Package ‘vineyard’

January 28, 2022

Type Package

Title Bud Break, Phenological and Yield Models for Vineyards

Version 0.1.3

Author D. Molitor [aut]
J. Junk [aut]
J. A. Torres-Matallana [cre, aut]
U. Leopold [ctb]
Luxembourg Institute of Science and Technology (LIST) [cph]

Maintainer J. A. Torres-Matallana <arturo.torres@list.lu>

Description Late frosts are a significant risk to grape production in frost-prone viticultural regions. Increasing air temperature because of climate change is likely to advance grape bud break and last frost events in spring. So far, it is unclear whether one trend will be more pronounced than the other, and hence, whether the risk of late frost damage will increase or decrease. The aim of this package is to provide tools for investigating e.g. the future frost risk in winegrowing regions by assessing the effect of simulated future climate conditions on the timing of bud break and last frost date. Late frost risk can be assessed by the implementation of phenological models for bud break of the grapevine.

License What license is it under?

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports xts,
zoo

Suggests knitr,
rmarkdown,
sp,
spacetime

VignetteBuilder knitr

R topics documented:

`cdd.lThresh` *Compute the cumulative degree days by the single (lower) temperature threshold*

Description

Implementation to compute the cumulative degree days by the single (lower) temperature threshold by Molitor et al., (2014).

Usage

```
cdd.lThresh(data, t.mean.col, a)
```

Arguments

`data` input data in xts format.

`t.mean.col` numeric, column position in data for the daily mean air temperature vector in Celsius degrees.

`a` numeric, threshold temperature (in Celsius degrees) for vine growth.

Value

list per year for the input data plus an additional column with the cumulative degree days (in Celsius degrees) for vine growth. The output for each year is a "xts" time series object.

References

Daniel Molitor, Jürgen Junk, Danièle Evers, Lucien Hoffmann, and Marco Beyer (2014). A high-resolution cumulative degree day-based model to simulate phenological development of grapevine. Am. J. Enol. Vitic., (65:1):72–80.

`cdd.lThresh.phenology` *Cumulative degree days (cdd) by the single (lower) threshold algorithm for phenology*

Description

Implementation to compute the cumulative degree days (cdd) by the single (lower) threshold algorithm by Molitor et al. (2014) for phenology.

Usage

```
cdd.lThresh.phenology(cdd.lt, chs.mean)
```

Arguments

`cdd.lt` list, cumulative degree days (in Celsius degrees) for vine growth in xts format per year as provided by "cdd.1Treshold" function.

`chs.mean` numeric, mean cumulative heat sum for bud break (in Celsius degrees).

Value

the cumulative degree days (in Celsius degrees) for vine growth plus an additional column with the cumulative degree days (in Celsius degrees) for phenology.

References

Daniel Molitor, Jürgen Junk, Danièle Evers, Lucien Hoffmann, and Marco Beyer (2014). A high-resolution cumulative degree day-based model to simulate phenological development of grapevine. Am. J. Enol. Vitic., (65:1):72–80.

`cdd.luhThresh` *Compute cumulative degree days by the lower, upper and heat temperature thresholds*

Description

Implementation to compute cumulative degree days by the lower, upper and heat temperature thresholds by Molitor et al., (2014).

Usage

```
cdd.luhThresh(data, t.mean.col, a, b, c)
```

Arguments

`data` input data in xts format.

`t.mean.col` numeric, column position in data for the daily mean air temperature vector in Celsius degrees.

`a` numeric, lower threshold temperature (in Celsius degrees) for vine growth.

`b` numeric, upper threshold temperature (in Celsius degrees) for vine growth.

`c` numeric, heat threshold temperature (in Celsius degrees) for vine growth.

Value

list per year for the input data plus an additional column with the cumulative degree days (in Celsius degrees) for vine growth. The output for each year is a "xts" time series object.

References

Daniel Molitor, Jürgen Junk, Danièle Evers, Lucien Hoffmann, and Marco Beyer (2014). A high-resolution cumulative degree day-based model to simulate phenological development of grapevine. Am. J. Enol. Vitic., (65:1):72–80.

cdd.luhThresh.phenology

Cumulative degree days (CDD) by the lower, upper and heat temperature thresholds for phenology

Description

Implementation to compute the cumulative degree days by the lower, upper and heat temperature thresholds by Molitor et al. (2014) for phenology.

Usage

```
cdd.luhThresh.phenology(cdd.bb, cdd.luht, chs.mean)
```

Arguments

`cdd.bb`	cumulative degree days (CDD) by the single triangle algorithm for bud break in xts format as provided by "cdd.single.triangle.budbreak" function.
`cdd.luht`	cumulative degree days (in Celsius degrees) for vine growth in xts format as provided by "cdd.luhThresh" function.
`chs.mean`	numeric, mean cumulative heat sum for bud break (in Celsius degrees).

Value

the cumulative degree days (in Celsius degrees) for vine growth plus an additional column with the cumulative degree days (in Celsius degrees) for phenology.

References

Daniel Molitor, Jürgen Junk, Danièle Evers, Lucien Hoffmann, and Marco Beyer (2014). A high-resolution cumulative degree day-based model to simulate phenological development of grapevine. Am. J. Enol. Vitic., (65:1):72–80.

cdd.luThresh

Compute the cumulative degree days by the double (lower and upper) temperature threshold

Description

Implementation to compute the cumulative degree days by the double (lower and upper) temperature threshold by Molitor et al., (2014).

Usage

```
cdd.luThresh(data, t.mean.col, a, b)
```

Arguments

`data` input data in xts format.

`t.mean.col` numeric, column position in data for the daily mean air temperature vector in Celsius degrees.

`a` numeric, lower threshold temperature (in Celsius degrees) for vine growth.

`b` numeric, upper threshold temperature (in Celsius degrees) for vine growth.

Value

list per year for the input data plus an additional column with the cumulative degree days (in Celsius degrees) for vine growth. The output for each year is a "xts" time series object.

References

Daniel Molitor, Jürgen Junk, Danièle Evers, Lucien Hoffmann, and Marco Beyer (2014). A high-resolution cumulative degree day-based model to simulate phenological development of grapevine. Am. J. Enol. Vitic., (65:1):72–80.

`cdd.luThresh.phenology` *Cumulative degree days (CDD) by the double (lower and upper) temperature thresholds for phenology*

Description

Implementation to compute the cumulative degree days by the double (lower and upper) temperature thresholds by Molitor et al. (2014) for phenology.

Usage

```
cdd.luThresh.phenology(cdd.lut, chs.mean)
```

Arguments

`cdd.lut` list, cumulative degree days (in Celsius degrees) for vine growth in xts format per year as provided by the double threshold temperature ("cdd.double.threshold" function).

