Package: musica (via r-universe)

November 5, 2024

Type Package

Title Multiscale Climate Model Assessment

Version 0.1.3

Description Provides functions allowing for (1) easy aggregation of multivariate time series into custom time scales, (2) comparison of statistical summaries between different data sets at multiple time scales (e.g. observed and bias-corrected data), (3) comparison of relations between variables and/or different data sets at multiple time scales (e.g. correlation of precipitation and temperature in control and scenario simulation) and (4) transformation of time series at custom time scales.

License GPL-2

LazyData TRUE

Depends R (>= 2.15.1), data.table

Imports magrittr, qmap, lubridate

RoxygenNote 5.0.1

Suggests knitr, rmarkdown, ggplot2

VignetteBuilder knitr

Repository https://hanel.r-universe.dev

RemoteUrl https://github.com/hanel/musica

RemoteRef HEAD

RemoteSha e394e552c05ab7ef1ca42e2bec3484ce4e28819a

Contents

`musica-package` *An R package for multiscale climate model assessment*

Description

Contains functions for flexible assessment of climate model bias and changes at multiple time scales. See documentation for `decomp`, `compare` and `vcompare`. In addition, musica provides functions for multiscale transformations of time series (see `msTrans_abs` and `msTrans_dif`)

Package options

Following option(s) are available:

additive_variables At several places the package compares values. The character vector `additive_values` specifies for which variables difference should be used for comparison instead of ratio. Defaults to `additive_values = "TAS"`. See `options` for setting or examining options.

Author(s)

Martin Hanel <hanel@fzp.czu.cz>

References

Hanel, M., Kozin, R. (2016) Bias correction for hydrological modelling, submitted.

basin_PT — *Basin average observed and simulated daily precipitation and temperature*

Description

A list of three data.tables with observed (`obs_ctrl`) and RCM simulated data for the control (`sim_ctrl`) and scenario (`sim_scen`) periods for Oslava basin (downto Cucice) in the Czech Republic. The basin average precipitation and temperature were obtained from griddedb observations and RCM simulation (EUR-11_CNRM-CERFACS-CNRM-CM5_rcp45_r1i1p1_CLMcom-CCLM4-8-17 simulation conducted within the CORDEX project).

Usage

```
basin_PT
```

Format

List of 3 data.tables:

obs_ctrl observed data for the basin for a period 1981-01-01 – 2005-21-31

sim_ctrl simulated data for the basin for a period 1981-01-01 – 2005-21-31

sim_scen simulated data for the basin for a period 2070-01-01 – 2099-21-31

Each data.table contains 3 variables:

DTM date

PR precipitation, mm

TAS temperature, degrees C

codes — *Conversion between period specification and codes*

Description

Conversion between period specification and codes

Usage

```
period2code(periods)

code2period(code)
```

Arguments

periods	period specification
code	period code

Details

Periods are specified using keywords "day", "month", "year" preceded by an integer and a space and optionally followed by "s" (the specification is further passed to `cut.Date`, see `cut.Date` for details). To fit in figures and for simplicity, periods can be also specified by codes, i.e. by `D`, `M`, `Y` (for "day", "month" and "year", respectively) and folowed by integer specifying the number of intervals. The functions `period2code` and `code2period` provide conversion between the two alternatives.

Examples

```
period2code(c('1 day', '23 days', '3 month', '2 years'))

code2period(c('D1', 'D23', 'M3', 'Y2'))
```

---

compare *Compare decomposed variables*

Description

The function evaluates distance between statistical characteristics of specified data sets. Distance is measured as difference for variables included in `getOption('additive_variables')`, i.e. temperature (`TAS`) by default, and as a ratio for other variables.

