Manuscript (2003), **met.no KLIMA 04/03**, pp. 1–85

clim.pact-V.1.0.

By Rasmus E. Benestad

The Norwegian Meteorological Institute, PO Box 43, 0313, Oslo, Norway *

February 7, 2003

ABSTRACT

An analysis tool, `clim.pact` , has been developed for empirical downscaling and analysis of local and regional climate on both a daily and a monthly basis. This tool is built in the data analysis environment R, and consists of a contributed package. The R-software is freely available (A "GNU version of Splus") from the Internet (URL: http://cran.r-project.org), and runs on both Linux and Windows platforms. Since R and Linux require no license, this approach provides a low-budget tool for data-analysis.

The `clim.pact` package is tested with both monthly and daily data and a number of different scenarios are compared. Empirical downscaling has been carried out with `clim.pact` on January mean temperature in Bergen, Copenhagen, Helsinki, Oslo, Stockholm, and Tromsø, using single field as well as mixed-field predictors and with predictors that cover different regions. The package can easily employ a new common EOF approach for empirical downscaling. A comparison has been made between the various results in order to study the robustness of the method. Furthermore, the residuals from the model calibration has been examined for remaining trends or biases. These tests indicate that `clim.pact` skillfully reproduces local climate variations with a linear model when these are near-normally distributed. Tests with skewed data such as daily precipitation show that a linear model does not give a good description of the local series. New models, however, are easily incorporated in the `clim.pact` framework, such as analogs and conditional weather generators. The empirical downscaling is fast, and `clim.pact` hence facilitates downscaling analysis on multi-model ensembles.

KEY WORDS: Downscaling analysis Daily values Monthly values common EOFs.

* *Corresponding author: R.E. Benestad, rasmus.benestad@met.no, The Norwegian Meteorological Institute, PO Box 43, 0313 Oslo, Norway, phone +47-22 96 31 70, fax +47-22 96 30 50*

1 Introduction

In climate change studies, one important question is what implications a global warming has for the local climate. The local climate can be regarded as the result of a combination of the local geography (physiography) and the large-scale climate (circulation). Until now, global climate models (GCMs) have not been able to answer this question since their spatial resolution is too coarse to give a realistic description of the local climate in most locations. Furthermore, local effects from valleys, mountains, lakes, etc are not sufficiently taken into account to give a representative description. It may nevertheless be possible to derive information for a local climate through the means of downscaling. One approach is nested modelling (dynamical downscaling), which involves limited area models with progressively higher spatial resolution that can account for more of the geographical features than the GCM. Another approach is to employ empirical downscaling, which extract information about statistical relationships between the large-scale climate and the local climate. It is important that there is a strong underlying physical mechanism that links the large and small scale climate, as the lack of a physical basis implies weak and coincidental correlations. Linear empirical downscaling provide approximate description of the relationships between the spatial scales. Furthermore, empirical downscaling facilitates a "correction" to the simulations, so that the observations and the downscaled simulations can be regarded as directly comparable. A nice feature of empirical downscaling is that this method is fast and computationally "cheap". The drawback of empirical downscaling is that the locations and elements are limited to those of historical observations.

A new tool, `clim.pact`†, tailored for climate data has been written in order to facilitate more efficient, easier, and faster analysis. This tool consists of an R-package (*Ellner*, 2001; *Gentleman & Ihaka*, 2000), and provides additional functions in the R environment for climate analysis, visualisation, and empirical downscaling. R is a statistical software freely available from the Internet (`http://www.R-project.org/`), sometimes referred to as the "a GNU-version of Splus". Documentation on R is also freely available from the Internet‡, but is also installed locally as (hyper text mark-up language) HTML pages, so that a Web-browser can be used in an online-help facility (activated through the R commando:"help.start()"). The user-support in R is impressive, especially considering it being a freely available software with no license restrictions. An advantage of R-packages is that they combine their functionality with the large number of functions available in the R-environment as well as from other packages.

A large number of contributed packages are available for the R environment, which are voluntary contributions from R-users world wide. These packages are also available from the Internet (`http://www.R-project.org/`). An R-package consists of the actual R-code as well as documentation in the HTML and portable document format (PDF). One such package is `clim.pact`, whose pilot version (V.0.9) has been documented in *Benestad* (2003). The work on `clim.pact` has progressed, and version 1.0 is now released. The new version has undergone dramatic changes with respect to the user-syntax§, but are based on the same mathematical and logical framework as version 0.9. Version 0.9 has been tested extensively, and the evaluation of `clim.pact`-V.0.9 is documented in *Benestad* (2003). The downscaling in `clim.pact` can easily be set up in a way that incorporates common principal components (*Flury*, 1988; *Sengupta & Boyle*, 1993, 1998; *Barnett*, 1999) similar to the work published by *Benestad* et al. (2002); *Benestad* (2002a,b, 2001). The difference main difference between method used to derive these published results and `clim.pact` is that the default method used by `clim.pact` is linear regression instead of CCA¶-based models. Furthermore, `clim.pact` utilises a stepwise screening procedure that aims to minimise the Akaike information criterion (AIC) (*Wilks*, 1995, 300–302), whereas the CCA-based results carried out a stepwise screening based on correlation coefficients in a cross-validation analysis (*Wilks*, 1995, p.194-198). This packages also incorporates polynomial descriptions of climatic trends (*Benestad*, 2002c) and provides an easy way to pre-process climate data (spatial maps and station records), as well as doing the "house keeping" in terms of matching time stamps, etc. Additional features include composites and spatial correlation analysis.

†Available from the CRAN Internet site (http://cran.r-project.org/) under the link to 'contributed packages'.
‡"An Introduction to R", "the R language definition", "Writing R Extensions", "R Data Import/Export", "R Installation and Administration", "The R Reference Index", in additional to frequently asked questions (FAQs), Contributed and newsletters.
§How the functions are called.
¶CCA = canonical correlation analysis.

The purpose of this report is primarily to give the readers an up-to-date documentation on the use of `clim.pact`-V.1.0.

Acknowledgments: This work was done under the Norwegian Regional Climate Development under Global Warming (RegClim) programme, and was supported by the Norwegian Research Council (Contract NRC-No. 120656/720) and the Norwegian Meteorological Institute. The analysis was carried out using the `R` (*Ellner*, 2001; *Gentleman & Ihaka*, 2000) data processing and analysis language, which is freely available over the Internet (URL http://www.R-project.org/). The

References

Barnett, T.P., 1999. Comparison of Near-Surface Air Temperature Variability in 11 Coupled Global Climate Models. *Journal of Climate*, **12**, 511–518.

Benestad, R.E., 2001. A comparison between two empirical downscaling strategies. *Int. J. Climatology*, **21**(November), 1645–1668. DOI 10.1002/joc.703.

Benestad, R.E., 2002a. Empirically downscaled multi-model ensemble temperature and precipitation scenarios for Norway. *Journal of Climate*, **15**(21), 3008–3027.

Benestad, R.E., 2002b. Empirically downscaled temperature scenarios for northern Europe based on a multi-model ensemble. *Climate Research*, **21 (2)**(June), 105–125.

Benestad, R.E., 2002c. What can present climate models tell us about climate change? *Climatic Change*, **accepted**.

Benestad, R.E., 2003. *Downscaling analysis for daily and monthly values using clim.pact-V.0.9.* KLIMA 01/03. met.no, PO Box 43 Blindern, 0313 Oslo, Norway.

Benestad, R.E., Hanssen-Bauer, I., & Førland, E.J., 2002. Empirically downscaled temperature scenarios for Svalbard. *Atmospheric Science Letters*, September 18, doi.10.1006/asle.2002.0051.

Ellner, Stephen P., 2001. Review of R, Version 1.1.1. *Bulletin of the Ecological Society of America*, **82**(2), 127–128.

Flury, B., 1988. *Common Principal Components and Related Multivariate Models.* Wiley Series in Probability and Mathematical Statistics. New York: Wiley.

Gentleman, R., & Ihaka, R., 2000. Lexical Scope and Statistical Computing. *Journal of Computational and Graphical Statistics*, **9**, 491–508.

Sengupta, S., & Boyle, J. S., 1998. Using Common Principal Components in Comparing GCM Simulations. *Journal of Climate*, **11**, 816–830.

Sengupta, S. K., & Boyle, J. S., 1993 (November). *Statistical Intercomparison of Global Climate Models: A Common Principal Component Approach.* Tech. rept. 13. PCMDI, Lawrence Livermore National Laboratory, California, USA, [http://www-pcmdi.llnl.gov/pcmdi/pubs/pdf/13.pdf].

Wilks, D.S., 1995. *Statistical Methods in the Atmospheric Sciences.* Orlando, Florida, USA: Academic Press.