`chs.mean` numeric, mean cumulative heat sum for bud break (in Celsius degrees).

Value

the cumulative degree days (in Celsius degrees) for vine growth plus an additional column with the cumulative degree days (in Celsius degrees) for phenology.

References

Daniel Molitor, Jürgen Junk, Danièle Evers, Lucien Hoffmann, and Marco Beyer (2014). A high-resolution cumulative degree day-based model to simulate phenological development of grapevine. Am. J. Enol. Vitic., (65:1):72–80.

`cdd.simple.budbreak` *Cumulative degree days (CDD) by the simple algorithm*

Description

Implementation to compute the cumulative degree days by the simple algorithm by Nendel (2010).

Usage

```
cdd.simple.budbreak(data, t.mean.col, start.month, start.day)
```

Arguments

`data`	input data in xts format.
`t.mean.col`	numeric, column position in data for the daily mean air temperature vector in Celsius degrees.
`start.day`	

Value

list per year for the input data plus an additional column with the cumulative degree days (in Celsius degrees) for vine growth. The output for each year is a "xts" time series object.

References

Nendel, Class (2010). Grapevine bud break prediction for cool winter climates. Int. J. Biometeorol., 54, 231–241.

`cdd.simple.luhThresh.phenology` *Cumulative degree days (CDD) by the simple algorithm and lower, upper and heat temperature thresholds for phenology*

Description

Implementation to compute the cumulative degree days by the simple algorithm and the lower, upper and heat temperature thresholds by Molitor et al. (2014) for phenology.

Usage

```
cdd.simple.luhThresh.phenology(cdd.bb, cdd.luht, chs.mean)
```

Arguments

`cdd.bb` cumulative degree days (CDD) by the simple algorithm for bud break in xts format as provided by "cdd.simple.budbreak" function.

`cdd.luht` cumulative degree days (in Celsius degrees) for vine growth in xts format as provided by "cdd.luhThresh" function.

`chs.mean` numeric, mean cumulative heat sum for bud break (in Celsius degrees).

Value

the cumulative degree days (in Celsius degrees) for vine growth plus an additional column with the cumulative degree days (in Celsius degrees) for phenology.

References

Daniel Molitor, Jürgen Junk, Danièle Evers, Lucien Hoffmann, and Marco Beyer (2014). A high-resolution cumulative degree day-based model to simulate phenological development of grapevine. Am. J. Enol. Vitic., (65:1):72–80.

`cdd.single.triangle` *Cumulative degree days (CDD) by the single triangle algorithm*

Description

Implementation to compute the cumulative degree days by the single triangle algorithm by Nendel (2010).

Usage

```
cdd.single.triangle(data, t.zero, t.min.col, t.mean.col, t.max.col)
```

Arguments

`data` input data in xts format.

`t.zero` numeric, threshold temperature (in Celsius degrees) for vine growth.

`t.min.col` numeric, column position in data for the daily minimum air temperature vector in Celsius degrees.

`t.mean.col` numeric, column position in data for the daily mean air temperature vector in Celsius degrees.

`t.max.col` numeric, column position in data for the daily maximum air temperature vector in Celsius degrees.

Value

list per year for the input data plus an additional column with the cumulative degree days (in Celsius degrees) for vine growth. The output for each year is a "xts" time series object.

References

Nendel, Class (2010). Grapevine bud break prediction for cool winter climates. Int. J. Biometeorol., 54, 231–241.

`cdd.single.triangle.budbreak` *Cumulative degree days (CDD) by the single triangle algorithm for bud break*

Description

Implementation to compute the cumulative degree days by the single triangle algorithm by Nendel (2010) for bud break.

Usage

```
cdd.single.triangle.budbreak(cdd, start.month, start.day)
```

Arguments

`cdd`	cumulative degree days (in Celsius degrees) for vine growth in xts format as provided by "cdd.single.triangle" function.
`start.month`	numeric, calculated optimum starting month of year.
`start.day`	numeric, calculated optimum starting day of start.month.

Value

the cumulative degree days (in Celsius degrees) for vine growth plus an additional column with the cumulative degree days (in Celsius degrees) for bud break.

References

Nendel, Class (2010). Grapevine bud break prediction for cool winter climates. Int. J. Biometeorol., 54, 231–241.

compare.stage *Compare by growth stage from phenology output*

Description

Implementation to compare observations versus computations for growth stage from phenology output.

Usage

```
compare.stage(ref.data, phen, growth.stage)
```

Arguments

`ref.data`	data.frame, reference dataset to define the observations for the phenological stage to compare e.g. "data_remich_bbch09" or "data_remich_bbch81" datasets.
`phen`	list per year, with each list containing a data.frame with the phenological stages for vine growth as the output from the "phenology.stages" function.
`growth.stage`	numeric, one of the growth stages to compute the summary.

Value

data.frame, with each row containing a year with the comparison of the phenological stage computed and observed. The last colunm indicates the "Difference" between observed and computed day of year (DOY).

data_boundary_grevenmacher *A 'sp' object for the admininistrative boundaries*

Description

A dataset containing a 'sp' object for the admininistrative boundaries of the District of Grevenmacher in the Grand-Duchy of Luxembourg. It comprises the Commune and Canton levels as provided by the Luxembourgish Data Platform. The data is provided as a SpatialPolygonsDataFrame from "sp" package.

Usage

```
data(data_boundary_grevenmacher)
```

Format

```
Formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5 slots
..@ data :'data.frame': 23 obs. of 4 variables:
.. ..$ COMMUNE : Factor w/ 23 levels "Beaufort","Bech",..: 14 16 1 18 9 12 8 10 4 2 ...
.. ..$ CANTON : Factor w/ 3 levels "Echternach","Grevenmacher",..: 2 3 1 1 1 2 3 2 2 1 ...
.. ..$ DISTRICT: Factor w/ 1 level "Grevenmacher": 1 1 1 1 1 1 1 1 1 1 ...
.. ..$ LAU2 : Factor w/ 23 levels "1001","1002",..: 13 19 1 6 5 12 17 10 8 2 ...
..@ polygons :List of 23
.. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
.. .. .. ..@ Polygons :List of 1
.. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
.. .. .. .. .. .. ..@ labpt : num [1:2] 97715 87463
.. .. .. .. .. .. ..@ area : num 27853866
.. .. .. .. .. .. ..@ hole : logi FALSE
.. .. .. .. .. .. ..@ ringDir: int 1
#' .. .. .. .. .. .. ..@ coords : num [1:1611, 1:2] 96023 96027 96037 96070 96078 ...
.. .. .. ..@ plotOrder: int 1
.. .. .. ..@ labpt : num [1:2] 97715 87463
.. .. .. ..@ ID : chr "0"
.. .. .. ..@ area : num 27853866
.. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
.. .. .. ..@ Polygons :List of 1
.. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
.. .. .. .. .. .. ..@ labpt : num [1:2] 88254 64603
.. .. .. .. .. .. ..@ area : num 13591512
.. .. .. .. .. .. ..@ hole : logi FALSE
.. .. .. .. .. .. ..@ ringDir: int 1
.. .. .. .. .. .. ..@ coords : num [1:1012, 1:2] 89679 89677 89674 89673 89673 ...
.. .. .. ..@ plotOrder: int 1
.. .. .. ..@ labpt : num [1:2] 88254 64603
.. .. .. ..@ ID : chr "1"
.. .. .. ..@ area : num 13591512
.. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
.. .. .. ..@ Polygons :List of 1
.. .. .. ..[list output truncated]
```

Details

The spatial representation for the data corresponds to a SpatialPolygonsDataFrame with coordinate reference system (CRS) EPSG:2169 - Luxembourg 1930 / Gauss - Projected.