Usage

```
compare(x, compare_to, fun = mean, wet_int_only = TRUE, wet_int_thr = 0.1,
  exclude_below = 0.9)
```

Arguments

x	List of decomposed variables to be compared
compare_to	Decomposed variable used as a reference
fun	Function used for comparison
wet_int_only	(logical) Should only the wet intervals be considered?
wet_int_thr	Numeric value specifying the minimum depth to be considered wet
exclude_below	Some of the intervals might not be of required length, e.g. D10 interval may have less than 10 days available. The `exclude_below` argument controls the minimum fraction of the interval that has to be available in order to be considered in the summary statistics. Set to 0 for monthly data.

Value

data.table summarizing the differences with columns:

variable factor indicating the variable

period specification of the averaging length with 'D' - day(s), 'M' - month(s), 'Y' - year(s) and 'G1' - the overall mean

TS averaging length in hours

sub_period indication of the aggregating scale specified by agg_by argument

comp factor indicating the data sets from x with labels given by `names(x)`

DIF distance between data sets from x and `compare_to`. Distance is measured as difference for variables included in `getOption('additive_variables')`, i.e. temperature (TAS) by default, and as a ratio for other variables, see `dif`

Examples

```
library(ggplot2)
data(basin_PT)
## Not run:
dobs = decomp(basin_PT[['obs_ctrl']])
dctrl = decomp(basin_PT[['sim_ctrl']])
dscen = decomp(basin_PT[['sim_scen']])
d = compare(x = list(CTRL = dctrl, SCEN = dscen), compare_to = dobs, fun = max)
ggplot(d) +
 geom_line(aes(x = TS, y = DIF, col = factor(sub_period))) +
 facet_grid(variable ~ comp, scale = 'free') +
 scale_x_log10()

## End(Not run)
```

`dcompare` *Compare distribution function of variables*

Description

Compare distribution function of variables

Usage

```
dcompare(x, compare_to, p = seq(0, 1, 0.01), wet_int_only = TRUE,
  wet_int_thr = 0.1, exclude_below = 0.9)
```

Arguments

x	List of decomposed objects
wet_int_only	(logical) Should only the wet intervals be considered?
wet_int_thr	Numeric value specifying the minimum depth to be consider wet
exclude_below	Some of the intervals might not be of required length, e.g. D10 interval may have less than 10 days available. The exclude_below argument controls the minimum fraction of the interval that has to be available in order to be considered in the summary statistics.

Value

data.table summarizing the relation with columns:

variable factor indicating the variable

period specification of the averaging length with 'D' - day(s), 'M' - month(s), 'Y' - year(s) and 'G1' - the overall mean

TS averaging length in hours

sub_period indication of the aggregating scale specified by agg_by argument

comp factor indicating the data sets from x with labels given by names(x)

DIF distance between quantiles of x and compare_to. Distance is measured as difference for variables included in getOption('additive_variables'), i.e. temperature (TAS) by default, and as a ratio for other variables, see dif

Examples

```
library(ggplot2)
data(basin_PT)
## Not run:
dobs = decomp(basin_PT[['obs_ctrl']])
dctrl = decomp(basin_PT[['sim_ctrl']])
d = dcompare(x = list(CTRL = dctrl), compare_to = dobs)
ggplot(d[variable=='TAS' & period!='G1']) +
  geom_line(aes(x = p, y = DIF, col = comp)) +
  facet_grid(sub_period~period, scale = 'free') +
  theme(legend.position = 'top', axis.text.x = element_text(angle = 90, vjust = .5))

## End(Not run)
```

---

decomp *Decomposition of time-series*

Description

Calculate series of averages over the periods specified in the period argument into the inpur data.table.

