2 then by f1 and with rows containing NA's removed.
na.rows	Rows in D that were omitted due to missing values.

Author(s)

Joerg Schaber

See Also

model.matrix matrix.csr

Examples

```
data(DWD)
ddm1 <- pheno.ddm(DWD)
attach(DWD)
y <- factor(DWD[[2]])
s <- factor(DWD[[3]])
ddm2 <- as.matrix.csr(model.matrix(~ y + s -1, contrasts=list(s=("contr.sum"))))
identical(ddm1$ddm,ddm2)
```

---

pheno.flm.fit   *Fits a two-way linear fixed model*

Description

Fits a two-way linear fixed model. The model assumes the first factor f1 the second factor f2 to be fixed. Errors are assumed to be i.i.d. No general mean and sum of f2 is constrained to be zero.

Usage

```
pheno.flm.fit(D,limit=1000)
```

Arguments

D   Data frame with three columns (x, f1, f2) or a matrix where rows are ranks of factor f1 levels and columns are ranks of factor f2 levels and missing values are assumed to be NA or 0.

limit   Integer that determines which algorithm to use (see Details).

Details

This function is basically a wrapper for the slm.fit() function form the SparseM package, adapted for the estimation of combined phenological time series. In phenological application, x should be the julian day of observation of a certain phase, factor f1 should be the observation year and factor f2 should be a station-id. For large problems length(x)>limit, the linear model is calculated for treatment contrasts for efficiency reasons, and the constraint that the sum of f2 is zero, is adjusted afterwards. This results in a slight over-estimation of standard errors. Note that the input data is sorted before fitting, such that subsequent analyses using the input data should be done using the sorted output data frame.

Value

f1	Estimated fixed effects f1, in phenology this is precisely the combined time series.
f1.se	f1 estimated standard error.
f1.lev	Levels of f1. Should be the same order as f1.
f2	Estimated fixed effects f2, in phenology these are the station effects.
f2.se	f2 estimated standard error.
f2.lev	Levels of f2. Should be the same order as f2.
resid	Residuals
lclf1	Lower 95 percent confidence limit of factor f1.
uclf1	Upper 95 percent confidence limit of factor f1.
lclf2	Lower 95 percent confidence limit of factor f2.
uclf2	Upper 95 percent confidence limit of factor f2.
D	The input as ordered data frame, ordered first by f2 then by f1
fit	The fitted lm model object.

Author(s)

Joerg Schaber

References

Searle (1997) 'Linear Models'. Wiley. Schaber J, Badeck F-W (2002) 'Evaluation of methods for the combination of phenological time series and outlier detection'. Tree Physiology 22:973-982

See Also

[lm](#)

Examples

```
data(DWD)
R <- pheno.flm.fit(DWD)      # parameter estimation
```

---

pheno.lad.fit	*Fits a robust two-way linear model*

Description

Fits a robust two-way linear model. The model assumes both factors (f1 and f2) to be fixed. Errors are assumed to be i.i.d. No general mean and sum of f2 is constrained to be zero.

Usage

```
pheno.lad.fit(D,limit=1000)
```

Arguments

D   Data frame with three columns (x, f1, f2) or a matrix where rows are ranks of
                   factor f1 levels and columns are ranks of factor f2 levels and missing values are
                   assumed to be NA or 0.

limit   Integer that determines which algorithm to use (see Details).

Details

The function minimizes the least absolute deviations (LAD or L1 norm) of the residuals of a two-way linear model. This function is basically a wrapper for the `rq.fit()` or `rq.fit.sfn()` functions of the `quantreg` package, respectively, adapted for the estimation of combined phenological time series. Depending on the size of the problem `length(x)<=limit` either the `rq.fit()` function using the Barrodale-Roberts algorithm is used or (length(x)>1000) the corresponding dense matrix implementation with `rq.fit.sfn()` using the Interior-Point method. In phenological applications, x should be the julian day of observation of a certain phase, factor f1 should be the observation year and factor f2 should be a station-id. For efficiency reasons, the linear model is calcualted for treatment contrasts and the constraint that the sum of f2 is zero, is adjusted afterwards. Note that the input data is sorted before fitting, such that subsequent analyses using the input data should be done using the sorted output data frame.

Value

f1   Estimated parameters of factor f1, in phenology this is precisely the combined
                   time series.

f1.lev   Levels of f1. Should be the same order as f1.

f2   Estimated parameters of factor f2, in phenology these are precisely the station
                   effects.

f2.lev   Levels of f2. Should be the same order as f2.

resid   Residuals

ierr   For length(x) > 1000 this is the return error code of `rq.fit.sfn()`

D   The input as ordered data frame, ordered first by f2 then by f1

fit   The fitted rq.fit model object.

Author(s)

Joerg Schaber

References

Rousseeuw PJ, Leroy AM (1987) 'Robust estimation and outlier detection'. Wiley. Schaber J, Badeck F-W (2002) 'Evaluation of methods for the combination of phenological time series and outlier detection'. Tree Physiology 22:973-982

See Also

[rq.fit](rq.fit) [rq.fit.sfn](rq.fit.sfn)

Examples

```
data(DWD)
R <- pheno.lad.fit(DWD)    # robust parameter estimation
plot(levels(factor(R$D[[2]])),R$p1,type="l") # plot combined time series
R$D[R$resid >= 30,]    # observation whose residuals
       # are > 30 days (outliers)
```

| pheno.mlm.fit | *Fits a two-way linear mixed model* |

Description

Fits a two-way linear mixed model. The model assumes the first factor f1 to be fixed and the second factor f2 to be random. Errors are assumed to be i.i.d. No general mean and sum of f2 is constrained to be zero.