Source

https://data.public.lu/fr/datasets/limites-administratives-du-grand-duche-de-luxembourg/

Examples

```
data(data_boundary_grevenmacher)
str(data_boundary_grevenmacher)
```

```
#' plot stfdf object
plot(data_boundary_grevenmacher)
```

`data_lm_botrytis_riesling` *Botrytis cinerea model for Riesling cultivar*

Description

A sample dataset containing the Linear regression model calculated by Molitor et al. (2020) for predicting the CDD_7;18;24 reaching 5% disease severity (Botrytis cinerea) for Riesling cultivar.

Usage

```
data(data_lm_botrytis_riesling)
```

Format

```
List of 12
$ coefficients : Named num [1:6] 951.47 7.22 -16.32 10.79 21.99 ...
..- attr(*, "names")= chr [1:6] "(Intercept)" "a" "b" "c" ...
$ residuals : Named num [1:21] 30.9 43.8 38.7 -52.6 -55.9 ...
..- attr(*, "names")= chr [1:21] "1" "2" "3" "4" ...
$ effects : Named num [1:21] -4372.2 249.2 -175.3 134.1 -77.4 ...
..- attr(*, "names")= chr [1:21] "(Intercept)" "a" "b" "c" ...
$ rank : int 6
$ fitted.values: Named num [1:21] 1055 972 1066 834 941 ...
..- attr(*, "names")= chr [1:21] "1" "2" "3" "4" ...
$ assign : int [1:6] 0 1 2 3 4 5
$ qr :List of 5
..- attr(*, "class")= chr "qr"
$ df.residual : int 15
$ xlevels : Named list()
$ call : language lm(formula = observed ~ a + b + c + d + e, data = lr.data)
$ terms :Classes ’terms’, ’formula’ language observed ~ a + b + c + d + e
.. ..- attr(*, "variables")= language list(observed, a, b, c, d, e)
.. ..- attr(*, "factors")= int [1:6, 1:5] 0 1 0 0 0 0 0 0 1 0 ...
.. .. ..- attr(*, "dimnames")=List of 2
.. ..- attr(*, "term.labels")= chr [1:5] "a" "b" "c" "d" ...
.. ..- attr(*, "order")= int [1:5] 1 1 1 1 1
.. ..- attr(*, "intercept")= int 1
.. ..- attr(*, "response")= int 1
.. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
.. ..- attr(*, "predvars")= language list(observed, a, b, c, d, e)
.. ..- attr(*, "dataClasses")= Named chr [1:6] "numeric" "numeric" "numeric" "numeric" ...
.. .. ..- attr(*, "names")= chr [1:6] "observed" "a" "b" "c" ...
$ model :'data.frame': 21 obs. of 6 variables:
..- attr(*, "terms")=Classes 'terms', 'formula' language observed ~ a + b + c + d + e
.. .. ..- attr(*, "variables")= language list(observed, a, b, c, d, e)
.. .. ..- attr(*, "factors")= int [1:6, 1:5] 0 1 0 0 0 0 0 0 1 0 ...
.. .. .. ..- attr(*, "dimnames")=List of 2
.. .. ..- attr(*, "term.labels")= chr [1:5] "a" "b" "c" "d" ...
.. .. ..- attr(*, "order")= int [1:5] 1 1 1 1 1
.. .. ..- attr(*, "intercept")= int 1
.. .. ..- attr(*, "response")= int 1
.. .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
.. .. ..- attr(*, "predvars")= language list(observed, a, b, c, d, e)
.. .. ..- attr(*, "dataClasses")= Named chr [1:6] "numeric" "numeric" "numeric" "numeric" ...
.. .. .. ..- attr(*, "names")= chr [1:6] "observed" "a" "b" "c" ...
- attr(*, "class")= chr "lm"
```

References

Daniel Molitor, Ottmar Baus, Yoanne Didry, Jürgen Junk, Lucien Hoffmann and Marco Beyer (2020). BotRisk: simulating the annual bunch rot risk on grapevines (Vitis vinifera L. cv. Riesling) based on meteorological data. International Journal of Biometeorology. DOI 10.1007/s00484-020-01938-5

`data_lm_yield_riesling` *Yield model for Riesling cultivar*

Description

A sample dataset containing the Linear regression model calculated by Daniel Molitor and Markus Keller (2016) for predicting the yield for Riesling cultivar.

Usage

```
data(data_lm_yield_riesling)
```

Format

List of 12

References

Daniel Molitor and Markus Keller (2016). Yield of Müller-Thurgau and Riesling grapevines is altered by meteorological conditions in the current and previous growing seasons OENO One, 2016, 50, 4, 245 - 258.

`data_lm_yield_rivaner` *Yield model for Rivaner cultivar*

Description

A sample dataset containing the Linear regression model calculated by Daniel Molitor and Markus Keller (2016) for predicting the yield for Rivaner cultivar.

Usage

```
data(data_lm_yield_rivaner)
```

Format

List of 12

References

Daniel Molitor and Markus Keller (2016). Yield of Müller-Thurgau and Riesling grapevines is altered by meteorological conditions in the current and previous growing seasons OENO One, 2016, 50, 4, 245 - 258.

`data_pik_observ` *Sample of meteorological data for the Moselle region in Luxembourg*

Description

A sample dataset containing time series of meteorological data for the Moselle region located in the Grand-Duchy of Luxembourg obtained by the Potsdam Institute for Climate Impact Research (PIK) in Germany. The data covers the period from 1961 to 2018 at daily time interval. The data is provided as a space-time full data frame (STFDF) object from "spacetime" package.

