Package: MetGen (via r-universe)

September 9, 2024

Type Package

Title Stochastic Weather Generator

Version 0.5

Date 2020-05-29

Author Julie Carreau, Nesrine Farhani

Maintainer Nesrine Farhani <nesrine.farhani@ird.fr>

Description An adaptation of the multi-variable stochastic weather generator proposed in 'Rglimclim' to perform gap-filling and temporal extension at sub-daily resolution. Simulation is performed based on large scale variables and climatic observation data that could be generated from different gauged stations having geographical proximity. SWG relies on reanalyses. Multi-variable dependence is taking into account by using the decomposition of the product rule (in statistics) into conditional probabilities. See <https://hal.archives-ouvertes.fr/hal-02554676>.

License GPL (>= 2.0)

LazyLoad yes

LazyData true

Depends R (>= 3.5.0), chron, glmnet, MASS

URL www.r-project.org

NeedsCompilation no

Repository CRAN

Date/Publication 2020-06-04 09:40:10 UTC

Contents

`MetGen-package` *Stochastic Weather Generator*

Description

An adaptation of the multi-variable stochastic weather generator (SWG) proposed in Rglimclim to perform gap-filling and temporal extension at sub-daily resolution. Simulation is performed based on large scale variables and climatic observation data that could be generated from different gauged stations having geographical proximity. SWG relies on ERA5 reanalyses. Multivariable dependance is taking into account by using the decomposition of the product rule (in statistics) into conditional probabilities. See Farhani et al 2020 https://hal.archives-ouvertes.fr/hal-02554676

Details

The DESCRIPTION file:

Package:	MetGen
Type:	Package
Title:	Stochastic Weather Generator
Version:	0.5
Date:	2020-05-29
Author:	Julie Carreau, Nesrine Farhani
Maintainer:	Nesrine Farhani <nesrine.farhani@ird.fr>
Description:	An adaptation of the multi-variable stochastic weather generator proposed in 'Rglimclim' to perform gap-fillin
License:	GPL (>=2.0)
LazyLoad:	yes
LazyData:	true
Depends:	chron, glmnet, MASS
URL:	www.r-project.org

The SWG is based on generalized linear models (GLMs) for each hydro-meteorological variable with a suitable probability distribution (Normal, Gamma or Binomial) and appropriate covariates.

Covariates are considered in the GLMs to account for temporal and spatial variability. The inter-variable dependencies are taken into account by including a subset of hydro-meteorological variables (excluding the one being modelled) in the covariates of the GLMs. Large-scale atmospheric variables and deterministic effects (seasonal and diurnal cycles, geographical information and temporal persistence) are also included in the covariates.

Index of help topics:

```
MetGen-package          Stochastic Weather Generator
diurnal.effect          diurnal and seasonal effects
fit.glm                 Fitting Generalized Linear Models
fit.lasso               Lasso regression
formating.var           Format variables
imputation.lagged       Imputation
lagged.effect           lag effect
movave.effect           Moving average
myclimatic_data         climatic data
mycovariates            Covariates data
projection.lagged       climatic variable simulation
rm.buffer               Remove buffer
sim.glm                 Simulation of glm
spatave.effect          Spatial average
```

1- This package helps to define the probability distribution and covariates for each hydro-meteorological variables using `fit.lasso` function.

2- The package can be used either in a gap filling mode in which missing values in observation period are imputed using `imputation.lagged` function, or in a projection mode in which the generator simulates values on a period with no observations to perform temporal extension using `projection.lagged` function.

Author(s)

Julie Carreau, Nesrine Farhani

Maintainer: Nesrine Farhani <nesrine.farhani@ird.fr>

References

Chandler, R. 2015. A multisite, multivariate daily weather generator based on Generalized Linear 9 Models. User guide:R package.

Farhani N, Carreau J, Kassouk Z, Mougenot B, Le Page M, et al.. Sub-daily stochastic weather generator based on reanalyses for water stress retrieval in central Tunisia. 2020. (hal-02554676)

See Also

`fit.glm`

Cycle.effect *diurnal and seasonal effects*

Description

Create covariates to reproduce seasonal and diurnal variability basing on the time

of the year or of the day.

