Package 's2dverification'

December 2, 2013

Type Package

Title Set of common tools for model diagnostics.

Version 2.0

Date 2013-11-15

Author Virginie Guemas <virginie.guemas@ic3.cat>, Isabel Andreu-Burillo <isabel.andreu-burillo@ic3.cat>, Fabian Lienert <fabian.lienert@ic3.cat>, Javier Garcia-Serrano <jgarcia@ic3.cat>, Luis Rodrigues <lrodrigues@ic3.cat>, Caio A. S. Coelho <caio.coelho@cptec.inpe.br>, Chris Ferro <c.a.t.ferro@readir
han Steinskog <dag.johan.steinskog@nersc.no>, Christoph Frei <Christoph.Frei@meteoswiss.ch>, Ludovic Auger <ludovic.auger@meteo.fr>, Nicolau Manubens Gil <nicolau.manubens@ic3.cat>

Maintainer Virginie Guemas <virginie.guemas@ic3.cat>

Description Set of tools to score a model basing on one or more observational datasets and the experimental dataset obtained from the model simulation.

License Unlimited

Depends R (>= 2.14.1), ncdf, GEOmap, geomapdata, maps, mapproj

Encoding UTF-8

R topics documented:

`s2dverification-package`

Prediction Model Score

Description

This package contains a set of tools to score prediction models by comparing experimental data with observational data.

Details

Package:	s2dverification
Type:	Package
Version:	2.0
Date:	2013-08-21
License:	Unlimited

First, data has to be loaded from the repository with the function Load().
It is needed to specify a variable to load, names for the experimental and observational datasets to load the data from and the starting dates, among other arguments.
This will automatically provide two matrices with the observational and experimental data.
From then on, you can compute the anomalies, climatologies, etc.

Author(s)

Virginie Guemas <virginie.guemas@ic3.cat>, Isabel Andreu-Burillo <isabel.andreu-burillo@ic3.cat>, Fabian Lienert <fabian.lienert@ic3.cat>, Javier Garcia-Serrano <jgarcia@ic3.cat>, Luis Rodrigues <lrodrigues@ic3.cat>, Caio A. S. Coelho <caio.coelho@cptec.inpe.br>, Chris Ferro <c.a.t.ferro@reading.ac.uk>, Dag Johan Steinskog <dag.johan.steinskog@nersc.no>, Christoph Frei <Christoph.Frei@meteoswiss.ch>, Ludovic Auger <ludovic.auger@meteo.fr>, Nicolau Manubens Gil <nicolau.manubens@ic3.cat>

References

Please, for more information load the package and check the help for each function, or check the wiki page on common diagnostics.

ACC *Compute Anomaly Correlation Coefficient*

Description

Matrix var_exp & var_obs should have dimensions (nexp/nobs, nsdates, nltimes, nlat, nlon). ACC computes the Anomaly Correlation Coefficient for each jexp in 1:nexp and each jobs in 1:nobs which gives nexp x nobs ACC for each startdate and each leadtime. The confidence interval is computed by a Fisher transformation. The significance level relies on a one-sided student-T distribution. A domain can be selected by providing the list of longitudes/latitudes (lon/lat) of the grid together with the corner of the domain: lonlatbox = c(lonmin, lonmax, latmin, latmax)

Usage

```
ACC(var_exp, var_obs, lon = NULL, lat = NULL, lonlatbox = NULL)
```

Arguments

var_exp	Matrix of experimental data.
var_obs	Matrix of observational data, same dimensions as var_exp except along the first dimension.
lon	Array of longitudes of the var_exp/var_obs grids, optional.
lat	Array of latitudes of the var_exp/var_obs grids, optional.
lonlatbox	Domain to select c(lonmin, lonmax, latmin, latmax), optional.

Value

ACC	Matrix with c(nexp, nobs, nsdates, nleadtimes, 4) dimensions. The fourth dimension of length 4 corresponds to the lower limit of the 95 confidence interval, the computed ACC, the upper limit of the 95 confidence interval and the 95 T-test.
MACC	Mean Anomaly Correlation Coefficient with c(nexp, nobs, nleadtimes) dimensions.

Author(s)

History: 1.0 - 2013-08 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

Alpha *Estimate Correlation At Lag 1*

Description

This function, based on the fit_acfcoef function, estimates the autocorrelation at lag 1 in the xdata array following the method described in Guemas V., Auger L., Doblas-Reyes F., JAMC, 2013.

Usage

```
Alpha(xdata, detrend = F, filter = F)
```

Arguments

xdata	Array of data to estimate the autocorrelation from.
detrend	T applies a linear detrending to xdata prior to the estimation of the autocorrelation at lag 1.
filter	T applies a filtering of any cycle prior to the estimation of the autocorrelation at lag 1.

Value

Correlation at lag 1.

Author(s)

History:
1.0 - 2012-06 (V. Guemas, <vguemas@ic3.cat>) - Original code

`AnimVsLTime` *Animate Variable Values Over Forecast Time*

Description

Create animations showing the anomalies, the climatologies, the mean InterQuartile Range, Maximum-Mininum, Standard Deviation, Median Absolute Deviation, the trends, the RMSE or correlation between modelled and observed data along the forecast time (versus the lead-times) for all input experiments and input observational datasets.

Usage

```
AnimVsLTime(var, lat, lon, toptitle = c("", "", "", "", "", "", "", "", "",
"", ""), units = "", monini = 1, freq = 12, msk95lev = FALSE, brk = NULL,
col = NULL, lonmin = 0, lonmax = 360, latmin = -90, latmax = 90,
center = NULL, equi = TRUE, fileout = c("output1_animvsltime",
"output2_animvsltime", "output3_animvsltime"))
```

Arguments

`var`	Matrix of dimensions (nltime, nlat, nlon) or (nexp/nmod, nltime, nlat, nlon) or (nexp/nmod, 3, nltime, nlat, nlon) or (nexp/nmod, nobs, 3, nltime, nlat, nlon)
`lat`	Vector containing latitudes (°)
`lon`	Vector containing longtitudes (°)
`toptitle`	c("," , ...) array of main title for each animation, optional. If RMS or correlations: first exp with successive obs, then second exp with successive obs, etc ...
`units`	Units, optional
`monini`	Starting month between 1 and 12. Default = 1
`freq`	1 = yearly, 12 = monthly, 4 = seasonal ...
`msk95lev`	T/F grid points masked if 95 Default = F
`brk`	Limits of colour levels, optional. For example: seq(min(var), max(var), (max(var) - min(var)) / 10)
`col`	Vector of colours of length(brk) - 1, optional.
`lonmin`	Westward limit of the domain to plot (> 0 or < 0). Default : 0°
`lonmax`	Eastward limit of the domain to plot (> 0 or < 0). lonmax > lonmin. Default : 360°
`latmin`	Southward limit of the domain to plot. Default : -90°
`latmax`	Northward limit of the domain to plot. Default : 90°

center	(latitude, longitude, angle) for stereographic projection
equi	TRUE/FALSE == cylindrical equidistant/stereographic projection. Default: TRUE
fileout	c(", ", ...) array of output file name for each animation without extension. If RMS or correlations : first exp with successive obs, then second exp with successive obs, etc ...

Details

Examples of input: ——————

1- Outputs from clim (exp, obs, memb=F): (nmod, nltime, nlat, nlon) or (nobs, nltime, nlat, nlon)

2- Model output from load/ano/smoothing: (nmod, nmemb, sdate, nltime, nlat, nlon) then passed through spread(var, posdim = 2, narm = T) & mean1dim(var, posdim = 3, narm = T) or through trend(mean1dim(var, 2), posTR = 2): (nmod, 3, nltime, nlat, nlon) animates average along start dates of IQR/MaxMin/SD/MAD across members or trends of the ensemble-mean computed accross the start dates.