Usage

```
data(data_pik_observ)
```

Format

Formal class 'STFDF' [package "spacetime"] with 4 slots
..@ data :'data.frame': 466048 obs. of 9 variables:
.. ..$ ta : num [1:466048] 1 1 1 1 1 1 1 1 1 1 ...
.. ..$ mo : num [1:466048] 1 1 1 1 1 1 1 1 1 1 ...
.. ..$ jahr : num [1:466048] 1961 1961 1961 1961 1961 ...
.. ..$ tmax : num [1:466048] 3.5 3.3 3 2.6 2.8 2.5 2 2.5 2.2 2 ...
.. ..$ tmit : num [1:466048] 2.3 2.2 1.7 1.3 1.6 1.1 0.5 1.5 1.1 0.7 ...

```
.. ..$ tmin : num [1:466048] 1.2 1 0.5 0 -0.1 -0.4 -0.9 -0.7 -0.9 -1.1 ...
.. ..$ prec : num [1:466048] 0 0 0 0 0 0 0 0 0 0 ...
.. ..$ ludr : num [1:466048] 984 982 978 973 980 ...
.. ..$ station_id: Factor w/ 22 levels "00809","00810",..: 1 2 3 4 5 6 7 8 9 10 ...
..@ sp :Formal class 'SpatialPoints' [package "sp"] with 3 slots
.. .. ..@ coords : num [1:22, 1:2] 90823 102882 115006 127122 100946 ...
.. .. .. ..- attr(*, "dimnames")=List of 2
.. .. .. .. ..$ : NULL
.. .. .. .. ..$ : chr [1:2] "x" "y"
.. .. ..@ bbox : num [1:2, 1:2] 81069 43466 127122 109652
.. .. .. ..- attr(*, "dimnames")=List of 2
.. .. .. .. ..$ : chr [1:2] "x" "y"
.. .. .. .. ..$ : chr [1:2] "min" "max"
.. .. ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
.. .. .. .. ..@ projargs: chr "+proj=tmerc +lat_0=49.83333333333334 +lon_0=6.166666666666667
+k=1 +x_0=80000 +y_0=100000 +ellps=intl +units=m +no_defs"
..@ time :An ‘xts’ object on 1961-01-01/2018-12-31 containing:
Data: int [1:21184, 1] 1 2 3 4 5 6 7 8 9 10 ...
- attr(*, "dimnames")=List of 2
..$ : NULL..$ : chr "timeIndex"
Indexed by objects of class: [POSIXct,POSIXt] TZ:
xts Attributes:
NULL
..@ endTime: POSIXct[1:21184], format: "1961-01-02" "1961-01-03" "1961-01-04" "1961-01-05"
...
```

Details

The spatial representation for the data corresponds to a SpatialPoints with coordinate reference system (CRS) EPSG:2169 - Luxembourg 1930 / Gauss - Projected.

References

Christoph Menz and Thomas Kartschall, 2019. Project CLIM4VITIS: Climate change impact mitigation for European viticulture. Potsdam Institut fuer Klimafolgenforschung (PIK), Germany. https://clim4vitis.eu/.

Examples

```
#' Cohercion to data.frame
pik_observ.df <- as.data.frame(data_pik_observ)
head(pik_observ.df)

#' Cohercion to xts
library(xts)
pik_observ.xts <- xts(x = pik_observ.df, order.by = pik_observ.df$time)
head(pik_observ.xts)

#' plot stfdf object
```

```
year.ini <- which(index(data_pik_observ@time) == as.POSIXct("1973-06-01", format = "%Y-%m-%d"))
year.end <- which(index(data_pik_observ@time) == as.POSIXct("1973-06-30", format = "%Y-%m-%d"))

stplot(data_pik_observ[, year.ini:year.end, "tmit"], scales=list(draw=TRUE))
stplot(data_pik_observ[, year.ini:year.end, "tmit"], scales=list(draw=TRUE), mode = "ts")
```

data_remich *Sample of meteorological data for Remich station*

Description

A sample dataset containing time series of meteorological data for Remich station located in the Grand-Duchy of Luxembourg obtained by the Institut Viti-vinicole in Remich. The data covers the period from 1970 to 2017 at daily time interval. The data is providad as a space-time full data frame (STFDF) object from "spacetime" package.

Usage

```
data(data_remich)
```

Format

```
Formal class 'STFDF' [package "spacetime"] with 4 slots
..@ data :'data.frame': 17532 obs. of 8 variables:
.. ..$ Year : num [1:17532] 1970 1970 1970 1970 1970 1970 1970 1970 1970 1970 ...
.. ..$ Month : num [1:17532] 1 1 1 1 1 1 1 1 1 1 ...
.. ..$ Day : num [1:17532] 1 2 3 4 5 6 7 8 9 10 ...
.. ..$ DayYear : num [1:17532] 1 2 3 4 5 6 7 8 9 10 ...
.. ..$ T.max : num [1:17532] -3.5 -1 1 0.8 -0.1 -0.7 0 -0.3 3.6 4.3 ...
.. ..$ T.min : num [1:17532] -4.8 -6.5 -2.5 -0.4 -4.1 -4.5 -4.8 -5.4 -1.6 2.5 ...
.. ..$ T.mean : num [1:17532] -4.15 -3.75 -0.75 0.2 -2.1 -2.6 -2.4 -2.85 1 3.4 ...
.. ..$ Rainfall: num [1:17532] 0 0 2.7 0.8 4 3.2 0 0 1.8 5.8 ...
..@ sp :Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots
.. .. ..@ data :'data.frame': 1 obs. of 4 variables:
.. .. .. ..$ id: Factor w/ 1 level "1": 1
.. .. .. ..$ x : num 6.35
.. .. .. ..$ y : num 49.5
.. .. .. ..$ z : Factor w/ 1 level "207": 1
.. .. ..@ coords.nrs : num(0)
.. .. ..@ coords : num [1, 1:2] 93626 67967
.. .. .. ..- attr(*, "dimnames")=List of 2
.. .. .. .. ..$ : NULL
.. .. .. .. ..$ : chr [1:2] "coords.x1" "coords.x2"
.. .. ..@ bbox : num [1:2, 1:2] 93626 67967 93626 67967
.. .. .. ..- attr(*, "dimnames")=List of 2
.. .. .. .. ..$ : chr [1:2] "coords.x1" "coords.x2"
.. .. .. .. ..$ : chr [1:2] "min" "max"
.. .. ..@ proj4string:Formal class 'CRS' [package "sp"] with 1 slot
```

.. ..@ projargs: chr "+proj=tmerc +lat_0=49.8333333333334 +lon_0=6.16666666666667 +k=1 +x_0=80000 +y_0=100000 +ellps=intl +units=m +no_defs"
..@ time :An ‘xts’ object on 1970-01-01/2017-12-31 containing:
Data: int [1:17532, 1] 1 2 3 4 5 6 7 8 9 10 ...
- attr(*, "dimnames")=List of 2
..$: NULL
..$: chr "timeIndex"
Indexed by objects of class: [POSIXct,POSIXt] TZ:
xts Attributes:
NULL
..@ endTime: POSIXct[1:17532], format: "1970-01-02" "1970-01-03" "1970-01-04" "1970-01-05" ...