Seasonal effects are expressed by cos(2*pi*d/k) and sin(2*pi*d/k),

with d is the day of the year and k=(365, 183, 96 or 30),

Diurnal effects are expressed by cos(2*pi*h/k) and sin(2*pi*h/k),

with h is the hour of the day and k=(24, 12 or 6).

Usage

```
diurnal.effect(var.mat, period = 24)
seasonal.effect(var.mat, period = 365.25)
```

Arguments

var.mat	data frame that contains chron variable, possibly including times, geographical information for the measurement site and climatic variables for each time step,
period	a numeric vector that contains the period of the cycle desired, could be (183, 365, 91 or 30) for seasonal variation or (24, 12 or 6) for diurnal variation

Value

The same data frame defined in the arguments is returned with supplementary number of columns that contains cycle effects. The number of additional columns depends on the period defined in arguments.

See Also

myclimatic_data

Examples

```
##Perform cycle effects
seasonal_effects <- seasonal.effect(myclimatic_data, period=c(365,183,91,30))
diurnal_effects <- diurnal.effect(myclimatic_data, period=c(24,12,6))
```

fit.glm *Fitting Generalized Linear Models*

Description

fit.glm is used to fit generalized linear models, taking account of covariates specified in the argument list

Usage

```
fit.glm(var.name, dep.var=NULL, geocov=TRUE, large.var, seasonal = TRUE,
speriod = 365.25, diurnal = TRUE, dperiod = 24, spatave=TRUE, lagspat,
movave = TRUE, bwM = 48, lagmov, spatmovave= TRUE, bwSM = 48, lagspatmov,
lagvar, add.cov= FALSE, others=NULL,  fam.glm = "gaussian", data)
```

Arguments

var.name	character that forms the name of the climatic variable to be fitted
dep.var	character that forms the name of depending variables
geocov	logical value indicating whether geographical information is part of covariates set
large.var	character object that forms the name of the large scale variable
seasonal	logical value indicating whether seasonal effects are among the covariates defined for the variable that will be fitted
speriod	A numeric vector that contains the lenght of seasonal cycles defined for the variable of interest
diurnal	logical value indicating whether diurnal effects are among the covariates defined for the variable that will be fitted
dperiod	A numeric vector that contain the lenght of diurnal cycles defined for the variable of interest
spatave	logical value indicating whether spatial average effects are among the covariates defined for the variable that will be fitted
lagspat	Numeric vector indicating values of lags performed for the spatial average
movave	logical value indicating whether moving average effects are among the covariates defined for the variable that will be fitted
bwM	A numeric vector that contains the bandwidth defined for the moving average
lagmov	Numeric vector indicating values of lags performed for the moving average
spatmovave	logical value indicating whether spatial moving average effects are among the covariates defined for the variable that will be fitted
bwSM	A numeric vector that contains the bandwidth defined for the spatial moving average
lagspatmov	Numeric vector indicating values of lags performed for the spatial moving average

lagvar	Numeric vector indicating values of lags performed for the variable
add.cov	logical value indicatng whether we have an additional covariates defined for the variable to fit
others	character that forms the name of additional covariates specified for the variable to fit
fam.glm	family objects to specify probability distribution that will be used for the simulation mode ("gaussian", "gaussian-hetero", "binomial" or "Gamma")
data	data frame that contains a variable named "dates" in chron format, possibly including times, geographical information for the measurement site, climatic variable to be fitted and all different covariates that will be used to fit the variable of interest

Value

value returned is a variable fitted which inherits from the class "lm"

See Also

diurnal.effect, seasonal.effect, lagged.effect

Examples

```
##Create a new data that contains climatic series and all effects that will be
##used as covariates for the variable to fit
mat_effects <- seasonal.effect(myclimatic_data, period=c(365,183))
mat_effects <- diurnal.effect(mat_effects, period=24)
mat_effects <- lagged.effect(mat_effects, "temp",2, nstat=3)
##Add a large scale variable
mat_effects$t2m <- rnorm(nrow(myclimatic_data), mean=25, sd=1)
temp_fitted <- fit.glm("temp", dep.var = "Rh", geocov=TRUE, large.var="t2m",
seasonal = TRUE, speriod = c(365, 183), diurnal = TRUE, dperiod = 24,
spatave = FALSE, movave = FALSE, spatmovave= FALSE, add.cov = FALSE, others = NULL,
lagvar=2, fam.glm = "gaussian", data=mat_effects)
```