3- model and observed output from load/ano/smoothing: (nmod, nmemb, sdate, nltime, nlat, nlon) & (nobs, nmemb, sdate, nltime, nlat, nlon) then averaged along members mean1dim(var_exp/var_obs, posdim = 2): (nmod, sdate, nltime, nlat, nlon) (nobs, sdate, nltime, nlat, nlon) then passed through corr(exp, obs, posloop = 1, poscor = 2) or RMS(exp, obs, posloop = 1, posRMS = 2): (nmod, nobs, 3, nltime, nlat, nlon) animates correlations or RMS between each exp & each obs against leadtime.

Author(s)

History: 1.0 - 2012-04 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

Ano *Compute Raw Anomalies*

Description

This function computes raw anomalies from experimental or observational matrix output from `load()` and their climatologies output from clim().

Usage

```
Ano(var, clim)
```

Arguments

var	Model or observational data: c(nmod/nexp/nobs, nmemb/nparam, nsdates, nltime) up to c(nmod/nexp/nobs, nmemb/nparam, nsdates, nltime, nlevel, nlat, nlon)
clim	Climatologies from clim: c(nmod/nexp/nobs, nmemb/nparam, nltime) up to c(nmod/nexp/nobs, nmemb/nparam, nltime, nlevel, nlat, nlon) or c(nmod/nexp/nobs, nltime) up to c(nmod/nexp/nobs, nltime, nlevel, nlat, nlon)

Value

Matrix with same dimensions as var

Author(s)

History: 1.0 - 2012-03 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

`Ano_CrossValid` *Compute Raw Anomalies From Experimental And Observational Data*

Description

This function computes raw anomalies from experimental and observational matrix output from `load()` by subtracting the climatologies computed in a cross-validation mode and with a per-pair method.

Usage

```
Ano_CrossValid(var_exp, var_obs, memb = T)
```

Arguments

`var_exp`	Model data: c(nmod/nexp, nmemb/nparam, nsdates, nltime) up to c(nmod/nexp, nmemb/nparam, nsdates, nltime, nlevel, nlat, nlon)
`var_obs`	Observational data: c(nobs, nmemb, nsdates, nltime) up to c(nobs, nmemb, nsdates, nltime, nlevel, nlat, nlon)
`memb`	memb: T/F (1 climatology for each member). Default = T.

Value

`$ano_exp`	Matrix with same dimensions as var_exp
`$ano_obs`	Matrix with same dimensions as var_obs

Author(s)

History: 1.0 - 2011-12 (V. Guemas, <vguemas@ic3.cat>) - Original code

BlueRed *Generate Colour Scale For Plotting*

Description

Returns a colour scale corresponding to specified data values or a specified number of colours.

Usage

```
BlueRed(data, scale = c("equal", "linear", "split"), white = median(data),
yellow = 0, cyan = 0, invert = FALSE, format = c("hex", "rgb"))
```

Arguments

data	A vector of distinct data values corresponding to the colours or the number of colors (>= 1) to be created.
scale	Controls the colour gradients (see Details). Must be either '"equal"' (the default), '"linear"' or '"split"' (or any unique partial match).
white	The data value corresponding to the colour white.
yellow	A number between 0 and 1 indicating the amount of yellow to insert between white and red. If 0 (the default) then no yellow is used. Larger values indicate more yellow.
cyan	A number between 0 and 1 indicating the amount of cyan to insert between white and blue. If 0 (the default) then no cyan is used. Larger values indicate more cyan. Not implemented.
invert	Logical: if 'TRUE', negative data values are red and positive values are blue; if 'FALSE' (the default) then vice-versa.
format	The format in which the colours are returned (see Value). Must be either '"hex"' (the default) or '"rgb"' (or any unique partial match).

Details

One colour is created for each value in 'data'. Colours range from blue for data values less than 'white' to red for data values greater than 'white'. If 'scale' equals '"linear"' then colour intensity scales linearly with absolute data value: highest intensity corresponding to the maximum absolute value of 'data'. If 'scale' equals '"split"' then linear intensity scales are constructed separately for blue and red colours so that the maximum intensity is attained by both blue and red. If 'scale' equals '"equal"' then intensities are equally spaced regardless of the data values.

Value

If 'format' is '"hex"' then returns a character vector of hexadecimal RGB numbers. If 'format' is '"rgb"' then returns a matrix with three columns containing RGB intensities.

Author(s)

History: 1.0 - 2005-05 (C. Ferro, - Original code - <c.a.t.ferro@reading.ac.uk>) -

Clim *Comput Per-pair Climatologies*

Description

This function computes per-pair climatologies from experimental and observational matrix output from load.

Usage

```
Clim(var_exp, var_obs, memb = T, kharin = F, NDV = F)
```

Arguments

var_exp	Model data: c(nmod/nexp, nmemb/nparam, nsdates, nltime) up to c(nmod/nexp, nmemb/nparam, nsdates, nltime, nlevel, nlat, nlon)
var_obs	Observational data: c(nobs, nmemb, nsdates, nltime) up to c(nobs, nmemb, nsdates, nltime, nlevel, nlat, nlon)
memb	memb: T/F (1 climatology for each member). Default = T.
kharin	T/F (if Kharin method is applied or not). Default = F.
NDV	

Value

clim_exp	Matrix with same dimensions as var_exp
clim_obs	Matrix with same dimensions as var_obs

Author(s)

History: 1.0 - 2011-03 (V. Guemas, <vguemas@ic3.cat>) - Original code

ColorBar *Draw Color Bar*

Description

Creates a horizontal or vertical colorbar to introduce in multipanels.

Usage

```
ColorBar(brks, cols = NULL, vert = TRUE, subsampleg = 1)
```

Arguments

brks	Levels.
cols	List of colours, optional.
vert	TRUE/FALSE for vertical/horizontal colorbar.
subsampleg	Supsampling factor of the interval between ticks on colorbar. Default: 1 = every level

Author(s)

History: 1.0 - 2012-04 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

Consist_Trend *Compute Trends*

Description

Compute trends by least square fitting and the associated error interval for both the observational and model data. Provide also the detrended observational and modelled data. The trend is computed along the start date dimension.

Usage

```
Consist_Trend(var_exp, var_obs, interval = 1)
```

Arguments

var_exp	Ensemble mean of model hindcasts with dimensions: c(nmod/nexp, nsdates, nltime) up to c(nmod/nexp, nsdates, nltime, nlevel, nlat, nlon)
var_obs	Ensemble mean of observational data with dimensions: c(nobs, nsdates, nltime) up to c(nobs, nsdates, nltime, nlevel, nlat, nlon)
interval	Number of months between 2 start dates. Default = 1.

Value

$trend	Trends of model and observational data same dimensions with dimensions: c(nmod/nexp + nobs, 3, nltime) up to c(nmod/nexp + nobs, 3, nltime, nlevel, nlat, nlon) The length 3 dimension corresponds to the lower limit of the 95 interval, the computed trends and the upper limit of the 95 interval.
$detrendedmod	Same dimensions as var_exp with linearly detrended var along the start date dimension.
$detrendedobs	Same dimensions as var_exp with linearly detrended var along the start date dimension.

Author(s)

History: 1.0 - 2011-11 (V. Guemas, <vguemas@ic3.cat>) - Original code

Corr *Compute Correlations*

Description

Matrix var_exp & var_obs should have the same dimensions except along posloop where the length can be different (nexp & nobs). Corr computes correlation for each jexp in 1:nexp and each jobs in 1:nobs which gives nexp x nobs correlation for each other grid point of the matrix. The correlations are computed along the poscor dimension. If compROW is given, the correlations are computed only if rows along the (compROW)th dimension are complete between limits[1] and limits[2], that mean with no NA between limits[1] and limits[2]. Default: limits[1] = 1 and limits[2] = length(compROW dimension). The confidence interval is computed by a Fisher transformation. The significance level relies on a one-sided student-T distribution.