APPENDIX

2 Sample sessions.

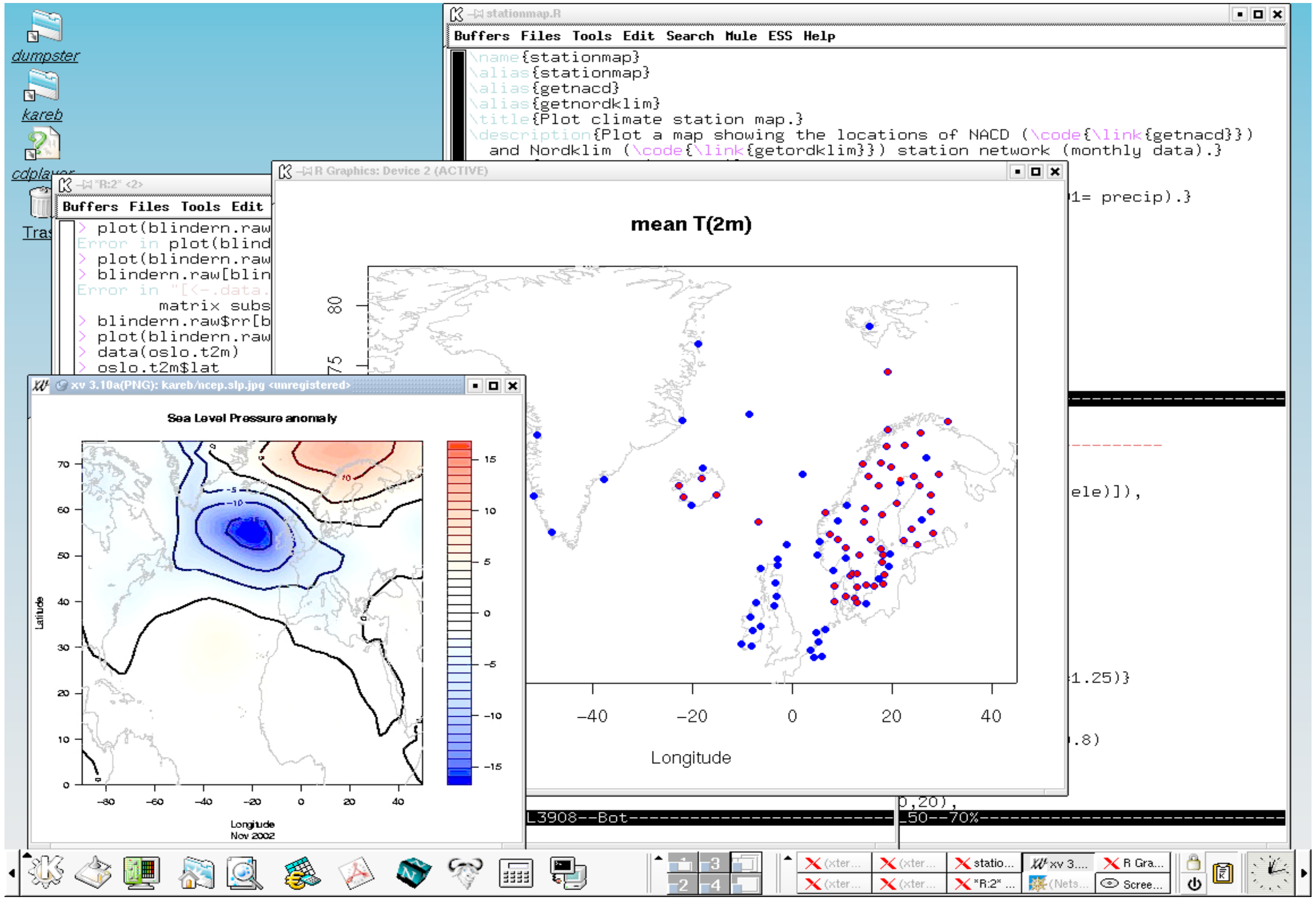
3 R manual for clim.pact

Clim.pact

Http://cran.p-project.org/

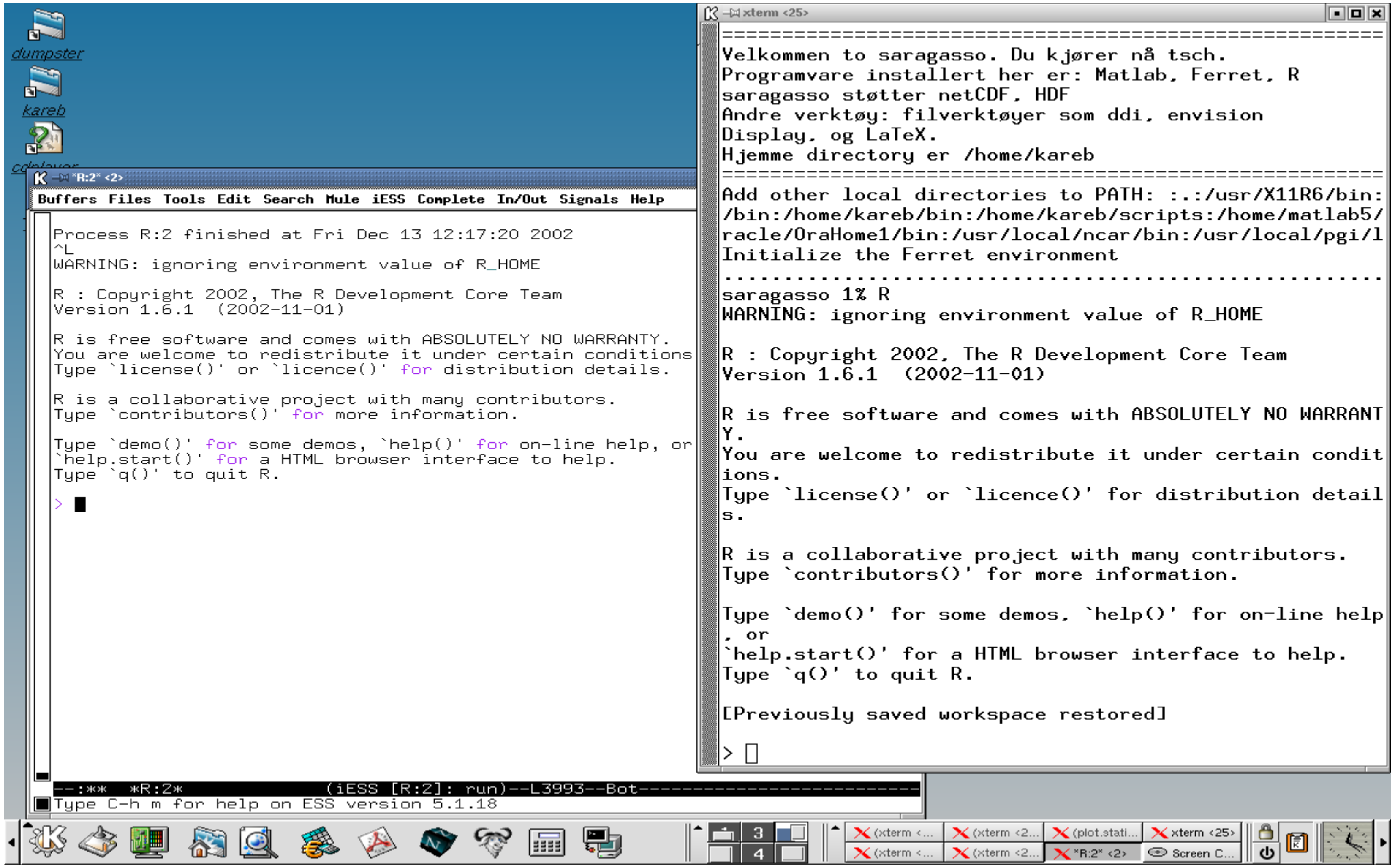
An analytical tool for climate analysis

R.E. Benestad



A cook book recipe

Can be run on PCs (click), in emacs on Linux (with the ESS package installed, or in ordinary shells:



To start R

- In emacs:
  - Type 'Esc' & 'X'
  - Type 'R'
  - <return> or name of working directory.
- In Linux shell
  - Type 'R'

In Windows: click, click click..

Basic commands:

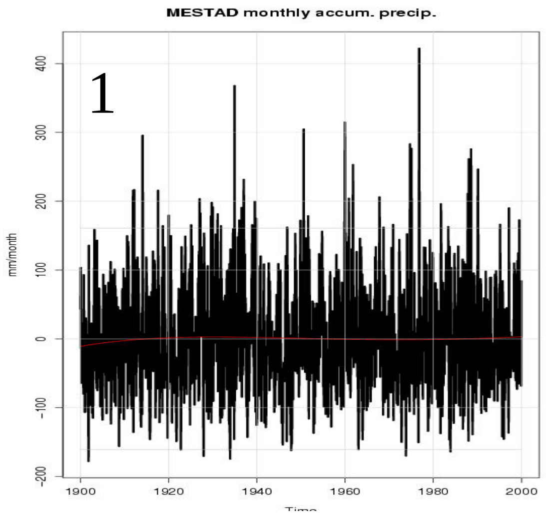
- help.start()
- quit()
- library(clim.pact)
- ls()
- A<-3+5
- print(A)
- source(<file name>)
- data(<data name>)

- To give a help menu
- To quit R.
- Load clim.pact
- Show the variables.
- Assign 3+5(=8) to A
- Prints value of A
- Runs a script
- Reads in data

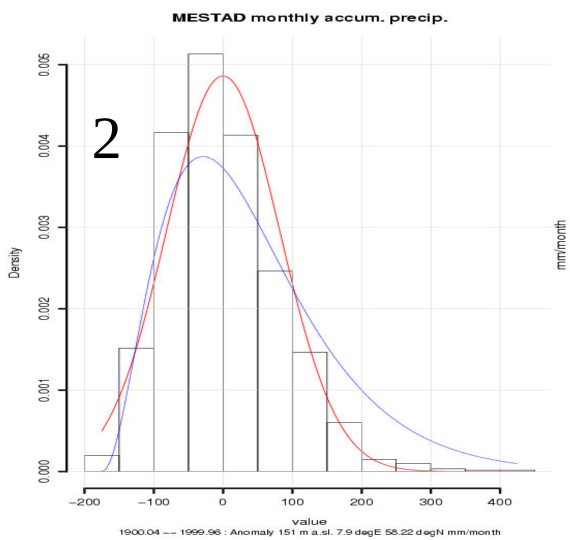
Selected clim.pact functions.

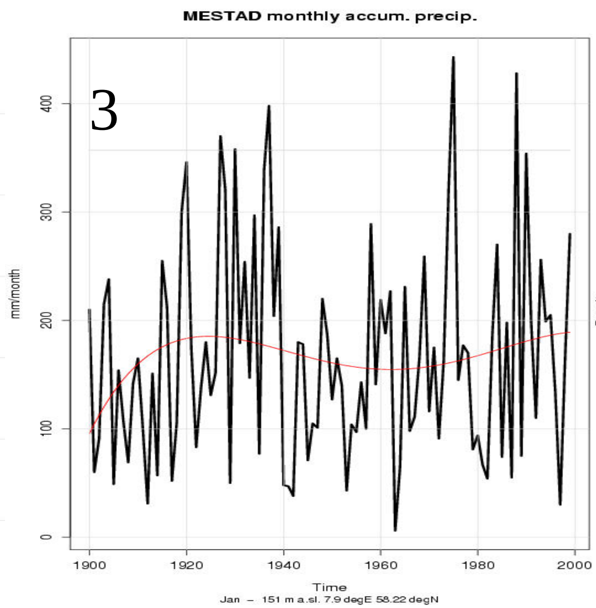
- DS() — Empirical DownScaling.
- EOF() — EOF analysis.
- getnacd() — Reads NACD data.
- mixFields() — Mixes data fields (eg SLP & T2m).
- catFields() — Concatinates fields (obs + model).
- mapField() — Shows a map of fields values
- plotDS()/plotEOF() — .Plots the results (DS or EOF).
- optint() — Optimal interpolation.
- satellite() — Polarstereographic view.
- corField() — Correlation analysis.

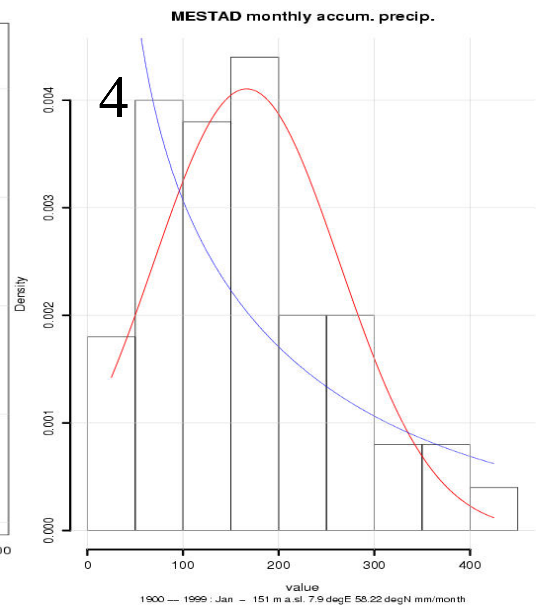
Sample session:Station record



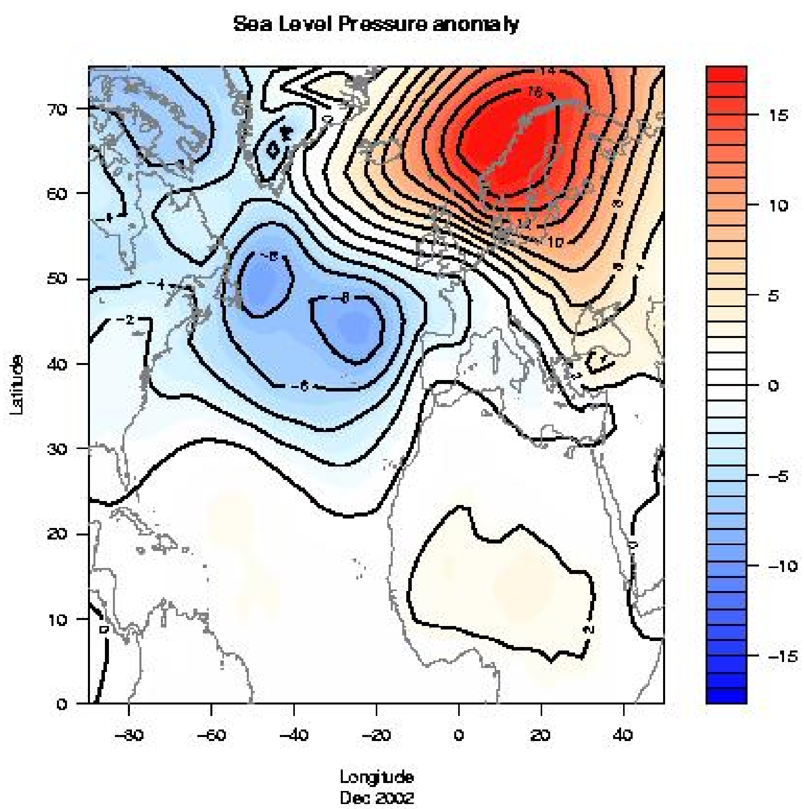

```
rm(list=ls())

avail.locs()

mestad.rr <- getnordklim("MESTAD",ele=601)

plotStation(mestad.rr)            # -> Figs 1 & 2

plotStation(mestad,mon=1)   # -> Figs 3 & 4
```





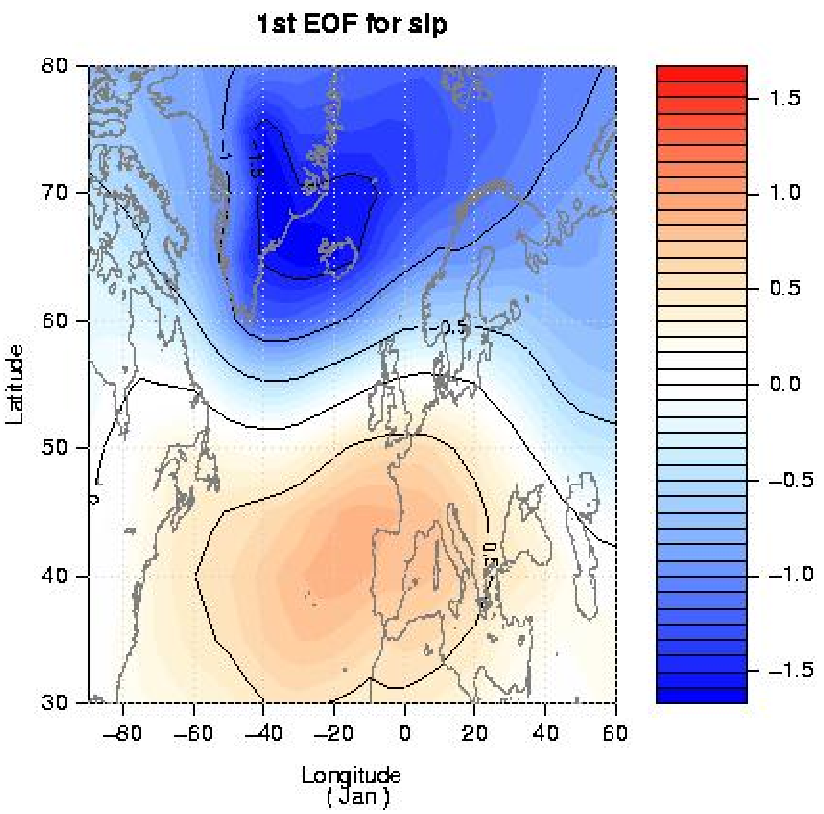


Sample session: Presenting maps



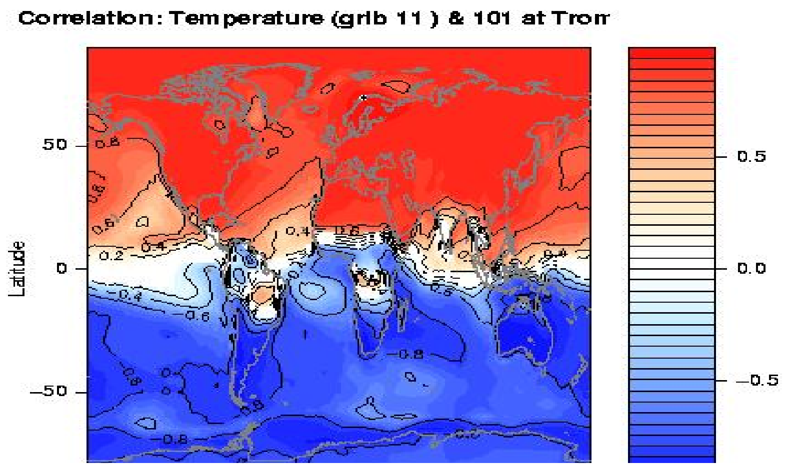

```
slp<-retrieve.nc("slp.mon.mean.nc",
            x.rng=c(-90,50),y.rng=c(0,75))
bitmap("ncep.slp.jpg",type="jpeg")
mapField(slp)
```

