Package ‘meteoland’

October 3, 2016

Type Package

Title Landscape Meteorology Tools

Version 0.4.3

Date 2016-09-28

Author Miquel De Caceres, Nicolas Martin, Antoine Cabon

Maintainer Miquel De Caceres <miquelcaceres@gmail.com>

Description Functions to estimate weather variables at any position of a landscape.

LazyData TRUE

Depends R (>= 2.10), sp, qmap

Imports spdep, rgdal, ncdf4, ncdf4.helpers, methods, Rcpp (>= 0.11.0)

LinkingTo Rcpp

NeedsCompilation yes

Repository CRAN

Date/Publication 2016-10-03 08:17:46

R topics documented:

`defaultDownscalingParams`

Default downscaling parameters

Description

Returns a list with the default parameterization for staistical downscaling.

Usage

```
defaultDownscalingParams()
```

Value

A list with the following items (default values in brackets):

- `fill_wind [= TRUE]`: A logical flag to fill wind speed values with coarse-scale values when fine-scale data is missing.
- `wind_height  [= 10]`: Wind measurement height (in m).

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

`MeteorologyInterpolationData`

`defaultInterpolationParams`

Default interpolation parameters

Description

Returns a list with the default parameterization for interpolation. Most parameter values are set according to Thornton et al. (1997).

Usage

```
defaultInterpolationParams()
```

Value

A list with the following items (default values in brackets):

- `initial_Rp [= 140000]`: Initial truncation radius.
- `iterations [= 3]`: Number of station density iterations.
- `alpha_MinTemperature [= 3.0]`: Gaussian shape parameter for minimum temperature.
- `alpha_MaxTemperature [= 3.0]`: Gaussian shape parameter for maximum temperature.
- `alpha_DewTemperature [= 3.0]`: Gaussian shape parameter for dew-point temperature.
- `alpha_PrecipitationEvent [= 5.0]`: Gaussian shape parameter for precipitation events.
- `alpha_PrecipitationAmount [= 5.0]`: Gaussian shape parameter for the regression of precipitation amounts.
- `alpha_Wind [= 3.0]`: Gaussian shape parameter for wind.
- `N_MinTemperature [= 30]`: Average number of stations with non-zero weights for minimum temperature.
- `N_MaxTemperature [= 30]`: Average number of stations with non-zero weights for maximum temperature.
- `N_DewTemperature [= 30]`: Average number of stations with non-zero weights for dew-point temperature.
- `N_PrecipitationEvent [= 5]`: Average number of stations with non-zero weights for precipitation events.
- `N_PrecipitationAmount [= 20]`: Average number of stations with non-zero weights for the regression of precipitation amounts.
- `N_Wind [= 2]`: Average number of stations with non-zero weights for wind.
- `St_Precipitation [= 5]`: Number of days for the temporal smoothing of precipitation.
- `St_TemperatureRange [= 15]`: Number of days for the temporal smoothing of temperature range.

- `pop_crit` [= 0.50]: Critical precipitation occurrence parameter.
- `f_max` [= 0.95]: Maximum value for precipitation regression extrapolations.
- `wind_height` [= 10]: Wind measurement height (in m).

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

References

Thornton, P.E., Running, S.W., White, M. A., 1997. Generating surfaces of daily meteorological variables over large regions of complex terrain. J. Hydrol. 190, 214–251. doi:10.1016/S0022-1694(96)03128-9.

See Also

`MeteorologyInterpolationData`

`downscalinggrid` *Statistical downscaling of meteorological variables over a grid*

Description

Performs statistical downscaling of meteorological data over grids using objects of class `MeteorologyDownscalingData-cla` by applying bias correction to all meteorological variables excepting precipitation, which is downscaled using quantile mapping correction.

Usage

```
downscalinggrid(object, gridfiles, elevation = NULL, dates = NULL,
                maxreadcells = 50, export = FALSE,
                exportDir=getwd(), exportFormat = "netCDF",
                metadatafile = "MG.txt", verbose=TRUE)
```

Arguments

`object`	An object of class `MeteorologyDownscalingData-class`.
`gridfiles`	Either a vector of filename strings or a data frame with two columns: 'dir' and 'filename, to indicate the path to the historical meteorological data over a grid that will be used to calibrate downscaling.
`elevation`	A numeric vector with elevation (in m) for all grid cells. If `elevation = NULL` then Penman's potential evapotranspiration is not calculated.
`dates`	An object of class `Date`. If `dates = NULL` then all dates in `object` are processed.
`maxreadcells`	Maximum number of cells to be read simultaneously when calculating cell biases for the reference period. Reading historical meteorology for the whole grid simultaneously can cause memory problems.

`export`	If `export = FALSE` the result of interpolation is stored in memory. Otherwise the result is written in the disk (using the format specified in `exportFormat`).
`exportDir`	Output directory for downscaled meteorology data.
`exportFormat`	Format of meteorological data. Current accepted format is `"netCDF"` only.
`metadatafile`	The name of the file that will store the meta data describing all written files.
`verbose`	Boolean flag to print process information.

Details

The function first determines the predicted coarse-scale cell where each target grid cell falls. Then, the function calculates the bias for each cell between the meteorological data in `gridfiles` and the historic (reference) meteorological data in the corresponding coarse-scale cell of `object`. Finally, it uses this biases for the correction of future (climatic) data in `object`. For temperature, wind speed and radiation, the mean bias calculated for each of the twelve months in the historical period is used to correct the corresponding month in the future period. Relative humidity is first transformed to specific humidity, then it is corrected, and finally back-transformed to relative humidity. In the case of precipitation downscaling is performed by applying a quantile mapping between observed and modelled data (Gudmundsson et al. 2012), also on a monthly basis.

Value

If `export = FALSE`, function `downscalinggrid` returns an object of `SpatialGridMeteorology-class`. If `export = TRUE`, the function writes the results in files and a `data.frame` with columns 'dir' and 'filename' is returned.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

References

Gudmundsson L, Bremnes JB, Haugen JE, Engen-Skaugen T (2012) Technical Note: Downscaling predicted climatic precipitation to the station scale using statistical transformations - A comparison of methods. Hydrology and Earth System Sciences 16, 3383–3390. doi:10.5194/hess-16-3383-2012.

See Also

`penmanpoint`, `SpatialGridMeteorology-class`, `MeteorologyDownscalingData`

Examples

```
## TO BE DONE ##
```

downscalingpoints *Statistical downscaling of meteorological variables for a set of points*

Description

This function performs downscaling of predicted climatic data by applying bias correction to all meteorological variables excepting precipitation, which is downscaled using quantile mapping correction.