Usage

```
Corr(var_exp, var_obs, posloop = 1, poscor = 2, compROW = NULL, limits = NULL)
```

Arguments

var_exp	Matrix of experimental data.
var_obs	Matrix of observational data, same dimensions as var_exp except along posloop.
posloop	Dimension nobs and nexp.
poscor	Dimension along which correlation are to be computed.
compROW	Data taken into account only if (compROW)th row is complete. Default = NULL.
limits	Complete between limits[1] & limits[2]. Default = NULL.

Value

Matrix with c(length(posloop) in var_exp, length(posloop) in var_obs, 4, all other dimensions of var_exp & var_obs except poscor). The third dimension of length 4 corresponds to the lower limit of the 95 confidence interval, the computed correlation, the upper limit of the 95 confidence interval and the 95 T-test.

Author(s)

History: 1.0 - 2011-04 (V. Guemas, <vguemas@ic3.cat>) - Original code

Enlarge *Extend Number Of Dimensions*

Description

Extends the number of dimensions of var to numdims (the added dimensions have length 1).

Usage

```
Enlarge(var, numdims)
```

Arguments

var	Array to extend.
numdims	Desired number of dimensions.

Value

Extended array.

Author(s)

History: 1.0 - 2011-03 (V. Guemas, <vguemas@ic3.cat>) - Original code

Eno *Compute Effective Number Of Independant Observations*

Description

Compute the effective number of independant observations along the (posdim)th dimension of an observational matrix. This effective number of independant observations may be required to perform statistical tests. Based on eno function from Caio Coelho.

Usage

```
Eno(obs, posdim)
```

Arguments

obs	Observational matrix of any number of dimensions up to 10.
posdim	Position along which to compute the effective number of observations.

Value

Same dimensions as var except without posdim dimension.

Author(s)

History: 1.0 - 2011-05 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

`EnoNew` *Compute Equivalent Number Of Independant Data*

Description

This function computes the equivalent number of independant data in the xdata array following the method described in Guemas V., Auger L., Doblas-Reyes F., JAMC, 2013.

Usage

```
EnoNew(xdata, detrend = F, filter = F)
```

Arguments

`xdata`	Array of data.
`detrend`	T applies a linear detrending to xdata prior to the estimation of the number of independant data.
`filter`	T applies a filtering of any cycle prior to the estimation of the number of independant data.

Author(s)

History:
1.0 - 2012-06 (V. Guemas, <vguemas@ic3.cat>) - Original code

`Filter` *Filter Signal From Data*

Description

This function filters from the xdata array, the signal of frequency freq.
The filtering is performed by dichotomal seek for the frequency around freq and the phase that maximizes the signal to subtract to xdata.
The maximization of the signal to subtract relies on a minimization of the mean square differences between xdata and a cosine of given frequency and phase.

Usage

```
Filter(xdata, freq)
```

Arguments

xdata	Array of data to be filtered.
freq	Frequency to filter.

Value

Array of filtered data.

Author(s)

History:
1.0 - 2012-02 (V. Guemas, <vguemas@ic3.cat>) - Original code

FitAcfCoef *Fit Acf Coefficients*

Description

This function finds the minimum point of the fourth order polynom (a - x)2 + 0.25(b - x2)2 written to fit the two autoregression coefficients a and b.
Thanks to the Cardano formula provided a and b in [0 1] the problem is well posed delta > 0 and there is only one solution to the minimum.

This function is called in Alpha() to minimize the mean square differences between the theoretical autocorrelation function of an AR1 and the first guess of estimated autocorrelation function estacf, using only the first two lags.

Usage

```
FitAcfCoef(a, b)
```

Arguments

a	Coefficient a.
b	Coefficient b.

Value

Minimum value of the polynom.

Author(s)

History:
1.0 - 2012-06 (L. Auger, <ludovic.auger@meteo.fr>) - Original code

FitAutocor *Fit Theoretical Autocorrelation Of An AR1*

Description

This function fits the theoretical autocorrelation function of an AR1 to the first guess of estimated autocorrelation function estacf containing any number of lags. The fitting relies on a dichotomial minimisation of the mean square differences between both autocorrelation functions. It returns the autocorrelation at lag 1 of the fitted AR1 process.

Usage

```
FitAutocor(estacf, window = c(-1, 1), prec = 0.01)
```

Arguments

estacf	Estimated autocorrelation function
window	Epecifies the interval in which lies the autocorrelation at lag 1 which is sought for.
prec	Determines the precision to which the autocorrelation function at lag 1 is to be estimated.

Value

Autocorrelation at lag 1 of the fitted AR1 process

Author(s)

History:
1.0 - 2012-02 (V. Guemas, <vguemas@ic3.cat>) - Original code

GenSeries *Generate AR1 Process*

Description

This functions generates AR1 processes containing n data, with alpha as autocorrelation at lag 1, and mean and standard deviation provided by the mean and std arguments.

Usage

```
GenSeries(n, alpha, mean, std)
```

Arguments

n	Length of data to be generated.
alpha	Autocorrelation at lag 1.
mean	Mean of the data.
std	Standard deviation of the data.

Value

AR1 process.

Author(s)

History:
1.0 - 2012-04 (L. Auger, <ludovic.auger@meteo.fr>) - Original code

---

Histo2Hindcast *Chunk Long Simulations*

Description

This function reorganizes a long run (historical typically) with only one start date into chunks corresponding to a set of start dates. The expected input structure is the one output from load with 4 to 7 dimensions.

Usage

```
Histo2Hindcast(varin, sdatesin, sdatesout, nleadtimesout)
```

Arguments

varin	Input model or observational data: c(nmod/nexp/nobs, nmemb/nparam, nsdates, nltime) up to c(nmod/nexp/nobs, nmemb/nparam, nsdates, nltime, nlevel, nlat, nlon)
sdatesin	Start date of the input matrix 'YYYYMMDD'.
sdatesout	List of start dates of the output matrix c('YYYYMMDD', 'YYYYMMDD', ...).
nleadtimesout	Number of leadtimes in the output matrix.

Value

A matrix with the same number of dimensions as the input one, the same dimensions 1 and 2 and potentially the same dimensions 5 to 7. Dimensions 3 and 4 are set by the arguments sdatesout and nleadtimesout.

Author(s)

History: 1.0 - 2012-11 (V. Guemas, <vguemas@ic3.cat>) - Original code

IniListDims *Create A List Of Integer Ranges*

Description

This function generates a list of arrays where those arrays contain integers from 1 to various numbers. This list of arrays is used in the other functions as a list of indices of the elements of the matrices.

Usage

```
IniListDims(dims, lenlist)
```

Arguments

dims	The dimensions of a matrix the elements of which we want to generate the indices to. For exemple, if the dimensions sent are c(3,2,5), the following list of arrays will be generated: list(c(1:3), c(1:2), c(1:5))
lenlist	lenlist is the length of the list because the list will be complemented above length(dims) by arrays of length 1. For example, if lenlist is set to 7, the previous list of arrays will be extended to: list(c(1:3), c(1:2), c(1:5), 1, 1, 1, 1)

Value

A list with lenlist elements, each with arrays with integers from 1 to the corresponding number in dims array.

Author(s)

History: 1.0 - 2011-04 (V. Guemas, <vguemas@ic3.cat>) - Original code

InsertDim *Add A Dimension To A Matrix*

Description

Add one dimension to matrix var in position posdim with length lendim and which correspond to (lendim x var matrix).

Usage

```
InsertDim(var, posdim, lendim)
```

Arguments

var	Matrix to be add a dimension to.
posdim	Position of the new dimension.
lendim	Length of the new dimension.

Value

Matrix with the added dimension.

Author(s)

History: 1.0 - 2011-03 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

LeapYear *Check Whether A Year Is Leap Year*

Description

This function tells whether a year is leap year or not.

Usage

```
LeapYear(year)
```

Arguments

year	The year to tell whether is leap year or not.

