Package ‘meteo’

September 24, 2015

Type Package

Title Spatio-Temporal Analysis and Mapping of Meteorological Observations

Version 0.1-5

Date 2015-09-22

Author Milan Kilibarda, Aleksandar Sekulic, Tomislav Hengl, Edzer Pebesma, Benedikt Graeler

Maintainer Milan Kilibarda <kili@grf.bg.ac.rs>

Description Spatio-temporal geostatistical mapping of meteorological data. Global spatio-temporal models calculated using publicly available data are stored in package.

License GPL (>= 2.0) | file LICENCE

URL http://meteo.r-forge.r-project.org/

LazyLoad yes

Depends R (>= 3.0.0)

Imports methods, utils, stats, sp, spacetime, rgdal, gstat, raster, snowfall, plyr

NeedsCompilation no

Repository CRAN

Date/Publication 2015-09-24 11:58:52

R topics documented:

`dprec` *Daily precipitation amount in mm for July 2011*

Description

Sample data set showing values of merged daily precipitation amount measurements from the Global Surface Summary of Day data (GSOD) with European Climate Assessment & Data set (ECA&D) for the month July 2011.

Usage

```
data(dprec)
```

Format

The `dprec` contains the following columns:

`staid` character; station ID from GSOD or ECA&D data set

`time` Date; day of the measurement

`prec` numeric; daily precipitation amount in mm

Note

The data summaries provided here are based on data exchanged under the World Meteorological Organization (WMO) World Weather Watch Program. To prepare a point map, merge with the `stations` table containing stations' coordinates.

Author(s)

Milan Kilibarda and Tomislav Hengl

References

- Global Surface Summary of the day data (ftp://ftp.ncdc.noaa.gov/pub/data/gsod/)
- European Climate Assessment & Dataset (http://eca.knmi.nl/dailydata/predefinedseries.php)

Examples

```
## load data
data(dprec)
str(dprec)
```

---

dslp — *Mean sea level pressure in hPa for July 2011*

Description

Sample data set showing values of merged mean sea level pressure measurements from the Global Surface Summary of Day data (GSOD) with European Climate Assessment & Data set (ECA&D) for the month July 2011.

Usage

```
data(dslp)
```

Format

The `dslp` contains the following columns:

`staid` character; station ID from GSOD or ECA\&D data set

`time` Date; day of the measurement

`slp` numeric; mean sea level pressure amount in hPa

Note

The data summaries provided here are based on data exchanged under the World Meteorological Organization (WMO) World Weather Watch Program. To prepare a point map, merge with the `stations` table containing stations' coordinates.

Author(s)

Milan Kilibarda and Tomislav Hengl

References

- Global Surface Summary of the day data (ftp://ftp.ncdc.noaa.gov/pub/data/gsod/)
- European Climate Assessment & Dataset (http://eca.knmi.nl/dailydata/predefinedseries.php)

Examples

```
## load data
data(dslp)
str(dslp)
```

---

dsndp *Daily snow depth in cm for July 2011*

Description

Sample data set showing values of merged daily snow depth measurements from the Global Surface Summary of Day data (GSOD) with European Climate Assessment & Data set (ECA&D) for the month July 2011.

Usage

```
data(dsndp)
```

Format

The `dsndp` contains the following columns:

`staid` character; station ID from GSOD or ECA&D data set

`time` Date; day of the measurement

`sndp` numeric; daily snow depth in cm

Note

The data summaries provided here are based on data exchanged under the World Meteorological Organization (WMO) World Weather Watch Program. To prepare a point map, merge with the `stations` table containing stations' coordinates.

Author(s)

Milan Kilibarda and Tomislav Hengl

References

- Global Surface Summary of the day data (`ftp://ftp.ncdc.noaa.gov/pub/data/gsod/`)
- European Climate Assessment & Dataset (`http://eca.knmi.nl/dailydata/predefinedseries.php`)

Examples

```
## load data
data(dsndp)
str(dsndp)
```

`dtempc` *Mean daily temperature in degrees Celsius for July 2011*

Description

Sample data set showing values of merged mean daily temperature measurements from the Global Surface Summary of Day data (GSOD) with European Climate Assessment & Dataset (ECA&D) data for the month July 2011.

Usage

```
data(dtempc)
```

Format

The `dtempc` contains the following columns:

`staid` character; station ID from GSOD or ECA&D dataset

`time` Date; day of the measurement

`tempc` numeric; mean daily temperature in degrees Celsius

Note

The data summaries provided here are based on data exchanged under the World Meteorological Organization (WMO) World Weather Watch Program. To prepare a point map, merge with the `stations` table containing stations' coordinates.

Author(s)

Milan Kilibarda and Tomislav Hengl

References

- Global Surface Summary of the day data (`ftp://ftp.ncdc.noaa.gov/pub/data/gsod/`)
- European Climate Assessment & Dataset (`http://eca.knmi.nl/dailydata/predefinedseries.php`)

Examples

```
## load data
data(dtempc)
str(dtempc)
```

`dtemp_maxc` *Maximum daily temperature in degrees Celsius for July 2011*

Description

Sample data set showing values of merged maximum daily temperature measurements from the Global Surface Summary of Day data (GSOD) with European Climate Assessment & Dataset (ECA&D) data for the month July 2011.

Usage

```
data(dtemp_maxc)
```

Format

The `dtemp_maxc` contains the following columns:

`staid` character; station ID from GSOD or ECA&D dataset

`time` Date; day of the measurement

`temp_minc` numeric; maximum daily temperature in degrees Celsius

Note

The data summaries provided here are based on data exchanged under the World Meteorological Organization (WMO) World Weather Watch Program. To prepare a point map, merge with the `stations` table containing stations' coordinates.