Usage

```
downscalingpoints(object, points, elevation = NULL, dates = NULL,
                export = FALSE, exportDir = getwd(), exportFormat = "meteoland",
                  metadatafile = "MP.txt", verbose = TRUE)
```

Arguments

`object`	An object of class `MeteorologyDownscalingData-class`.
`points`	An object of class `SpatialPointsMeteorology-class` with the coordinates and historical meteorological data of the locations for which downscaling of predicted climatic data has to be done. Alternatively, an object of class `SpatialPointsDataFrame-class` containing the meta data (columns `dir` and `filename`) of meteorological files that will be read from the disk.
`elevation`	A numeric vector with elevation (in m) for all points. If `elevation = NULL` then Penman's potential evapotranspiration is not calculated.
`dates`	An object of class `Date`. If `dates = NULL` then all dates in `object` are processed.
`export`	If `export = FALSE` the result of downscaling is stored in memory. Otherwise the result is written in the disk (using the format specified in `exportFormat`).
`exportDir`	Output directory for downscaled meteorology.
`metadatafile`	The name of the file that will store the meta data describing all written files.
`exportFormat`	Format of meteorological data. Current accepted formats are `"castanea"` and `"meteoland"`.
`verbose`	Boolean flag to print process information.

Details

Function `downscalingpoints` performs statistical downscaling of predicted climatic data for all points supplied in `points`. Observed meteorological data for each point typically comes from a nearby meteorological station, but they can be the result of interpolating the meteorology of several stations (see `MeteorologyInterpolationData`) or they can be extracted from reanalyzed meteorology (e.g. EU-WATCH) (see `extractNetCDF`).

For each target point, the function first determines the predicted cell where the point falls. Then it determines the dates that are shared in observed and predicted data for the historical period. These meteorological data of dates are used to conduct the correction of predicted climatic data for the future period. For temperature, wind speed and radiation, the mean bias calculated for each of the twelve months in the historical period is used to correct the corresponding month in the future period. Relative humidity is first transformed to specific humidity, then it is corrected, and finally back-transformed to relative humidity. In the case of precipitation downscaling is performed by applying a quantile mapping between observed and modelled data (Gudmundsson et al. 2012), also on a monthly basis.

Value

If `export = FALSE`, the function returns an object of class `SpatialPointsMeteorology-class` with the downscaled meteorology for each point. Otherwise the function returns an object of class `SpatialPointsDataFrame-class` containing the meta data of the files written in the disk.

Author(s)

Nicolas Martin, INRA-Avignon

Miquel De Cáceres Ainsa, Centre Tecnologic Forestal de Catalunya

References

Gudmundsson L, Bremnes JB, Haugen JE, Engen-Skaugen T (2012) Technical Note: Downscaling predicted climatic precipitation to the station scale using statistical transformations - A comparison of methods. Hydrology and Earth System Sciences 16, 3383–3390. doi:10.5194/hess-16-3383-2012.

See Also

`penmanpoint`, `extractNetCDF`, `SpatialPointsMeteorology-class`, `writemeteorologypointfiles`, `MeteorologyDownscalingData`, `MeteorologyInterpolationData`

Examples

```
data(examplegridtopography)
data(exampleinterpolationdata)
data(exampledownscalingdata)

#Creates spatial topography points from the grid
p = 1:2
points = as(examplegridtopography,"SpatialPoints")[p]
points = spTransform(points, exampleinterpolationdata@proj4string)
spt = SpatialPointsTopography(points, examplegridtopography$elevation[p],
                              examplegridtopography$slope[p],
                              examplegridtopography$aspect[p])

#Interpolation of two points for the whole time period (2000-2003)
historical = interpolationpoints(exampleinterpolationdata, spt)

#Downscaling of future predictions (RCM models, period 2020-2023)
predicted = downscalingpoints(exampledownscalingdata, historical, spt@data$elevation)

#Plot predicted mean temperature for point 1
```

```
meteoplot(predicted, 1, "MeanTemperature", ylab="Mean temperature (Celsius)", ylim=c(-5,40))
#Add uncorrected mean temperature data (cell #5)
lines(exampledownscalingdata@dates,
      exampledownscalingdata@futuredata[[5]]$MeanTemperature,
      col="red")
legend("topright", legend=c("corrected","uncorrected"), col=c("black","red"), lty=c(1,1), bty="n")

#Scatter plot
plot(exampledownscalingdata@futuredata[[5]]$MeanTemperature,
     predicted@data[[1]]$MeanTemperature, cex=0.1, asp=1,
     ylab="Corrected mean temperature", xlab="Uncorrected mean temperature")
abline(a=0,b=1,col="gray")

#Plot predicted precipitation for point 1
meteoplot(predicted, 1, "Precipitation", ylab="Precipitation (mm)", ylim=c(0,120))
#Add uncorrected mean temperature data (cell #5)
lines(exampledownscalingdata@dates,
      exampledownscalingdata@futuredata[[5]]$Precipitation,
      col="red", lty=3)
legend("topleft", legend=c("corrected","uncorrected"), col=c("black","red"), lty=c(1,3), bty="n")

#Scatter plot
plot(exampledownscalingdata@futuredata[[5]]$Precipitation,
     predicted@data[[1]]$Precipitation, cex=0.1, asp=1,
     ylab="Corrected precipitation (mm)", xlab="Uncorrected precipitation (mm)")
abline(a=0,b=1,col="gray")
```

`exampledownscalingdata` *Example data set for statistical downscaling of RCM predictions*

Description

Example data set including the predictions of Regional Climate Model (CCLM4-8-17; driving global model CNRM-CERFACS-CNRM-CM5) for 12 model cells in a small area in Catalonia (NE Spain). Meteorological data covers an historical period (2000-2003) and a future period (2020-2023), the latter simulated under rcp4.5 scenario.

Usage

```
data("exampledownscalingdata")
```

Format

Formal class 'MeteorologyDownscalingData-class'

Source

ESFG web site (http://esgf.llnl.gov/) that centralizes climate data from GCM and RCM uploaded in the frame of different international consortium, including the EURO-CORDEX regionalisation project.

Examples

```
data(exampledownscalingdata)
```

---

examplegridtopography *Example spatial grid topography*

Description

'SpatialGridTopography' object describing topographic features for a grid of 5 km x 5 km and cell size of 100 m in Catalonia (NE Spain).

Usage

```
data("examplegridtopography")
```

Format

Formal class 'SpatialGridTopography'

Source

'Institut Cartogràfic de Catalunya' (ICC)

Examples

```
data(examplegridtopography)
```

---

exampleinterpolationdata *Example data set for interpolation from weather stations*

Description

Example data set of spatial location, topography and daily meteorological records from 38 weather stations in Catalonia (NE Spain) corresponding to years 2000-2003.

Usage

```
data("exampleinterpolationdata")
```

Format

Formal class '`MeteorologyInterpolationData-class`'

Source

'Servei Meteorològic de Catalunya' (SMC) and 'Agencia Española de Meteorología' (AEMET)

Examples

```
data(exampleinterpolationdata)
```

`extractNetCDF` *Extraction of climatic data from NetCDF files*

Description

This function reads a set of NetCDF files (one per variable) and extracts data for a set of NetCDF cells that are specified using a boundary box (in lon/lat format).