Details

The spatial representation for the data corresponds to a SpatialPointsDataFrame with coordinate reference system (CRS) EPSG:2169 - Luxembourg 1930 / Gauss - Projected.

Source

https://agriculture.public.lu/de/weinbau-oenologie.html

Examples

```
#' Cohercion to data.frame
remich.df <- as.data.frame(data_remich)
head(remich.df)

#' Cohercion to xts
library(xts)
remich.xts <- xts(x = remich.df, order.by = remich.df$time)
head(remich.xts)
```

data_remich_bbch09 *Observed day of year (DOY) of stage BBCH-09 in Remich*

Description

A sample dataset containing the Observed day of year (DOY) of stage BBCH-09 in Remich for the period 1972 to 2019 from Molitor et al. (?).

Usage

```
data(data_remich_bbch09)
```

Format

```
'data.frame': 48 obs. of 9 variables:
$ Year : int 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 ...
$ Elbling : int 123 132 103 123 120 124 105 133 124 103 ...
$ Rivaner : int 124 133 104 123 122 124 107 133 126 104 ...
$ Auxerrois: int 124 132 104 124 122 125 107 133 126 105 ...
$ P.Blanc : int 123 135 104 123 122 124 107 134 127 105 ...
$ P.Gris : int 123 134 104 124 122 124 106 134 126 106 ...
$ Riesling : int 125 135 105 124 123 125 108 134 127 105 ...
$ Gew.Tr. : int 123 134 104 123 120 125 104 133 127 104 ...
$ Average : num 124 134 104 123 122 ...
```

References

Daniel Molitor, ...

`data_remich_bbch81` *Observed day of year (DOY) of stage BBCH-81 in Remich*

Description

A sample dataset containing the Observed day of year (DOY) of stage BBCH-81 in Remich for the period 1972 to 2015 from Molitor et al. (?).

Usage

```
data(data_remich_bbch81)
```

Format

```
'data.frame': 44 obs. of 9 variables:
$ Year : int 1972 1973 1974 1975 1976 1977 1978 1979 1980 1981 ...
$ Elbling : int 268 247 254 242 233 NA NA NA NA NA ...
$ Rivaner : int 240 236 234 228 224 NA 246 NA 253 229 ...
$ Auxerrois: int 244 237 246 232 227 NA NA NA NA NA ...
$ P.Blanc : int 249 246 249 242 232 NA NA NA NA NA ...
$ P.Gris : int 248 239 237 232 230 NA NA NA NA NA ...
$ Riesling : int 266 242 252 241 232 NA NA NA NA NA ...
$ Gew.Tr. : int 249 236 249 236 228 NA NA NA NA NA ...
$ Average : num 252 240 246 236 229 ...
```

References

Daniel Molitor, ...

`data_remich_riesling_d68` *Average dates of BBCH 09, 63, and 68 for Riesling in Remich*

Description

A sample dataset containing the average dates of BBCH 09 (budbreak), BBCH 63 (early bloom: 30% of caps fallen), and BBCH 68 (80% of caps fallen) in Luxembourg for the wine grape cultivar Riesling. Data were arranged relative to 1 January (DOY) or relative to the date of BBCH 68 (D68). Table 2 from Molitor ad Keller (2016).

Usage

```
data(data_remich_riesling_d68)
```

Format

'data.frame': 23 obs. of 7 variables:
$ Year : int 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 ...
$ BBCH.09_DOY: int 116 119 120 119 115 120 119 115 126 113 ...
$ BBCH.09_D68: int -49 -59 -61 -62 -63 -52 -54 -51 -50 -58 ...
$ BBCH.63_DOY: int 159 173 177 171 166 166 167 162 172 167 ...
$ BBCH.63_D68: int -6 -5 -4 -10 -12 -6 -6 -4 -4 -4 ...
$ BBCH.68_DOY: int 165 178 181 181 178 172 173 166 176 171 ...
$ BBCH.68_D68: int 0 0 0 0 0 0 0 0 0 0 ...

Source

Institut Viti-vinicole, Remich.

References

Daniel Molitor, and Markus Keller. Yield of Müller-Thurgau and Riesling grapevines is altered by meteorological conditions in the current and previous growing seasons. Vine and Wine Open Access Journal, Oneo One, 50(4):245-258, 2016.

`data_remich_rivaner_d68` *Average dates of BBCH 09, 63, 68 and 81 for Rivaner in Remich*

Description

A sample dataset containing the average dates of BBCH 09 (budbreak), 63 (early bloom: 30% of caps fallen), 68 (80% of caps fallen) and 81 (beginning of ripening) in Luxembourg for the wine grape cultivars Müller-Thurgau (Rivaner). Data were arranged relative to 1 January (DOY) or relative to the date of BBCH 68 (D68). Table 2 from Molitor ad Keller (2016).

Usage

```
data(data_remich_rivaner_d68)
```

Format

'data.frame': 23 obs. of 9 variables:
$ Year : int 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 ...
$ BBCH.09_DOY: int 116 118 120 119 115 120 117 117 126 113 ...
$ BBCH.09_D68: int -48 -58 -61 -61 -63 -54 -56 -49 -50 -58 ...
$ BBCH.63_DOY: int 158 172 177 172 166 169 167 162 172 167 ...
$ BBCH.63_D68: int -6 -4 -4 -8 -12 -5 -6 -4 -4 -4 ...
$ BBCH.68_DOY: int 164 176 181 180 178 174 173 166 176 171 ...
$ BBCH.68_D68: int 0 0 0 0 0 0 0 0 0 0 ...
$ BBCH.81_DOY: int 224 227 230 236 230 232 230 228 229 227 ...
$ BBCH.81_D68: int 60 51 49 56 52 58 57 62 53 56 ...

Source

Institut Viti-vinicole, Remich.

References

Daniel Molitor, and Markus Keller. Yield of Müller-Thurgau and Riesling grapevines is altered by meteorological conditions in the current and previous growing seasons. Vine and Wine Open Access Journal, Oneo One, 50(4):245-258, 2016.

`data_remich_yield` *Average annual yield for Rivaner and Riesling, and meteorological variables in Remich*

Description

Average annual yield (hl/ha) in Luxembourg for the wine grape cultivars Müller-Thurgau and Riesling, as well as key annual and growing season (April – October) meteorological variables (meteorological data from Remich). Table 1 from Molitor ad Keller (2016).