Author(s)

Milan Kilibarda and Tomislav Hengl

References

- Global Surface Summary of the day data (`ftp://ftp.ncdc.noaa.gov/pub/data/gsod/`)
- European Climate Assessment & Dataset (`http://eca.knmi.nl/dailydata/predefinedseries.php`)

Examples

```
## load data
data(dtemp_maxc)
str(dtemp_maxc)
```

dtemp_minc — *Minimum daily temperature in degrees Celsius for July 2011*

Description

Sample data set showing values of merged minimum daily temperature measurements from the Global Surface Summary of Day data (GSOD) with European Climate Assessment & Data set (ECA&D) for the month July 2011.

Usage

```
data(dtemp_minc)
```

Format

The `dtemp_minc` contains the following columns:

`staid` character; station ID from GSOD or ECA&D data set

`time` Date; day of the measurement

`temp_minc` numeric; minimum daily temperature in degrees Celsius

Note

The data summaries provided here are based on data exchanged under the World Meteorological Organization (WMO) World Weather Watch Program. To prepare a point map, merge with the `stations` table containing stations' coordinates.

Author(s)

Milan Kilibarda and Tomislav Hengl

References

- Global Surface Summary of the day data (`ftp://ftp.ncdc.noaa.gov/pub/data/gsod/`)
- European Climate Assessment & Dataset (`http://eca.knmi.nl/dailydata/predefinedseries.php`)

Examples

```
## load data
data(dtemp_minc)
str(dtemp_minc)
```

dwdsp *Daily mean wind speed in m/s for July 2011*

Description

Sample data set showing values of merged daily mean wind speed measurements from the Global Surface Summary of Day data (GSOD) with European Climate Assessment & Data set (ECA&D) for the month July 2011.

Usage

```
data(dwdsp)
```

Format

The dwdsp contains the following columns:

staid character; station ID from GSOD or ECA&D data set

time Date; day of the measurement

wdsp numeric; daily mean wind speed in m/s

Note

The data summaries provided here are based on data exchanged under the World Meteorological Organization (WMO) World Weather Watch Program. To prepare a point map, merge with the stations table containing stations' coordinates.

Author(s)

Milan Kilibarda and Tomislav Hengl

References

- Global Surface Summary of the day data (ftp://ftp.ncdc.noaa.gov/pub/data/gsod/)
- European Climate Assessment & Dataset (http://eca.knmi.nl/dailydata/predefinedseries.php)

Examples

```
## load data
data(dwdsp)
str(dwdsp)
```

meteo2STFDF — *Create an object of STFDF class from two data frames (observation and stations)*

Description

The function creates an object of STFDF class, spatio-temporal data with full space-time grid, from two data frames (observation and stations). Observations data frame minimum contains station ID column, time column (day of observation) and measured variable column. Stations data frame contains at least station ID column, longitude (or x) and latitude (or y) column.

Usage

```
meteo2STFDF(obs, stations, obs.staid.time = c(1, 2),
        stations.staid.lon.lat = c(1, 2, 3), crs=CRS(as.character(NA)), delta=NULL )
```

Arguments

`obs`	data.frame; observations data frame minimum contains station ID column, time column (day of observation) and measured variable column. It can contain additional variables (columns).
`stations`	data.frame; Stations data frame contains at least station ID column, longitude (or x) and latitude (or y) column.It can contain additional variables (columns).
`obs.staid.time`	numeric; records the column positions where in `obs` (observation) data frame the station ID and time values are stored.
`stations.staid.lon.lat`	numeric; records the column positions where in `stations` data frame the station ID, longitude (x) and latitude (y) values are stored.
`crs`	CRS; coordinate reference system (see CRS) of `stations` coordinates
`delta`	time; time interval to end points in seconds

Value

STFDF object

Note

The function is intended for conversion of meteorological data to STFDF object, but can be used for similar spatio temporal data stored in two separated tables.

Author(s)

Milan Kilibarda <`kili@grf.bg.ac.rs`>

See Also

`tgeom2STFDF` `pred.strk`

Examples

```
# prepare data
# load observation - data.frame of mean temperatures
data(dtempc)
str(dtempc)
data(stations)
#
str(stations)
lonmin=18 ;lonmax=22.5 ; latmin=40 ;latmax=46
library(sp)
library(spacetime)
serbia = point.in.polygon(stations$lon, stations$lat, c(lonmin,lonmax,lonmax,lonmin),
                          c(latmin,latmin,latmax,latmax))
st= stations[ serbia!=0, ] # stations in Serbia approx.
# create STFDF
temp <- meteo2STFDF(dtempc,st, crs= CRS('+proj=longlat +datum=WGS84'))
str(temp)
```

`nlmodis20110704` *MODIS LST 8 day images image for the Netherlands ('2011-07-04')*

Description

The original 8 day MODIS LST images were also converted from Kelvin to degrees Celsius using the formula indicated in the MODIS user's manual.SpatialGridDataFrame.

Usage

```
data(nlmodis20110704)
```

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

References

Wan, Z., Y. Zhang, Q. Zhang, and Z.-L. Li (2004), Quality assessment and validation of the MODIS global land surface temperature, Int. J. Remote Sens., 25(1), 261-274

Examples

```
data(nlmodis20110704)
# spplot(nlmodis20110704)
```

`nlmodis20110712` *MODIS LST 8 day images image for the Netherlands ('2011-07-12')*

Description

The original 8 day MODIS LST images were also converted from Kelvin to degrees Celsius using the formula indicated in the MODIS user's manual.

Usage

```
data(nlmodis20110712)
```

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

References

Wan, Z., Y. Zhang, Q. Zhang, and Z.-L. Li (2004), Quality assessment and validation of the MODIS global land surface temperature, Int. J. Remote Sens., 25(1), 261-274

Examples

```
data(nlmodis20110712)
# spplot(nlmodis20110712)
```

`NLpol` *The Netherlands border polygon from WCAB*

Description

SpatialGridDataFrame from World Country Admin Boundary Shapefile

Usage

```
data(NLpol)
```

Examples

```
data(NLpol)
##  plot(NLpol) ...
```

pred.strk *Spatio-temporal regression kriging*

Description

Function for spatio-temporal regression kriging prediction based on krigeST. The prediction is made for raster objects, i.e. for STFDF-class objects.