Usage

```
extractNetCDF(ncdf_files, bbox, export = TRUE, exportDir = getwd(),
              exportFormat = "meteoland", mpfilename = "MP.txt")
```

Arguments

`ncdf_files`	Character vector containing files to read
`bbox`	Boundary box (2 x 2 matrix) specifying the limit coordinates of a study area (in lon/lat format).
`export`	If `export = FALSE` the extracted data is stored in memory. Otherwise the result is written in the disk (using the format specified in `exportFormat`).
`exportFormat`	Format of meteorological data. Current accepted formats are `"castanea"` and `"meteoland"`.
`exportDir`	Output directory for extracted meteorology.
`mpfilename`	The name of the file that will store the meta data describing all written files.

Details

Function `extractNetCDF` first identifies which cells in NetCDF data should be extracted, and the overall period (days). For each cell to be processed, the function loops over all files (which can describe different variables and time periods) and extracts the corresponding data. The function transforms units to the units used in `meteoland`. If specific humidity and mean temperature are available, the function calculates mean relative humidity.

Extracted meteorological data (a data frame with days in rows and meteorological variables in columns) can be stored in an object `SpatialPointsMeteorology-class` or it can be written in the disk (one file per cell). In the latter case, the output format can be chosen and the function also writes a supplementary file containing the meta data (i.e. the coordinates and filename of each file).

Value

If `export = FALSE`, the function returns an object of class `SpatialPointsMeteorology-class` with the meteorological series for each cell (represented by a spatial point). Otherwise the function returns an object of class `SpatialPointsDataFrame-class` containing the meta data of the files written in the disk.

Author(s)

Miquel De Cáceres Ainsa, Centre Tecnologic Forestal de Catalunya

Nicolas Martin, INRA-Avignon

See Also

`downscalingpoints`, `writemeteorologypointfiles`, `SpatialPointsMeteorology-class`

`extractSpatialData` *Extracts spatial data from coordinates.*

Description

Extracts spatial data contained in a `SpatialGridDataFrame` or `SpatialPixelsDataFrame` corresponding to the coordinates of spatial classes.

Usage

```
extractSpatialData(x, y)
```

Arguments

x	An object of class `GridTopology`, `SpatialGrid` and `SpatialPoints`.
y	An object of class `SpatialGridDataFrame` or `SpatialPixelsDataFrame`

Value

A data frame with the contents of slot 'data' corresponding to the coordinates in `x`.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

`SpatialGridDataFrame`

Examples

```
# TO BE DONE
```

interpolation.cv *Calibration and validation of interpolation procedures*

Description

Function `interpolation.calibration` determines optimal interpolation parameters 'N' and 'alpha' for a given meteorological variable. Optimization is done by minimizing mean absolute error (MAE) (Thornton et al. 1997). Function `interpolation.cv` calculates average mean absolute errors (MAE) for the prediction period of an object of class '`MeteorologyInterpolationData`'. Function `plot.interpolation.cv` plots cross-validation results. In both calibration and validation procedures, predictions for each weather station are made using a leave-one-out procedure (i.e. after excluding the station from the predictive set).

Usage

```
interpolation.calibration(object, stations = NULL, variable="Tmin",
                     N_seq = seq(5,30, by=5), alpha_seq = seq(0.25,10, by=0.25),
                        verbose = FALSE)
interpolation.cv(object, stations = NULL, verbose = FALSE)
## S3 method for class 'interpolation.cv'
plot(x, type = "stations", ...)
```

Arguments

`object`	An object of class `MeteorologyInterpolationData-class`.
`stations`	A numeric vector containing the indices of stations to be used to calculate mean absolute errors (MAE) in the calibration or cross-validation analysis. All the stations with data are included in the training set but predictive MAE are calculated for the 'stations' subset only.
`variable`	A string indicating the meteorological variable for which interpolation parameters 'N' and 'alpha' will be calibrated. Accepted values are 'Tmin' (for minimum temperature), 'Tmax' (for maximum temperature), 'Tdew' (for dew-point temperature), 'PrecEvent' (for precipitation events),'Prec' (for regression of precipitation amounts),'Prec' (for precipitation with the same values for precipitation events and regression of precipitation amounts).
`N_seq`	Set of average number of points to be tested.
`alpha_seq`	Set of alpha values to be tested.
`verbose`	A logical flag to generate additional console output.
`x`	A S3 object of class `interpolation.cv` with cross-validation results.
`type`	A string of the plot type to be produced (either "stations" or "dates").
`...`	Additional parameters passed to plot functions.

Value

Function `interpolation.calibration` returns an object of class `'interpolation.calibration'` with the following items:

- `MAE`: A numeric matrix with the mean absolute error values (averaged across stations) for each combination of parameters 'N' and 'alpha'.
- `minMAE`: Minimum MAE value.
- `N`: Value of parameter 'N' corresponding to the minimum MAE.
- `alpha`: Value of parameter 'alpha' corresponding to the minimum MAE.
- `Observed`: A matrix with observed values.
- `Predicted`: A matrix with predicted values for the optimum parameter combination.

Function `interpolation.cv` returns a list of class `'interpolation.cv'` with the following items:

- `stations`: A data frame with as many rows as weather stations and the following columns:
  - `MinTemperature-Bias`: Bias (in degrees), calculated over the prediction period, of minimum temperature estimations in weather stations.
  - `MinTemperature-MAE`: Mean absolute errors (in degrees), averaged over the prediction period, of minimum temperature estimations in weather stations.
  - `MaxTemperature-Bias`: Bias (in degrees), calculated over the prediction period, of maximum temperature estimations in weather stations.
  - `MaxTemperature-MAE`: Mean absolute errors (in degrees), averaged over the prediction period, of maximum temperature estimations in weather stations.
  - `Precipitation-Total`: Difference in the total precipitation of the studied period.
  - `Precipitation-DPD`: Difference in the proportion of days with precipitation.
  - `Precipitation-Bias`: Bias (in mm), calculated over the days with precipitation, of precipitation amount estimations in weather stations.
  - `Precipitation-MAE`: Mean absolute errors (in mm), averaged over the days with precipitation, of precipitation amount estimations in weather stations.
  - `RelativeHumidity-Bias`: Bias (in percent), calculated over the prediction period, of relative humidity estimations in weather stations.
  - `RelativeHumidity-MAE`: Mean absolute errors (in percent), averaged over the prediction period, of relative humidity estimations in weather stations.
  - `Radiation-Bias`: Bias (in MJ/m2), calculated over the prediction period, of incoming radiation estimations in weather stations.
  - `Radiation-MAE`: Mean absolute errors (in MJ/m2), averaged over the prediction period, of incoming radiation estimations in weather stations.
- `dates`: A data frame with as many rows as weather stations and the following columns:
  - `MinTemperature-Bias`: Daily bias (in degrees), averaged over the stations, of minimum temperature estimations.
  - `MinTemperature-MAE`: Daily mean absolute error (in degrees), averaged over the stations, of minimum temperature estimations.
  - `MaxTemperature-Bias`: Daily bias (in degrees), averaged over the stations, of maximum temperature estimations.