---

fit.lasso *Lasso regression*

Description

Fit a generalized linear model via penalized maximum likelihood. LASSO regression is applied to perform a preliminary screening for a large covariate set. It solves a regularized least squares problem recognized for its potential to perform simultaneously variable selection and parameter estimation. It might be used only for gaussian cases to yield parsimonious solutions, i.e. with few covariates.

Usage

```
fit.lasso(cov.mat, var.name)
```

Arguments

`cov.mat`	data frame that contains only the variable to be fitted, and a large set of covariates that will be tested to choose the parsimonious ones relying on Lasso coefficient estimates
`var.name`	character object that forms the name of the variable to be fitted

Value

an object of class "cv.glmnet" is returned, which is a list with the ingredients of the cross-validation fit.

See Also

`glmnet`, `cv.glmnet`

Examples

```
##Create a new data that contains climatic series and all effects that will be
##used as covariates for the variable to be computed
mat_effects <- seasonal.effect(myclimatic_data, period=c(365,183))
mat_effects <- diurnal.effect(mat_effects, period=24)
mat_effects <- spatave.effect(mat_effects, "temp", nstat = 3, na.proc = TRUE)
mat_effects <- lagged.effect(mat_effects, "temp",2, nstat=3)
mat_effects$t2m <- rnorm(nrow(myclimatic_data), mean=25, sd=1)

fit.lasso(mat_effects[,3:length(mat_effects)], "temp")
```

`formating.var` *Format variables*

Description

Extraction of a variable of interest for each site separetly

Usage

```
formating.var(var.mat, var.name, nstat)
```

Arguments

`var.mat`	data frame containing chron variable, geographical information and different climatic variables for each time step and for each site
`var.name`	character object that forms the name of the climatic variable to be picked up from the data frame according to sites
`nstat`	numeric vector that contains the number of gauged sites included in the data frame

Value

a new data frame is returned with time steps and temporal series of the variable of interest arranged in columns according to sites

Examples

```
nstat <- 3
temp_mat <- formating.var(myclimatic_data,"temp", nstat)
```

---

imputation.lagged *Imputation*

Description

Gap filling on time steps with no data during the observation period.

Usage

```
imputation.lagged(fit, var.name, maxlag, coord, cov = NULL,
seasonal = TRUE, speriod = 365.25, diurnal = TRUE, dperiod = 24,
spatave = TRUE, movave = TRUE, bw = 48, na.proc = TRUE,
fam.glm = "gaussian", occ.cond = NULL, init.buff = 1440)
```

Arguments

`fit`	Fitted model derived from `fit.glm`
`var.name`	Charcter object that forms the name of the climatic variable to be imputed
`maxlag`	Numeric vector that forms the maximum amount of lag defined for the fitted model
`coord`	Data frame of two columns (x and y) that contains geographical coordinates of each site
`cov`	Data frame that contains chron object and all external covariates used to fit each climatic variables that will be imputed. These covariates must be available for a buffer period in addition of the period that will be imputed.
`seasonal`	A logical value indicating whether seasonal effects are among the covariates defined for the variable that will be imputed
`speriod`	A numeric vector that contains the lenght of seasonal cycles defined for the variable of interest
`diurnal`	A logical value indicating whether diurnal effects are among the covariates defined for the variable that will be imputed
`dperiod`	A numeric vector that contains the lenght of diurnal cycles defined for the variable of interest
`spatave`	A logical value indicating whether spatial average effects are among the covariates defined for the variable that will be imputed

movave	A logical value indicating whether moving average effects are among the covariates defined for the variable that will be imputed
bw	A numeric vector that contains the bandwidth defined for the moving average
na.proc	A logical value indicating whether NA values should be stripped before imputation
fam.glm	Family objects to specify probability distribution that will be used for the simulation in the missing period ("gaussian", "gaussian-hetero", "binomial" or "Gamma")
occ.cond	character object that specifies the occurrence variable if that exists
init.buff	A buffer time is an extra time added before simulation to keep the simulation on track. the init.buffer is numeric vector, defined according to the number of climatic observations per the day and the number of few days that we choose as a buffer time before starting the simulation