Value

Boolean telling whether the year is a leap year or not.

Author(s)

History: 1.0 - 2011-03 (V. Guemas, <vguemas@ic3.cat>) - Original code

Load *Load Experimental And Observational Data*

Description

This function loads experimental data and corresponding observational data from the specified experimental and observational datasets into two matrices with similar structures. Only data of a specified variable and set of starting dates is loaded. Once the two matrices are filled by calling this function, other functions of the Common Diagnostics package that require this data structure can be executed (e.g: Ano() to compute anomalies, Clim() to compute climatologies).

Usage

```
Load(var, exp, obs = NULL, sdates, lonmin = 0, lonmax = 360, latmin = -90,
latmax = 90, nleadtime = NULL, nmember = NULL, leadtimemin = 1,
leadtimemax = NULL, storefreq = "monthly", sampleperiod = 1,
output = "areave", method = "conservative", grid = NULL, maskmod = list(NULL,
NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL), maskobs = list(NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL, NULL,
NULL, NULL, NULL, NULL, NULL, NULL))
```

Arguments

var	One of the following: 'tas', 'prlr', 'tos', 'g500', 'g200', 'ta50', 'psl', 'hflsd', 'hfssd', 'rls', 'rss', 'rsds', 'uas', 'vas', 'siaN', 'sieN', 'sivN', 'siaS', 'sieS', 'sivS', 'moc_40N55N_1-2km', 'moc_30N40N_1-2km', 'max_moc_38N50N_500m-2km', 'max_moc_40N', 'heatc', '0-315_heatc', '373-657_heatc', '800-5350_heatc', 'mxl_heatc', 'NAtl_10N65N_heatc', 'NAtl_10N65N_0-315_heatc', 'NAtl_10N65N_373-657_heatc', 'NAtl_10N65N_800-5350_heatc', 'mxl_NAtl_10N65N_heatc', 'TAtl_30S30N_heatc', 'TAtl_30S30N_0-315_heatc', 'TAtl_30S30N_373-657_heatc', 'TAtl_30S30N_800-5350_heatc', 'mxl_TAtl_30S30N_heatc', 'NPac_10N70N_heatc', 'NPac_10N70N_0-315_heatc', 'NPac_10N70N_373-657_heatc', 'NPac_10N70N_800-5350_heatc', 'mxl_NPac_10N70N_heatc', 'TPac_30S30N_heatc', 'TPac_30S30N_0-315_heatc', 'TPac_30S30N_373-657_heatc', 'TPac_30S30N_800-5350_heatc', 'mxl_TPac_30S30N_heatc', 'Arc_65N90N_heatc', 'Arc_65N90N_0-315_heatc', 'Arc_65N90N_373-657_heatc', 'Arc_65N90N_800-5350_heatc', 'mxl_Arc_65N90N_heatc', 'Ant_90S60S_heatc', 'Ant_90S60S_0-315_heatc', 'Ant_90S60S_373-657_heatc', 'Ant_90S60S_800-5350_heatc', 'mxl_Ant_90S60S_heatc', 'TInd_30S30N_heatc', 'TInd_30S30N_0-315_heatc', 'TInd_30S30N_373-657_heatc', 'TInd_30S30N_800-5350_heatc', 'mxl_TInd_30S30N_heatc'.
exp	IMPORTANT: Place first the experiment with the largest number of members and, if possible, with the largest number of leadtimes. If not possible, it is mandatory to specify the argument nleadtime. c('EnsEcmwfDec', 'EnsUkmoDec', 'EnsCerfacsDec', 'EnsIfmDec', 'EnsEcmwfSeas', 'EnsCmccSeas', 'EnsIfmSeas', 'EnsMetfrSeas', 'EnsUkmoSeas', 'DePreSysAsimDec', 'DePreSysNoAsimDec', 'DePreSysAsimSeas', 'ECMWF_S3Seas', 'ECMWF_S4_sea', 'ECMWF_S4_ann', 'hadcm3dec', 'miroc4dec', 'miroc5dec', 'mri-cgcm3dec', 'cancm4dec1', 'cancm4dec2',

	'cnrm-cm5dec', 'knmidec', 'smhidec', 'mpimdec', 'gfdldec', 'cmcc-cmdec', 'ipsldec', 'bccdec', 'gfdlhis', 'ipslhis', 'cmcc-cmhis', 'bcchis', 'i00k', 'b013', 'b014', 'yve2', ...)
obs	c('ERA40', 'NCEP', 'ERAint', '20thCv2', 'GHCN', 'GHCNERSSTGISS', 'ERSST', 'HadISST', 'GPCP', 'GPCC', 'CRU', 'HadSLP', 'NSIDC', 'PIOMAS', 'UCL', 'DS94', 'OAFlux', 'DFS4.3', 'NCDCglo', 'NCDCland', 'NCDCoc', 'GISSglo', 'GISSland', 'GISSoc', 'HadCRUT3glo', 'HadCRUT4', 'HadSST2oc', 'CRUTEM3land')
sdates	c('ERA40', 'NCEP', 'ERAint', 'GHCN', 'GHCNERSSTGISS', 'ERSST', 'HadISST', 'GPCP', 'GPCC', 'CRU', 'HadSLP', 'NSIDC', 'PIOMAS', 'UCL', 'DS94', 'OAFlux', 'DFS4.3', 'NCDCglo', 'NCDCland', 'NCDCoc', 'GISSglo', 'GISSland', 'GISSoc', 'HadCRUT3glo', 'HadSST2oc', 'CRUTEM3land')
lonmin	>= 0, default: 0
lonmax	<= 360, default: 360
latmin	>= -90, default: -90
latmax	<= 90, default: 90
nleadtime	Optional argument needed only if the first experiment in the parameter exp does not have the largest number of leadtimes. Default: number of leadtimes of the first experiment.
nmember	Some experiments have more members in starting dates other than the first. If it is the case in the first experiment specified, fill nmember with the largest number of members.
leadtimemin	Load into the matrices only the leadtimes from leadtimemin. Default: 1.
leadtimemax	Load into the matrices only the leadtimes before leadtimemax. Default: nleadtime.
storefreq	Frequency at which the data to be loaded are stored in the repository. Can take values 'monthly' or 'daily'. Default: 'monthly'.
sampleperiod	To load only a subset between leadtimemin and leadtimemax with a specified period of subsampling. Default: 1 (all leadtimes are loaded).
output	'areave' / 'lon' / 'lat' / 'lonlat'. 1) Time series of area-averaged variables over the specified domain. 2) Time series of meridional averages as a function of longitudes. 3) Time series of zonal averages as a function of latitudes. 4) Time series of 2d fields. Default: 'areave'
method	'bilinear' / 'bicubic' / 'conservative' / 'distance-weighted' Method of interpolation for 'lon' / 'lat' / 'lonlat' output options. Default: 'conservative'.
grid	To choose the output grid. Possible options: rNXxNY or tTRgrid, ex: r96x72, t106grid. Default: model grid, argument need to be filled if various exp on various grids.
maskmod	list(mask[lon,lat]) = 1/0: kept/removed grid cell over the entire model domains. Warning: list() compulsory even if 1 model!!! Default: 1 everywhere.
maskobs	list(mask[lon,lat]) = 1/0: kept/removed grid cell over the entire observed domains, only necessary for 'areave' output option. Warning: list() compulsory even if 1 dataset !!! Default: 1 everywhere.

Details

The two output data matrices are similar and have 7 dimensions each with the following lengths: 1) Number of experimental/observational datasets. 2) Number of members. 3) Number of starting dates. 4) Number of lead-times. 5) Number of latitudes (optional). 6) Number of longitudes (optional).

For a detailed explanation of the process read the documentation attached to the package or check the comments in the code.