  - `MaxTemperature-MAE`: Daily mean absolute error (in degrees), averaged over the stations, of maximum temperature estimations.
  - `Precipitation-Bias`: Daily bias (in mm), averaged over the stations, of precipitation amount estimations.
  - `Precipitation-MAE`: Daily mean absolute error (in mm), averaged over the stations, of precipitation amount estimations.
  - `RelativeHumidity-Bias`: Daily bias (in percent), averaged over the stations, of relative humidity estimations.
  - `RelativeHumidity-MAE`: Daily mean absolute error (in percent), averaged over the stations, of relative humidity estimations.
  - `Radiation-Bias`: Daily bias (in MJ/m2), averaged over the stations, of incoming radiation estimations.
  - `Radiation-MAE`: Daily mean absolute errors (in MJ/m2), averaged over the stations, of incoming radiation estimations.
- `MinTemperature`: A data frame with predicted minimum temperature values.
- `MinTemperatureError`: A matrix with predicted minimum temperature errors.
- `MaxTemperature`: A data frame with predicted maximum temperature values.
- `MaxTemperatureError`: A matrix with predicted maximum temperature errors.
- `Precipitation`: A data frame with predicted precipitation values.
- `PrecipitationError`: A matrix with predicted precipitation errors.
- `RelativeHumidity`: A data frame with predicted relative humidity values.
- `RelativeHumidityError`: A matrix with predicted relative humidity errors.
- `Radiation`: A data frame with predicted radiation values.
- `RadiationError`: A matrix with predicted radiation errors.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

References

Thornton, P.E., Running, S.W., 1999. An improved algorithm for estimating incident daily solar radiation from measurements of temperature, humidity, and precipitation. Agric. For. Meteorol. 93, 211–228. doi:10.1016/S0168-1923(98)00126-9.

See Also

`MeteorologyInterpolationData`

Examples

```
data(exampleinterpolationdata)

#Calibration procedure
prec_cal = interpolation.calibration(exampleinterpolationdata, variable="Prec",
```

```
                                              stations = 1:5,
                                              N_seq=c(5,10,15), alpha_seq=seq(0.25,1.0, by=0.25),
                                              verbose = TRUE)

#Set 'alpha' and 'N' parameters to values found in calibration
exampleinterpolationdata@params$N_Precipitation = prec_cal$N
exampleinterpolationdata@params$alpha_Precipitation = prec_cal$alpha

#Run cross validation
cv = interpolation.cv(exampleinterpolationdata, stations = 1:5, verbose = TRUE)

#Plot results
plot(cv)
```

`interpolationgrid` *Interpolates meteorological variables over a grid.*

Description

Interpolates meteorological data over grids using objects of class `MeteorologyInterpolationData-class`.

Usage

```
interpolationgrid(object, grid, dates, export = FALSE,
                  exportDir=getwd(), exportFormat = "netCDF",
                  metadatafile = "MG.txt", verbose=TRUE)
```

Arguments

`object`	An object of class `MeteorologyInterpolationData-class`.
`grid`	An object of class `SpatialGridTopography-class` representing the target landscape.
`dates`	An object of class `Date`.
`export`	If `export = FALSE` the result of interpolation is stored in memory. Otherwise the result is written in the disk (using the format specified in `exportFormat`).
`exportDir`	Output directory for interpolated meteorology.
`exportFormat`	Format of meteorological data. Current accepted format is `"netCDF"` only.
`metadatafile`	The name of the file that will store the meta data describing all written files.
`verbose`	Boolean flag to print process information.

Value

If `export = FALSE`, function `interpolationgrid` returns an object of `SpatialGridMeteorology-class`. If `export = TRUE`, the function writes the results in files and a `data.frame` with columns 'dir' and 'filename' is returned.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

SpatialGridTopography, MeteorologyInterpolationData

Examples

```
data(examplegridtopography)
data(exampleinterpolationdata)

#Interpolation of meteorology over a grid for two days
examplegridtopography@proj4string = exampleinterpolationdata@proj4string
ml = interpolationgrid(exampleinterpolationdata, examplegridtopography,
                       as.Date(c("2001-02-03", "2001-06-03")))
#Plot PET corresponding to 2001-06-03
spplot(ml,2,"PET")
```

---

interpolationpoints *Interpolates meteorological variables for a set of points.*

Description

Interpolates meteorological data for points using an object of class MeteorologyInterpolationData-class.

Usage

```
interpolationpoints(object, points, dates = NULL, export = FALSE,
                    exportDir = getwd(), exportFormat = "meteoland",
                    metadatafile = "MP.txt", verbose=TRUE)
```

Arguments

object	An object of class MeteorologyInterpolationData-class.
points	An object of class SpatialPointsTopography-class.
dates	An object of class Date. If this is NULL then all dates in object are processed.
export	If export = FALSE the result of interpolation is stored in memory. Otherwise the result is written in the disk (using the format specified in exportFormat).
exportDir	Output directory for interpolated meteorology data.
exportFormat	Format of meteorological data. Current accepted formats are "castanea" and "meteoland".
metadatafile	The name of the file that will store the meta data describing all written files.
verbose	Boolean flag to print process information.

Value

If `export = FALSE`, function `interpolationpoints` returns an object of `SpatialPointsMeteorology-class`. If `export = TRUE`, the function returns an object of class `SpatialPointsDataFrame-class` containing the meta data of the files written in the disk.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

`penmanpoint`, `SpatialPointsTopography-class`, `MeteorologyInterpolationData`

Examples

```
data(examplegridtopography)
data(exampleinterpolationdata)

#Creates spatial topography points from the grid
p = 1:2
points = as(examplegridtopography,"SpatialPoints")[p]
points = spTransform(points, exampleinterpolationdata@proj4string)
spt = SpatialPointsTopography(points, examplegridtopography$elevation[p],
                              examplegridtopography$slope[p],
                              examplegridtopography$aspect[p])

#Interpolation of two points for the whole time period (2000-2003)
mp = interpolationpoints(exampleinterpolationdata, spt)

#Plot interpolated meteorological series
meteoplot(mp,1, ylab="Mean temperature")
```

`meteoplot` *Plots meteorological series*

Description

Simple plotting of meteorological series from an object of class `SpatialPointsMeteorology-class` or `SpatialGridMeteorology-class`.

Usage

```
meteoplot(object, index, variable="MeanTemperature", add = FALSE,...)
```

Arguments

object	An object of class SpatialPointsMeteorology-class or SpatialGridMeteorology-class.
index	An integer to indicate the point in a SpatialPointsMeteorology-class object or the grid cell in a SpatialGridMeteorology-class object.
variable	The meteorological variable to be plotted.
add	A flag to indicate wether drawing should be done on the current plot (using function lines).
...	Additional parameters for functions plot or lines.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

---

MeteorologyDownscalingData

Creates an object of class 'MeteorologyDownscalingData'

Description

Initializes an object for statistical downscaling of meteorological data over landscapes.