Usage

```
pred.strk(temp,zcol = 1, newdata, pred.id = "tempPred", zero.tol = 0,
  dynamic.cov = c(1, 2), static.cov = c(1, 2),
reg.coef=list(
tmean=c(-0.126504415,0.4051734447,0.4943247727,0.0001837527,-0.0189207588),
tmin = c(-0.9825601517,0.5672140021,0.3344561638, 0.0003119777,-0.0243629638),
tmax = c(1.7873573081,0.350228076, 0.5569091092, 0.0002571338,-0.0012988123)
)[['tmean']],
vgm.model=list( tmean=vgmST("sumMetric",
                                space=vgm( 14.13, "Sph", 5903, 1.933),
                                time =vgm(0, "Sph",  0.1, 0),
                                joint=vgm(9.06, "Sph", 2054, 0.474),
                                 stAni=497.9),
                   tmin = vgmST("sumMetric",
                                space=vgm( 22.682, "Sph", 5725, 3.695),
                                time =vgm(0, "Sph",  0.1, 0),
                                joint=vgm(9.457, "Sph",1888, 1.67),
                                 stAni=485),
                    tmax = vgmST("sumMetric",
                                 space=vgm( 8.31, "Sph", 4930, 2.872),
                                 time =vgm(0, "Sph",  0.1, 0),
                                 joint=vgm(11.175, "Sph", 2117, 1.75),
                                 stAni=527) ) [['tmean']] ,
tiling = FALSE, ntiles = 64, parallel.processing = FALSE, cpus = 3,
sp.nmax = 18, time.nmax = 2,fast = FALSE, computeVar = FALSE,
do.cv = FALSE, only.cv = FALSE, out.remove = FALSE, threshold.res = 15,progress=TRUE)
```

Arguments

`temp`	object of STFDF-class containing dependent variable (observations) in space and time.
`zcol`	variable column name or number showing position of dependent variable in `temp@data`
`newdata`	dynamic and static covariates as STFDF-class object; spatial and temporal overlay with `temp` object must be possible
`pred.id`	identifier of new variable

- `zero.tol` — distance values less than or equal to this threshold value locations are considered as duplicates, see rm.dupl, duplicates are removed to avoid singular covariance matrices in kriging.
- `dynamic.cov` — vector of variable column names or numbers showing position of dynamic covariates in `newdat@data`; dynamic covariates are spatio-temporal explanatory variables, changing in space and time domain
- `static.cov` — vector of variable column names or numbers showing position of static covariates in `newdata@data@sp`; static covariates are spatial explanatory variables changing just in space; static in time dimension
- `reg.coef` — linear regression coefficients; order is assumed as intercept, `dynamic.cov`, `static.cov`. Coefficients can be specified by user; depending on type, number and order of dynamic and static covariates. At the moment the function contains regression coefficient for mean, minimum and maximum temperature calculated globally for GSOD and ECA&D data set on geometrical temperature trend, MODIS LST-8 day, elevation and TWI, see regdata. Coefficients for mean temperature are defined by default.
- `vgm.model` — spatio-temporal variogram of regression residuals, see vgmST. At the moment the function contains spatio-temporal variogram model on residuals for mean, minimum and maximum temperature calculated globally for GSOD and ECA&D data set. Regression residuals on geometrical temperature trend, MODIS LST-8 day, elevation and TWI, see regdata. Ranges are in km. Spatio-temporal variogram for mean temperatures is defined by default. User can specified own variogram model as vgmST object.
- `tiling` — for simplified local kriging. Area is divided in tiles and kriging calculation is done for each tile separately, number of observation used per tile is defined with `sp.nmax` and `time.nmax`. Default is TRUE. If `FALSE` just temporal local kriging will be applied defined with `time.nmax`, `sp.nmax` will be ignored.
- `ntiles` — number of tiles. Default is 64. Each tile at minimum should contain less observations than `sp.nmax`, ideally each tile should contain observations falling in neighboring tiles.
- `parallel.processing` — if TRUE parallel processing is performed via sfLapply
- `cpus` — number of processing units
- `sp.nmax` — number of nearest spatial observations that should be used for a kriging prediction for each tile
- `time.nmax` — number of nearest time observations that should be used for a kriging prediction
- `fast` — if TRUE tiling, tiling is done twice to avoid edge effect
- `computeVar` — if TRUE, just variance is computed
- `do.cv` — if TRUE, cross validation leave-one-station-out is performed
- `only.cv` — if TRUE, only cross validation leave-one-station-out is performed without prediction
- `out.remove` — if TRUE, potential outliers are removed. Removing procedure is iterative, all location with residual higher than defined threshold (`treshold.res`) are selected. Only location with highest cross validation residual is removed, than cross validation is done again, the procedure removing one by one location run until all locations have residuals under defined threshold.

threshold.res	critical threshold for removing potential outliers
progress	if FALSE remove progress bar

Value

An list object containing:

pred	an object of STFDF-class with column contains prediction or variance
cv	cross validation information for points used in prediction, as object of STFDF-class
out	potential outliers, returned as vector of row names of x$cv@sp, only returned if out.remove=FALSE
remst	removed locations as an object of Spatial-class, if out.remove=TRUE
remobs	removed locations with observations as an object of STFDF-class, if out.remove=TRUE

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

References

Kilibarda, M., T. Hengl, G. B. M. Heuvelink, B. Graeler, E. Pebesma, M. Percec Tadic, and B. Bajat (2014), Spatio-temporal interpolation of daily temperatures for global land areas at 1 km resolution, J. Geophys. Res. Atmos., 119, 2294-2313, doi:10.1002/2013JD020803.