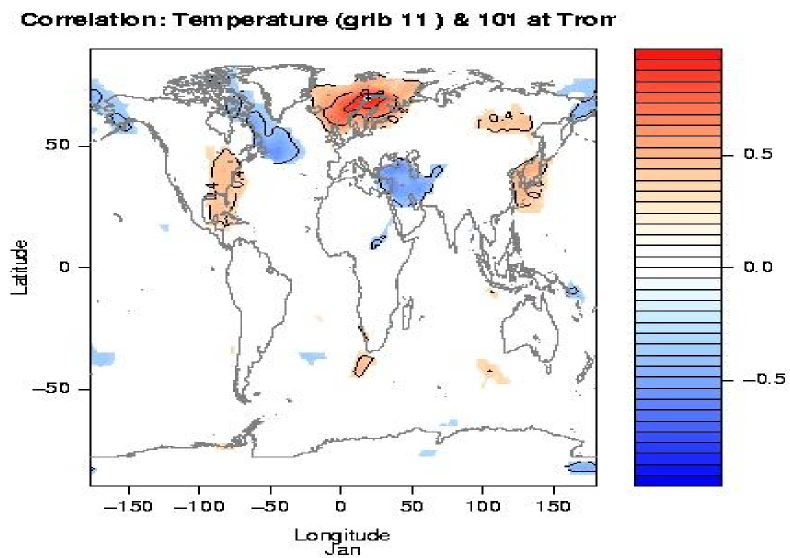
Sample session: EOF analysis




```
Slp.obs <-  retrieve.nc('SLP_oi.nc')

eof.slp <-  eof(slp.obs,mon=1)

dev2bitmap(file="slp_jan_eof.jpg",type
     ="jpeg")
```

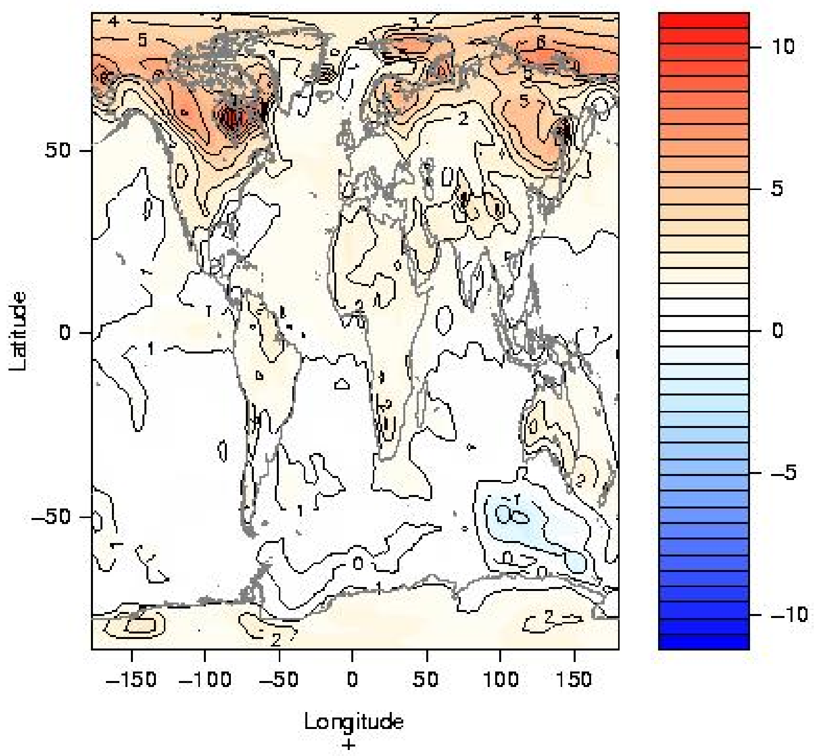
Sample session: spatial correlation





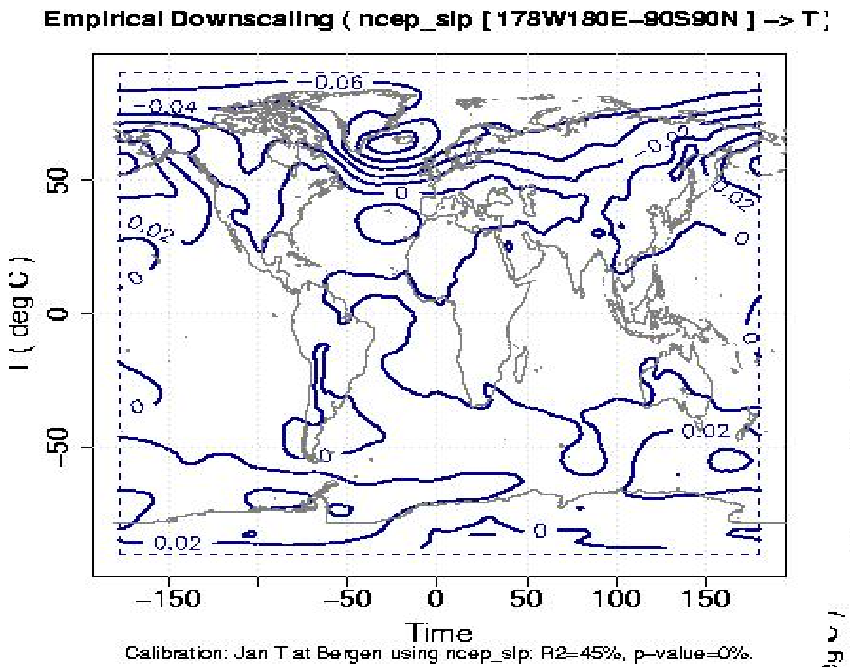

```
data(tromsoe.t2m)

obs.t2m <- retrieve.nc("ncep_t2m.nc")

corField(obs.t2m,tromsoe.t2m)

corField(obs.t2m,tromsoe.t2m,mon=1)
```

Sample session: climate change




```
tmp <- retrieve.nc("EH4OPYC_B2_temp.nc")

tmp.jan <-
    cat.fields(tmp,field.2=NULL,mon=1)

t.ctl <-
    meanField(tmp.jan,yy.rng=c(1900,2000))

t.sce <-
    meanField(tmp.jan,yy.rng=c(2001,2100))

map(t.sce,t.ctl)
```

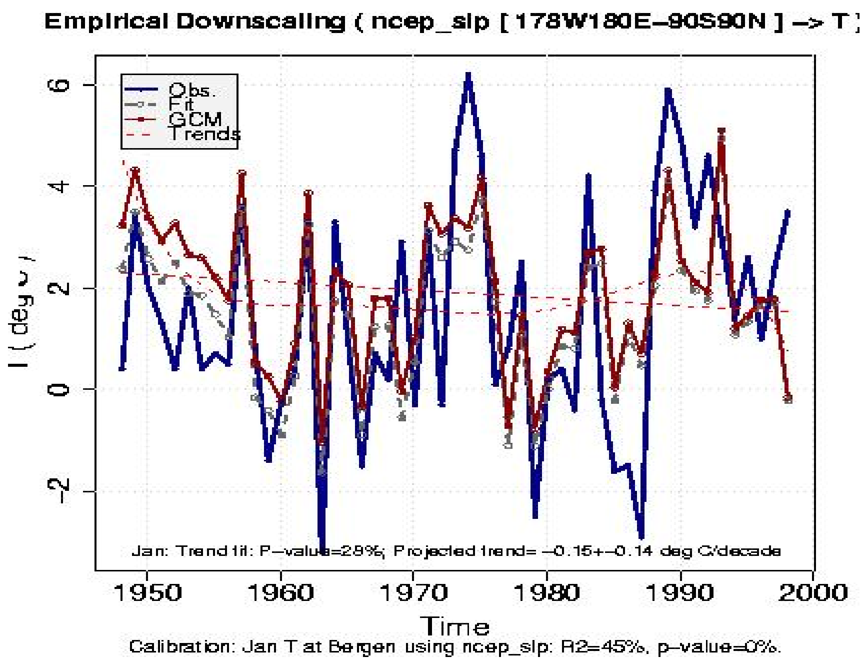
Downscaling with real observations.



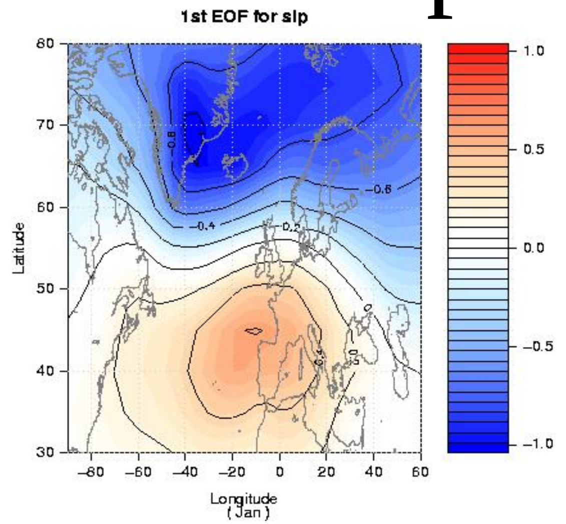
data(eof.slp)

data(bergen.t2m)

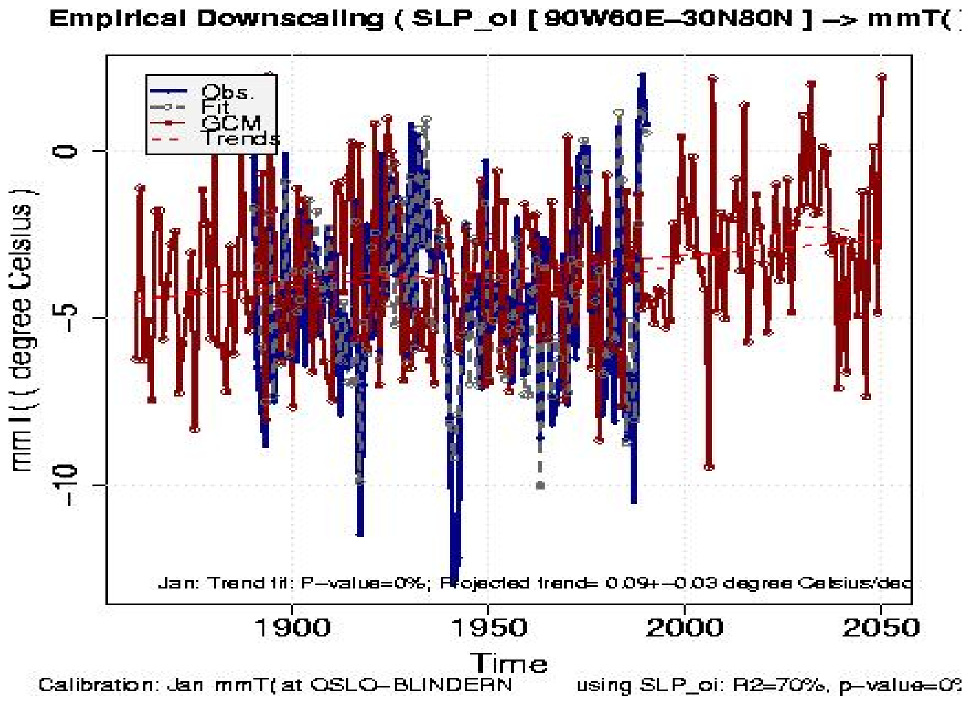
DS(preds=eof.slp,dat=bergen.t2m)

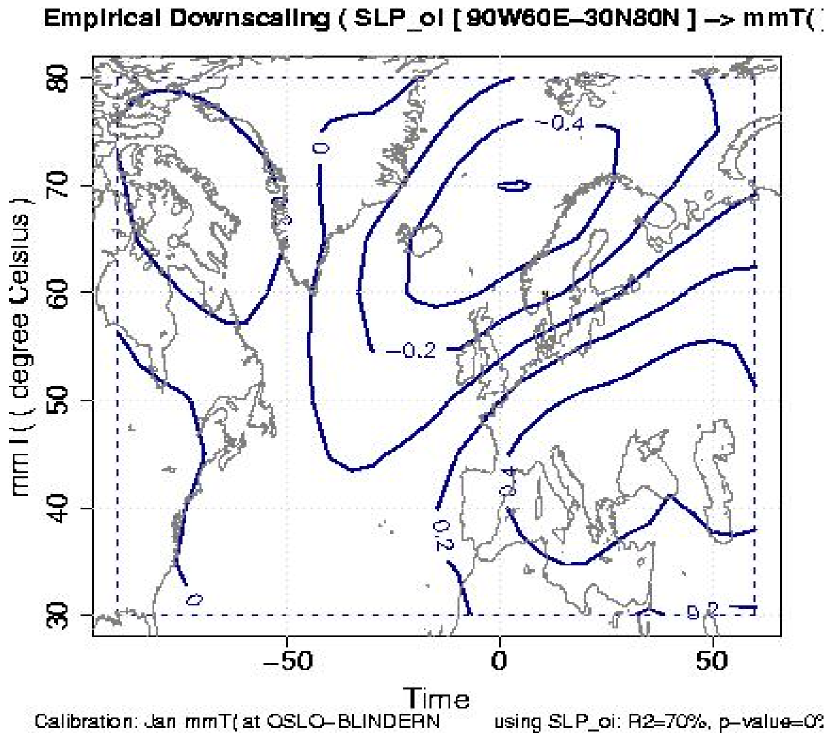


Sample session: downscaling




```
slp.obs <-  retrieve.nc('SLP_oi.nc')

slp.mod <-  retrieve.nc('mpi-gsdio_slp.nc')

slp <-  catFields(slp.obs,slp.mod,mon=1)

eof.slp <-  EOF(slp)

oslo.t2m <-  getnacd('OSLO-BLINDERN')

ds.t2m.oslo <- DS(dat=oslo.t2m, preds=eof.slp)
```





Documentation

- Online manual & PDF: clim.pact.pdf
- met.no KLIMA report 01/03 for V0.9.