Usage

```
MeteorologyDownscalingData(points, historicdata, futuredata, dates,
                           params = defaultDownscalingParams())
```

Arguments

points	An object of class SpatialPoints.
historicdata	Historic (reference) meteorological data used to calibrate correction factors when compared with observations. A vector of data frames (one per point) or a single data frame containing the meta data (columns dir and filename) of meteorological files that will be read from the disk.
futuredata	Future (predicted) meteorological data to be corrected. A vector of data frames (one per point) or a single data frame containing the meta data (columns dir and filename) of meteorological files that will be read from the disk.
dates	Object of class "Date" describing the time period for which meteorological downscaling is possible (corresponding to futuredata).
params	A "list" containing downscaling parameters.

Details

See downscaling details in vignettes or in downscalingpoints.

Value

An object of class `MeteorologyDownscalingData`.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

`MeteorologyDownscalingData`, `exampledownscalingdata`, `defaultDownscalingParams`.

`MeteorologyDownscalingData-class`
Class `"MeteorologyDownscalingData"`

Description

An S4 class to conduct statistical downscaling of meteorology over a landscape.

Objects from the Class

Objects can be created by calls of the form `new("MeteorologyDownscalingData", ...)`, or by calls to the function `MeteorologyDownscalingData`.

Slots

`dates`**:** Object of class `"Date"` describing the time period for which meteorological estimates are possible.

`bbox`**:** Object of class `"matrix"` with the boundary box that sets meteorological estimation boundaries.

`proj4string`**:** Object of class `"CRS"` with the projection string of station spatial coordinates.

`coords`**:** Object of class `"matrix"` containing the coordinates of weather stations (each row is a point).

`historicdata`**:** Historic (reference) meteorological data used to calibrate correction factors when compared with observations. A vector of data frames (one per point) or a single data frame containing the meta data (columns `dir` and `filename`) of meteorological files that will be read from the disk.

`futuredata`**:** Future (predicted) meteorological data to be corrected. A vector of data frames (one per point) or a single data frame containing the meta data (columns `dir` and `filename`) of meteorological files that will be read from the disk.

`params`**:** A `"list"` containing downscaling parameters.

Extends

Class "MeteorologyProcedureData", directly. Class "Spatial", by class "MeteorologyProcedureData", distance 2

Methods

meteogrid `signature(object = "MeteorologyDownscalingData")`: Generates SpatialGridDataFrame objects with raster maps of meteorological variables for an input set of days.

meteopoints `signature(object = "MeteorologyDownscalingData")`: Generates data.frame objects with time series of meteorological variables for a given set of points.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

MeteorologyDownscalingData, MeteorologyProcedureData-class, exampledownscalingdata

Examples

```
#Structure of the S4 object
showClass("MeteorologyDownscalingData")
```

MeteorologyInterpolationData

Creates an object of class 'MeteorologyInterpolationData'

Description

Initializes an object for meteorology interpolation over landscapes using weather station data and the methods described in Thornton et al. (1997) and Thornton & Running (1999).

Usage

```
MeteorologyInterpolationData(points, elevation, slope, aspect,
                                 MinTemperature, MaxTemperature,
                                 Precipitation, RelativeHumidity,
                                 Radiation = NULL, WindSpeed = NULL,
                                 WindDirection = NULL, WindFields = NULL,
                                 params = defaultInterpolationParams())
```

Arguments

points	An object of class SpatialPoints.
elevation	A numeric vector with elevation values of weather stations (in meters).
slope	A numeric vector with slope values of weather stations (in degrees). Needed for cross-validation only.
aspect	A numeric vector with aspect values of weather stations (in degrees from North). Needed for cross-validation only.
MinTemperature	A matrix with minimum temperature recordings (in degrees Celsius) for all weather stations (in rows) and all days (in columns).
MaxTemperature	A matrix with maximum temperature recordings (in degrees Celsius) for all weather stations (in rows) and all days (in columns).
Precipitation	A matrix with precipitation recordings (in mm of water) for all weather stations (in rows) and all days (in columns).
RelativeHumidity	A matrix with relative humidity recordings (in percent) for all weather stations (in rows) and all days (in columns).
Radiation	A matrix with relative radiation recordings (in MJ/m2) for all weather stations (in rows) and all days (in columns). Needed for cross-validation only.
WindSpeed	A matrix with wind speed recordings (in m/s) for all weather stations (in rows) and all days (in columns).
WindDirection	A matrix with wind direction recordings (in degrees from North) for all weather stations (in rows) and all days (in columns).
WindFields	Object of class "list". See function readWindNinjaWindFields.
params	A "list" containing interpolation parameters.

Details

See interpolation details in vignettes.

Value

An object of class MeteorologyInterpolationData.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

References

Thornton, P.E., Running, S.W., 1999. An improved algorithm for estimating incident daily solar radiation from measurements of temperature, humidity, and precipitation. Agric. For. Meteorol. 93, 211–228. doi:10.1016/S0168-1923(98)00126-9.

Thornton, P.E., Running, S.W., White, M. a., 1997. Generating surfaces of daily meteorological variables over large regions of complex terrain. J. Hydrol. 190, 214–251. doi:10.1016/S0022-1694(96)03128-9.

See Also

`MeteorologyInterpolationData`, `defaultInterpolationParams`.

Examples

```
## TO BE DONE ##
```

MeteorologyInterpolationData-class
Class "MeteorologyInterpolationData"

Description

An S4 class to interpolate meteorology over a landscape.

Objects from the Class

Objects can be created by calls of the form `new("MeteorologyInterpolationData", ...)`, or by calls to the function `MeteorologyInterpolationData`.

Slots

- `dates`: Object of class "Date" describing the time period for which meteorological estimates are possible.
- `bbox`: Object of class "matrix" with the boundary box that sets meteorological estimation boundaries.
- `proj4string`: Object of class "CRS" with the projection string of station spatial coordinates.
- `coords`: Object of class "matrix" containing the coordinates of weather stations (each row is a point).
- `elevation`: A numeric vector with elevation values of weather stations (in meters).
- `slope`: A numeric vector with slope values of weather stations (in degrees). Needed for cross-validation only.
- `aspect`: A numeric vector with aspect values of weather stations (in degrees from North). Needed for cross-validation only.
- `MinTemperature`: Object of class "data.frame" with minimum temperature recordings (in degrees Celsius) for all weather stations (in rows) and all days (in columns).
- `MaxTemperature`: Object of class "data.frame" with maximum temperature recordings (in degrees Celsius) for all weather stations (in rows) and all days (in columns).
- `SmoothedTemperatureRange`: Object of class "matrix" with temporally smoothed temperature range recordings (in degrees Celsius) for all weather stations (in rows) and all days (in columns).
- `Precipitation`: Object of class "matrix" with precipitation recordings (in mm of water) for all weather stations (in rows) and all days (in columns).
- `SmoothedPrecipitation`: Object of class "matrix" with temporally smoothed precipitation recordings (in mm of water) for all weather stations (in rows) and all days (in columns).