Usage

```
pheno.mlm.fit(D)
```

Arguments

D   Data frame with three columns (x, f1, f2) or a matrix where rows are ranks of factor f1 levels and columns are ranks of factor f2 levels and missing values are set to 0.

Details

This function is basically a wrapper for the `lme()` function of the `nlme` package, adapted for the estimation of combined phenological time series. Estimation method: restricted maximum likelihood (REML) In phenological application, x should be the julian day of observation of a certain phase, factor f1 should be the observation year and factor f2 should be a station-id. Note that the input data is sorted before fitting, such that subsequent analyses using the input data should be done using the sorted output data frame.

Value

fixed	Estimated fixed effects, in phenology this is precisely the combined time series.
fixed.lev	Levels of fixed effects. Should be the same order as fixed effects.
random	Estimated random effects, in phenology these are the station effects.
random.lev	Levels of random effects. Should be the same order as random effects.
SEf1	Standard error group f1, i.e. square root of variance component fixed effect.

SEf2	Standard error group f2, i.e. square root of variance component random effect.
lclf	Lower 95 percent confidence limit of fixed effects.
uclf	Upper 95 percent confidence limit of fixed effects.
D	The input as ordered data frame, ordered first by f2 then by f1
fit	The fitted lme model object.

Author(s)

Joerg Schaber

References

Searle (1997) 'Linear Models'. Wiley. Schaber J, Badeck F-W (2002) 'Evaluation of methods for the combination of phenological time series and outlier detection'. Tree Physiology 22:973-982

See Also

[lme](#)

Examples

```
data(DWD)
R <- pheno.mlm.fit(DWD)    # parameter estimation
plot(levels(factor(DWD[[2]])),R$fixed,type="l") # plot combined time series
tr <- lm(R$fixed~rank(levels(factor(DWD[[2]]))))# trend estimation
summary(tr)$coef[2]    # slope of trend
summary(tr)$coef[4]    # standard error of trend
```

---

raw2matrix   *Converts a numeric data frame to matrix*

Description

Converts a numeric data frame D with three columns (x, factor 1, factor 2) to a numeric matrix M where rows are ranks of levels of factor 1 and columns are ranks of levels of factor 2, missing values are set to NA.

Usage

```
raw2matrix(D)
```

Arguments

| D | Data frame with three columns (x, factor 1, factor 2) |

Value

M   Numeric matrix where rows are ranks of levels of factor 1 and columns are ranks of levels of factor 2, missing values are set to NA.

Author(s)

Joerg Schaber

Examples

```
data(DWD)
raw2matrix(DWD)
```

Searle   *Example of a two-way classification table*

Description

Example of a two-way classification table where lacking data creates three distinct connected sets.

Usage

```
data(Searle)
```

Format

R source file

References

Searle (1997) 'Linear Models'. Wiley. 324p.

---

seqMK   *Sequential Mann-Kendall test for time series.*

Description

The sequential Mann-Kendall test on time series x detects approximate potential trend turning points in time series.

Usage

```
seqMK(x)
```

Arguments

x   Numeric vector x.

Details

Implicitly assumes a equidistant time series x. Calculates a progressive and a retrograde series of Kendall normalized tau's. Points where the two lines cross are considered as approximate potential trend turning points. When either the progressive or retrograde row exceed certain confidence limits before and after the crossing points, this trend turning point is considered significant at the corresponding level, i.e. 1.96 for 95

Value

prog	Progressive row of Kendall's normalized tau's
retr	Retrograde row of Kendall's normalized tau's
tp	Boolean vector indicating at what indices of the original timeseries the prog and retr cross, i.e. TRUE at potential trend turning points.

Author(s)

Joerg Schaber

References

Kendall M, Gibbons JD (1990) 'Rank correlation methods'. Arnold. Sneyers R (1990) 'On statistical analysis of series of observations. Technical Note No 143. Geneva. Switzerland. World Meteorological Society. Schaber J (2003) 'Phenology in German in the 20th Century: Methods, analyses and models. Ph.D. Thesis. University of Potsdam. Germany. https://nbn-resolving.org/urn:nbn:de:kobv:517-0000532

Simple	*Simple example of a two-way classification table*

Description

Simple example of a two-way classification table where missing data creates two distinct connected sets.

Usage

```
data(Simple)
```

Format

R source file

---

tau   *Kendall's normalized tau*

Description

Kendall's normalized tau for time series x

Usage

```
tau(x)
```

Arguments

x   Numeric vector x.

Details

Implicitly assumes a equidistant time series x.

Value

t   Kendall's normalized tau.

Author(s)

Joerg Schaber

References

Kendall M, Gibbons JD (1990) 'Rank correlation methods'. Arnold. Sneyers R (1990) 'On statistical analysis of series of observations. Technical Note No 143. Geneva. Switzerland. World Meteorological Society. Schaber J (2003) 'Phenology in German in the 20th Century: Methods, analyses and models. Ph.D. Thesis. University of Potsdam. Germany. https://nbn-resolving.org/urn:nbn:de:kobv:517-0000532

Index