Usage

```
decomp(x, period = c("Y1", "M6", "M3", "M1", "D15", "D1"), agg_by = quarter,
  full_return = FALSE, remove_incomplete = TRUE, year_starts = months(2))
```

Arguments

x	data.table with columns DTM (date), variable and value. Any number of variables are in principle allowed.
period	The periods over which the averages will be calculated, see Details
agg_by	Function for specification of the period (season, month) to be additionaly included in output, see Details
full_return	(logical) Should the average be repeated for each scale along with original time series? Default is FALSE (e.g. for M1 only monthly and not daily time series is returned)
remove_incomplete	Should the incomplete years be removed from results? Default is TRUE. For use with monthly data always set to FALSE.
year_starts	Indication of the start of the year - determines how the months will be grouped into the seasons. Note that the sub_period output variable is with respect to the year_starts

Details

The original time series in daily time step is decomposed into series of averages ove periods specified in periods argument using letter codes 'D' - day(s), 'M' - month(s), 'Y' - year(s) followed by number corresponding to number of periods and 'G1' the overall mean. The periods must be given in order from longest to shortest, the overall mean is always included (and needs not to be specified in period). Shorter periods are always identified within the closest longer periods, i.e. each shorter period is included in exactly one longer period. As a result, the averages may be calculated over shorter periods than specified. This is due to varying length of "month" and "year" periods. The actual length used for averaging is included in the output. To make further assessment of the decomposed objects easier, indicator of period within the year (e.g. quarter or month) as specified by agg_by argument is included in the output.

Value

data.table with variables:

variable factor indicating the variable

DTM date

period specification of the averaging length with 'D' - day(s), 'M' - month(s), 'Y' - year(s) and 'G1' - the overall mean

value value of the variable for given averaging length

sub_period indication of the aggregating scale specified by agg_by argument

period_pos average date of the interval

N real length of the vectors used for calculating averages

TS averaging length in hours

Examples

```
data(basin_PT)
str(basin_PT)
basin_PT[['obs_ctrl']]
dobs = decomp(basin_PT[['obs_ctrl']], period = c('1 year', '1 month', '1 day'))
```

`difs` *Functions for evaluating distance between variables*

Description

Functions for evaluating distance between variables

Usage

```
dif(x, y, var)

rev_dif(x, y, var)

rev_difv(x, var)
```

Arguments

`x, y`	variables to be compared
`var`	variable code

Value

Difference or ratio of `x` and `y` (for `dif`) and sum or product (for `rev_dif` and `rev_difv`). Distance is measured as difference for variables included in `getOption('additive_variables')`, i.e. temperature (`TAS`) by default, and as a ratio for other variables.

While `rev_dif` returns `sum(x, y)` or `prod(x, y)`, `rev_difv` takes single vector `x` and returns `sum(x)` or `prod(x)`.

Used mainly in other functions of the package.

Examples

```
getOption('additive_variables')

# calculate distance of 2 vectors
dif(c(10, 20, 30), c(11, 18, 3), 'TAS')
dif(c(10, 20, 30), c(11, 18, 3), 'PR')

# inverse for 2 vectors
rev_dif(c(10, 20, 30), c(11, 18, 3), 'TAS')

# inverse for 1 vector
rev_difv(c(10, 1.1, .9), 'TAS')
```

m2s *Indication of a season*

Description

Indication of a season

Usage

```
month2sea(dtm, year_starts = months(0))

sscale2sea(sub_scale, year_starts = months(-1))
```

Arguments

dtm	a Date object
year_starts	Month object indicating the start of the year
sub_scale	integer indicating the season

Value

3 letter code (as DJF, JJA etc.) specifying the season

Examples

```
month2sea(as.Date('2000-01-01') + months(1:10) )

sscale2sea(c(1, 1, 2, 2, 2, 3, 3), year_starts = months(-1))
```

msTrans_abs *Multiscale quantile mapping bias correction*

Description

Applies standard quantile mapping at custom time scales.