See Also

regdata meteo2STFDF tgeom2STFDF

Examples

```
# prepare data
# load observation - data.frame of mean temperatures
data(dtempc)
# str(dtempc)
data(stations)
library(sp)
library(spacetime)
library(gstat)

# str(stations)
## lonmin,lonmax,lonmax, lonmin   latmin, latnmin,latmax,latmax
serbia= point.in.polygon(stations$lon, stations$lat, c(18,22.5,22.5,18), c(40,40,46,46))
st= stations[ serbia!=0, ]
# create STFDF
temp <- meteo2STFDF(dtempc,st)
rm(dtempc)
# str(temp)
# Adding CRS
temp@sp@proj4string <- CRS('+proj=longlat +datum=WGS84')
```

```
# load covariates for mean temperatures
data(regdata)
# str(regdata)
regdata@sp@proj4string <- CRS('+proj=longlat +datum=WGS84')

# Calculate prediction of mean temperatures for "2011-07-05"
# global model is used for regression and variogram
# load precalculated variograms
data(tvgms)
data(tregcoef)
res= pred.strk(temp,zcol=1, newdata= regdata[,1,drop=FALSE],
              reg.coef=tregcoef[[1]] ,vgm.model=tvgms[[1]], progress=FALSE )

## plot prediction
# stplot(res$pred, col.regions=bpy.colors())


# t1= temp[regdata@sp,]
# # create fake observations
# t1@data$tempc[seq(1,120,by=8)] =35
#
#
# res= pred.strk(t1,zcol=1, newdata= regdata[,1:2],
#                 reg.coef=tregcoef[[1]], vgm.model=tvgms[[1]] ,
#                 threshold.res=5, do.cv=T, out.remove = T)
# # plot cross validation residuals
# stplot(res$cv[,,'resid.cv'] , col.regions=bpy.colors())
#
# # plot locations of removed stations
# spplot(res$remst, zcol='station_name' , col.regions=bpy.colors())
# #plot removed stations as time-series
# row.names(res$remobs@sp) = res$remst$station_name
# res$remobs[,1:2,c('tempc','pred.cv')]
# stplot(res$remobs[,1:2,c('tempc','pred.cv')], mode='tp')

## Calculate prediction of mean temperature for "2011-07-05" "2011-07-06"
## only MODIS is used as covariate

# modisVGM =vgmST("sumMetric",space=vgm( 18.27, "Sph", 6000, 3.22),
#                                         time =vgm(0, "Sph",  0.1, 0),
#                                         joint=vgm(8.34, "Sph", 2349, 1.80),
#                                         stAni=583)
# attr(modisVGM,"temporal unit") = "days"

# rkmod <-  pred.strk(temp,zcol=1, newdata= STFDF(regdata@sp,
#                 time=as.POSIXct("2011-07-05"), endTime=as.POSIXct("2011-07-06"),
#                 data=regdata[,1]@data) , threshold.res=10,
#               dynamic.cov='modis', static.cov=NULL,
#                reg.coef= c(-0.23,0.7303284),
#                vgm.model= modisVGM  )

## coefficients and variogram is calculated globally for GSOD and ECA&D obs. for 2011 year
```

```
# stplot(rkmod$pred, col.regions=bpy.colors())

## parallel processing
# library(snowfall)
# rkmod <-  pred.strk(temp,zcol=1,
#                   newdata= STFDF(regdata@sp,
#                    time=as.POSIXct("2011-07-05"), endTime=as.POSIXct("2011-07-06"),
#                    data=regdata[,1]@data) ,
#                     threshold.res=10,
#                     dynamic.cov='modis', static.cov=NULL,
#                     reg.coef= c(-0.23,0.7303284),
#                     vgm.model= modisVGM, parallel.processing=TRUE)
```

---

pred.strk1 *Spatio-temporal regression kriging*

Description

Function for spatio-temporal regression kriging prediction based on krigeST. The prediction is made for points given in data.frame.

Usage

```
pred.strk1(obs, stations, newdata, zero.tol=0,
reg.coef=list(
tmean=c(-0.126504415,0.4051734447,0.4943247727,0.0001837527,-0.0189207588),
tmin = c(-0.9825601517,0.5672140021,0.3344561638, 0.0003119777,-0.0243629638),
tmax = c(1.7873573081,0.350228076, 0.5569091092, 0.0002571338,-0.0012988123)
)[['tmean']],
vgm.model=list(tmean=vgmST("sumMetric",
                                    space=vgm( 14.13, "Sph", 5903, 1.933),
                                    time =vgm(0, "Sph",  0.1, 0),
                                    joint=vgm(9.06, "Sph", 2054, 0.474),
                                    stAni=497.9),
                         tmin = vgmST("sumMetric",
                                    space=vgm( 22.682, "Sph", 5725, 3.695),
                                    time =vgm(0, "Sph",  0.1, 0),
                                    joint=vgm(9.457, "Sph",1888, 1.67),
                                    stAni=485),
                         tmax = vgmST("sumMetric",
                                    space=vgm( 8.31, "Sph", 4930, 2.872),
                                    time =vgm(0, "Sph",  0.1, 0),
                                    joint=vgm(11.175, "Sph", 2117, 1.75),
                                    stAni=527) ) [['tmean']] ,
computeVar=FALSE, out.remove=FALSE, threshold.res=15, returnList=FALSE)
```