- `RelativeHumidity`: Object of class `matrix` with relative humidity recordings (in percent) for all weather stations (in rows) and all days (in columns).
- `Radiation`: Object of class `matrix` with relative radiation recordings (in MJ/m2) for all weather stations (in rows) and all days (in columns). Needed for cross-validation only.
- `WindSpeed`: Object of class `"matrix"` with wind speed recordings (in m/s) for all weather stations (in rows) and all days (in columns).
- `WindDirection`: Object of class `"matrix"` with wind direction recordings (in degrees from North) for all weather stations (in rows) and all days (in columns).
- `WindFields`: Object of class `"list"`. See function `readWindNinjaWindFields`.
- `WFIndex`: Object of class `"matrix"` with the closest windfield index for each stations in each day.
- `WFFactor`: Object of class `"matrix"` with the multiplication factor for the wind speed of each stations in each day.
- `params`: A `"list"` containing interpolation parameters.

Extends

Class `"MeteorologyProcedureData"`, directly. Class `"Spatial"`, by class "MeteorologyProcedureData", distance 2

Methods

- **meteogrid** `signature(object = "MeteorologyInterpolationData")`: Generates `SpatialGridDataFrame` objects with raster maps of meteorological variables for an input set of days.
- **meteopoints** `signature(object = "MeteorologyInterpolationData")`: Generates data.frame objects with time series of meteorological variables for a given set of points.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

`MeteorologyInterpolationData`, `MeteorologyProcedureData-class`

Examples

```
#Structure of the S4 object
showClass("MeteorologyInterpolationData")
```

MeteorologyProcedureData-class

Class "MeteorologyProcedureData"

Description

A virtual class for estimating meteorology over landscapes

Objects from the Class

A virtual Class: No objects may be created from it.

Slots

- `dates`**:** Object of class "Date" describing the time period for which meteorological estimates are possible.
- `bbox`**:** Object of class "matrix" with the boundary box that sets meteorological estimation boundaries.
- `proj4string`**:** Object of class "CRS" with the projection string of accepted coordinates.

Methods

- **meteogrid** `signature(object = "MeteorologyProcedureData")`: Generates `SpatialGridDataFrame` objects with raster maps of meteorological variables for an input set of days.
- **meteopoints** `signature(object = "MeteorologyProcedureData")`: Generates data.frame objects with time series of meteorological variables for a given set of points.

Extends

Class "`Spatial`", directly.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

`MeteorologyInterpolationData-class`, `MeteorologyDownscalingData-class`

Examples

```
showClass("MeteorologyProcedureData")
```

`penmanpoint` *Penman's potential evapotranspiration*

Description

Calculates Penman's potential evapotranspiration for a single point.

Usage

```
penmanpoint(latitude, elevation, J,
            Tmin, Tmax, RHmin, RHmax, R_s,  u,
            z=2.0, z0 = 0.001, alpha = 0.08, windfun="1956")
```

Arguments

`latitude`	Latitude in degrees.
`elevation`	Elevation (in m).
`J`	An integer vector with Julian days.
`Tmax`	Numeric vector with maximum temperature (degrees Celsius).
`Tmin`	Numeric vector with minimum temperature (degrees Celsius).
`RHmin`	Numeric vector with minimum relative humidity (percent).
`RHmax`	Numeric vector with maximum relative humidity (percent).
`R_s`	Numeric vector with solar radiation (MJ/m2).
`u`	Numeric vector with wind speed (m/s).
`z`	Wind measuring height (m).
`z0`	Roughness height (m).
`alpha`	Albedo.
`windfun`	Wind speed function version, either "1948" or "1956".

Details

The code was taken from package 'Evapotranspiration', which follows McMahon et al. (2013). If wind speed is not available, an alternative formulation for potential evapotranspiration is used as an approximation (Valiantzas 2006)

Value

A numeric vector with potential evapotranspiration (in mm of water).

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

References

Penman, H. L. 1948. Natural evaporation from open water, bare soil and grass. Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences, 193, 120-145.

Penman, H. L. 1956. Evaporation: An introductory survey. Netherlands Journal of Agricultural Science, 4, 9-29.

McMahon, T.A., Peel, M.C., Lowe, L., Srikanthan, R., McVicar, T.R. 2013. Estimating actual, potential, reference crop and pan evaporation using standard meteorological data: a pragmatic synthesis. Hydrology \& Earth System Sciences 17, 1331–1363. doi:10.5194/hess-17-1331-2013.

See Also

`interpolationpoints`

`readmeteorologygrid` *Reads grid meteorology from the disk*

Description

Functions to read grid meteorological data from the disk.

Usage

```
readmeteorologygrid(file, format = "netCDF")
readmeteorologygridfiles(files, format = "netCDF")
readmeteorologygridcells(files, cellIndices, format = "netCDF")
```

Arguments

`file`	A string of the file name to be read.
`format`	Format of meteorological data. The only accepted format is `"netCDF"`.
`files`	Either a vector of filename strings or a data frame with two columns: 'dir' and 'filename.
`cellIndices`	An integer vector with grid cell indices indicating the cells for which meteorological data has to be read.

Details

Function `readmeteorologygrid` reads a file containing the meteorology over a grid for a single day. Function `readmeteorologygridfiles` reads several files, each containing the meteorology over the same grid for a different day. Function `readmeteorologygridcell` also reads several grid meteorology files, but it keeps the meteorology of a set of cells only.

Value

Function `readmeteorologygrid` returns an object `SpatialGridDataFrame-class` where the data frame has grid cells as rows and meteorological variables as columns. Function `readmeteorologygrid` returns an object `SpatialGridMeteorology-class` and `readmeteorologygridcells` returns an object `SpatialPointsMeteorology-class`.

Author(s)

Miquel De Cáceres Ainsa, Centre Tecnologic Forestal de Catalunya

See Also

`writemeteorologygrid`, `SpatialGridMeteorology-class`

`readmeteorologypoint` *Reads point meteorology from the disk*

Description

Functions to read point meteorological data from ascii files in different formats.

Usage

```
readmeteorologypoint(file, dates = NULL, format = "meteoland")
readmeteorologypointfiles(points, files=NULL, dates = NULL, format="meteoland")
```

Arguments

`file`	A string of the file to be read.
`points`	An object of class `SpatialPoints-class` (in this case `files` cannot be `NULL`) or object of class `SpatialPointsDataFrame-class` with two data columns: '`dir`' and '`filename`'.
`files`	A vector of strings to be read (when `points` is of class `SpatialPoints-class`). Length and order must match `points`.
`dates`	Object of class `"Date"` describing a subset of dates to be extracted from meteorological series. If `NULL` the whole period read from files is kept.
`format`	Format of meteorological data. Current accepted formats are `"castanea"` or `"meteoland"`.