Usage

```
data(data_remich_yield)
```

Format

'data.frame': 23 obs. of 7 variables:
$ Year : int 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 ...
$ Yield_MuellerThurgau_hl_ha : int 137 147 130 118 49 142 162 109 122 140 ...
$ Yield_Riesling_hl_ha : int 96 94 77 78 64 102 110 95 72 98 ...

```
$ Mean_annual_temp_C : num 9.9 11.1 10.6 9 10.4 10.2 11.3 11.3 10.7 11.1 ...
$ Mean_growing_season.temp_C : num 14.4 14.9 15.3 13.9 14.7 14.6 15.3 15.3 15.3 15 ...
$ Annual_precip_sum_mm : int 737 741 816 534 785 790 732 948 994 776 ...
$ Growing_season_precip_sum_mm: int 437 451 387 280 461 560 409 596 562 414 ...
```

References

Daniel Molitor, and Markus Keller. Yield of Müller-Thurgau and Riesling grapevines is altered by meteorological conditions in the current and previous growing seasons. Vine and Wine Open Access Journal, Oneo One, 50(4):245-258, 2016.

`data_vineyards2018` *A 'sp' object for the vineyards along the Mosel in Luxemburg in 2018*

Description

A dataset containing a 'sp' object for the vineyards along the Mosel in the Grand-Duchy of Luxembourg in 2018. The data is provided by the Luxembourgish Data Platform, and provided as a SpatialPolygonsDataFrame from "sp" package.

Usage

```
data(data_vineyards2018)
```

Format

```
Formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5 slots
..@ data :'data.frame': 4966 obs. of 7 variables:
.. ..$ Weinbergsn: Factor w/ 4950 levels "1","10005","10007",..: 1359 4035 3194 2950 2952 1083
2849 2871 2874 2951 ...
.. ..$ CODE_ELEM : Factor w/ 1 level "V": 1 1 1 1 1 1 1 1 1 1 ...
.. ..$ CODE_COM : Factor w/ 14 levels "?","022","023",..: 12 4 14 6 12 12 14 14 14 14 ...
.. ..$ CODE_SECT : Factor w/ 8 levels "A","b","B","c",..: 1 7 8 7 3 3 7 7 7 7 ...
.. ..$ Shape_Leng: num [1:4966] 164 247 231 311 282 ...
.. ..$ Shape_Le_1: num [1:4966] 164 247 231 311 282 ...
.. ..$ Shape_Area: num [1:4966] 504 3773 3000 6016 2686 ...
..@ polygons :List of 4966
.. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
.. .. .. ..@ Polygons :List of 1
.. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
.. .. .. .. .. .. ..@ labpt : num [1:2] 93823 71477
.. .. .. .. .. .. ..@ area : num 504
.. .. .. .. .. .. ..@ hole : logi FALSE
.. .. .. .. .. .. ..@ ringDir: int 1
.. .. .. .. .. .. ..@ coords : num [1:6, 1:2] 93793 93797 93853 93851 93849 ...
.. .. .. ..@ plotOrder: int 1
.. .. .. ..@ labpt : num [1:2] 93823 71477
.. .. .. ..@ ID : chr "0"
.. .. .. ..@ area : num 504
.. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
.. .. .. ..@ Polygons :List of 1
.. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
.. .. .. .. .. .. ..@ labpt : num [1:2] 96817 77073
.. .. .. .. .. .. ..@ area : num 3773
.. .. .. .. .. .. ..@ hole : logi FALSE
.. .. .. .. .. .. ..@ ringDir: int 1
.. .. .. .. .. .. ..@ coords : num [1:8, 1:2] 96857 96848 96839 96827 96777 ...
.. .. .. ..@ plotOrder: int 1
.. .. .. ..@ labpt : num [1:2] 96817 77073
.. .. .. ..@ ID : chr "1"
.. .. .. ..@ area : num 3773
.. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
.. .. .. ..@ Polygons :List of 1
.. .. .. ..[list output truncated]
```

Details

The spatial representation for the data corresponds to a SpatialPolygonsDataFrame with coordinate reference system (CRS) EPSG:2169 - Luxembourg 1930 / Gauss - Projected.

Source

https://data.public.lu/fr/datasets/vineyards/

Examples

```
data(data_vineyards2018)
str(data_vineyards2018, list.len = 15)

#' plot sp object
library(sp)
spplot(data_vineyards2018, zcol="Weinbergsn", col = "Brown", scales = list(draw = TRUE),
       colorkey = FALSE)
```

data_water_surface   *A 'sp' object for the water surfaces in Grevenmacher*

Description

A dataset containing a 'sp' object for the water surfaces in the District of Grevenmacher in the Grand-Duchy of Luxembourg. The dataset contains streches from the Moselle (Mosel) and the S\^ure rivers. The data is provided by the Luxembourgish Data Platform, and provided as a SpatialPolygonsDataFrame from "sp" package.

Usage

```
data(data_water_surface)
```

Format

```
Formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5 slots
..@ data :'data.frame': 1373 obs. of 5 variables:
.. ..$ cat : int [1:1373] 1 2 3 4 5 6 7 8 9 10 ...
.. ..$ id : int [1:1373] 1 1 1 1 1 1 1 1 1 1 ...
.. ..$ ID_2 : Factor w/ 1353 levels "2?443?282","2?443?944",..: 2 4 9 16 19 35 39 45 51 56 ...
.. ..$ NATURE : int [1:1373] 3 3 3 3 3 3 3 3 3 3 ...
.. ..$ TOPONYME: Factor w/ 4 levels "Bassin d'?puration",..: NA NA NA NA NA NA NA NA NA
NA ...
..@ polygons :List of 1373
.. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
.. .. .. ..@ Polygons :List of 1
.. .. .. .. ..$ :Formal class 'Polygon' [package "sp"] with 5 slots
.. .. .. .. .. .. ..@ labpt : num [1:2] 93462 66333
.. .. .. .. .. .. ..@ area : num 145
.. .. .. .. .. .. ..@ hole : logi FALSE
.. .. .. .. .. .. ..@ ringDir: int 1
.. .. .. .. .. .. ..@ ringDir: int 1
.. .. .. .. .. .. ..@ coords : num [1:11, 1:2] 93468 93468 93466 93464 93462 ...
.. .. .. ..@ plotOrder: int 1
.. .. .. ..@ labpt : num [1:2] 93462 66333
.. .. .. ..@ ID : chr "0"
.. .. .. ..@ area : num 145
.. ..$ :Formal class 'Polygons' [package "sp"] with 5 slots
.. .. .. ..@ Polygons :List of 1
.. .. .. ..[list output truncated]
```

Details

The spatial representation for the data corresponds to a SpatialPolygonsDataFrame with coordinate reference system (CRS) EPSG:2169 - Luxembourg 1930 / Gauss - Projected.