R documentation

of all in 'clim.pact/man'

February 6, 2003

R topics documented:

`COn0E65N` *Convert long-lat to km-km*

Description

The function returns the distance in km from 0E 65N given the longitude and latitude. See also `km2lon` and `km2lat`.

Usage

```
COn0E65N(lon, lat)
```

Arguments

`lon`	longitude
`lat`	latitude

Value

list(y=latitudes distance,x= longitudes distance)

Author(s)

R.E. Benestad

Examples

```
library(clim.pact)
data(oslo.t2m)
print(c(oslo.t2m$lon,oslo.t2m$lat))
#[1] 10.71667 59.95000
xy<-COn0E65N(oslo.t2m$lon,oslo.t2m$lat)
oslo.t2m$lon<-xy$x
oslo.t2m$lat<-xy$y
print(c(oslo.t2m$lon,oslo.t2m$lat))
#[1]  595.4086 -560.3004
lon<-km2lon(oslo.t2m$lon,oslo.t2m$lat,x.centre=0,y.centre=65)
lat<-km2lat(oslo.t2m$lon,oslo.t2m$lat,x.centre=0,y.centre=65)
print(c(lon,lat))
#[1] 10.71667 59.95000
```

DS *Downscaling of monthly means*

Description

Identifies statistical relationships between large-scale spatial climate patterns and local climate variations for monthly and daily data series. Calibrates a linear regression model using step-wise screening and common EOFs (EOF) as basis functions. Evaluates the statistical relationship. Predicts local climate parameter from predictor fields. Works with ordinary EOFs, common EOFs (catFields) and mixed-common EOFs (mixFields). The rationale for using mixed-common EOFs is that the coupled structures described by the mixed-field EOFs may have a more physical meaning than EOFs of single fields [Benestad et al. (2002), "Empirically downscaled temperature scenarios for Svalbard", *Atm. Sci. Lett.*, doi.10.1006/asle.2002.0051].

The downscaling analysis returns a time series representing the local climate, patterns of large-scale anomalies associated with this, ANOVA, and analysis of residuals. Care must be taken when using this routine to infer local scenarios: check the R2 and p-values to check wether the calibration yielded an appropriate model. It is also important to examine the spatial structures of the large-scale anomalies assocaiated with the variations in the local climate: do these patterns make physical sense? Experiment with both single and mixed fields. It is also a good idea to check whether there are any structure in the residuals: if so, then a linear model for the relationship between the large and small-scale structures may not be appropriate. It is furthermore important to experiment with predictors covering different regions [ref: Benestad (2001), "A comparison between two empirical downscaling strategies", *Int. J. Climatology*, **vol 21**, Issue 13, pp.1645-1668. DOI 10.1002/joc.703].

The function `ds()` is a generic routine which in principle works for when there is any real statistical relationship between the predictor and predictand. The predictand is therefore not limited to a climate variable, but may also be any quantity affected by the regional climate. *It is important to stress that the downscaling model must reflect a well-understood (physical) relationship.*

The trend-estimation uses regression to fit a 5th-order polynomial (in time) to fit the observed time series. The rate-of-change is estimated by taking the time-derivative of this equation. If

$$y = c_0 + c_1 x + c_2 x^2 + c_3 x^3 + c_4 x^4 + c_5 x^5,$$

where x is the time, then the rate-of-change is:

$$y = c_1 + 2c_2 x + 3c_3 x^2 + 4c_4 x^3 + 5c_5 x^4.$$

[ref: Benestad (2002), What can present climate models tell us about climate change?, *Climatic Change*, accepted.]

The routine uses a step-wise regression (step) using the leading EOFs. The calibration is by default carried out on de-trended data [ref: Benestad (2001), "The cause of warming over Norway in the ECHAM4/OPYC3 GHG integration", *Int. J. Clim.*, 15 March, **vol 21**, p.371-387.].

The downscaled scenario is saved in a text file in the output directory (default: 'output').

The course notes from Environmental statistics for climate researchers `http://www.gfi.uib.no/~nilsg/kurs/notes/course.html` is a useful reference to statistical modelling and regression.

Usage

```
DS(dat,preds,mon=NULL,direc="output/",cal.id=NULL,
              ldetrnd=TRUE,i.eofs=seq(1,8,by=1),ex.tag="",
              method="lm",plot=TRUE,leps=FALSE,param="t2m",
              plot.res=FALSE,plot.rate=FALSE,xtr.args="")
```

Arguments

`dat`	A climate.station object (`station.obj` or `station.obj.dm`). [e.g. from `getnacd`, `getnordklim` or `station.obj`].
`preds`	The predictor `EOF`.
`mon`	month or season to downscale, this is automatically changes if predictor only contains a different month (this is normally a redundant feature).
`direc`	name of directory inwhich the output is dumped (e.g. figures, tables).
`cal.id`	ID tag used for calibration. By default use the first field (`catFields`) for calibration.
`ldetrnd`	F for no detrending; T for removing linear trends before model calibration.
`i.eofs`	select which EOFs to include in the setp-wise screening.
`ex.tag`	Extra labelling tag for file names for experiments.
`method`	Sets the method to use for regression.
`plot`	'TRUE' produces figures.
`leps`	'TRUE' produces EPS figures (files).

`param`	Name of parameter (for plot labels).
`plot.res`	'TRUE' shows statistics for residuals.
`plot.rate`	'TRUE' shows analysis of rate-of-change.
`xtr.args`	Extra/additional arguments in the formula.

Value

A 'ds' object - a list of following elements:

X.1 .. X.n	1..nth predictor pattern for n fields (mixFields).
lon.1 ..	Longitude coordinate of spatial fields (a vector).
lat.1 ..	Latitude coordinate of spatial fields (a vector).
n.fld	Number of fields (different types of predictors, mixFields).
unit	Unit of quantity in station series.
pred.name	Name of predictor.
lon.loc	Longitude of predictand location.
lat.loc	Latitude of predictand location
yy.gcm	Years corresponding to scenario (GCM).
mm.gcm	Months corresponding to scenario (GCM).
dd.gcm	Days corresponding to scenario (GCM).
yy.cal	Years corresponding to observation (Calibration).
mm.cal	Months corresponding to observation (Calibration).
dd.cal	Days corresponding to observation (Calibration).
yy.o	Years corresponding to station series (obs.).
mm.o	Months corresponding to station series (obs.).
dd.o	Days corresponding to station series (obs.).
rate.ds	Estimated linear rate of change of downscaled scenario.
rate.err	Error estimate for rate.ds.
gcm.trnd.p	P-value of linear trend in downscaled scenario.
fit.p	ANOVA p-value for fit between large-scale and small-scale variability(from regression analysis).
fit.r2	ANOVA R2 for fit between large-scale and small-scale variability (from regression analysis).
pre.gcm	The downscaled scenario (a vector).
pre.y	The downscaled results using the calibration data.
location	Nsme of location of predictor.
gcm.stat	ANOVA of linear trend fit to scrnario.
month	Month of study (0-> all months).
v.name	Name of downscaled element.
region	Region used for downscaling.
pre.fit	Linear fit to prediction (downscaled scenario) (a vector).
pre.p.fit	Polynomial fit to the downscaled scenario.
tr.est.p.fit	Rate of change derived from a fifth-order polynomial trend-fit to prediction (downscaled scenario) (a vector).
id.1, id.2	IDs labelling which data was used for calibration (id.1).

Author(s)

R.E. Benestad

Examples

```
library(clim.pact)
data("oslo.t2m")
data("eof.mc")
a<-DS(dat=oslo.t2m,preds=eof.mc,plot=FALSE)
```

EOF *Empirical Orthogonal Functions (EOFs).*

Description

Computes EOFs (a type of principal component analysis) for combinations of data sets, typically from the NCEP reanalysis and corresponding data from climate models. Preprocessing by `catFields` allows for common EOF analysis [ref: Benestad (2001), "A comparison between two empirical downscaling strategies", *Int. J. Climatology*, **vol 21**, Issue 13, pp.1645-1668. DOI 10.1002/joc.703]. and `mixFields` prepares for mixed-field EOF analysis [ref. Bretherton et al. (1992) "An Intercomparison of Methods for finding Coupled Patterns in Climate Data", *J. Climate*, **vol 5**, 541-560; Benestad et al. (2002), "Empirically downscaled temperature scenarios for Svalbard", *Atm. Sci. Lett.*, doi.10.1006/asle.2002.0051].

Uncertainty estimates are computed according to North et al. (1982), "Sampling Errors in the Estimation of Empirical Orthogonal Functions", *Mon. Weather Rev.*, **vol 110**, 699-706.

NB: This routine may be computer-intensive! The computation of the EOFs tends to take some time, especially on computers/PCs with little memory (less than 128Mb) and slow processors less than 800MHz.

See the course notes from Environmental statistics for climate researchers `http://www.gfi.uib.no/~nilsg/kurs/notes/course.html` for a discussion on EOF analysis.

Usage

```
EOF(fields,l.wght=TRUE,lc180e=FALSE,direc="data/",
              lon=NULL,lat=NULL,l.stndrd=TRUE,las=1,
              mon=NULL,plot=TRUE,neofs=20,l.rm.ac=TRUE)
```

Arguments

`fields`	A field object (eg from retrieve.nc).
`l.wght`	'TRUE' applies a geographical weighting.
`lc180e`	'TRUE' centers the maps on date line (180 deg E).
`direc`	Directory for the output.
`lon`	longitudinal region of interest.
`lat`	latitudinal region of interest.

`l.stndrd`	Not yet used.
`las`	Used by filled.contour, see par.
`mon`	Month (1-12) [season (1-4) for daily data] to extract.
`plot`	'TRUE' plots the results.
`neofs`	Number of leading EOFs to retain.
`l.rm.ac`	'TRUE' removes the annual cycle.

Value

File containing an 'eof.c' object:

EOF	EOF patterns.
W	Eigen values.
PC	Principal components of common PCA.
n.fld	Number of different predictors (see mixFields).
tot.var	Sum of all W squared.
id.t	Time labels for the fields (see catFields) - used in DS.
id.x	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lon	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lat	Spatial labels for the fields (see mixFields) - used in plotEOF.
region	Describes the region analysed.
tim	Time information (usually redundant).
lon	Longitudes associated with EOF patterns.
lat	Latitudes associated with EOF patterns.
var.eof	Fractional variances associated with EOF patterns.
yy	years.
mm	months.
dd	days.
v.name	Name of element.
c.mon	Month-season information.
f.name	File name of original data.

The data is also saved as files.

Author(s)

R.E. Benestad

Examples

```
# Computes a set of mixed-common EOFs (overnight work..). This takes a while...

library(clim.pact)
x.1 <- retrieve.nc("/home/kareb/data/ncep/ncep_t2m.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
x.2 <- retrieve.nc("/home/kareb/data/ncep/ncep_slp.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
print(x.1$v.name)
```

```
print("Read GCM predictor data.")
X.1 <- retrieve.nc("data/mpi-gsdio_t2m.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
X.2 <- retrieve.nc("data/mpi-gsdio_slp.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
print(X.1$v.name)
print("Cat fields.")
xX.1 <- cat.fields(x.1,X.1,interval.1=c(1958,1998),interval.2=c(1958,2050))
xX.2 <- cat.fields(x.2,X.2,interval.1=c(1958,1998),interval.2=c(1958,2050))
xX <- mix.fields(xX.1,xX.2,mon=1,
                 interval=c(1900,2050))
print("EOF")
eof.c <- eof(xX.1,mon=1)
eof.mc <- eof(xX,mon=1)
```

`addland` *Add land contours to map.*

Description

The function superimposes land contours on a map.

Usage

```
addland(col="grey50",lwd=1)
```

Arguments

`col`	colour
`lwd`	line width

Author(s)

R.E. Benestad

Examples

```
plot(c(-90,90),c(0,80),type="n")
addland()
grid()
```

`addland` *Coordinates of coast line.*

Description

Coordinates of coast line: `lat.cont` and `lon.cont` give the latitude and longitude coordinates of the coasts (continents).

Usage

```
data(addland)
```

Format

Two vectors, lat.cont and lon.cont, of length 124622.

Source

Ferret web site

References

URL http://ferret.wrc.noaa.gov/Ferret/

Examples

```
library(clim.pact)
data(addland)
ls()                 # [1] "lat.cont" "lon.cont"
```

`anomaly.field` *Anomalies of a field object.*

Description

Estimates the anomalies of a field object. Also see `anomaly.field`.

Usage

```
anomaly.field(x,period=NULL)
```

Arguments

`x`	A field object.
`period`	Period to use as climatology NULL -> the entire series as reference clim.

Value

A field object.

Author(s)

R.E. Benestad

Examples

```
slp<-retrieve.nc("ncep_slp.nc")
slp.a<-anomaly.field(slp)
```

`anomaly.station` *Anomaly.station*

Description

Computes anomalies of a station series by subtracting the climatology. The climatology is estimated either by taking the average of the respective months over a given reference period or a least squares fit to the 6 leading harmonics, depending on the appropriateness. Also see `anomaly.field`.

Usage

```
anomaly.station(obs,period=c(1961,1990))
```

Arguments

`obs`	Monthly station series
`period`	Period to use as climatology NULL -> the entire series as reference clim.

Value

Station series object.