Arguments

`obs`	data.frame; observations data frame contains station ID column, time column (day of observation), measured variable column and ovariates columns (in exactly that order). It can contain additional variables (columns).
`stations`	data.frame; Stations data frame contains at least station ID column, longitude (or x) and latitude (or y) column (in exactly that order).It can contain additional variables (columns).
`newdata`	data.frame; predictions data frame contains longitude (or x) and latitude (or y) column, time column (day of prediction) and covariates columns (in exactly that order)
`zero.tol`	distance values less than or equal to this threshold value locations are considered as duplicates, see rm.dupl, duplicates are removed to avoid singular covariance matrices in kriging.
`reg.coef`	linear regression coefficients; order is assumed as intercept, `dynamic.cov`, `static.cov`. Coefficients can be specified by user; depending on type, number and order of dynamic and static covariates. At the moment the function contains regression coefficient for mean, minimum and maximum temperature calculated globally for GSOD and ECA&D data set on geometrical temperature trend, MODIS LST-8 day, elevation and TWI, see regdata. Coefficients for mean temperature are defined by default.
`vgm.model`	spatio-temporal variogram of regression residuals, see vgmST. At the moment the function contains spatio-temporal variogram model on residuals for mean, minimum and maximum temperature calculated globally for GSOD and ECA&D data set. Regression residuals on geometrical temperature trend, MODIS LST-8 day, elevation and TWI, see regdata. Ranges are in km. Spatio-temporal variogram for mean temperatures is defined by default. User can specified own variogram model as vgmST object.
`computeVar`	if TRUE, just variance is computed
`out.remove`	if TRUE, potential outliers are removed. Removing procedure is iterative, all location with residual higher than defined threshold (`treshold.res`) are selected. Only location with highest cross validation residual is removed, than cross validation is done again, the procedure removing one by one location run until all locations have residuals under defined threshold.
`threshold.res`	critical threshold for removing potential outliers
`returnList`	if TRUE, result islist; if FALSE, result is data frame

Value

An list object containing:

`pred`	an object of SpatialPointsDataFrame-class with column contains prediction or variance
`cv`	cross validation information for points used in prediction, as object of STFDF-class
`remst`	removed locations as an object of Spatial-class, if `out.remove=TRUE`

`remobs` removed locations with observations as an object of STFDF-class, if `out.remove=TRUE`

A data frame object contains longitude (or x) and latitude (or y) column, time column (day of prediction) and prediction value.

Author(s)

Aleksandar Sekulic <asekulic@grf.bg.ac.rs>, Milan Kilibarda <kili@grf.bg.ac.rs>

References

Kilibarda, M., T. Hengl, G. B. M. Heuvelink, B. Graeler, E. Pebesma, M. Percec Tadic, and B. Bajat (2014), Spatio-temporal interpolation of daily temperatures for global land areas at 1 km resolution, J. Geophys. Res. Atmos., 119, 2294-2313, doi:10.1002/2013JD020803.

See Also

`regdata` `meteo2STFDF` `tgeom2STFDF`

Examples

```
## prepare data
## load observation - data.frame of mean temperatures
data(dtempc)
str(dtempc)
data(stations)
library(sp)
library(spacetime)
library(gstat)

#str(stations)
## lonmin,lonmax,lonmax, lonmin   latmin, latnmin,latmax,latmax
serbia= point.in.polygon(stations$lon, stations$lat, c(18,22.5,22.5,18), c(40,40,46,46))
st= stations[ serbia!=0, ]
## create STFDF
temp <- meteo2STFDF(dtempc,st)
rm(dtempc)
# str(temp)
## Adding CRS
temp@sp@proj4string <- CRS('+proj=longlat +datum=WGS84')

## load covariates for mean temperatures
data(regdata)
# str(regdata)
regdata@sp@proj4string <- CRS('+proj=longlat +datum=WGS84')

#creating newdata
pred <- data.frame(regdata@sp@coords[1:5,1],regdata@sp@coords[1:5,2],
                  '2011-07-05', regdata@data$temp_geo[1:5], regdata@data$modis[1:5],
                  regdata@sp@data$dem[1:5], regdata@sp@data$twi[1:5])
## pred names
names(pred)=c("x", "y", "time", "temp_geo", "modis", "dem", "twi")
```

```
#creating observation
gg <- regdata
time <- gg@time
gg@data$dem = rep(gg@sp@data[,1],length(time))
gg@data$twi = rep(gg@sp@data[,2],length(time))

temp_geo <- sapply(1:length(time),
               function(i) over(temp@sp,as(gg[,i,'temp_geo'],'SpatialPixelsDataFrame')))
modis <- sapply(1:length(time),
               function(i) over(temp@sp,as(gg[,i,'modis'],'SpatialPixelsDataFrame' ) ) )
dem <- sapply(1:length(time),
              function(i) over(temp@sp,as(gg[,i,'dem'],'SpatialPixelsDataFrame' ) ) )
twi <- sapply(1:length(time),
                function(i) over(temp@sp,as(gg[,i,'twi'],'SpatialPixelsDataFrame' ) ) )

temp_geo <- do.call('cbind',temp_geo)
temp_geo <- as.vector(temp_geo)
modis <- do.call('cbind',modis)
modis <- as.vector(modis)
dem <- do.call('cbind',dem)
dem <- as.vector(dem)
twi <- do.call('cbind',twi)
twi <- as.vector(twi)

t1 <- which(as.character(index(time[1])) == as.character( index(temp@time)) )
t2 <- which(as.character(index(time[ length(time) ])) == as.character(index(temp@time)) )

temp <- temp[,t1:t2, drop=FALSE]

temp$temp_geo = temp_geo
temp$modis = modis
temp$dem = dem
temp$twi = twi

obs = as.data.frame(temp)[,c(7,4,11,12,13,14,15)]

# Calculate prediction of points in pred data frame
# global model is used for regression and variogram
# load precalculated variograms
data(tvgms)
data(tregcoef)
res= pred.strk1(obs,st, newdata= pred,
              reg.coef=tregcoef[[1]] ,vgm.model=tvgms[[1]], returnList=TRUE )

#str(res)

# res1= pred.strk1(obs,st, newdata= pred, reg.coef=tregcoef[[1]] ,vgm.model=tvgms[[1]],
#                returnList=FALSE, out.remove=TRUE, threshold.res=15 )

# str(res1)
```

regdata — *Dynamic and static covariates for spatio-temporal regression kriging*

Description

Dynamic and static covariates for spatio-temporal regression kriging of STFDF-class. The `regdata` contains geometrical temperature trend, MODIS LST 8-day splined at daily resolution, elevation and topographic wetness index.