Value

An additional column in the cov data frame of the variable of interest with no more missing values

See Also

fit.glm, glm

Examples

```
myclimatic_data$dates=myclimatic_data$JD

##random removal of 30 percent of climatic observations to comput artificially
##missing values
n.miss=round(nrow(myclimatic_data)*0.30)
ind_miss=sample(nrow(myclimatic_data), n.miss)
myclimatic_data$temp[ind_miss]=NA

##Create a new data that contains climatic series and all effects that will be used
##as covariates for the variable to be computed
temp.effects <- seasonal.effect(myclimatic_data, period=c(365,183))
temp.effects <- diurnal.effect(temp.effects, period=24)
temp.effects <- lagged.effect(temp.effects, "temp",2, nstat=3)
temp.effects$t2m <- rnorm(nrow(myclimatic_data),mean=25,sd=1)

coord <- data.frame(x=c(9.92,9.93,10.04),y=c(35.55,35.62,35.57))
nstat=3
init.buff=48*7 ##48 time step per day and 7 days will be considered as buffer time

##fitted variable
temp.fitted <- fit.glm("temp", dep.var = NULL, geocov=TRUE, large.var="t2m",
seasonal = TRUE, speriod = c(365, 183), diurnal = TRUE, dperiod = 24,
spatave = FALSE, movave = FALSE, spatmovave= FALSE, lagvar=2, add.cov = FALSE,
others = NULL, fam.glm = "gaussian", data= temp.effects)

temp.imputation <- imputation.lagged(temp.fitted, "temp", maxlag=2, coord,
cov=mycovariates, seasonal = TRUE, speriod = c(365,183), diurnal = TRUE,
dperiod = 24, spatave = FALSE, movave=FALSE,bw = 0, fam.glm = "gaussian")
```

lagged.effect *lag effect*

Description

Introduce lag effects for variables

Usage

```
lagged.effect(var.mat, var.name, maxlag, nstat = NULL)
```

Arguments

var.mat	A data frame containing a chron variable and different climatic variables for each time step
var.name	Character object that forms the name of the climatic variable to be lagged
maxlag	Numeric vector that specifies the maximum amount of lag requested
nstat	Numeric vector that specifies the number of sites considered

Value

lagged effect for the variable var.name, will be observed, according to the amount of maxlag defined in the data frame introduced

Examples

```
data_lagged4 <- lagged.effect(myclimatic_data, "temp",4, nstat=3)
data_lagged8 <- lagged.effect(myclimatic_data, "temp",8, nstat=3)
```

movave.effect *Moving average*

Description

Averaging previous time steps using different bandwidth to perform temporal auto-correlation

Usage

```
movave.effect(var.mat, var.name, bw, nstat = NULL, na.proc = FALSE)
```

Arguments

`var.mat`	Data frame containing chron variable and different climatic variables for each time step
`var.name`	character object to define the name of the climatic variable that we will perform the moving average
`bw`	Numeric vector that forms the bandwidth to be used for the moving average
`nstat`	Numeric vector specifying the number of gauged stations used to obtain observation data
`na.proc`	A logical value indicating whether NA values should be stripped before the computation proceeds

Value

An additional column will be added in the data frame introduced containing the average of the variable of interest over previous time step specified by the bandwith for each site

Examples

```
temp_movave <- movave.effect(myclimatic_data, "temp", 48, nstat = 3,
na.proc = TRUE)
```

`myclimatic_data` *climatic data*

Description

A synthetic climatic data that contains time series, geographical coordinates and different climatic variables for a subdaily resolution. Each row corresponds for a time step, and each time is available for a site.

Usage

```
myclimatic_data
```

Format

A data frame with 4320 observations on the following 6 variables.