Author(s)

R.E. Benestad

Examples

```
library(clim.pact)
data(oslo.t2m)
oslo.t2ma<-anomaly.station(oslo.t2m)
```

avail.elem — *Available elements*

Description

The function `avail.preds` searches for available predictors and returns a list of file names. `avail.elem` returns a list of available elements and `avail.locs` a list of available locations. These functions are support functions for `getnacd` and `getnordklim`.

Usage

```
avail.ds(direc="output")
avail.eofs(direc="data")
avail.preds(direc="data")
avail.elem()
avail.locs(ele)
```

Arguments

`direc` String containing the data directory.

`ele` Integer code for element in the Nordklim dataset:

101	mean T(2m)
111	mean maximum T(2m)
112	highest maximum T(2m) [Th]
113	day of Th date Thd
121	mean minimum T(2m)
122	lowest minimum T(2m) [Tl]
123	day of Tl date Tld
401	mean SLP
601	monthly accum. precip.
602	maximum precip.
701	Number of days with snow cover (> 50% covered) days dsc
801	Mean cloud cover % N
911	mean snow depth

Value

avail.preds	vector of characters
avail.locs	a list with name, lons, lats, country, ident
avail.elem	a list with data.set, ele, name

Author(s)

R.E. Benestad

Examples

```
library(clim.pact)
avail.elem()$name
# [1] "mean T(2m)"
# [2] "mean maximum T(2m)"
# [3] "highest maximum T(2m)"
# [4] "day of Th date Thd"
# [5] "mean minimum T(2m)"
# [6] "lowest minimum T(2m)"
# [7] "day of Tl date Tld"
# [8] "mean SLP"
# [9] "monthly accum. precip."
#[10] "maximum precip."
#[11] "Number of days with snow cover (> 50
#[12] "Mean cloud cover
#[13] "mean snow depth"

# The following assumes that the subdirectory 'data' exists

avail.locs()$name[avail.locs()$country=="FIN"]
# [1] "HELSINKI"      "TURKU"         "TAMPERE"       "LAPPEENRANTA"
# [5] "JYVASKYLA"     "KUOPIO"        "KAJAANI"       "OULU"
# [9] "KUUSAMO"       "SODANKYLA"     "Maarianhamina" "Helsinki"
#[13] "Turku"         "Huittinen"     "Tampere"       "Hattula"
#[17] "Heinola"       "Virolahti"     "Lappeenranta"  "Lavia"
#[21] "Virrat"        "Orivesi"       "Jyvaeskylae"   "Vaasa"
#[25] "Ylistaro"      "Aehtaeri"      "Kuopio"        "Maaninka"
#[29] "Joensuu"       "Kestilä"       "Kajaani"       "Oulu"
#[33] "Yli-Ii"        "Pudasjärvi"    "Kuusamo"       "Sodankylae"

avail.preds()
# [1] "eof.dc.Rdata"
# [2] "eof.dmc.Rdata"
# [3] "eof.mc2.Rdata"
# [4] "eof.nn.dc.Rdata"
# [5] "eof.nn.dmc.Rdata"
# ...
```

---

bergen.dm   *Daily Bergen record.*

Description

A station record of daily mean temperature and daily precipitation from Bergen-Florida.

Usage

```
data(bergen.dm)
```

Format

a ”daily.station.record”-class object.

t2m	a vector holding daily mean temperature.
precip	a vector holding daily precipitation.
dd	a vector holding day of month.
mm	a vector holding the month.
yy	a vector holding the year.
obs.name	the name of observation: eg c(”Daily mean temperature”,”Daily precipitation”).
unit	the unite of observation: eg c(”deg C”,”mm/day”).
ele	element code: eg c(”tam”,”rr”).
station	local (national) station number.
lat	latitude.
lon	longitude.
alt	altitude.
location	name of location.
wmo.no	WMO number of station.
start	start of measurements.
yy0	first year of record.
country	name of country.
ref	reference to the data.

Source

The Norwegian Meteorological Institute, Climatology deivision.

References

The Norwegian Meteorological Institute, P.O. Box 43, 0313 Oslo, Norway (http://www.met.no).

Examples

`bergen.t2m` *Monthly mean temperature in Bergen.*

Description

A station record of monthly mean temperature from Bergen-Florida.

Usage

```
data(bergen.t2m)
```

Format

a <- list of ”monthly.station.record” class:

val	The monthly values (a 12-column matrix with one column for each year).
station	station number.
yy	The years of observation (vector).
lat,lon	Latitude and longitude of the location.
x.0E65N,y.0E65N	Distance in km from 0E, 65N.
location	Name of location .
wmo.no	WMO number.
start	Start of observatins from this location.
yy0	First year in current record.
ele	Code of theelement.
obs.name	Name of the element.
unit	Unit of the element.
country	The country in which the location is located.
quality	Code/description for data quality.
found	Flag: T - the data requested was found.
ref	Reference for the data set.

Source

The Nordklim data set: http://www.smhi.se/hfa_coord/nordklim/

References

Tuomenvirta et al. (2001), "Nordklim data set 1.0", DNMI KLIMA 08/01, pp. 26;
The Norwegian Meteorological Institute, P.O. Box 43, 0313 Oslo, Norway (http://www.met.no).

Examples

`catFields` *catFields*

Description

Concatinates fields two different gridded sets of observation. The two fields must be stored on the same spatial grid, and the routine performs a bilinear spatial interpolation to place the data on the same grid. Observations/data for representing values at n different locations at a given time (t) can be described in terms of a vector

$$\vec{x}(t) = [x_1, x_2, ...x_n].$$

The data set consists of a time series of vectors which can be represented by the means of matrices

$$X = [\vec{x}(t_1), \vec{x}(t_2), ...\vec{x}(t_n)].$$

Two different sets of observations can be represented by two matrices Y and Z with dimensions k x n and k x m respectively (k is the number of spatial points, whereas n and mIindicate the number of observations in time). The information in these two data sets are combined combining the two matrices using rbind. The major difference between this routine and rbind is that this routine takes care of all the 'house keeping' in terms of grid, time and variable information.

cat.field can be used to process single fields by setting 'field.2=NULL'. This option allows for interpolation and extraction of sub-regions or sub-intervals, removing the mean values, and selecting a paricular month or season.

The output from cat.fields can by further analysed in `EOF`. By using a concatination of two fields of similar data, eg observed and simulated sea level pressure (SLP), it is possible to carry out a common EOF analysis. The application of `DS` to the EOFs of concatinated fields provides an analysis similar to the common EOH method described in Benestad (2001), "A comparison between two empirical downscaling strategies", *Int. J. Climatology*, **vol 21**, 1645-1668, DOI 10.1002/joc.703.

Usage

```
catFields(field.1,field.2,lat=NULL,lon=NULL,plot.interp=FALSE,
          interval.1=NULL,interval.2=NULL,mon=NULL,demean=TRUE)
```

Arguments

`field.1`	A 'field.object'.
`field.2`	A 'field.object'. A 'field.2=NULL' processes single fields.
`lat`	Latitudes to extract. If NULL, use the latitudes from the first field. Otherwise interpolate both fields to latitudes.
`lon`	Longitudes to extract. See 'lat'.
`plot.interp`	Flag: 'TRUE' plots the interploation results - Used for checking interpolation.
`interval.1`	Extract the time interval for the 1st field.
`interval.2`	Extract the time interval for the 2nd field.
`mon`	Calendar month or season to extract. eg January or DJF.
`demean`	Flag: 'TRUE' subtracts the mean values. This flag should be set to 'FALSE' if for instance two time slices are concatinated and the object is to investigate the mean change between these periods (see examples in `eof.dc` or `eof.dmc`).

Value

A 'field.object'.

Author(s)

R.E. Benestad

Examples

```
library(clim.pact)
x.1 <- retrieve.nc("/home/kareb/data/ncep/ncep_t2m.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
x.2 <- retrieve.nc("/home/kareb/data/ncep/ncep_slp.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
print(x.1$v.name)

print("Read GCM predictor data.")
X.1 <- retrieve.nc("data/mpi-gsdio_t2m.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
X.2 <- retrieve.nc("data/mpi-gsdio_slp.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
print(X.1$v.name)
print("Cat fields.")
xX.1 <- catFields(x.1,X.1,interval.1=c(1958,1998),interval.2=c(1958,2050))
xX.2 <- catFields(x.2,X.2,interval.1=c(1958,1998),interval.2=c(1958,2050))
xX <- mixFields(xX.1,xX.2,mon=1,
                 interval=c(1900,2050))
print("EOF")
eof.c <- eof(xX.1,mon=1)
eof.mc <- eof(xX,mon=1)
```

---

composite.field *Composite maps*

Description

Produce composites of maps.

Usage

```
composite.field(x,y,lsig.mask=TRUE,sig.lev=0.05,s=0.42,mon=NULL,
                        lty=1,col="black",lwd=1)
```

Arguments

x	A field object.
y	A station object, a vector consisting of [-1,0,+1] or a vector consisting of the years (negative values are used for negative phase, eg. c(1991,1993,1998,-1961,-1963,-1994).
lsig.mask	FLAG: mask the regions not statistically significant.
sig.lev	Level of significance.
s	Threshold for defining whether y is high or low = s * sd(y).
mon	Month to analyse.

`lty`	Contour line type.
`col`	Contour line colour.
`lwd`	Contour line width.

Value

A map object.

Author(s)

R.E. Benestad

Examples

```
slp <- retrieve.nc("ncep_slp.nc")
data(oslo.t2m)
composite.field(slp,oslo.t2m)
```

`corEOF` *Field correlation*

Description

Produces maps of spatial correlation patterns from the EOF products

$$X = \mathbf{UWV}^T$$

according to following formula:

$$r(\mathbf{X}_{r,t},\vec{y}) = \frac{\sum_k \mathbf{W}_k \mathbf{U}_{r,k} \mathbf{W}_k \sum_t [\mathbf{V}^T_{k,t} y'_t]}{\sqrt{\sum_k \mathbf{W}_k \mathbf{U}_{r,k} \mathbf{W}_k \sum_t (\mathbf{V}^T_{k,t})^2 \times \sum_t (y'_t)^2}}.$$

Reference: Environmental statistics for climate researchers http://www.gfi.uib.no/~nilsg/kurs/notes/course.html

NOTE: This routine is not finished yet - still contains some bugs, sometimes resulting in absolute correlation values greater than unity.

Usage

```
corEOF(x,y,lsig.mask=TRUE,sig.lev=0.05,neofs=20,
                lty=1,col="black",lwd=1)
```

Arguments

`x`	An EOF.
`y`	A station series.
`lsig.mask`	Mask out values that are not statistically significant.
`sig.lev`	Level of significanse.
`neofs`	Number of modes to include.
`lty`	Contour line type.
`col`	Contour line colour.
`lwd`	Contour line width.

Value

A map object.

Author(s)

R.E. Benestad

Examples

```
data(oslo.t2m)
data(eof.slp)
corEOF(eof.slp,oslo.t2m)
```

`corField` *Field correlation*

Description

Produces maps of spatial correlation patterns.

Usage

```
corField(x,y,lsig.mask=TRUE,sig.lev=0.05,mon=NULL,
                 lty=1,col="black",lwd=1)
```

Arguments

`x`	A field object.
`y`	A station series.
`lsig.mask`	Mask out values that are not statistically significant.
`sig.lev`	Level of significanse.
`mon`	Month to analyse.
`lty`	Contour line type.
`col`	Contour line colour.
`lwd`	Contour line width.

Value

A map object.

Author(s)

R.E. Benestad

Examples

```
slp <- retrieve.nc("ncep_slp.nc")
data(oslo.t2m)
corField(slp,oslo.t2m)
```

---

delta *Delta function*

Description

The dirac delta function: delta (i,j) returns 1 if i==j, 0 otherwise. Used in `optint`.

Usage

```
delta(i,j)
```

Arguments

`i`	first index.
`j`	second index.

Value

0 or 1

Author(s)

R.E. Benestad

Examples

`eof.c` *Monthly common EOF.*

Description

Common EOFs for January mean 2-meter temperature (T(2m)).

Usage

```
data(eof.c)
```

Format

EOF	EOF patterns.
W	Eigen values.
PC	Principal components of common PCA.
n.fld	Number of different predictors (see mixFields).
tot.var	Sum of all W squared.
id.t	Time labels for the fields (see catFields) - used in DS.
id.x	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lon	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lat	Spatial labels for the fields (see mixFields) - used in plotEOF.
region	Describes the region analysed.
tim	Time information (usually redundant).
lon	Longitudes associated with EOF patterns.
lat	Latitudes associated with EOF patterns.
var.eof	Fractional variances associated with EOF patterns.
yy	years.
mm	months.
dd	days.
v.name	Name of element.
c.mon	Month-season information.
f.name	File name of original data.

Source

The common EOF was produced using `EOF`, with the combined January 2-meter air temperature data field from National Center for Environmental Prediction (NCEP; USA) reanalysis (Kalnay et al., (1996) "The NCEP/NCAR 40-Year Reanalysis Project", *Bul. Am. Met. Soc.*, **vol 77**, no 3, 437-471; e.g. see URL: `http://www.cdc.noaa.gov/index.html`) and the ECHAM4-GSDIO scenario (Max-Planck Institute for Meteorology, Hamburg, Germany; URL: `http://www.mpimet.mpg.de/`). The region is 60W - 40E, 50N - 75N.

References

Reference to methodology: R.E. Benestad (2001), "A comparison between two empirical downscaling strategies", *Int. J. Climatology*, **vol 210**, pp.1645-1668. [DOI 10.1002/joc.703].

Examples

```
library(clim.pact)
data(eof.c)
```

`eof.dc` *Daily common EOF.*

Description

Common EOFs for daily December-February 2-meter temperature (T(2m)).