Usage

```
msTrans_abs(dta, agg_by = month, wet_int_thr = 0.1, maxiter = 10,
  tol = 1e-04, qstep = 0.001, period = c("G1", "Y1", "M3", "M1", "D1"))
```

Arguments

dta	List with components FROM (simulated data for the control period), TO (observed data) and NEWDATA (data to be corrected). Each component is a data.table with columns DTM (date) and the climate variables (typically PR - precipitation and TAS - temperature)
agg_by	Function for specification of the period (season, month) to be additionaly included in output, see Details
wet_int_thr	Numeric value specifying the minimum depth to be considered wet
maxiter	Maximum number of iterations, see Details
tol	Stoping criterion of the iteration cycle, see Details
qstep	A numeric value between 0 and 1. The quantile mapping is fitted only for the quantiles defined by quantile(0,1,probs=seq(0,1,by=qstep). Passed to doQmapQUANT.
period	Specification of the aggregation lengths the correction is applied at with 'D' - day(s), 'M' - month(s), 'Y' - year(s) and 'G1' - the overall mean

Details

The procedure utilizes standard quantile mapping from the qmap-package, but at multiple time scales. Since correction at particular temporal scale influences values at other aggregations, the procedure is applied iterativelly until the maximum number of iterations (maxiter) is reached or the difference between succesive iteration step is smaller than tol. Differences between corrected and uncorrected variable at longer time scales are used to modify daily values after each iteration step (see e.g. Mehrorta and Sharma, 2016; Pegram et al. 2009). To make further assessment of the decomposed objects easier, indicator of period within the year (e.g. quarter or month) as specified by agg_by argument is included in the output.

Value

data.table with corrected data

References

Hanel, M., Kozin, R., 2016. Bias and projected changes in climate model simulations at multiple time scales: consequences for hydrological impact assessment. Environmental Modelling and Software, submitted.

Mehrotra, R., Sharma, A., 2016. A multivariate quantile-matching bias correction approach with auto-and cross-dependence across multiple time scales: Implications for downscaling. Journal of Climate 29, 3519-3539.

Pegram, G.G., et al., 2009. A nested multisite daily rainfall stochastic generation model. Journal of Hydrology 371, 142-153.

Examples

```
data("basin_PT")
scen = basin_PT$sim_scen
ctrl = basin_PT$sim_ctrl
obs = basin_PT$obs_ctrl
```

```
dta = list(TO = obs, FROM = ctrl, NEWDATA = scen)
## Not run:
msTrans_abs(dta,  maxiter = 10, period = 'D1')

## End(Not run)
```

`msTrans_dif` *Multiscale delta method*

Description

Transforms observed data such that the changes in summary statistics of variables at custom time scales are similar to those obtained from climate model simulation. Number of functions can be used to summarize the variables.

Usage

```
msTrans_dif(dta, model = "const", model_par = list(NULL), agg_by = month,
  wet_int_thr = 0.1, maxiter = 10, tol = 1e-04, period = c("G1", "Y1",
  "M1", "D1"), qstep = 0.001)
```

Arguments

`dta`	List with components `FROM` (simulated data for the control period), `TO` (simulated data for the scenario period) and `NEWDATA` (observed data to be transformed). Each component is a `data.table` with columns `DTM` (date) and the climate variables (typically `PR` - precipitation and `TAS` - temperature)
`model`	One of `loess`, `const`, `identity`, `lm`, `smooth`, `runmed`, `smooth.spline`. The model is used to provide statistical summary of the empirical cumulative distribution function.
`model_par`	optional parameters of the `model`
`agg_by`	Function for specification of the period (season, month) to be additionaly included in output, see Details
`wet_int_thr`	Numeric value specifying the minimum depth to be considered wet
`maxiter`	Maximum number of iterations, see Details
`tol`	Stoping criterion of the iteration cycle, see Details
`period`	Specification of the aggregation lengths the correction is applied at with 'D' - day(s), 'M' - month(s), 'Y' - year(s) and 'G1' - the overall mean
`qstep`	A numeric value between 0 and 1. The ecdf is calculated only for the quantiles defined by quantile(0, 1, probs = seq(0, 1, by = qstep).

Value

transformed data.table

References

Hanel, M., Kozin, R., 2016. Bias and projected changes in climate model simulations at multiple time scales: consequences for hydrological impact assessment. Environmental Modelling and Software, submitted.