`dates` a chron object to define time steps

`JD` a numeric vector sepcifying julian days

`coord.x` a numeric vector to define the x coodinates of different sites

`coord.y` a numeric vector to define the y coodinates of different sites

`temp` a numeric vector for a climatic variable (temperature)

`Rh` a numeric vector for a climatic variable (relative humidity) mycovariates *Covariates data*

Description

Covariates data that contains time series, geographical coordinates and different climatic variables for a subdaily resolution. Each row corresponds for a time step, and each time is available for a site.

Usage

```
mycovariates
```

Format

A data frame with 4320 observations on the following 18 variables.

`dates` a numeric vector for the time steps

`coord.x` a numeric vector to define the x coodinates of different sites

`coord.y` a numeric vector to define the y coodinates of different sites

`t2m` a numeric vector for a large scale variable obtained from reanalysis

`cos.365d` a numeric vector for the seasonal variabiliy throughout 365 days

`sin.365d` a numeric vector for the seasonal variabiliy throughout 365 days

`cos.183d` a numeric vector for the seasonal variabiliy throughout 183 days

`sin.183d` a numeric vector for the seasonal variabiliy throughout 183 days

`cos.91d` a numeric vector for the seasonal variabiliy throughout 91 days

`sin.91d` a numeric vector for the seasonal variabiliy throughout 91 days

`cos.30d` a numeric vector for the seasonal variabiliy throughout 30 days

`sin.30d` a numeric vector for the seasonal variabiliy throughout 30 days

`cos.24h` a numeric vector for diurnal variabiliy during a day

`sin.24h` a numeric vector for diurnal variabiliy during a day

`cos.12h` a numeric vector for diurnal variabiliy during half a day

`sin.12h` a numeric vector for diurnal variabiliy during half a day

`cos.6h` a numeric vector for diurnal variabiliy during 6 hours

`sin.6h` a numeric vector for diurnal variabiliy during 6 hours projection.lagged *climatic variable simulation*

Description

Projection of climatic variables for a specific period defined by the user. It forms a free simulation from fitted model constraining by covariates that must be available for the period of projection.

Usage

```
projection.lagged(dstart, dend, fit, var.name, maxlag, coord, cov = NULL,
seasonal = TRUE, speriod = 365.25, diurnal = TRUE, dperiod = 24,
spatave = TRUE, movave = TRUE, bw = 48, na.proc = TRUE,
fam.glm = "gaussian", occ.cond = NULL, init.buff = 1440)
```

Arguments

dstart	Numeric that defines the first time step of the simulation
dend	Numeric that defines the last time step of the simulation
fit	fitted model derived from glm or fit.glm
var.name	Character that defines the name of the climatic variable to be simulated
maxlag	Numeric vector that specifies the maximum amount of lag defined for the fitted model
coord	Data frame of two columns (x and y) that contains geographical coordinates of each site
cov	Data frame that contains dates and all external covariates used to fit each climatic variables that will be simulated. These covariates must be available for a buffer period in addition of the period of simulation that start at "dstart" and finish at "dend".
seasonal	A logical value indicating whether seasonal effects are among the covariates defined for the variable that will be simulated
speriod	A vector that contains the lenght of seasonal cycles defined for the variable of interest
diurnal	A logical value indicating whether diurnal effects are among the covariates defined for the variable that will be simulated
dperiod	A vector that contains the lenght of diurnal cycles defined for the variable of interest
spatave	A logical value indicating whether spatial average effects are among the covariates defined for the variable that will be simulated
movave	A logical value indicating whether moving average effects are among the covariates defined for the variable that will be simulated
bw	A numeric vector that contains the bandwidth defined for the moving average

na.proc	A logical value indicating whether NA values should be stripped before simulation
fam.glm	family objects to specify probability distribution that will be used for the simulation mode ("gaussian", "gaussian-hetero", "binomial" or "Gamma")
occ.cond	Character that specify the name of the occurrence variable if that exists
init.buff	A buffer time is an extra time added before simulation to keep the simulation on track. the init.buffer is numeric vector, defined according to the number of climatic observations per the day and the number of few days that we choose as a buffer time before starting the simulation

Value

Returns a data frame that contains the covariates data frame with an additional column that contains the simulation of the variable of interest. If the covariates data frame is not specified among arguments, the function will create its own data frame of covariates that contains deterministic effects: geographical, diurnal and seasonal effects.