Value

`$mod`	Model outputs. If output = 'areave', matrix with dimensions c(nmod/nexp or nobs, nmemb/nparam, nsdates, nltime) If output = 'lat', matrix with dimensions c(nmod/nexp or nobs, nmemb/nparam, nsdates, nltime, nlat) If output = 'lon', matrix with dimensions c(nmod/nexp or nobs, nmemb/nparam, nsdates, nltime, nlon) If output = 'lonlat', matrix with dimensions c(nmod/nexp or nobs, nmemb/nparam, nsdates, nltime, nlat, nlon)
`$obs`	Observations. Matrix with same dimensions as '$mod'.
`$lat`	Latitudes of the model grid.
`$lon`	Longitudes of the model grid.

Author(s)

History: 1.0 - 2011-03 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

`Mean1Dim` *Compute Matrix Mean Along A Dimension*

Description

Averages the matrix var along the posdim dimension between limits [1] and limits [2] if limits argument is given.

Usage

```
Mean1Dim(var, posdim, narm = T, limits = NULL)
```

Arguments

`var`	Matrix to average.
`posdim`	Position of the dimension to average along.
`narm`	Ignore NA values or not.
`limits`	Limits to average between.

Value

Matrix with one dimension less containing the average.

Author(s)

History: 1.0 - 2011-04 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

MeanListDim *Average Matrix Along Various Dimensions*

Description

Averages the matrix var along a set of dimensions given as argument dims.

Usage

```
MeanListDim(var, dims, narm = T)
```

Arguments

var	Matrix to average.
dims	List of dimensions to average along.
narm	Ignore NA values or not.

Value

Matrix with the averages applied.

Author(s)

History: 1.0 - 2011-04 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

Plot2VarsVsLTime *Plot Two Variables In A Common Plot*

Description

Plots two input variables havind the same dimensions in a common plot. One plot for all experiments. Input variables should have dimensions (nexp/nmod, nltime).

Usage

```
Plot2VarsVsLTime(var1, var2, toptitle = "", ytitle = "", monini = 1,
                 freq = 12, nticks = NULL, limits = NULL,
                 listexp = c("exp1", "exp2", "exp3"),
                 listvars = c("var1", "var2"), biglab = F, hlines = NULL,
                 leg = T, siglev = F, sizetit = 1,
                 fileout = "output_plot2varsvsltime.eps", show_conf = T)
```

Arguments

var1	Matrix of dimensions (nexp/nmod, nltime).
var2	Matrix of dimensions (nexp/nmod, nltime).
toptitle	Main title, optional.
ytitle	Title of Y-axis, optional.
monini	Starting month between 1 and 12. Default = 1.
freq	1 = yearly, 12 = monthly, 4 = seasonal, ... Default = 12.
nticks	Number of ticks and labels on the x-axis, optional.
limits	c(lower limit, upper limit): limits of the Y-axis, optional.
listexp	List of experiment names, up to three, optional.
listvars	List of names of input variables, optional.
biglab	T/F for presentation/paper plot. Default = F.
hlines	c(a, b, ...) Add horizontal black lines at Y-positions a, b, ... Default: NULL.
leg	T/F if legend should be added or not to the plot. Default = T.
siglev	T/F if significance level should replace confidence interval. Default = F.
sizetit	Multiplicative factor to change title size, optional.
fileout	Name of output ps file.
show_conf	T/F to show/not confidence intervals for input variables.

Details

Examples of input: ————————

RMSE error for a number of experiments and along lead-time: (nexp, nltime)

Author(s)

History: 1.0 - 2013-03 (I. Andreu-Burillo, <isabel.andreu-burillo@ic3.cat>) - Original code

---

PlotACC *Plot Time-Series Of Anomaly Correlation Coefficient*

Description

Plots time-series of ACC in matrix with dimensions: c(nexp, nobs, nsdates, nltime, 4) with the fourth dimension of length 4 containing the lower limit of the 95 confidence interval, the computed ACC, the upper limit of the 95 interval and the 95

Usage

```
PlotACC(ACC, sdates, toptitle = "", sizetit = 1, ytitle = "", limits = NULL,
        legends = NULL, freq = 12, biglab = F, fill = F, linezero = F,
        points = T, vlines = NULL, fileout = "output_PlotACC.eps")
```

Arguments

ACC	Matrix with with dimensions: c(nexp, nobs, nsdates, nltime, 4) with the fourth dimension of length 4 containing the lower limit of the 95 confidence interval, the computed ACC, the upper limit of the 95 confidence interval and the 95
sdates	List of starting dates: c('YYYYMMDD','YYYYMMDD').
toptitle	Main title, optional.
sizetit	Multiplicative factor to scale title size, optional.
ytitle	Title of Y-axis for each experiment: c("","), optional.
limits	c(lower limit, upper limit): limits of the Y-axis, optional.
legends	List to be written in the legend as characters, optional.
freq	1 = yearly, 12 = monthly, 4 = seasonal, ... Default: 12.
biglab	T/F for presentation/paper plot, Default = F.
fill	T/F if filled confidence interval. Default = F.
linezero	T/F if a line at y=0 should be added. Default = F.
points	T/F if points instead of lines. Default = T.
vlines	List of x location where to add vertical black lines, optional.
fileout	Name of output eps file.

Author(s)

History: 1.0 - 2013-08 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

PlotAno *Plot Raw Or Smoothed Anomalies*

Description

Plots time-series of raw or smoothed anomalies of any index output from load or ano or smoothed and organized in matrix with dimensions: c(nmod/nexp, nmemb/nparam, nsdates, nltime) c(nobs, nmemb, nsdates, nltime)

Usage

```
PlotAno(exp_ano, obs_ano = NULL, sdates, toptitle = c("", "", "", "", "", "",
        "", "", "", "", "", "", "", "", "", ""), ytitle = c("", "", "", "", "",
        "", "", "", "", "", "", "", "", "", "", ""), limits = NULL,
        legends = NULL, freq = 12, biglab = F, fill = T, memb = T,
        ensmean = T, linezero = F, points = F, vlines = NULL,
        fileout = c("output1_plotano.eps", "output2_plotano.eps",
        "output3_plotano.eps", "output4_plotano.eps", "output5_plotano.eps"),
        sizetit = 1)
```

Arguments

exp_ano	Matrix with experimental data: c(nmod/nexp, nmemb/nparam, nsdates, nltime).
obs_ano	Optional matrix with observational data: c(nobs, nmemb, nsdates, nltime)
sdates	List of starting dates: c('YYYYMMDD','YYYYMMDD').
toptitle	Main title for each experiment: c("","), optional.
ytitle	Title of Y-axis for each experiment: c("","), optional.
limits	c(lower limit, upper limit): limits of the Y-axis, optional.
legends	List of observational dataset names, optional.
freq	1 = yearly, 12 = monthly, 4 = seasonal, ... Default: 12.
biglab	T/F for presentation/paper plot. Default = F.
fill	T/F if filled spread between members. Default = T.
memb	T/F if all members/ensemble-mean only should be plotted. Default = T.
ensmean	T/F if the ensemble-mean should be plotted. Default = T.
linezero	T/F if a line at y=0 should be added. Default = F.
points	T/F if points instead of lines. Default = F.
vlines	List of x location where to add vertical black lines, optional.
fileout	Name of output ps file for each experiment: c("",").
sizetit	Multiplicative factor to scale title size, optional.