Usage

```
data(eof.dc)
```

Format

EOF	EOF patterns.
W	Eigen values.
PC	Principal components of common PCA.
n.fld	Number of different predictors (see mixFields).
tot.var	Sum of all W squared.
id.t	Time labels for the fields (see catFields) - used in DS.
id.x	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lon	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lat	Spatial labels for the fields (see mixFields) - used in plotEOF.
region	Describes the region analysed.
tim	Time information (usually redundant).
lon	Longitudes associated with EOF patterns.
lat	Latitudes associated with EOF patterns.
var.eof	Fractional variances associated with EOF patterns.
yy	years.
mm	months.
dd	days.
v.name	Name of element.
c.mon	Month-season information.
f.name	File name of original data.

Source

The common EOF was produced using EOF, with the combined December-February (DJF) 2-meter air temperature data field from European Centre for Medium-Range Weather Forecasts (ECMWF; UK) reanalysis (see URL: http://www.ecmwf.int/) and HIRHAM dynamically dowwnscaled scenarios from the ECHAM4-GSDIO scenario (Max-Planck Institute for Meteorology, Hamburg, Germany; URL: http://www.mpimet.mpg.de/). The region is 5E - 25E, 58N - 65N.

References

Reference to methodology: R.E. Benestad (2001), ”A comparison between two empirical downscaling strategies”, *Int. J. Climatology*, **vol 210**, pp.1645-1668. [DOI 10.1002/joc.703].

Examples

```
#The EOFs were produced using the following code:
library(clim.pact)

x.1.dm<-retrieve.nc("/data1/era15/ERA-15_t2m.nc",x.rng=c(5,25),y.rng=c(58,65))
X.1.dm<-retrieve.nc("/data1/hirham/T2M_198001-199912.nc",x.rng=c(5,25),
                    y.rng=c(58,65))
Y.1.dm<-retrieve.nc("/data1/hirham/T2M_203001-204912.nc",x.rng=c(5,25),
                    y.rng=c(58,65))
Y.1.dm$yy <- Y.1.dm$yy + 50
# It is important that demean=FALSE when concatinating the two time slices
# from the model simulations, if a study of climate change is the objective.
xX.1.dm <- catFields(X.1.dm,Y.1.dm,demean=FALSE)
xX.1.dm <- catFields(x.1.dm,xX.1.dm)
eof.dc <- eof(xX.1.dm,mon=1)

# To read the data:
data(eof.dc)
```

---

eof.dmc *Daily common EOF.*

Description

Common EOFs for daily December-February 2-meter temperature (T(2m)) and sea level pressure (SLP).

Usage

```
data(eof.dmc)
```

Format

EOF	EOF patterns.
W	Eigen values.
PC	Principal components of common PCA.
n.fld	Number of different predictors (see mixFields).
tot.var	Sum of all W squared.
id.t	Time labels for the fields (see catFields) - used in DS.
id.x	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lon	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lat	Spatial labels for the fields (see mixFields) - used in plotEOF.
region	Describes the region analysed.
tim	Time information (usually redundant).
lon	Longitudes associated with EOF patterns.
lat	Latitudes associated with EOF patterns.
var.eof	Fractional variances associated with EOF patterns.
yy	years.
mm	months.
dd	days.
v.name	Name of element.
c.mon	Month-season information.
f.name	File name of original data.

Source

The common EOF was produced using `EOF`, with the combined December-February (DJF) 2-meter air temperature data field from European Centre for Medium-Range Weather Forecasts (ECMWF; UK) reanalysis (see URL: `http://www.ecmwf.int/`) and HIRHAM dynamically dowwnscaled scenarios from the ECHAM4-GSDIO scenario (Max-Planck Institute for Meteorology, Hamburg, Germany; URL: `http://www.mpimet.mpg.de/`). The region is 5E - 25E, 58N - 65N.

References

Reference to methodology: R.E. Benestad (2001), "A comparison between two empirical downscaling strategies", *Int. J. Climatology*, **vol 210**, pp.1645-1668. [DOI 10.1002/joc.703].

Examples

```
library(clim.pact)

x.1.dm<-retrieve.nc("/data1/era15/ERA-15_t2m.nc",x.rng=c(5,25),y.rng=c(58,65))
X.1.dm<-retrieve.nc("/data1/hirham/T2M_198001-199912.nc",x.rng=c(5,25),
                    y.rng=c(58,65))
Y.1.dm<-retrieve.nc("/data1/hirham/T2M_203001-204912.nc",x.rng=c(5,25),
                    y.rng=c(58,65))
Y.1.dm$yy <- Y.1.dm$yy + 50
# It is important that demean=FALSE when concatinating the two time slices
```

```
# from the model simulations, if a study of climate change is the objective.
xX.1.dm <- catFields(X.1.dm,Y.1.dm,demean=FALSE)
xX.1.dm <- catFields(x.1.dm,xX.1.dm)
x.2.dm<-retrieve.nc("/data1/era15/ERA-15_slp.nc",x.rng=c(5,25),y.rng=c(58,65))
X.2.dm<-retrieve.nc("/data1/hirham/PSL_198001-199912.nc",x.rng=c(5,25),
                    y.rng=c(58,65))
Y.2.dm<-retrieve.nc("/data1/hirham/PSL_203001-204912.nc",x.rng=c(5,25),
                    y.rng=c(58,65))
Y.2.dm$yy <- Y.2.dm$yy + 50
# It is important that demean=FALSE when concatinating the two time slices
# from the model simulations, if a study of climate change is the objective.
xX.2.dm <- catFields(X.2.dm,Y.2.dm,demean=FALSE)
xX.2.dm <- catFields(x.2.dm,xX.2.dm)

xX.dm <- mix.fields(xX.1.dm,xX.2.dm,mon=1)
eof.dmc <- eof(xX.dm,mon=1)

# To read the data:
data(eof.dmc)
```

`eof.mc` *Monthly mixed-common EOF.*

Description

Monthly mixed-common EOFs for January mean 2-meter temperature (T(2m)) and sea-level-Pressure (SLP).

Usage

```
data(eof.mc)
```

Format

EOF	EOF patterns.
W	Eigen values.
PC	Principal components of common PCA.
n.fld	Number of different predictors (see mixFields).
tot.var	Sum of all W squared.
id.t	Time labels for the fields (see catFields) - used in DS.
id.x	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lon	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lat	Spatial labels for the fields (see mixFields) - used in plotEOF.
region	Describes the region analysed.
tim	Time information (usually redundant).
lon	Longitudes associated with EOF patterns.
lat	Latitudes associated with EOF patterns.
var.eof	Fractional variances associated with EOF patterns.
yy	years.

mm	months.
dd	days.
v.name	Name of element.
c.mon	Month-season information.
f.name	File name of original data.

Source

The common EOF was produced using EOF, with the combined January 2-meter air temperature and sea level pressure data field from National Center for Environmental Prediction (NCEP; USA) reanalysis (Kalnay et al., (1996) "The NCEP/NCAR 40-Year Reanalysis Project", *Bul. Am. Met. Soc.*, **vol 77**, no 3, 437-471; e.g. see URL: http://www.cdc.noaa.gov/index.html) and the ECHAM4-GSDIO scenario (Max-Planck Institute for Meteorology, Hamburg, Germany; URL: http://www.mpimet.mpg.de/). The region is 60W - 40E, 50N - 75N.

References

Reference to methodology: R.E. Benestad (2001), "A comparison between two empirical downscaling strategies", *Int. J. Climatology*, **vol 210**, pp.1645-1668. [DOI 10.1002/joc.703].

Examples

```
library(clim.pact)
data(eof.mc)
```

`eof.slp` *EOF of NCEP reanalysis SLP.*

Description

EOF of NCEP reanalysis January mean sea level pressure

Usage

```
data(eof.slp)
```

Format

EOF	EOF patterns.
W	Eigen values.
PC	Principal components of common PCA.
n.fld	Number of different predictors (see mixFields).
tot.var	Sum of all W squared.
id.t	Time labels for the fields (see catFields) - used in DS.

id.x	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lon	Spatial labels for the fields (see mixFields) - used in plotEOF.
id.lat	Spatial labels for the fields (see mixFields) - used in plotEOF.
region	Describes the region analysed.
tim	Time information (usually redundant).
lon	Longitudes associated with EOF patterns.
lat	Latitudes associated with EOF patterns.
var.eof	Fractional variances associated with EOF patterns.
yy	years.
mm	months.
dd	days.
v.name	Name of element.
c.mon	Month-season information.
f.name	File name of original data.

Source

The common EOF was produced using `EOF`, with the combined January 2-meter air temperature data field from National Center for Environmental Prediction (NCEP; USA) reanalysis (Kalnay et al., (1996) "The NCEP/NCAR 40-Year Reanalysis Project", *Bul. Am. Met. Soc.*, **vol 77**, no 3, 437-471; e.g. see URL: `http://www.cdc.noaa.gov/index.html`).

References

Reference to methodology: R.E. Benestad (2001), "A comparison between two empirical downscaling strategies", *Int. J. Climatology*, **vol 210**, pp.1645-1668. [DOI 10.1002/joc.703].

Examples

```
library(clim.pact)
data(eof.slp)
```

`getdnmi` *Retrieve station record from DNMI database filed.*

Description

Retrieve station record from DNMI (The Norwegian Meteorological Institute, met.no) database filed. URL `http://www.met.no`. Also see `getnacd` and `getnordklim`.

Usage

```
getdnmi(location="prompt",ele.c='101',direc)
```

Arguments

`location`	name of climate station location.
`ele.c`	name of element [e.g. avail.elem(), or 't2m', 'rr', 'slp'].
`direc`	name of directory in which the data are stored.

Value

a list of "monthly.station.record" class:

val	The monthly values (a 12-column matrix with one column for each year).
station	station number.
yy	The years of observation (vector).
lat,lon	Latitude and longitude of the location.
x.0E65N,y.0E65N	Distance in km from 0E, 65N.
location	Name of location .
wmo.no	WMO number.
start	Start of observatins from this location.
yy0	First year in current record.
ele	Code of theelement.
obs.name	Name of the element.
unit	Unit of the element.
country	The country in which the location is located.
quality	Code/description for data quality.
found	Flag: T - the data requested was found.
ref	Reference for the data set.

Author(s)

R.E. Benestad

Examples

```
oslo.t2m.dnmi <- getdnmi("oslo")
ferder.t2m.dnmi <- getdnmi("ferder")
```

`getnacd` *Retreave station record from the NACD set.*

Description

Retrieve station record from the North Atlantic Climate Data (NACD) set: URL http://www.dmi.dk/. Also see getdnmi and getnordklim.

Usage

```
getnacd(location="prompt",ele.c='101',ascii=FALSE,silent=FALSE,direc="data")
```

Arguments

`location`	name of climate station location.
`ele.c`	name of element [e.g. avail.elem(), or 't2m', 'rr', 'slp'].
`ascii`	Flag. T -> force ascii read, otherwise look for R-formatted verion (faster).
`silent`	Flag. F -> print error messages.
`direc`	name of directory in which the data are stored.

Value

a <- list of "monthly.station.record" class:

val	The monthly values (a 12-column matrix with one column for each year).
station	station number.
yy	The years of observation (vector).
lat,lon	Latitude and longitude of the location.
x.0E65N,y.0E65N	Distance in km from 0E, 65N.
location	Name of location .
wmo.no	WMO number.
start	Start of observatins from this location.
yy0	First year in current record.
ele	Code of theelement.
obs.name	Name of the element.
unit	Unit of the element.
country	The country in which the location is located.
quality	Code/description for data quality.
found	Flag: T - the data requested was found.
ref	Reference for the data set.

Author(s)

R.E. Benestad

Examples

```
helsinki.rr <- getnacd("helsinki",ele=601)
obs.t2m <- getnacd()
```

`getnordklim` *Retrieve station record from the Nordklima set.*

Description

Reads the data from Nordklim available at URL: `http://www.smhi.se/hfa_coord/nordklim/`. Also see `getdnmi` and `getnacd`.

Usage

```
getnordklim(location="prompt",ele.c='101',ascii=FALSE,silent=FALSE,direc="data")
```

Arguments

`location`	name of climate station location.
`ele.c`	name of element [e.g. avail.elem(), or 't2m', 'rr', 'slp'].
`ascii`	Flag. T -> force ascii read, otherwise look for R-formatted verion (faster).
`silent`	Flag. F -> print error messages.
`direc`	name of directory in which the data are stored.

Value

a <- list of "monthly.station.record" class:

val	The monthly values (a 12-column matrix with one column for each year).
station	station number.
yy	The years of observation (vector).
lat,lon	Latitude and longitude of the location.
x.0E65N,y.0E65N	Distance in km from 0E, 65N.
location	Name of location .
wmo.no	WMO number.
start	Start of observatins from this location.
yy0	First year in current record.
ele	Code of theelement.
obs.name	Name of the element.
unit	Unit of the element.
country	The country in which the location is located.
quality	Code/description for data quality.
found	Flag: T - the data requested was found.
ref	Reference for the data set.

Author(s)

R.E. Benestad

Examples

```
helsinki.rr <- getnordklim("helsinki",ele=601)
```

`grd.box.ts` *Grid box time series*

Description

Plots time series from field objects, eg from `retrieve.nc`.

Usage

```
grd.box.ts(x,lon,lat,what="abs",greenwich=TRUE,mon=NULL,
                         col="grey10",lwd=1,lty=1,pch=26,add=FALSE)
```

Arguments

`x`	A field object.
`lon`	Longitude to plot.
`lat`	Latitude to plot.
`what`	What to draw: "ano"-> anomalies, "cli"-> climatological values, "abs" -> absolute values.
`greenwich`	Maps centre on the Greenwich meridian.
`mon`	Month to extract
`col`	Colour.
`lwd`	Line width
`lty`	Line style.
`pch`	Plot character.
`add`	'TRUE' adds curve to old plot.