Author(s)

Miquel De Cáceres Ainsa, Centre Tecnologic Forestal de Catalunya

Nicolas Martin, INRA-Avignon

See Also

`writemeteorologypoint`, `SpatialPointsMeteorology-class` readWindNinjaWindFields

Reads WindNinja results

Description

Reads the wind fields generated by 'WindNinja' (http://www.firelab.org/project/windninja) for combinations of domain-level wind speed and wind direction classes.

Usage

```
readWindNinjaWindFields(filebase, resolution = "100m",
                        directionClasses = c(0, 45, 90, 135, 180, 225, 270, 315),
                     speedClasses = c(5, 15, 25), proj4string = CRS(as.character(NA)))
```

Arguments

`filebase`	A string to indicate the template for accessing WindNinja files. Resolution, wind directions and wind speed class values are appended to this string to obtain the filename to read.
`resolution`	Resolution string.
`directionClasses`	A vector of wind speed directions (in degrees).
`speedClasses`	A vector of wind class values (in m/s).
`proj4string`	Object of class `"CRS"` with the projection string of wind field rasters.

Value

A list with the following items:

- `directionClasses`: The vector of wind direction classes.
- `speedClasses`: The vector of wind speed classes.
- `indexTable`: A numeric matrix indicating the raster index of each combination of domain-level wind directions and wind speed classes.
- `windSpeed`: An object of class `SpatialGridDataFrame` containing wind speed rasters (in m/s) for each combination of domain-level wind direction and wind speed.
- `windDirection`: An object of class `SpatialGridDataFrame` containing wind direction rasters (in degrees from North) for each combination of domain-level wind direction and wind speed.

Note

WindNinja should be run with m/s as wind speed units and for all the combinations of domain-level wind speed and wind direction required.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

References

Forthofer, J.M., Butler, B.W., Wagenbrenner, N.S., 2014. A comparison of three approaches for simulating fine-scale surface winds in support of wildland fire management. Part I. Model formulation and comparison against measurements. Int. J. Wildl. Fire 23, 969–981.

See Also

MeteorologyInterpolationData

Examples

```
## TO BE DONE ##
```

SpatialGridMeteorology *Creates a 'SpatialGridMeteorology'*

Description

Initializes an object of class `SpatialGridMeteorology-class`

Usage

```
SpatialGridMeteorology(grid, proj4string=CRS(as.character(NA)), data, dates)
```

Arguments

grid	An object of class GridTopology-class
proj4string	Object of class "CRS" with the projection string.
data	A vector of data frames (one per date).
dates	Object of class "Date" describing the time period of meteorological estimates.

Value

An object of class SpatialGridMeteorology-class

Author(s)

Miquel De Cáceres Ainsa, Centre Tecnologic Forestal de Catalunya

See Also

SpatialGridMeteorology-class

SpatialGridMeteorology-class
Class "SpatialGridMeteorology"

Description

An S4 class that represents a spatial grid with meteorology daily data.

Objects from the Class

Objects can be created by calls of the form `new("SpatialGridMeteorology", ...)`, or by calls to the function `SpatialGridMeteorology`.

Slots

`dates`: Object of class `"Date"` describing the time period for which meteorological estimates are available.

`data`: A vector of `"data.frame"` objects, each one containing the grid data for one date.

`bbox`: Object of class `"matrix"` with the boundary box.

`proj4string`: Object of class `"CRS"` with the projection string.

Extends

Class `"SpatialGrid"`, directly. Class `"Spatial"`, by class "SpatialGrid", distance 2.

Author(s)

Miquel De Cáceres Ainsa, Centre Tecnologic Forestal de Catalunya

See Also

`SpatialGridTopography`, `SpatialGridDataFrame-class`

Examples

```
#Structure of the S4 object
showClass("SpatialGridMeteorology")
```

SpatialGridTopography *Creates a 'SpatialGridTopography'*

Description

Function `SpatialGridTopography` creates an object of class `SpatialGridTopography-class` containing topographic variables over a landscape.

Usage

```
SpatialGridTopography(grid, elevation, proj4string = CRS(as.character(NA)))
```

Arguments

`grid`	An object of class `GridTopology-class` or `SpatialGrid-class`.
`elevation`	A vector of elevation values for all cells of the grid (in m).
`proj4string`	An object of class `CRS-class`.

Details

Slope and aspect calculations were adapted from functions in package 'SDMTools', which used the approach described in Burrough & McDonell (1998).

The rate of change (delta) of the surface in the horizontal `(dz/dx)` and vertical `(dz/dy)` directions from the center cell determines the slope and aspect. The values of the center cell and its eight neighbors determine the horizontal and vertical deltas. The neighbors are identified as letters from 'a' to 'i', with 'e' representing the cell for which the aspect is being calculated. The rate of change in the x direction for cell 'e' is calculated with the algorithm:

`[dz/dx] = ((c + 2f + i) - (a + 2d + g) / (8 * x_cell_size)`

The rate of change in the y direction for cell 'e' is calculated with the following algorithm:

`[dz/dy] = ((g + 2h + i) - (a + 2b + c)) / (8 * y_cell_size)`

The algorithm calculates slope as: `rise_run = sqrt ( [dz/dx]2 + [dz/dy]2 ]`).

From this value , one can calculate the slope in degrees or radians as:

`slope_degrees = ATAN (rise_run) * 57.29578`

`slope_radians = ATAN (rise_run)`

Taking the rate of change in both the x and y direction for cell 'e', aspect is calculated using:

`aspect = 57.29578 * atan2 ([dz/dy], -[dz/dx])`

The aspect value is then converted to compass direction values (0-360 degrees).

Value

Function `SpatialGridTopography` returns an object '`SpatialGridTopography-class`'.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

References

Burrough, P. A. and McDonell, R.A., 1998. Principles of Geographical Information Systems (Oxford University Press, New York), p. 190.

See Also

`extractSpatialData`, `SpatialGridTopography-class`

Examples

```
data(examplegridtopography)

#Display data
spplot(examplegridtopography, type="elevation", scales=list(draw=TRUE))
spplot(examplegridtopography, type="slope", scales=list(draw=TRUE))
spplot(examplegridtopography, type="aspect", scales=list(draw=TRUE))
```

SpatialGridTopography-class — *Class* "SpatialGridTopography"

Description

An S4 class that represents topography over a grid of coordinates.

Objects from the Class

Objects can be created by calls of the form `new("SpatialGridTopography", ...)`, or by calls to the function `SpatialGridTopography`.

Slots

`grid`: Object of class `GridTopology`.

`data`: Object of class `"data.frame"` containing the elevation (in m), slope (in degrees) and aspect (in degrees from North) of every cell.

`bbox`: Object of class `"matrix"` with the boundary box.

`proj4string`: Object of class `"CRS"` with the projection string.

Extends

Class "`SpatialGridDataFrame`", directly. Class "`SpatialGrid`", by class "SpatialGridDataFrame", distance 2. Class "`Spatial`", by class "SpatialGridDataFrame", distance 3.