See Also

fit.glm

Examples

```
##Create a new data that contains climatic series and all effects that will be
##used as covariates for the variable to be projected or simulated
temp.effects <- seasonal.effect(myclimatic_data, period=c(365,183))
temp.effects <- diurnal.effect(temp.effects, period=24)
temp.effects <- spatave.effect(temp.effects, "temp", nstat = 3, na.proc = TRUE)
temp.effects <- lagged.effect(temp.effects, "temp",2, nstat=3)
temp.effects$t2m <- rnorm(length(myclimatic_data),mean=25,sd=1)

coord <- data.frame(x=c(9.92,9.93,10.04),y=c(35.55,35.62,35.57))
nstat=3
init.buff=48*7 ##48 time step per day and 7 days will be considered as buffer time
dstart=as.numeric(mycovariates[(init.buff*nstat)+1,1])
dend=as.numeric(mycovariates$dates[nrow(mycovariates)])

##fitted variable
temp.fitted <- fit.glm("temp", dep.var = NULL, geocov=TRUE, large.var="t2m",
seasonal = TRUE, speriod = c(365, 183), diurnal = TRUE, dperiod = 24,
spatave = FALSE, movave = FALSE, spatmovave= FALSE, lagvar=2, add.cov = FALSE,
others = NULL, fam.glm = "gaussian", data= temp.effects)

temp.projection <- projection.lagged(dstart, dend, temp.fitted, "temp", maxlag=2,
coord, cov=mycovariates, seasonal = TRUE, speriod = c(365,183),  diurnal = TRUE,
dperiod = 24, spatave = FALSE, movave=FALSE,bw = 0, fam.glm = "gaussian",
occ.cond = NULL)

##Remove buffer
temp.projection <- rm.buffer(temp.projection, nstat, bi.length=init.buff)
```

rm.buffer *Remove buffer*

Description

Remove buffer after using it for simulation

Usage

```
rm.buffer(simdata, nstat, bi.length = 480)
```

Arguments

`simdata`	Data frame obtained after simulation using the function `projection.lagged`
`nstat`	Numeric vector specifying the number of sites used
`bi.length`	Numeric vector that specifies the period of buffer (number of days* number of climatological observations per day)

Value

Data frame derived from `projection.lagged` that start from dstart and finish on dend

See Also

`projection.lagged`

sim.glm *Simulation of glm*

Description

function for predictions from the results of model fitting functions (`fit.glm`) or (`glm`)

Usage

```
sim.glm(fit, datapred, fam.glm = "gaussian", occ.cond = NULL)
```

Arguments

`fit`	climatic variable fitted returned from "glm" which inherits from the class "lm".
`datapred`	Data frame that contains dates, climatic variables and all covariates used to fit the variable to simulate
`fam.glm`	family objects to specify probability distribution used for the model ("gaussian", "gaussian-hetero", "binomial" or "Gamma")
`occ.cond`	character object that specifies the name of the occurrence variable if that exists

Value

Value returned belong to the same class of its first argument

See Also

glm, fit.glm

Examples

```
temp_fitted=glm(temp~Rh, family=gaussian, data=myclimatic_data)
temp.sim=sim.glm(temp_fitted, myclimatic_data, fam.glm = "gaussian", occ.cond = NULL)
```

---

spatave.effect *Spatial average*

Description

Performs average value of climatic variable over different sites

Usage

```
spatave.effect(var.mat, var.name, nstat = NULL, na.proc = FALSE)
```

Arguments

var.mat	A data frame containing a variable named "dates" in chron format and different climatic variables for each time step and for each site
var.name	Character object specifying the name of the climatic variable for which we will perfom spatial average
nstat	Numeric vector to define the number of sites used to obtain climatic series
na.proc	A logical value indicating whether NA values should be stripped before the computation proceeds

Value

An additional column will be added in the data frame introduced to contain the average of the variable of interest over the different sites for each time step

Examples

```
temp_spatave <- spatave.effect(myclimatic_data, "temp", nstat = 3, na.proc = TRUE)
```

Index