Examples

```
data("basin_PT")
scen = basin_PT$sim_scen
ctrl = basin_PT$sim_ctrl
obs = basin_PT$obs_ctrl
dta = list(TO = scen, FROM = ctrl, NEWDATA = obs)
## Not run:
msTrans_dif(dta,  maxiter = 10, period = 'D1')

## End(Not run)
```

---

prob *Evaluation of empirical cumulative distribution function*

Description

Evaluation of empirical cumulative distribution function

Usage

```
prob(x)
```

Arguments

x	vector of values

Value

value of the empirical distribution function evaluated at x

Examples

```
prob(rnorm(10))
```

Q *Convenience function for calculation of quantiles*

Description

The typical use is in compare to avoid anonymous functions in specification of its fun argument.

Usage

```
Q(p, ...)
```

Arguments

p	Specification of the quantile
...	other arguments passed to quantile

Value

function calculating the p-th quantile

Examples

```
q90 = Q(.9)
class(q90)
q90(rnorm(10))
```

sscale2sea *Provides the letter code for months*

Description

Provides the letter code for months

Usage

```
sscale2sea(sub_scale, year_starts = months(-1))
```

Arguments

sub_scale	Typically the sub_period variable from decomposed object
year_starts	The start of the year

Details

Typical workflow is to set `year_starts` in the decomp function e.g. to `months(-1)` or `months(2)`. These both result in climatological seasons (December-January-February - DJF, etc.). The latter in addition results in grouping of warm and cold seasons together at M6 scale. The sub_period field of the decomposed object is with respect to `year_starts`, i.e. when `year_starts` = `months(-1)` then `sub_period` = `1` corresponds to December. To obtain the three-letter codes back, `sscale2sea` is used. The function is typically used for plotting.

Value

Vector of three-letter codes for seasons

Examples

```
sscale2sea(1:12, year_starts = months(-1))
```

---

tscale *Convert averaging length code to hours*

Description

Period durations are calculated by the `lubridate` package.

Usage

```
tscale(x, nyears = 30)
```

Arguments

`x`	Vector of the averaging period codes
`nyears`	Overall number of years - used for conversion of the overall mean

Value

numerical vector of durations in hours

Examples

```
tscale('M1')
tscale('G1', nyears = 25)
```

vcompare *Assess the relations between two decomposed variables*

Description

Assess the relations between two decomposed variables

Usage

```
vcompare(x, fun = cor, wet_int_only = TRUE, wet_int_thr = 0.1,
  exclude_below = 0.9)
```

Arguments

x	List of decomposed objects
fun	Function to sumarize dependence (like cor, cov)
wet_int_only	(logical) Should only the wet intervals be considered?
wet_int_thr	Numeric value specifying the minimum depth to be consider wet
exclude_below	Some of the intervals might not be of required length, e.g. D10 interval may have less than 10 days available. The exclude_below argument controls the minimum fraction of the interval that has to be available in order to be considered in the summary statistics.

Details

vcompare compares the relation between all pairs of variables included in x, typically precipitation and temperature, but other variables may be included also (e.g. runoff).

Value

data.table summarizing the relation with columns:

variable factor indicating the variable

period specification of the averaging length with 'D' - day(s), 'M' - month(s), 'Y' - year(s) and 'G1' - the overall mean

TS averaging length in hours

sub_period indication of the aggregating scale specified by agg_by argument

comp factor indicating the data sets from x with labels given by names(x)

value value of fun

Examples

```
library(ggplot2)
data(basin_PT)
## Not run:
dobs = decomp(basin_PT[['obs_ctrl']])
dctrl = decomp(basin_PT[['sim_ctrl']])
d = vcompare(x = list(OBS = dobs, CTRL = dctrl), fun = cov)
ggplot(d[period!='G1']) +
 geom_line(aes(x = TS, y = value, col = factor(sub_period))) +
 facet_grid(VARS~ID) +
 scale_x_log10()

## End(Not run)
```

Index