Methods

spplot `signature(object = "SpatialGridTopography")`: allows plotting topography maps.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

`SpatialGridTopography`, `SpatialGridDataFrame-class`

Examples

```
#Structure of the S4 object
showClass("SpatialGridTopography")
```

`SpatialPointsMeteorology` *Creates a 'SpatialPointsMeteorology'*

Description

Initializes an object of class `SpatialPointsMeteorology-class`

Usage

```
SpatialPointsMeteorology(points, data, dates)
```

Arguments

`points`	An object of class `SpatialPoints-class`
`data`	A vector of data frames (one per spatial point).
`dates`	Object of class `"Date"` describing the time period of meteorological estimates.

Value

An object of class `SpatialPointsMeteorology-class`

Author(s)

Miquel De Cáceres Ainsa, Centre Tecnologic Forestal de Catalunya

See Also

`SpatialPointsMeteorology-class`

SpatialPointsMeteorology-class
Class "SpatialPointsMeteorology"

Description

An S4 class that represents a set of points with meteorology data series.

Objects from the Class

Objects can be created by calls of the form `new("SpatialPointsMeteorology", ...)`, or by calls to the function `SpatialPointsMeteorology`.

Slots

`dates`: Object of class `"Date"` describing the time period for which meteorological estimates are available.

`data`: A vector of `"data.frame"` objects, each one corresponding to one spatial point.

`coords`: Object of class `"matrix"` with the spatial coordinates.

`bbox`: Object of class `"matrix"` with the boundary box.

`proj4string`: Object of class `"CRS"` with the projection string.

Extends

Class "`SpatialPoints`", directly. Class "`Spatial`", by class "SpatialPoints", distance 2.

Author(s)

Miquel De Cáceres Ainsa, Centre Tecnologic Forestal de Catalunya

See Also

`SpatialPointsTopography-class`, `SpatialPoints-class`

Examples

```
#Structure of the S4 object
showClass("SpatialPointsMeteorology")
```

SpatialPointsTopography *Creates a 'SpatialPointsTopography'*

Description

Function SpatialPointsTopography creates an object of class SpatialPointsTopography-class containing topographic variables for a set of points.

Usage

```
SpatialPointsTopography(points, elevation, slope, aspect)
```

Arguments

points	An object of class SpatialPoints-class.
elevation	Elevation values (in m) of the points.
slope	Slope values (in degrees) of the points.
aspect	Aspect values (in degrees from North) of the points.

Value

Function SpatialPointsTopography returns an object 'SpatialPointsTopography-class'.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

SpatialPointsTopography-class

Examples

```
data(examplegridtopography)

#Creates spatial topography points from the grid
p = 1:2
points = as(examplegridtopography,"SpatialPoints")[p]
spt = SpatialPointsTopography(points, examplegridtopography$elevation[p],
                              examplegridtopography$slope[p],
                              examplegridtopography$aspect[p])
```

SpatialPointsTopography-class

Class "SpatialPointsTopography"

Description

An S4 class that represents topography over a grid of coordinates.

Objects from the Class

Objects can be created by calls of the form `new("SpatialPointsTopography", ...)`, or by calls to the function `SpatialPointsTopography`.

Slots

`data`: Object of class "`data.frame`" containing the elevation (in m), slope (in degrees) and aspect (in degrees from North) of every point.

`coords`: Object of class "`matrix`" with the spatial coordinates.

`bbox`: Object of class "`matrix`" with the boundary box.

`proj4string`: Object of class "`CRS`" with the projection string.

Extends

Class "`SpatialPointsDataFrame`", directly. Class "`SpatialPoints`", by class "SpatialPointsDataFrame", distance 2. Class "`Spatial`", by class "SpatialPointsDataFrame", distance 3.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

`SpatialPointsTopography`, `SpatialPointsDataFrame-class`

Examples

```
#Structure of the S4 object
showClass("SpatialPointsTopography")
```

spplot — *Spatial grid plots*

Description

Function spplot for SpatialGridTopography-class objects allows drawing maps of topographic attributes. Function spplot for SpatialGridMeteorology-class objects allows drawing maps of meteorological variables corresponding to specific dates.

Usage

```
## S4 method for signature 'SpatialGridTopography'
spplot(obj, variable="elevation",...)
## S4 method for signature 'SpatialGridMeteorology'
spplot(obj, date, variable="MeanTemperature", ...)
```

Arguments

obj	An object of class SpatialGridTopography.
variable	A string of the variable to be plotted (only type="elevation", type="slope", type="aspect" are allowed).
...	Additional parameters to function spplot.
date	A string or an integer for the date to be plotted.

Author(s)

Miquel De Cáceres Ainsa, Biodiversity and Landscape Ecology Laboratory, Centre Tecnologic Forestal de Catalunya

See Also

meteoplot

Examples

```
data(examplegridtopography)

#Display data
spplot(examplegridtopography, type="elevation", scales=list(draw=TRUE))
spplot(examplegridtopography, type="slope", scales=list(draw=TRUE))
spplot(examplegridtopography, type="aspect", scales=list(draw=TRUE))
```

writemeteorologygrid *Writes grid meteorology to the disk*

Description

Functions to write grid meteorological data to the file system.

Usage

```
writemeteorologygrid(object, date, file, format = "netCDF")
writemeteorologygridfiles(object, dir=getwd(), format = "netCDF",
                          metadatafile = "MG.txt")
```

Arguments

`object`	An object of class `SpatialGridMeteorology-class` with the meteorological data to be written.
`date`	A `Date` object or a character string indicating the date of the meteorological grid to be written.
`file`	A string with the file name to be written.
`format`	Format of meteorological data. The only accepted format is `"netCDF"`.
`dir`	Output directory for meteorology data.
`metadatafile`	The name of the file that will store the meta data describing all written files.

Author(s)

Miquel De Cáceres Ainsa, Centre Tecnologic Forestal de Catalunya

See Also

`readmeteorologygrid`, `SpatialGridMeteorology-class`

writemeteorologypoint *Writes point meteorology to the disk*

Description

Functions to write point meteorological data to ascii files in different formats.

Usage

```
writemeteorologypoint(data, file, format = "meteoland")
writemeteorologypointfiles(object, dir=getwd(), format = "meteoland",
                           metadatafile="MP.txt")
```

Arguments

`data`	An data frame with meteorological data.
`file`	A string with the file name to be written.
`format`	Output format of meteorological data. Current accepted formats are `"castanea"` or `"meteoland"`.
`object`	An object of class `SpatialPointsMeteorology-class` with the meteorological data to be written.
`dir`	Output directory for meteorology data.
`metadatafile`	The name of the file that will store the meta data describing all written files.

Author(s)

Miquel De Cáceres Ainsa, Centre Tecnologic Forestal de Catalunya

Nicolas Martin, INRA-Avignon

See Also

`readmeteorologypoint`, `SpatialPointsMeteorology-class`

Index