Usage

```
data(regdata)
```

Format

The `regdata` contains the following dynamic and static covariates:

- `regdata$temp_geo` numeric; geometrical temperature trend for mean temperature, calculated with `tgeom2STFDF` ; from 2011-07-05 to 2011-07-09, in degree Celsius
- `regdata$modis` numeric; MODIS LST 8-day splined at daily resolution, missing pixels are filtered by spatial splines and 8-day values are splined at daily level; from 2011-07-05 to 2011-07-09, in degree Celsius
- `regdata@sp$dem` numeric; elevation data obtained from Worldgrids (`http://worldgrids.org/`)
- `regdata@sp$twi` numeric; SAGA Topographic Wetness Index (TWI) from Worldgrids (`http://worldgrids.org/`)

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

Examples

```
data(regdata)
str(regdata)
library(sp) # spplot
library(spacetime) # stplot

stplot(regdata[,,'modis']) # plot modis data

spplot(regdata@sp,zcol='twi', col.regions = bpy.colors() ) # plot TWI

spplot(regdata@sp,zcol='dem', col.regions = bpy.colors() ) # plot dem
```

rfillspgaps — *Close gaps of a grid or raster Layer data*

Description

The function close gaps of a grid or raster Layer data by using IDW.

Usage

```
rfillspgaps(rasterLayer,maskPol=NULL,nmax =50,zcol=1, ...)
```

Arguments

`rasterLayer`	raster Layer or SpatiaGrid(Pixels)DF containing NAs
`maskPol`	SpatialPolygons or SpatialPolygonsDataFrame
`nmax`	see idw
`zcol`	variable column name or number showing position of variable in `rasterLayer` to be interpolated
`...`	arguments passed to idw

Value

raster Layer or SpatiaGrid(Pixels)DF object with NA replaced using IDW

Author(s)

Milan Kilibarda `<kili@grf.bg.ac.rs>`

References

Kilibarda, M., T. Hengl, G. B. M. Heuvelink, B. Graeler, E. Pebesma, M. Percec Tadic, and B. Bajat (2014), Spatio-temporal interpolation of daily temperatures for global land areas at 1 km resolution, J. Geophys. Res. Atmos., 119, 2294-2313, doi:10.1002/2013JD020803;

Kilibarda M., M. Percec Tadic, T. Hengl, J. Lukovic, B. Bajat - Spatial Statistics (2015), Global geographic and feature space coverage of temperature data in the context of spatio-temporal interpolation, doi:10.1016/j.spasta.2015.04.005.

See Also

`rfilltimegaps` `pred.strk`

Examples

```
library(raster)
data(nlmodis20110712)
data(NLpol)

# spplot(nlmodis20110712, col.regions=bpy.colors())
# fill spatial gaps
r=rfillspgaps(nlmodis20110712,NLpol)
# spplot(r, col.regions=bpy.colors())
```

---

rfilltimegaps — *Disaggregation in the time dimension through the use of splines for each pixel*

Description

The function creates an object of STFDF class, spatio-temporal data with full space-time grid, from another STFDF and fills attribute data for missing dates using splines.

Usage

```
rfilltimegaps(stfdf,tunits="day",attrname=1, ...)
```

Arguments

`stfdf`	STFDF object with time information of minimum length 2, and gap between
`tunits`	increment of the sequence used to generete time infromation for temporal gap. See 'Details'
`attrname`	varible from STFDF to be splined
`...`	arguments passed to spline

Details

`tunits` can be specified in several ways:

- A number, taken to be in seconds
- A object of class difftime
- A character string, containing one of `"sec"`, `"min"`, `"hour"`, `"day"`, `"DSTday"`, `"week"`, `"month"`, `"quarter"` or `"year"`. This can optionally be preceded by a (positive or negative) integer and a space, or followed by `"s"`

The difference between `"day"` and `"DSTday"` is that the former ignores changes to/from daylight savings time and the latter takes the same clock time each day. (`"week"` ignores DST (it is a period of 144 hours), but `"7 DSTdays"`) can be used as an alternative. `"month"` and `"year"` allow for DST.)

Value

STFDF object

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

References

Kilibarda, M., T. Hengl, G. B. M. Heuvelink, B. Graeler, E. Pebesma, M. Percec Tadic, and B. Bajat (2014), Spatio-temporal interpolation of daily temperatures for global land areas at 1 km resolution, J. Geophys. Res. Atmos., 119, 2294-2313, doi:10.1002/2013JD020803;

Kilibarda M., M. Percec Tadic, T. Hengl, J. Lukovic, B. Bajat - Spatial Statistics (2015), Global geographic and feature space coverage of temperature data in the context of spatio-temporal interpolation, doi:10.1016/j.spasta.2015.04.005.

See Also

rfillspgaps pred.strk

Examples

```
data(nlmodis20110704)
data(nlmodis20110712)


# fill spatial gaps
library(raster)
NLpol <- as(extent(5, 6, 51.5, 52), 'SpatialPolygons')
NLpol@proj4string <- nlmodis20110704@proj4string
nlmodis20110704 <- rfillspgaps(nlmodis20110704,NLpol)
nlmodis20110712 <- rfillspgaps(nlmodis20110712,NLpol)

nlmodis20110704 <- as(nlmodis20110704,"SpatialPixelsDataFrame")
names(nlmodis20110704)='m1'
nlmodis20110712 <- as(nlmodis20110712,"SpatialPixelsDataFrame")
names(nlmodis20110712)='m2'

nlmodis20110704@data <- cbind(nlmodis20110704@data, nlmodis20110712@data)

df<-reshape(nlmodis20110704@data , varying=list(1:2), v.names="modis",direction="long",
          times=as.Date(c('2011-07-04','2011-07-12')), ids=1:dim(nlmodis20110704)[1])

library(spacetime)
stMODIS<- STFDF(as( nlmodis20110704, "SpatialPixels"),
                time= as.Date(c('2011-07-04','2011-07-12')),
                data.frame(modis=df[,'modis']))

# stplot(stMODIS, col.regions=bpy.colors())
stMODIS <- rfilltimegaps(stMODIS)
# stplot(stMODIS, col.regions=bpy.colors())
```

rm.dupl *Find point pairs with equal spatial coordinates from STFDF object.*

Description

This function finds point pairs with equal spatial coordinates from STFDF object and remove locations with less observations.