Value

avail.preds	vector of characters
avail.locs	a list with name, lons, lats, country, ident
avail.elem	a list with data.set, ele, name

Author(s)

R.E. Benestad

Examples

```
slp <- retrieve.nc("ncep_slp.nc")
grd.box.ts(slp,0,60,what="ano",mon=1)
```

`helsinki.t2m` *Monthly mean temperature in Helsinki.*

Description

A station record of monthly mean temperature from Helsinki.

Usage

```
data(helsinki.t2m)
```

Format

a <- list of "monthly.station.record" class:

val	The monthly values (a 12-column matrix with one column for each year).
station	station number.
yy	The years of observation (vector).
lat,lon	Latitude and longitude of the location.
x.0E65N,y.0E65N	Distance in km from 0E, 65N.
location	Name of location .
wmo.no	WMO number.
start	Start of observatins from this location.
yy0	First year in current record.
ele	Code of theelement.
obs.name	Name of the element.
unit	Unit of the element.
country	The country in which the location is located.
quality	Code/description for data quality.
found	Flag: T - the data requested was found.
ref	Reference for the data set.

Source

The Nordklim data set: http://www.smhi.se/hfa_coord/nordklim/

References

Tuomenvirta et al. (2001), "Nordklim data set 1.0", DNMI KLIMA 08/01, pp. 26;
The Norwegian Meteorological Institute, P.O. Box 43, 0313 Oslo, Norway (http://www.met.no).

Examples

`instring` *instring*

Description

Finds the position of a character in a string (character vector). Similar to regexpr(), but a test with regexpr() failed with some characters. instring() returns all position with a character match, whereas regexpr() only returns the first position for a pattern match.

Usage

```
instring(c,target,case.match=TRUE)
```

Arguments

`c`	Character to look for.
`target`	string to search.
`case.match`	FALSE -> not case sensitive.

Value

vector of intigers.

Author(s)

R.E. Benestad

Examples

```
instring("e","efile.dat")
# 1 5
regexpr("e","efile.dat")
#[1] 1
#attr(,"match.length")
#[1] 1
# Case when regexpr() doesn't give the desired result:
regexpr(".","file.name")
#[1] 1
#attr(,"match.length")
#[1] 1
instring(".","file.name")
#[1] 5
```

`km2lat` *Convert long-lat to km-km*

Description

The function computes the latitude from given the distance from a reference point. See also `km2lon` and `COn0E65N`.

Usage

```
km2lat(x, y, x.centre=0, y.centre=65)
```

Arguments

`x`	not used.
`y`	distance from reference latitude in meridional direction.
`x.centre`	reference longitude.
`y.centre`	reference latitude.

Value

real

Author(s)

R.E. Benestad

Examples

```
library(clim.pact)
data(oslo.t2m)
print(c(oslo.t2m$lon,oslo.t2m$lat))
#[1] 10.71667 59.95000
xy<-COn0E65N(oslo.t2m$lon,oslo.t2m$lat)
oslo.t2m$lon<-xy$x
oslo.t2m$lat<-xy$y
print(c(oslo.t2m$lon,oslo.t2m$lat))
#[1]  595.4086 -560.3004
lon<-km2lon(oslo.t2m$lon,oslo.t2m$lat,x.centre=0,y.centre=65)
lat<-km2lat(oslo.t2m$lon,oslo.t2m$lat,x.centre=0,y.centre=65)
print(c(lon,lat))
#[1] 10.71667 59.95000
```

`km2lon` *Convert long-lat to km-km*

Description

The function computes the longitude from given the distance from a reference point. See also `km2lat` and `COn0E65N`.

Usage

```
km2lon(x, y, x.centre=0, y.centre=65)
```

Arguments

`x`	not used.
`y`	distance from reference latitude in meridional direction.
`x.centre`	reference longitude.
`y.centre`	reference latitude.

Value

real

Author(s)

R.E. Benestad

Examples

```
library(clim.pact)
data(oslo.t2m)
print(c(oslo.t2m$lon,oslo.t2m$lat))
#[1] 10.71667 59.95000
xy<-COn0E65N(oslo.t2m$lon,oslo.t2m$lat)
oslo.t2m$lon<-xy$x
oslo.t2m$lat<-xy$y
print(c(oslo.t2m$lon,oslo.t2m$lat))
#[1]  595.4086 -560.3004
lon<-km2lon(oslo.t2m$lon,oslo.t2m$lat,x.centre=0,y.centre=65)
lat<-km2lat(oslo.t2m$lon,oslo.t2m$lat,x.centre=0,y.centre=65)
print(c(lon,lat))
#[1] 10.71667 59.95000
```

`koebenhavn.t2m` *Monthly mean temperature in Copenhagen.*

Description

A station record of monthly mean temperature Copenhagen.

Usage

```
data(koebenhavn.t2m)
```

Format

list of "monthly.station.record" class:

val	The monthly values (a 12-column matrix with one column for each year).
station	station number.
yy	The years of observation (vector).
lat,lon	Latitude and longitude of the location.
x.0E65N,y.0E65N	Distance in km from 0E, 65N.
location	Name of location .
wmo.no	WMO number.
start	Start of observatins from this location.
yy0	First year in current record.
ele	Code of theelement.
obs.name	Name of the element.
unit	Unit of the element.
country	The country in which the location is located.
quality	Code/description for data quality.
found	Flag: T - the data requested was found.
ref	Reference for the data set.

Source

The Nordklim data set: http://www.smhi.se/hfa_coord/nordklim/

References

Tuomenvirta et al. (2001), "Nordklim data set 1.0", DNMI KLIMA 08/01, pp. 26; The Norwegian Meteorological Institute, P.O. Box 43, 0313 Oslo, Norway (http://www.met.no).

Examples

`lower.case` *convert to lower case*

Description

Converts characters to lower case.

Usage

```
lower.case(u.case)
```

Arguments

`u.case` Strings or arrays of strings

.

Value

converted strings or arrays of strings.

Author(s)

R.E. Benestad

Examples

```
print(upper.case(c("qwerty  e","asdf  rT")))      #  "QWERTY" "ASDF"
print(lower.case(c("QWERTY","ASDF")))             #  "qwErty" "asdf"
print(strip(c("Hello there!","Oslo"," ","NA ")))   # "Hello" "Oslo"  " "     "NA"
```

`map` *Produce a map*

Description

Produces maps.

Usage

```
map(x,y=NULL,col="black",lwd=1,lty=1,sym=TRUE,
               plot=TRUE,inv.col=FALSE)
```

Arguments

`x`	A map object.
`y`	A map object. If given, `map.map` plots the difference: x - y
`col`	Colour of contours.
`lwd`	Contour line width.
`lty`	Contour line style.
`sym`	Symmetry: if True, use zlimits c(-|max|,+|max|).
`plot`	TRUE gives graphics
`inv.col`	Inverse color scheme (e.g. 'TRUE' gives red for drier and blue for wetter conditions)

.

Value

A map object

Author(s)

R.E. Benestad

Examples

`map.eof` *Map eof*

Description

Draws maps of the spatial structures described by the EOF (`EOF` patterns. Is similar to `plotEOF`, but only plots the spatial information. Useful for comparing the spatial patterns in different EOFs.

Usage

```
map.eof(x,i.eof=1,nlevs=5,add=FALSE,
             col=c("red","blue","darkgreen","steelblue"),lwd=2,lty=1)
```

Arguments

`x`	A field object.
`i.eof`	The EOF to plot.
`nlevs`	Number of contour levels.
`add`	Add a map on pre-existing map - see contour.
`col`	Colour.
`lwd`	Line width.
`lty`	Line type.

Value

A map object (currently, only the last field in a mixed.field object).

Author(s)

R.E. Benestad

Examples

```
library(clim.pact)
data(eof.slp)
map.eof(eof.slp)
map.eof(eof.slp,i.eof=2,col="blue",add=TRUE)
```

`mapField` *MapField*

Description

Draws maps of fields in a field object, eg read using `retrieve.nc`.

Usage

```
mapField(x,l=NULL,greenwich=TRUE,what="ano",method="nice",
                        col="black",lwd=2,lty=1,add=FALSE,las = 1)
```

Arguments

`x`	A field object.
`l`	The field to map. Default: the last field in the record.
`what`	What to draw: "ano"-> anomalies, "cli"-> climatological values, "abs" -> absolute values.
`method`	"nice" -> `filled.contour`, otherwise use `image`.
`greenwich`	'TRUE' centres on Greenwich meridian.
`col`	Colour.
`lwd`	Line width.
`lty`	Line type.
`add`	Adds map to old figure.
`las`	See par

Value

Author(s)

R.E. Benestad

Examples

```
library(clim.pact)

skt<-retrieve.nc("skt.mon.mean.nc",
                 x.rng=c(-90,50),y.rng=c(0,75))
bitmap("ncep.skt.jpg",type="jpeg")
mapField(skt)
dev.off()
```

`meanField` *Optimal interpolation*

Description

.

Usage

```
meanField(x,lon.rng=NULL,lat.rng=NULL,t.rng=NULL)
```

Arguments

`x`	A field object.
`lon.rng`	Extract the longitude interval.
`lat.rng`	Extract the latitude interval.
`t.rng`	Extract the time interval.

Value

A map object

Author(s)

R.E. Benestad

Examples

```
slp <- retrieve.nc("ncep_slp.nc",x.rng=c(5,12),y.rng=c(58,63))
mslp <- meanField(slp)
```

mergeStation — *Merge climate station series.*

Description

Merges two series from different sources, eg from NACD (getnacd) and DNMI (getdnmi). The code is useful for updating long climate series with new observations from a different database. The routine cokpares data for overlapping times and prints out diagnostics about the two data sets.

Usage

```
mergeStation(x.1,x.2,plot=FALSE)
```

Arguments

x.1	1st series.
x.2	2nd series.
plot	'TRUE' plots the overlap.

Value

A climate station series object.

Author(s)

R.E. Benestad

Examples

```
oslo.1 <- getnacd("OSLO-BLINDERN")
oslo.2 <- getdnmi("oslo")
print(range(oslo.1$yy))
#[1] 1890 1990
print(range(oslo.2$yy))
#[1] 1937 2002
oslo <- mergeStation(oslo.1,oslo.2)
#[1] "Time intervals:"
#[1] 1890 1990
#[1] 1937 2002
#[1] 1937.042 1990.958
#[1] "RMSE:  0.04"
#
#Call:
#lm(formula = y ~ 1 + x, data = ovrlp)
#
#Residuals:
#     Min       1Q   Median       3Q      Max
```

```
#-7.24005 -0.03271  0.01161  0.06006  7.61593
#Coefficients:
#            Estimate Std. Error t value Pr(>|t|)
#(Intercept) 0.029044   0.047482   0.612    0.541
#x           0.993886   0.004866 204.231   <2e-16 ***
#---
#Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
#Residual standard error: 0.9738 on 644 degrees of freedom
#Multiple R-Squared: 0.9848,    Adjusted R-squared: 0.9848
#F-statistic: 4.171e+04 on 1 and 644 DF,  p-value: < 2.2e-16

print(range(oslo$yy))
#[1] 1890 2002
```

`mixFields` *mixFields*

Description

Mix fields by combining two different gridded sets of observation. Observations/data for representing values at n different locations at a given time (t) can be described in terms of a vector

$$\vec{x}(t) = [x_1, x_2, ...x_n].$$

Two different sets of observations can be represented by two vectors y and z of lengths n and m respectively. In mix.fields, the information in these two data sets are combined combining the two vectors:

$$\vec{x}(t) = [\vec{y}(t), \vec{z}(t)] = [y_1, y_2, ...y_n, z_1, z_2, ...z_m].$$

The length of the final vector of the mixed field is the sum of the lengths of the two respective vectors. The two data sets do not have to be on the same grid.

reference: Bretherton et al. (1992) "An Intercomparison of Methods for finding Coupled Patterns in Climate Data", *J. Climate*, **vol 5**, 541-560.

The output from `mixFields` can be used in `EOF` to compute mixed-common EOFs which subsecuently can be used as predictors in `DS` in order to downscale climate scenarios (Benestad et al. (2002), "Empirically downscaled temperature scenarios for Svalbard", *Atm. Sci. Lett.*, doi.10.1006/asle.2002.0051).

Usage

```
mixFields(field.1,field.2,mon=NULL,interval=NULL)
```

Arguments

`field.1`	A 'field.object'.
`field.2`	A 'field.object'.
`mon`	Calendar month to extract.
`interval`	Time interval to extract.

Value

A 'field.object'.

Author(s)

R.E. Benestad

Examples

```
library(clim.pact)
x.1 <- retrieve.nc("/home/kareb/data/ncep/ncep_t2m.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
x.2 <- retrieve.nc("/home/kareb/data/ncep/ncep_slp.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
print(x.1$v.name)

print("Read GCM predictor data.")
X.1 <- retrieve.nc("data/mpi-gsdio_t2m.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
X.2 <- retrieve.nc("data/mpi-gsdio_slp.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
print(X.1$v.name)
print("Cat fields.")
xX.1 <- catFields(x.1,X.1,interval.1=c(1958,1998),interval.2=c(1958,2050))
xX.2 <- catFields(x.2,X.2,interval.1=c(1958,1998),interval.2=c(1958,2050))
xX <- mixFields(xX.1,xX.2,mon=1,
                 interval=c(1900,2050))
print("EOF")
eof.c <- eof(xX.1,mon=1)
eof.mc <- eof(xX,mon=1)
```

---

mod *Modulus of a division.*

Description

Returns the modulus of a division: returns the remainder of the expression x/y.