The dataset identifies natural or artificial water surface, permanent or not. The field NATURE is defined as: 0 = stream surface; 1 = stagnant water surface; 2 = wet zone; 3 = basin; 4 = purification basin; 5 = pool; 6 = fish farming.
The selection criteria for defining the NATURE field is:
0: surface of stream of permanent flow. Minimum width 3.5 m.
1: permanent water surface without flow: lake, pond, pond whose smallest dimension is greater than 20 m. Dams are treated in this class and are updated if their area has increased 50%.
2: wetland area greater than 2 Ha, marsh, aquatic vegetation.
3: open basin. Treatment plant basins, swimming pools or fish farms are treated with a particular attribute value. Most small dimension must be greater than 10 m and the outline masonry.
4: sewage treatment plant basin, the smallest dimension of which must be greater than 10 m and the masonry outline.

5: swimming pool whose smallest dimension must be greater than 10 m.
6: fish pond with the smallest dimension greater than 10 m and the masonry outline.

Source

https://data.public.lu/en/datasets/bd-l-tc-surface-deau/#_

Examples

```
data(data_water_surface)
str(data_water_surface, list.len = 15)

#' plot stfdf object
library(sp)
spplot(data_water_surface, zcol="id", scales = list(draw = TRUE),
       colorkey = FALSE)
```

dd.lThresh *Compute degree days by the single (lower) temperature threshold*

Description

Implementation to compute degree days by the single (lower) temperature threshold by Molitor et al., (2014).

Usage

```
dd.lThresh(t.mean, a)
```

Arguments

`t.mean`	daily mean air temperature vector in Celsius degrees.
`a`	numeric, single (lower) threshold temperature (in Celsius degrees) for vine growth.

Value

a vector with the degree days.

References

Daniel Molitor, Jürgen Junk, Danièle Evers, Lucien Hoffmann, and Marco Beyer (2014). A high-resolution cumulative degree day-based model to simulate phenological development of grapevine. Am. J. Enol. Vitic., (65:1):72–80.

`dd.luhThresh` *Compute the degree days by a lower, upper and heat temperature thresholds*

Description

Implementation to compute the degree days by a lower, upper and heat temperature thresholds by Molitor et al., (2014).

Usage

```
dd.luhThresh(t.mean, a, b, c)
```

Arguments

`t.mean`	daily mean air temperature vector in Celsius degrees.
`a`	numeric, lower threshold temperature (in Celsius degrees) for vine growth.
`b`	numeric, upper threshold temperature (in Celsius degrees) for vine growth.
`c`	numeric, heat threshold temperature (in Celsius degrees) for vine growth.

Value

a vector with the degree days (in Celsius degrees) for vine growth.

References

Daniel Molitor, Jürgen Junk, Danièle Evers, Lucien Hoffmann, and Marco Beyer (2014). A high-resolution cumulative degree day-based model to simulate phenological development of grapevine. Am. J. Enol. Vitic., (65:1):72–80.

`dd.luThresh` *Compute the degree days by the double (lower and upper) temperature thresholds*

Description

Implementation to compute the degree days by the double (lower and upper) temperature thresholds by Molitor et al., (2014).

Usage

```
dd.luThresh(t.mean, a, b)
```

Arguments

`t.mean` vector, daily mean air temperature in Celsius degrees.

`a` numeric, lower threshold temperature (in Celsius degrees) for vine growth.

`b` numeric, upper threshold temperature (in Celsius degrees) for vine growth.

Value

a vector with the degree days (in Celsius degrees) for vine growth.

References

Daniel Molitor, Jürgen Junk, Danièle Evers, Lucien Hoffmann, and Marco Beyer (2014). A high-resolution cumulative degree day-based model to simulate phenological development of grapevine. Am. J. Enol. Vitic., (65:1):72–80.

`dd.single.triangle` *Degree days by the single triangle algorithm*

Description

Implementation to compute the degree days by the single triangle algorithm by Nendel (2010).

Usage

```
dd.single.triangle(t.zero, t.min, t.mean, t.max)
```

Arguments

`t.zero` numeric, threshold temperature (in Celsius degrees) for vine growth.

`t.min` daily minimum air temperature vector in Celsius degrees.

`t.mean` daily mean air temperature vector in Celsius degrees.

`t.max` daily maximum air temperature vector in Celsius degrees.

Value

a vector with the degree-days (in Celsius degrees) for vine growth.

References

Nendel, Class (2010). Grapevine bud break prediction for cool winter climates. Int. J. Biometeorol., 54, 231–241.

`fill.na` *Fill NA data in time series*

Description

Fill NA data in time series

Usage

```
fill.na(x)
```

Arguments

`x`	the input time series as xts object.

Value

a time series with the NAs replaced by data according to the na.locf zoo function.

`GrowthStage_CDD` *Cumulative degree days and BBCH growth stages*

Description

A sample dataset containing the cumulative degree days (CDD) for optimized lower, upper, and heat threshold triplets for Mueller-Thurgau (5°C, 20°C, 22°C), Riesling (7°C, 18°C, 24°C) and Pinot noir (3°C, 20°C, 24°C) per BBCH growth stages according to Supplementary Table 2 from Molitor and Junk (2019).

Usage

```
data(GrowthStage_CDD)
```

Format

'data.frame': 27 obs. of 3 variables:
$ Growth_stage: int 9 11 12 13 14 15 16 17 18 19 ...
$ Description : chr "Budburst: green shoot tips clearly visible"
"First leaf unfolded and spread away from shoot"
"Two leaves unfolded" "Three leaves unfolded" ...
$ CDD : num NA 34.9 58.6 80.8 110.2 ...

References

Daniel Molitor, Jürgen Junk. Climate change is implicating a two-fold impact on air temperature increase in the ripening period under the conditions of the Luxembourgish grapegrowing region. Vine and Wine Open Access Journal, Oneo One, (3):409-422, 2019.

id.na *Find indexes for NA data in time series*

Description

Find indexes for NA data in time series

Usage

```
id.na(x)
```

Arguments

x	the input time series as xts object.

Value

a vector with the index for NA data in the time series.

lm.botrytis.applied.riesling *Linear regression model applied for predicting the CDD_7;18;24 reaching 5% disease severity (Botrytis cinerea) for Riesling*

Description

Linear regression model applied for predicting the CDD_7;18;24 reaching 5% disease severity (Botrytis cinerea) for Riesling

Usage

```
lm.botrytis.applied.riesling(
  stat.tmean01,
  stat.tmean02,
  stat.tmean03,
  stat.rain01,
  stat.rain02,
  model
)
```

Arguments

`stat.tmean01`	numeric, vector with two columns: 1) year, and 2) first statistic for mean temperature.
`stat.tmean02`	numeric, vector with two columns: 1) year, and 2) second statistic for mean temperature.
`stat.tmean03`	numeric, vector with two columns: 1) year, and 2) third statistic for mean temperature.
`stat.rain01`	numeric, vector with two columns: 1) year, and 2) first statistic for precipitation.
`stat.rain02`	numeric, vector with two columns: 1) year, and 2) second statistic for precipitation.
`model`	the predefined model e.g. linear regression (lm). Should be a match between the linear regression coefficients and each statistic in the same definition order.