Author(s)

History: 1.0 - 2011-03 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

PlotClim *Plot Climatologies*

Description

Plots time-series of climatologies of any index output from clim and organized in matrix with dimensions: c(nmod/nexp, nmemb/nparam, nltime) c(nobs, nmemb, nltime)

Usage

```
PlotClim(exp_clim, obs_clim = NULL, toptitle = "", ytitle = "", monini = 1,
         freq = 12, limits = NULL, listexp = c("exp1", "exp2", "exp3"),
         listobs = c("obs1", "obs2", "obs3"), biglab = F, leg = T,
         fileout = "output_plotclim.eps", sizetit = 1)
```

Arguments

exp_clim	Matrix with experimental data: c(nmod/nexp, nmemb/nparam, nltime)
obs_clim	Matrix with observational data, optional: c(nobs, nmemb, nltime)
toptitle	Main title, optional
ytitle	Title of Y-axis, optional.
monini	Starting month between 1 and 12. Default = 1.
freq	1 = yearly, 12 = monthly, 4 = seasonal, ... Default = 12.
limits	c(lower limit, upper limit): limits of the Y-axis, optional.
listexp	List of experiment names, optional.
listobs	List of observational dataset names, optional.
biglab	T/F for presentation/paper plot. Default = F.
leg	Whether to plot legend or not.
fileout	Name of output ps file.
sizetit	Multiplicative factor to scale title size, optional.

Author(s)

History: 1.0 - 2011-03 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

PlotEquiMap *Map Two-Dimensional Variable On A Cylindrical Equidistant Projection*

Description

Map two dimensional matrix with (longitude, latitude) dimensions in cylindrical equidistant latitude and longitude projection.

Usage

```
PlotEquiMap(var, lon, lat, toptitle = "", sizetit = 1, units = "",
            brks = NULL, cols = NULL, square = TRUE,
            filled.continents = TRUE, contours = NULL, brks2 = NULL,
            dots = NULL, axelab = T, labW = F, intylat = 20, intxlon = 20,
            drawleg = T, subsampleg = 1, numbfig = 1, colNA = "white")
```

Arguments

var	Matrix to plot with (longitude, latitude) dimensions.
lon	Array of longitudes.
lat	Array of latitudes.
toptitle	Title, optional.

sizetit	Multiplicative factor to increase title size, optional.
units	Units, optional.
brks	Colour levels, optional.
cols	List of colours, optional.
square	Map with squares (TRUE) for each grid points or smoothing (FALSE). Default: TRUE.
filled.continents	Continents filled in grey (TRUE) or represented by a black line (FALSE). Default = TRUE. Filling unavailable if crossing Greenwich. Filling unavailable if square.
contours	Matrix to add to the plot with contours. Default = NULL.
brks2	Contour levels, optional.
dots	Matrix with TRUE / FALSE flags to add black dots option only available if square = TRUE.
axelab	TRUE/FALSE, label the axis. Default = TRUE.
labW	
intylat	Interval between latitude ticks on y-axis. Default: 20deg.
intxlon	Interval between longitude ticks on x-axis. Default: 20deg.
drawleg	Draw colorbar. Can be FALSE only if square = FALSE. Must be FALSE if numbfig > 1. Default: T.
subsampleg	Supsampling factor of the interval between ticks on colorbar. Default: 1 = every colour level.
numbfig	Number of figures in the final multipanel.
colNA	

Author(s)

History: 1.0 - 2011-11 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

PlotMap *Plot Two Dimensional Data Matrix On A Map*

Description

Map two dimensional data matrix, with first dimension p (longitude points) and second dimension q (latitude points), in either cylindrical equidistant latitude and longitude projection or stereographic projection

Usage

```
PlotMap(lon, lat, data, maintit = "", legtit = "", equi = TRUE, bw = FALSE,
        cont = FALSE, reg = FALSE, ..., lonlim = c(0, 360),
        latlim = c(50, 90), orientation = NULL, mapdat = "world",
        xmaplim = c(-180, 180), ymaplim = c(50, 90), longrds = NULL,
        latgrds = NULL, breaks = NULL, n = 11, colours = NULL)
```

Arguments

lon	Vector of p longitude coordinates in ascending order.
lat	Vector of q latitude coordinates in ascending order.
data	Matrix of p rows (longitudes) and q columns (latitudes) that contains the data values to be plotted.
maintit	String with the main title of the plot. If not provided no title is displayed.
legtit	String with the title of the colourbar. If not provided no title is displayed.
equi	Logical. If FALSE produces stereographic projection plot. Default is TRUE.
bw	Logical. If TRUE produces black, gray and white plot. Default is FALSE.
cont	Logical. If TRUE adds contours to the image plot. Default is FALSE.
reg	Logical. To be used only if plotting data in equidistant projection. If TRUE produces a plot using as much ares as possible of the display window. Default is FALSE (addequate to plot global data).
...	Additional arguments passed to 'contour'.
lonlim	Range of longitudes where data will be plotted. E.g. lonlim = c(0, 360) or lonlim = c(-180, 180)
latlim	Range of latitudes where data will be plotted. E.g. latlim = c(50, 90) or latlim = c(-90, -50)
orientation	Orientation parameter of mapproject. For Europe as a whole a reasonable setting is: orientation = c(45, 0, 7.5) The latter places the pole of projection to latitude 45 N. For the South Pole use, e.g. orientation = c(-90,0,0)
mapdat	Name of the dataset from which map data is taken from either maps or mpdata packages for producing stereographic plots. Default is "world". Other options could be for example mapdat = "worldHires" plots high resolution map, mapdat = "none" no map.
xmaplim	Range of longitudes where the map data will be drawn. In default, the map is drawn for the area of longitude and latitude provided in vectors lon and lat. Sometimes it is useful to define e.g. xmaplim = c(-180, 180), ymaplim = c(0,90), i.e. whole hemisphere. But this will be very slow if invoked with madat = "worldHires".
ymaplim	Range of latitudes where the map data will be drawn.
longrds	NA
latgrds	NA
breaks	Vector of values for the colour bar. Must always be of length(n) + 1 (see definition of n below). Default is NULL, meaning that the function produces an automatic vector of breaks based on n and the range of values of the map to be plotted.
n	Number of colours/intervals for the colour bar. Default is 11. For the default plot produced by this function n must be an odd number.
colours	Vector of colours of length n. Defauls is NULL, meaning that an automatically chosen colour (or gray) scale is produced.

Value

This function returns nothing.

Author(s)

History: 1.0 - 2005-11 (C. Coelho, - Original code - <c.a.d.s.coelho@reading.ac.uk>) - - xxxx-xx (D.J. Steinskog, - — - <dag.johan.steinskog@nersc.no>) -

---

`PlotSection` *Plot A Section*

Description

Plot a (longitude,depth) or (latitude,depth) section.

Usage

```
PlotSection(var, horiz, depth, toptitle = "", sizetit = 1, units = "",
            brks = NULL, cols = NULL, axelab = T, intydep = 200,
            intxhoriz = 20, drawleg = T)
```

Arguments

`var`	Matrix to plot with (longitude/latitude, depth) dimensions.
`horiz`	Array of longitudes/latitudes.
`depth`	Array of depths.
`toptitle`	Title, optional.
`sizetit`	Multiplicative factor to increase title size, optional.
`units`	Units, optional.
`brks`	Colour levels, optional.
`cols`	List of colours, optional.
`axelab`	TRUE/FALSE, label the axis. Default = TRUE.
`intydep`	Interval between depth ticks on y-axis. Default: 200m.
`intxhoriz`	Interval between longitude/latitude ticks on x-axis. Default: 20deg.
`drawleg`	Draw colorbar. Default: TRUE.

Author(s)

History: 1.0 - 2012-09 (V. Guemas, <vguemas@ic3.cat>) - Original code

PlotVsLTime *Plot Data Along Forecast Time*

Description

Plots mean InterQuartile Range, Maximum-Mininum, Standard Deviation, Median Absolute Deviation, the trends, the RMS and correlation between modelled and observed data against the forecast time for all input experiments.