Usage

```
rm.dupl(obj, zcol = 1, zero.tol = 0)
```

Arguments

obj	STFDF object
zcol	variable column name, or column number, from obj@data
zero.tol	distance values less than or equal to this threshold value are considered as duplicates; units are those of the coordinates for projected data or unknown projection, or km if coordinates are defined to be longitute/latitude

Value

STFDF object with removed duplicate locations. Stations with less observation is removed, if number of observation is the same for two stations the first is removed.

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

tgeom2STFDF pred.strk

Examples

```
# load observation - data frame
data(dtempc)
# load stations - data frame
data(stations)

str(dtempc)
str(stations)
# create STFDF
temp <- meteo2STFDF(dtempc,stations) # create STFDF from 2 data frames

nrow(temp@sp) # number of stations before removing dupl.

temp2 <-rm.dupl(temp)
nrow(temp2@sp) # number of stations after
```

stations *Data frame containing stations' information*

Description

Data frame containing stations' information of merged daily observations from the Global Surface Summary of Day (GSOD) with European Climate Assessment & Data set (ECA&D) for the month July 2011.

Usage

```
data(stations)
```

Format

The `stations` contains the following columns:

`staid` character; station ID from GSOD or ECA&D data set

`lon` numeric; longitude coordinate

`lat` numeric; longitude coordinate

`elev_1m` numeric; elevation derived from station metadata in m

`data_source` Factor; data source, GSOD or ECA&D

`station_name` character; station name

Author(s)

Milan Kilibarda and Tomislav Hengl

References

- Global Surface Summary of the day data (ftp://ftp.ncdc.noaa.gov/pub/data/gsod/)
- European Climate Assessment & Dataset (http://eca.knmi.nl/dailydata/predefinedseries.php)

Examples

```
## load data:
data(stations)
str(stations)
library(sp)
coordinates(stations) <-~ lon +lat
stations@proj4string <-CRS('+proj=longlat +datum=WGS84')
plot(stations)
```

`tgeom2STFDF` *Calculate geometrical temperature trend*

Description

Calculate geometrical temperature trend for mean, minimum or maximum temperature.

Usage

```
tgeom2STFDF(grid, time, variable = "mean",ab=NULL)
```

Arguments

`grid`	object of Spatial-class (Points, Grid or Pixels) with associated coordinate reference systems (CRS-class). If CRS is not defined longitude latitude is assumed.
`time`	object holding time information, reasonably it is day (calendar date), or vector of days
`variable`	character;`'mean'`, `'min'` or `'max'` ; geometrical temperature trend is calculated for mean, minimum or maximum; `'mean'` is default.
`ab`	Predefined coefficients to be used instead of incorporated.

Value

STFDF object with calculated `temp_geo` geometrical temperature trend. The calculated values are stored in `obj@data` slot.

Author(s)

Milan Kilibarda <`kili@grf.bg.ac.rs`>

References

Kilibarda, M., T. Hengl, G. B. M. Heuvelink, B. Graeler, E. Pebesma, M. Percec Tadic, and B. Bajat (2014), Spatio-temporal interpolation of daily temperatures for global land areas at 1 km resolution, J. Geophys. Res. Atmos., 119, 2294-2313, doi:10.1002/2013JD020803.

Examples

```
library(sp)
library(spacetime)
## create one point from lon lat
pos <- SpatialPoints(coords = cbind(19.22,45.33))
## temp_geom for 1st Jan 2011
tg1 <- tgeom2STFDF(pos,as.POSIXct("2011-01-01") )
tg1

## temp_geom for the 2011 at pos location
tg365<- tgeom2STFDF(pos,time = seq(as.POSIXct("2011-01-01"), as.POSIXct("2011-12-31"),
```

```
                    by="day") )
stplot(tg365, mode='ts')

data(regdata)
## DEM and TWI data for Serbia at 1 km resolution
# str(regdata@sp)
spplot(regdata@sp, zcol='dem', col.regions=bpy.colors() )

## temp_geom for Serbia 1st and 2nd Jully 2011
tgSrb<- tgeom2STFDF(regdata@sp,time = seq(as.POSIXct("2011-07-01"),
                    as.POSIXct("2011-07-02"), by="day") )

## temp_geom for "2011-07-01" , "2011-07-02"
# stplot(tgSrb, col.regions = bpy.colors() )
```

---

tiling — *Tiling raster or Spatial-class Grid or Pixels object*

Description

Tiling raster or Spatial-class Grid or Pixels (data frame) object to smaller parts with optional overlap.

Usage

```
tiling(filename,tilesize=500,overlapping=50, aspoints= FALSE,
       asfiles=FALSE,tilename="tile", format="GTiff",
      tiles_folder=paste(getwd(),'tiles',sep='/'), parallel.processing=FALSE, cpus=6)
```

Arguments

`filename`	raster object, SpatialPixels* object, SpatialGrid* object or file path of raster object stored on the disk (can be read via readGDAL), for more details see raster. The resolution of object should be the same in x and y direction.
`tilesize`	tile size in cells in x direction. nx = ny is assumed, total number of tile cells is `tilesize` by `tilesize`.
`overlapping`	overlapping cells in each direction
`aspoints`	if TRUE tiles are in form of `SpatialPointsDataFrame`
`asfiles`	if TRUE tiles are stored on local drive as raster objects
`tilename`	prefix given to file names
`format`	file format, see writeRaster
`tiles_folder`	folder to be created for tiles storage
`parallel.processing`	if TRUE parralel processing is performed via sfLapply
`cpus`	number of proccesing units