Value

data.frame, n obs. of 4 variables (year, predicted CDD for 7, 18 and 24 thresholds reaching 5% disease severity, Botrytis cinerea, for Riesling), and lower and upper confidence intervals. n is the total number of years for prediction.

`lm.yield.applied.riesling` *Linear regression model applied for predicting yield for Riesling*

Description

Linear regression model applied for predicting yield for Riesling

Usage

```
lm.yield.applied.riesling(
  stat.rain0.ries,
  stat.tmin1.ries,
  stat.tmin01.ries,
  stat.rain1.ries,
  stat.tmin02.ries,
  model
)
```

Arguments

`stat.rain0.ries`	numeric, vector with two columns: 1) year, and 2) first statistic for precipitation.
`stat.tmin1.ries`	numeric, vector with two columns: 1) year, and 2) first statistic for min temperature.

`stat.tmin01.ries`
: numeric, vector with two columns: 1) year, and 2) second statistic for min temperature.

`stat.rain1.ries`
: numeric, vector with two columns: 1) year, and 2) second statistic for precipitation.

`stat.tmin02.ries`
: numeric, vector with two columns: 1) year, and 2) third statistic for min temperature.

`model`
: the predefined model e.g. linear regression (lm). Should be a match between the linear regression coefficients and each statistic in the same definition order.

Value

data.frame, n obs. of 4 variables (year, predicted yield for Riesling), and lower and upper values represent the uncertainty band with 95 percent confidence level. n is the total number of years for prediction.

`lm.yield.applied.rivaner` *Linear regression model applied for predicting yield for Rivaner*

Description

Linear regression model applied for predicting yield for Rivaner

Usage

```
lm.yield.applied.rivaner(
  stat.tmin1,
  stat.tmean1,
  stat.tmax1,
  stat.tmean0,
  stat.rain01,
  stat.rain02,
  stat.tmin0,
  model
)
```

Arguments

`stat.tmin1`
: numeric, vector with two columns: 1) year, and 2) first statistic for min temperature.

`stat.tmean1`
: numeric, vector with two columns: 1) year, and 2) first statistic for mean temperature.

`stat.tmax1`
: numeric, vector with two columns: 1) year, and 2) first statistic for max temperature.

`stat.tmean0`	numeric, vector with two columns: 1) year, and 2) second statistic for mean temperature.
`stat.rain01`	numeric, vector with two columns: 1) year, and 2) first statistic for precipitation.
`stat.rain02`	numeric, vector with two columns: 1) year, and 2) second statistic for precipitation.
`stat.tmin0`	numeric, vector with two columns: 1) year, and 2) second statistic for min temperature.
`model`	the predefined model e.g. linear regression (lm). Should be a match between the linear regression coefficients and each statistic in the same definition order.

Value

data.frame, n obs. of 4 variables (year, predicted yield for Rivaner), and lower and upper values represent the uncertainty band with 95 percent confidence level. n is the total number of years for prediction.

`phen.dbbchx.df2` *Defining data.frame for yield and disease computation*

Description

Defining data.frame for yield and disease computation

Usage

```
phen.dbbchx.df2(data.years, year.ini, year.end, phen.luht, bbch.stage)
```

Arguments

`data.years`	list of zoo time series from the input data splitted by years.
`year.ini`	numeric, the initial year for computation.
`year.end`	numeric, the final year for computation.
`phen.luht`	data.frame for the phenology computation by lower, upper and heat thresholds (luht).
`bbch.stage`	numeric, the reference BBCH stage.

Value

a list with two objects: 1) data.years.subset, the input data.years data.frame subsetted according to the initial and ending year definition; and 2) phen.d_bbch_x.df2, a data.frame with two columns per reference BBCH stage defined, the first column is the actual BBCH DOY and the second column is the DOY for the reference BBCH (a vector of zeros, BBCH DOY reference minus BBCH DOY reference).

phenology.stages *Compute phenological stages*

Description

Implementation to compute phenological stages by Molitor et al., (2014).

Usage

```
phenology.stages(cdd.phen, ref.data, stage, cultivar)
```

Arguments

`cdd.phen`	list, cumulative degree days (in Celsius degrees) for vine growth in xts format as provided by any of the functions "cdd.1Tresh.phenology" or "cdd.2Tresh.phenology" or "cdd.3Tresh.phenology".
`ref.data`	data.frame, reference dataset to define the phenological stages e.g. "Growth-Stage_CDD" dataset.
`stage`	vector, growth stage(s) for which the phenology should be computed. One or more out of the 27 stages that range from 11 (First leaf unfolded and spread away from shoot) to 89 (Berries ripe for harvest) according to Molitor et al. (2014).

Value

list per year, with each list containing a data.frame with the phenological stages for vine growth.

References

Daniel Molitor, Jürgen Junk, Danièle Evers, Lucien Hoffmann, and Marco Beyer (2014). A high-resolution cumulative degree day-based model to simulate phenological development of grapevine. Am. J. Enol. Vitic., (65:1):72–80.

plot_na *Plot NA data in time series*

Description

Plot NA data in time series

Usage

```
plot_na(x, ids.na)
```

Arguments

`x`	the input time series as xts object.
`ids.na`	the vector which contains indexes for NA data as provided by the Id.na function.

Value

plots with the NAs highlighted.

`raw2xts` *Raw data to xts object*

Description

Raw data to xts object

Usage

```
raw2xts(data, year.name, month.name, day.name)
```

Arguments

`data`	the dataframe to convert to xts time series.
`year.name`	the column name for the variable year.
`month.name`	the column name for the variable month.
`day.name`	the column name for the variable day.

Value

the xts object for the input dataframe.

`window.stat` *Time window pane analysis*

Description

Time window pane analysis

Usage

```
## S3 method for class 'stat'
window(x, var, fun, t.ini, width, d.ref = FALSE)
```

Arguments

`x` zoo, time series from the input data for one individual year.

`var` character, the name of the variable for computing the window pane statistics.

`fun` character, name of the function for computation in the window pane.

`t.ini` numeric, the initial number in days from the reference to start the window pane statistics.

`width` numeric, the width in days of the window pane.

`d68` numeric or logical, one numeric value for the DOY of the reference BBCH stage to computing the statistic for the corresponding year or FALSE if reference BBCH stage day is not taken i.e. computation in actual DOY.

Value

numeric, a vector of two values: 1) the computation year; and 2) the computed statistic for the corresponding year.

Index