Usage

```
PlotVsLTime(var, toptitle = "", ytitle = "", monini = 1, freq = 12,
nticks = NULL, limits = NULL, listexp = c("exp1", "exp2", "exp3"),
listobs = c("obs1", "obs2", "obs3"), biglab = F, hlines = NULL, leg = T,
siglev = F, fileout = "output_plotvsltime.eps", sizetit = 1, show_conf = T)
```

Arguments

var	Matrix of dimensions: (nexp/nmod, 3 ,nltime) or (nexp/nmod, nobs, 3 ,nltime)
toptitle	Main title, optional.
ytitle	Title of Y-axis, optional.
monini	Starting month between 1 and 12. Default = 1.
freq	1 = yearly, 12 = monthly, 4 = seasonal, ... Default = 12.
nticks	Number of ticks and labels on the x-axis, optional.
limits	c(lower limit, upper limit): limits of the Y-axis, optional.
listexp	List of experiment names, optional.
listobs	List of observation names, optional.
biglab	T/F for presentation/paper plot. Default = F.
hlines	c(a,b, ..) Add horizontal black lines at Y-positions a,b, ... Default = NULL.
leg	T/F if legend should be added or not to the plot. Default = T.
siglev	T/F if significance level should replace confidence interval. Default = F.
fileout	Name of output ps file.
sizetit	Multiplicative factor to change title size, optional.
show_conf	T/F to show/not confidence intervals for input variables.

Details

Examples of input: ——————

1- Model output from load/ano/smoothing: (nmod, nmemb, sdate, nltime) then passed through spread(var, posdim = 2, narm = T) and mean1dim(var, posdim = 3, narm=T) or through trend(mean1dim(var, 2), posTR = 2): (nmod, 3, nltime) plots average along start dates of IQR/MaxMin/SD/MAD across members or trends of the ensemble-mean computed accross the start dates

2- Model and observed output from load/ano/smoothing: (nmod, nmemb, sdate, nltime) and (nobs, nmemb, sdate, nltime) then averaged along members mean1dim(var_exp/var_obs, posdim = 2): (nmod, sdate, nltime) and (nobs, sdate, nltime) then passed through corr(exp, obs, posloop = 1, poscor = 2) or RMS(exp, obs, posloop = 1, posRMS = 2): (nmod, nobs, 3, nltime) plots correlations or RMS between each exp & each obs against lead-time.

Author(s)

History: 1.0 - 2011-03 (V. Guemas, <vguemas@ic3.cat>) - Original code - 2013-03 (I. Andreu-Burillo, - Introduced parameter - <isabel.andreu-burillo@ic3.cat>) - sizetit - 2013-10 (I. Andreu-Burillo, - Introduced parameter - <isabel.andreu-burillo@ic3.cat>) - show_conf

---

`ProjMap` *Generate Pixel Plot Of Matrix Z*

Description

Generate a pixel plot of matrix z on a specified map projection. This routine deals with sterepgraphic and lambert projections from the mapproj package.

Usage

```
ProjMap(x, y, z, rotpol = c(0, 90), lonlim, latlim, projection = "",
        parameters = NULL, orientation = NULL, mapdat = "world", xmaplim,
        ymaplim, thin.map = 0, col = heat.colors(12), breaks,
        na.col = "white", longrds = seq(-180, 180, by = 20), longr = "yes",
        latgrds = seq(-80, +80, by = 20), latgr = "yes", ltygrds = 3,
        lwdgrds = 1, con = FALSE, ...)
```

Arguments

`x`	A vector of longitudes
`y`	A vector of latitudes
`z`	A matrix
`rotpol`	Defines the coordinates of the rotated pole. Default is geographical north pole (rotpol = c(0., 90.)).
`lonlim`	Defines the longitude window to be plotted. If specified the window of the projection is chosen such that the entire longitude window is comprised in the plot.
`latlim`	Defines the latitude window to be plotted. If specified the window of the projection is chosen such that the entire longitude window is comprised in the plot.
`projection`	Either lambert or stereographic provided by mapproject(). If an empty string is used, the projection used in the last application of mapproject will be used. The last projection settings are stored in variable .Last.projection.
`parameters`	NA

orientation	Is the orientation parameter of mapproject.
mapdat	Is the name of the dataset from which map data is taken. Default is "world".
xmaplim	Defines a longitude window from which map data shall be plotted. In default, all map data comprised in the area of x,y is used for plotting. In some cases this does not comprise all the map data of the projection window.
ymaplim	Define a latitude window from which map data shall be plotted. In default, all map data comprised in the area of x,y is used for plotting.
thin.map	Whether to thin out details of the map (political, continental outlines). thin.map = 1 means that every second point on the map is omitted. thin.map = 2 means that this thinning out is applied twice, etc. By default thin.map = 0, which means no thinning.
col	Breaks the color palette and breaks to be used for the pixels. Length of breaks needs to be one larger than length of colors. If breaks is not specified equidistant breaks from min to max are used.
breaks	NA
na.col	The color to use for pixels with NA (in image). Useful values are "white" and "grey80".
longrds	Array of longitude circles for which grid-lines shall be plotted.
longr	String indicating if the longitude circles should be plotted. Default is "no".
latgrds	Array of latitude circles for which grid-lines shall be plotted.
latgr	String indicating if the latitude circles should be plotted. Default is "no".
ltygrds	The type of line for longitude and latitude circles.
lwdgrds	The line thickness for longitude and latitude circles.
con	NA
...	NA

Details

Plots a pixel plot with either the lambert or stereographical projection. This script is used in the function PlotMap.R.

Value

A pixel plot of the chosen area and projection.

Author(s)

History: 1.0 - 2005-11 (D.J. Steinskog, - Original code - <dag.johan.steinskog@nersc.no>) - - xxxx-xx (C. Frei, - — - <Christoph.Frei@meteoswiss.ch>) -

RatioRMS *Compute Ratio Between Two RMSs.*

Description

Matrix var_exp1 / var_exp2 / var_obs should have the same dimensions. The ratio RMSE(var_exp1, var_obs) / RMSE(var_exp2, var_obs) is output. The p-value is provided by a two-sided Fischer test.

Usage

```
RatioRMS(var_exp1, var_exp2, var_obs, posRMS = 1)
```

Arguments

var_exp1	Matrix of experimental data 1.
var_exp2	Matrix of experimental data 2, same dimensions as var_exp1.
var_obs	Matrix of observational data, same dimensions as var_exp1.
posRMS	Dimension along which RMS are to be computed.

Value

Matrix with the same dimensions than var_exp1/var_exp2/var_obs except along posRMS where the dimension has length 2. The dimension 2 corresponds to the ratio between the computed RMSE (RMSE1/RMSE2) and the p.value of the two-sided Fisher test with Ho: RMSE1/RMSE2 = 1.

Author(s)

History: 1.0 - 2011-11 (V. Guemas, <vguemas@ic3.cat>) - Original code

RatioSDRMS *Compute Ratio Between Standard Deviations*

Description

Matrix var_exp & var_obs should have the common diagnostic structure: between c(nmod/nexp, nmemb/nparam, nsdates, nltime) and c(nmod/nexp, nmemb/nparam, nsdates, nltime, nlevel, nlat, nlon) The ratio between the standard deviation of the members around the ensemble mean in var_exp and the RMSE between var_exp and var_obs is output for each experiment and each observational dataset. The p-value is provided by a one-sided Fischer test.

Usage

```
RatioSDRMS(var_exp, var_obs)
```

Arguments

var_exp	Model data: c(nmod/nexp, nmemb/nparam, nsdates, nltime) up to c(nmod/nexp, nmemb/nparam, nsdates, nltime, nlevel, nlat, nlon)
var_obs	Observational data: c(nobs, nmemb, nsdates, nltime) up to c(nobs, nmemb, nsdates, nltime, nlevel, nlat, nlon)

Value

Matrix with c(nexp/nmod, nobs, 2, nltime) up to c(nexp/nmod, nobs, 2, nltime, nlevel, nlat, nlon) dimensions. The dimension 2 corresponds to the ratio (SD/RMSE) and the p.value of the one-sided Fisher test with Ho: SD/RMSE = 1.