Value

The list of tiles in raster format or in `SpatialPointsDataFrame` format if `aspoints=TRUE`

Author(s)

Milan Kilibarda <kili@grf.bg.ac.rs>

See Also

pred.strk

Examples

```
data(regdata)
str(regdata@sp) # DEM and TWI data for Serbia at 1 km resolution
dem=regdata@sp['dem']
library(sp)
spplot(dem,  col.regions=bpy.colors() )

str(dem)
# tiling dem in tiles 250x250 with 25 cells overlap
tiles = tiling(dem,tilesize=250,overlapping=25,aspoints=TRUE)
# number of tiles
length(tiles)

image(dem)
points(tiles[[1]] , pch='-' ,col ='green')
points(tiles[[2]], pch='.')

str(tiles[[1]])
```

---

tregcoef — *Multiple linear regression coefficients for global daily air temperatures*

Description

Multiple linear regression coefficients for mean, min., max. daily temperature on geometric temperature trend, MODIS LST, elevation, and topographic wetness index. The models is computed from GSOD and ECA&D and GHCN-Daily data.

Usage

```
data(tregcoef)
```

Format

A list of 8 multiple linear regression coefficients for daily air temperatures.

`tmeanGSODECAD` Multiple linear regression coefficients of mean daily temperature on geometric temperature trend, MODIS LST, elevation, and topographic wetness index, data used: GSOD and ECA&D

`tmeanGSODECAD_noMODIS` Multiple linear regression coefficients of mean daily temperature on geometric temperature trend, elevation, and topographic wetness index, data used: GSOD and ECA&D

`tminGSODECAD` Multiple linear regression coefficients of min. daily temperature on geometric temperature trend, MODIS LST, elevation, and topographic wetness index, data used: GSOD and ECA&D

`tminGHCND` Multiple linear regression coefficients of min. daily temperature on geometric temperature trend, MODIS LST, elevation, and topographic wetness index, data used: GHCN-Daily

`tminGSODECAD_noMODIS` Multiple linear regression coefficients of min. daily temperature on geometric temperature trend, elevation, and topographic wetness index, data used: GSOD and ECA&D

`tmaxGSODECAD` Multiple linear regression coefficients of max. daily temperature on geometric temperature trend, MODIS LST, elevation, and topographic wetness index, data used: GSOD and ECA&D

`tmaxGHCND` Multiple linear regression coefficients of max. daily temperature on geometric temperature trend, MODIS LST, elevation, and topographic wetness index, data used: GHCN-Daily

`tmaxGSODECAD_noMODIS` Multiple linear regression coefficients of max. daily temperature on geometric temperature trend, elevation, and topographic wetness index, data used: GSOD and ECA&D

Author(s)

Milan Kilibarda

References

Kilibarda, M., T. Hengl, G. B. M. Heuvelink, B. Graeler, E. Pebesma, M. Percec Tadic, and B. Bajat (2014), Spatio-temporal interpolation of daily temperatures for global land areas at 1 km resolution, J. Geophys. Res. Atmos., 119, 2294-2313, doi:10.1002/2013JD020803.

Examples

```
data(tregcoef)
tregcoef[[1]] # model for mean daily temp.
```

tvgms *Spatio-temporal variogram models for global daily air temperatures*

Description

Variograms of residuals from multiple linear regression of mean, min., max. daily temperatures on geometric temperature trend, MODIS LST, elevation, and topographic wetness index. The models is computed from GSOD and ECA&D and GHCN-Daily data. The obtained global models for mean, minimum, and maximum temperature can be used to produce gridded images of daily temperatures at high spatial and temporal resolution.

Usage

```
data(tvgms)
```

Format

A list of 8 space-time sum-metric models for daily air temperatures, units: space km, time days.

`tmeanGSODECAD` Variogram for residuals from multiple linear regression of mean daily temperature on geometric temperature trend, MODIS LST, elevation, and topographic wetness index, data used: GSOD and ECA&D

`tmeanGSODECAD_noMODIS` Variogram for residuals from multiple linear regression of mean daily temperature on geometric temperature trend, elevation, and topographic wetness index, data used: GSOD and ECA&D

`tminGSODECAD` Variogram for residuals from multiple linear regression of min. daily temperature on geometric temperature trend, MODIS LST, elevation, and topographic wetness index, data used: GSOD and ECA&D

`tminGHCND` Variogram for residuals from multiple linear regression of min. daily temperature on geometric temperature trend, MODIS LST, elevation, and topographic wetness index, data used: GHCN-Daily

`tminGSODECAD_noMODIS` Variogram for residuals from multiple linear regression of min. daily temperature on geometric temperature trend, elevation, and topographic wetness index, data used: GSOD and ECA&D

`tmaxGSODECAD` Variogram for residuals from multiple linear regression of max. daily temperature on geometric temperature trend, MODIS LST, elevation, and topographic wetness index, data used: GSOD and ECA&D

`tmaxGHCND` Variogram for residuals from multiple linear regression of max. daily temperature on geometric temperature trend, MODIS LST, elevation, and topographic wetness index, data used: GHCN-Daily

`tmaxGSODECAD_noMODIS` Variogram for residuals from multiple linear regression of max. daily temperature on geometric temperature trend, elevation, and topographic wetness index, data used: GSOD and ECA&D

Author(s)

Milan Kilibarda

References

Kilibarda, M., T. Hengl, G. B. M. Heuvelink, B. Graeler, E. Pebesma, M. Percec Tadic, and B. Bajat (2014), Spatio-temporal interpolation of daily temperatures for global land areas at 1 km resolution, J. Geophys. Res. Atmos., 119, 2294-2313, doi:10.1002/2013JD020803.

Examples

```
data(tvgms)
tvgms[[1]] # model for mean daily temp.
```

Index