Author(s)

History: 1.0 - 2011-12 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

Regression *Compute The Regression Of A Matrix On Another Along A Dimension*

Description

Compute the regression of matrice vary on matrice varx along the (posREG)th dimension by least square fitting. Provides the slope of the regression, the associated confidence interval, and the intercept. Provide also the vary data filtered out from the regression onto varx. The confidence interval relies on a student-T distribution.

Usage

```
Regression(vary, varx, posREG = 2)
```

Arguments

vary	Matrix of any number of dimensions up to 10.
varx	Matrix of any number of dimensions up to 10.
posREG	Position along which to compute the regression.

Value

$regression	Matrix with same dimensions as varx and vary except along posREG dimension which is replaced by a length 4 dimension, corresponding to the lower limit of the 95 the 95 along all the other dimensions.
$filtered	Same dimensions as vary filtered out from the regression onto varx along the posREG dimension.

Author(s)

History: 1.0 - 2013-05 (V. Guemas, <vguemas@ic3.cat>) - Original code

RMS *Compute Root Mean Square*

Description

Matrix var_exp & var_obs should have the same dimensions except along posloop where the length can be different (nexp & nobs). RMS compute for each grid point the Root Mean Square difference along the posRMS dimension between each model data in 1:nexp and each observation in 1:nobs which gives nexp x nobs RMS dif for each other grid point of the matrix. The confidence interval relies on a chi2 distribution.

Usage

```
RMS(var_exp, var_obs, posloop = 1, posRMS = 2, compROW = NULL, limits = NULL)
```

Arguments

var_exp	Matrix of experimental data.
var_obs	Matrix of observational data, same dimensions as var_exp except along posloop.
posloop	Dimension nobs and nexp.
posRMS	Dimension along which RMSE are to be computed.
compROW	Data taken into account only if (compROW)th row is complete. Default = NULL.
limits	Complete between limits[1] & limits[2]. Default = NULL.

Value

Matrix with c(length(posloop) in var_exp, length(posloop) in var_obs, 3, all other dimensions of var_exp & var_obs except posRMS). The dimension 3 corresponds to the lower limit of the 95 interval, the computed RMSE and the upper limit of the 95 interval.

Author(s)

History: 1.0 - 2011-05 (V. Guemas, <vguemas@ic3.cat>) - Original code

RMSSS *Compute Root Mean Square Skill Score*

Description

Matrix var_exp & var_obs should have the same dimensions except along posloop where the length can be different (nexp & nobs). RMSSS compute for each grid point the Root Mean Square Skill Score along the posRMS dimension for each model in 1:nexp against each observational data in 1:nobs which gives nexp x nobs RMSSS for each other grid point of the matrix. The p-value is provided by a one-sided Fisher test.

Usage

```
RMSSS(var_exp, var_obs, posloop = 1, posRMS = 2)
```

Arguments

var_exp	Matrix of experimental data.
var_obs	Matrix of observational data, same dimensions as var_exp except along posloop.
posloop	Dimension nobs and nexp.
posRMS	Dimension along which RMSE are to be computed.

Value

Matrix with c(length(posloop) in var_exp, length(posloop) in var_obs, 2, all other dimensions of var_exp & var_obs except posRMS). The dimension 2 corresponds to the RMSSS and the p.value of the one-sided Fisher test with Ho: RMSSS = 0.

Author(s)

History: 1.0 - 2012-04 (V. Guemas, <vguemas@ic3.cat>) - Original code

Season *Compute Seasonal Means*

Description

Computes seasonal means on time-series organized in a matrix of any number of dimensions up to 10 dimensions.

Usage

```
Season(var, posdim = 4, monini, moninf, monsup)
```

Arguments

var	Matrix in which are the time-series.
posdim	Rank of the dimension along with to compute seasonal means.
monini	Position in the year of the first month of the time-series: 1 to 12.
moninf	Position in the year of the month when to start the seasonal means: 1 to 12.
monsup	Position in the year of the month when to stop the seasonal means: 1 to 12.

Value

Matrix with same dimensions as var except along the (posdim)th dimension which length corresponds to the number of seasons. Partial seasons are not accounted for.

Author(s)

History: 1.0 - 2011-03 (V. Guemas, <vguemas@ic3.cat>) - Original code

---

SelIndices *Slice A Matrix Along A Dimension*

Description

This function allows to select a subensemble from a matrix of any dimensions, providing the dimension along which the user aims at cutting the matrix and between which indices.

Usage

```
SelIndices(var, posdim, limits)
```

Arguments

var	A matrix of any rank and any dimensions.
posdim	The dimension along which a submatrice should be selected.
limits	The lower and upper indice of the selection along the (posdim)th dimension.

Value

The sliced matrix.

Author(s)

History: 1.0 - 2011-04 (V. Guemas, <vguemas@ic3.cat>) - Original code

Smoothing *Smooth Time-Series*

Description

Smoothes time-series organized in a matrix of any number of dimensions up to 10 dimensions (arbitrary choice, can be enlarged if needed –> ask Virginie)

Usage

```
Smoothing(var, runmeanlen = 12, numdimt = 4)
```

Arguments

var	Matrix in which are the time-series to smooth.
runmeanlen	Running mean length in number of time-steps (typically months).
numdimt	Dimension along with to smooth.

Value

Matrix with same dimensions as var and smoothed time-series along the (numdimt)th dimension.

Author(s)

History: 1.0 - 2011-03 (V. Guemas, <vguemas@ic3.cat>) - Original code

Spectrum *Estimate Frequency Spectrum*

Description

This function estimates the frequency spectrum of the xdata array together with its 95 The spectrum estimation relies on a R built-in function and the significance levels are estimated by a Monte-Carlo method.

Usage

```
Spectrum(xdata)
```

Arguments

xdata	Input array of data.

Value

The frequency spectrum.

Author(s)

History:
1.0 - 2012-02 (V. Guemas, <vguemas@ic3.cat>) - Original code

`Spread` *Compute InterQuartile Range, Maximum-Minimum, Standard Deviation and Median Absolute Deviation*

Description

Compute InterQuartile Range, Maximum-Mininum, Standard Deviation and Median Absolute Deviation along the (posdim)th dimension of matrix var. The confidence interval is computed by bootstrapping.

Usage

```
Spread(var, posdim = 2, narm = T)
```

Arguments

`var`	Matrix of any number of dimensions up to 10.
`posdim`	Dimensions along which to compute IQR/MaxMin/SD/MAD.
`narm`	T/F if NA removed/kept before computation. Default = T.

Details

Example: ——— To compute IQR, Max-Min, SD & MAD accross members and start dates of var output from load or ano, call: spread(var, posdim = c(2, 3), narm = T)

Value

Matrix with same dimensions as var except along the first posdim dimension which is replaced by a length 3 dimension, corresponding to the lower limit of the 95 the upper limit of the 95 along all the other dimensions except for the other posdim dimensions which disappear.

`$iqr`	InterQuartile Range.
`$maxmin`	Maximum - minimum.
`$sd`	Standard deviation.
`$mad`	Median value.

Author(s)

History: 1.0 - 2011-03 (V. Guemas, <vguemas@ic3.cat>) - Original code

Trend *Compute Trends*

Description

Compute trends along the (posTR)th dimension of matrix var by least square fitting, and the associated an error interval. Provide also the detrended data. The confidence interval relies on a student-T distribution.

Usage

```
Trend(var, posTR = 2, interval = 1)
```

Arguments

var	Matrix of any number of dimensions up to 10.
posTR	Position along which to compute trends.
interval	Number of months between 2 points along posTR dimension. Default = 1.

Value

$trend	Same dimensions as var except along posTR dimension which is replaced by a length 3 dimension, corresponding to the lower limit of the 95 interval, the computed trends and the upper limit of the 95 interval for each point of the matrix along all the other dimensions.
$detrended	Same dimensions as var with linearly detrended var along the posTR dimension.

Author(s)

History: 1.0 - 2011-05 (V. Guemas, <vguemas@ic3.cat>) - Original code

Index