Usage

```
mod(x,y)
```

Arguments

x	nominator (integer).
y	denominator (integer).

Value

integer value from 0 .. (y-1)

Author(s)

R.E. Benestad

Examples

```
mod(101,10)   # 1
mod(4,12)     # 4
mod(123,12)   # 3
```

---

num2str *Convert numbers to string and format*

Description

Convert numbers to string and format. Similar to FORTRAN 'F8.2' format statement.

Usage

```
num2str(x,dec=2,f.width=NULL,d.point=".")
```

Arguments

`x`	Real numbers.
`dec`	number of decmal points.
`f.width`	width of field.
`d.point`	character marking the decimal point.

Value

string.

Author(s)

R.E. Benestad

Examples

```
print(num2str(c(1,23.4282,-3.14),dec=3))
#[1] "1.000"  "23.428" "-3.140"
```

`optint` *Optimum interpolation*

Description

Solves for w in:

$$\sum_j M_{ij} w_i k = \langle pi_j pi_k \rangle$$

Then computes a correction:

$$r = w_i q$$

where q = gridded observations - first guess.

Reference: Reynolds and Smith (1994), Improved global sea surface temperature analysis using optimum interpolation, *J. Climate*, **vol 7**, 929-948.

NB: The testing of this routine is not yet finished - there may still be some bugs in this function.

Usage

```
optint(lon,lat,obs,lon.grd,lat.grd,fguess,eps,
              lambda=50,M=NULL,piipij=NULL,w=NULL,tim=NULL,date=NULL)
```

Arguments

`lon`	A vector of longitude coordinates.
`lat`	A vector of latitude coordinates.
`obs`	A vector of observations.
`lon.grd`	A vector of longitude coordinates for gridded values.
`lat.grd`	A vector of latitude coordinates for gridded values.
`fguess`	First guess values on given grid (`lon.grd,lat.grd`).
`eps`	Error estimates on given grid (`lon.grd,lat.grd`).
`lambda`	Correlation length.
`M`	Interpolation model matrix.
`piipij`	Interpolation weights.
`w`	Interpolation weights..
`tim`	Time index.
`date`	Date stamp.

Value

A map object with additional elements: `M, piipij, w`

Author(s)

R.E. Benestad

Examples

```
# This example takes a little while to run..

lon <- c(12.02, 10.47,  9.28, 10.78, 10.71,  9.12,  8.20,  9.52,  8.18,  8.52,
          8.07,  6.15,  5.63,  6.37,  5.92,  5.33,  6.50,  7.42,  6.17, 10.00,
         10.02, 11.12, 10.93, 14.02, 13.98, 14.43, 15.42, 17.80, 18.53, 18.93,
         21.02, 21.88, 23.35, 23.52, 31.08, 15.47, 11.93)
lat <- c(60.61, 61.10, 62.12, 59.38, 59.95, 60.58, 60.52, 59.90, 59.83, 59.03,
         58.20, 58.88, 58.88, 59.65, 60.40, 60.38, 60.65, 62.23, 62.57,
         63.20, 62.82, 63.20, 63.47, 65.52, 66.82, 67.27, 68.70, 68.73,
         69.06, 69.65, 69.73, 69.83, 69.98, 69.33, 70.37, 78.25, 78.92)
val <- c(1.74, 1.43, 0.98, 1.31, 1.10, 1.39, 1.07, 1.33,-0.04, 1.07,
         1.07, 0.22, 0.89, 0.98, 0.77, 0.81, 1.43, 0.30, 0.69,-0.17,
         1.17, 0.32, 1.55, 1.65, 0.75, 0.78, 0.56, 0.04, 0.57, 0.52,
         0.51, 0.51, 0.96, 0.32, 0.35,-0.36, 0.00)
library(akima)
lon.grd <- seq(min(lon),max(lon),length=20)
lat.grd <- seq(min(lat),max(lat),length=20)
fguess <- interp(lon,lat,val,lon.grd,lat.grd)$z
fguess[is.na(fguess)]<-mean(fguess,na.rm=TRUE)
eps <- abs(fguess)*0 + 0.1
opt.int <- optint(lon,lat,val,lon.grd,lat.grd,fguess,eps)
image(opt.int$lon,opt.int$lat,opt.int$map)
contour(opt.int$lon,opt.int$lat,opt.int$map,lwd=1,add=TRUE)
contour(lon.grd,lat.grd,fguess,lty=2,lwd=2,col="darkblue",add=TRUE)
addland()
```

`oslo.dm` *Daily Oslo record.*

Description

A station record of daily mean temperature and daily precipitation from Oslo-Blindern.

Usage

```
data(oslo.dm)
```

Format

a ”daily.station.record”-class object.

t2m	a vector holding daily mean temperature.
precip	a vector holding daily precipitation.

dd	a vector holding day of month.
mm	a vector holding the month.
yy	a vector holding the year.
obs.name	the name of observation: eg c("Daily mean temperature","Daily precipitation").
unit	the unite of observation: eg c("deg C","mm/day").
ele	element code: eg c("tam","rr").
station	local (national) station number.
lat	latitude.
lon	longitude.
alt	altitude.
location	name of location.
wmo.no	WMO number of station.
start	start of measurements.
yy0	first year of record.
country	name of country.
ref	reference to the data.

Source

The Norwegian Meteorological Institute, Climatology deivision.

References

The Norwegian Meteorological Institute, P.O. Box 43, 0313 Oslo, Norway (http://www.met.no).

Examples

`oslo.t2m` *Monthly mean temperature in Oslo.*

Description

A station record of monthly mean temperature from Oslo-Blindern.

Usage

```
data(oslo.t2m)
```

Format

list of "monthly.station.record" class:

val	The monthly values (a 12-column matrix with one column for each year).
station	station number.
yy	The years of observation (vector).

lat,lon	Latitude and longitude of the location.
x.0E65N,y.0E65N	Distance in km from 0E, 65N.
location	Name of location .
wmo.no	WMO number.
start	Start of observatins from this location.
yy0	First year in current record.
ele	Code of theelement.
obs.name	Name of the element.
unit	Unit of the element.
country	The country in which the location is located.
quality	Code/description for data quality.
found	Flag: T - the data requested was found.
ref	Reference for the data set.

Source

The Nordklim data set: `http://www.smhi.se/hfa_coord/nordklim/`

References

Tuomenvirta et al. (2001), "Nordklim data set 1.0", DNMI KLIMA 08/01, pp. 26;
The Norwegian Meteorological Institute, P.O. Box 43, 0313 Oslo, Norway (http://www.met.no).

Examples

`plotStation` *Plots monthly station records.*

Description

Plots data in the monthly station records. The data may be read through getnacd, getnordklim, getdnmi, or created using station.obj. The commands avail.elem, avail.locs can be used to identify the station records available (in a given subdirectory).

Usage

```
plotStation(obs,l.anom=TRUE,mon=NULL,
                          leps=FALSE,out.dir="output")
```

Arguments

`obs`	A climate station series.
`l.anom`	flag: T -> plot anomalies.
`mon`	select month to plot, A value of 0 plots all months. mon=c(12,1,2) plots the DJF mean.
`leps`	Flag: T -> produce EPS files (hard copy).
`out.dir`	Directory where to store hard copies.

Value

a <- list of "monthly.station.record" class:

val	The monthly values (a 12-column matrix with one column for each year).
station	station number.
yy	The years of observation (vector).
lat,lon	Latitude and longitude of the location.
x.0E65N,y.0E65N	Distance in km from 0E, 65N.
location	Name of location .
wmo.no	WMO number.
start	Start of observatins from this location.
yy0	First year in current record.
ele	Code of theelement.
obs.name	Name of the element.
unit	Unit of the element.
country	The country in which the location is located.
quality	Code/description for data quality.
found	Flag: T - the data requested was found.
ref	Reference for the data set.

Author(s)

R.E. Benestad

Examples

```
data(tromsoe.t2m)
plotStation(tromsoe.t2m)
```

`plotDS` *Plot downscaled results*

Description

Plots the results from `DS`.

Usage

```
plotDS(ds.obj,leps=FALSE,plot.map=TRUE,plot.res=FALSE,
plot.rate=FALSE,add=FALSE,col="darkred",lwd=2,lty=1,
                    direc="output/")
```

Arguments

`ds.obj`	A `DS` object.
`leps`	'TRUE' produces EPS files.
`plot.map`	'TRUE' shows the spatial predictor pattern.
`plot.res`	'TRUE' shows statistics for residuals.
`plot.rate`	'TRUE' shows analysis of rate-of-change.
`add`	TRUE: adds a scenario to old time series plot.
`col`	Colour of scenario time series.
`lwd`	Line width of scenario time series.
`lty`	Line style of scenario time series.
`direc`	Directory for graphical output.

Value

Author(s)

R.E. Benestad

Examples

```
data(helsinki.t2m)
data(eof.mc)
ds.helsinki<-DS(dat=helsinki.t2m,preds=eof.mc,plot=FALSE)
plotDS(ds.helsinki,leps=TRUE)
```

---

plotEOF *Plot EOFs*

Description

Plots the results from an (mixed-common) EOF analysis (`EOF`).

Usage

```
plotEOF(x,i.eof=1,nlevs=5,
                    col=c("red","blue","darkgreen","steelblue"))
```

Arguments

`x`	An `EOF` object.
`i.eof`	EOF to plot.
`nlevs`	Contour levels.
`col`	Colour.

Value

Author(s)

R.E. Benestad

Examples

```
data(eof.mc)
plotEOF(eof.mc)
x11()
data(eof.dmc)
plotEOF(eof.dmc)
```

`plotField` *plotField*

Description

Produce 2D plots like maps and Hovmuller diagrams. A poor man's version of Ferret's http://ferret.wrc.noaa.gov/Ferret/ plot function. `plot.field` is a high level command that utilises `mapField` or `grd.box.ts` whenever appropriate.

Usage

```
plotField(x,lon=NULL,lat=NULL,tim=NULL,mon=NULL,
                          col="black",lty=1,lwd=1,what="ano")
```

Arguments

`x`	A field object.
`lon`	Position for longitude. One of `lon, lat, tim` must be set.
`lat`	Position for latitude.
`tim`	Position for time.
`mon`	Month to extract.
`col`	Contour line colour.
`lty`	Contour line type.
`lwd`	Contour line width.
`what`	Choose between "ano" - anomaly; "abs" absolute; "cli" climatological.

Value

Author(s)

R.E. Benestad

Examples

---

plumePlot *Plot downscaled time series as plumes*

Description

Retreives an ensemble of time series from DS and plots these as a plume. The routine retrieves the ds objects from disc.

Usage

```
plumePlot(ds.name.list=NULL,location,mon,direc="output",
                         t.rng=c(1850,2074),r2.th=50,p.th=0.05,
                         col="darkred",lwd=2,lty=1)
```

Arguments

`ds.name.list`	A list of file names holding the ds objects.
`location`	Name of location to plot.
`mon`	Month to plot.
`direc`	The subdirectory in which the ds objects are stored.
`t.rng`	Time interval to plot.
`r2.th`	R-squared theshold: only use scenarios that account for equal to or more than `r2.th` of the variance in %.
`p.th`	p-value threshold: only use scenarios that have p-values equal to or less than `p.th`.
`col`	Colour of scenario time series.
`lwd`	Line width of scenario time series.
`lty`	Line style of scenario time series.

Value

Author(s)

R.E. Benestad

Examples

```
ds.list<-avail.ds()
plumePlot(ds.list,location="OSLO-BLINDERN",mon=1)
```

---

retrieve.nc *Retrieve data from a netCDF file*

Description

Reads a netCDF file and picks out vectors that look like lngitude, latitude and time. Returns the first 3-D field in the file.

Usage

```
retrieve.nc(f.name="data/ncep_t2m.nc",v.nam="AUTO",
                         l.scale=TRUE,greenwich=TRUE,
                         x.nam="lon",y.nam="lat",t.nam="tim",
                         x.rng=NULL,y.rng=NULL,t.rng=NULL)
```

Arguments

f.name	name of netCDF file.
v.nam	name of variable. "AUTO" -> smart search.
l.scale	'TRUE' uses scaling.factor and add.offset.
greenwich	'TRUE' centres maps on Greenwhich meridian (0 deg E).
x.nam	Name of x-dimension.
y.nam	Name of y-dimension.
t.nam	Name of time-axis.
x.rng	Region to extract.
y.rng	Region to extract.
t.rng	Time interval to extract.

Value

A "field.object" list:

lon	a vector of longitudes
lat	a vector of latitudes
tim	a vector of times from time.0 (see attributes)
dat	a 3-D matrix with the data

.

Author(s)

R.E. Benestad

Examples

```
X.1 <- retrieve.nc("data/mpi-gsdio_t2m.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
X.2 <- retrieve.nc("data/mpi-gsdio_slp.nc",
                   x.rng=c(-60,40),y.rng=c(50,75))
```

`reverse` *Reverse*

Description

Reverses the order of a vector.`reverse.sort` returns a sorted vector in reverse order.

Usage

```
reverse(x)
reverse.sort(x)
```

Arguments

`x`	a vector

Value

a vector.

Author(s)

R.E. Benestad

Examples

```
reverse(c(1,3,5,7,2,4,6,8))      # 8 6 4 2 7 5 3 1
reverse.sort(c(1,3,5,7,2,4,6,8)) # 8 7 6 5 4 3 2 1
```

`satellite` *Satellite view / polar stereographic*

Description

Produces polar stereographic maps / satellite views.

Usage

```
satellite(map.obj,col="black",lwd=2,lty=1,add=FALSE,
                      las = 1,lon.0=0,lat.0=90,
                      ni=100,nj=100, n.nearest=4,max.dist=3)
```

Arguments

`map.obj`	a map object (`mapField`).
`col`	contour colours.
`lwd`	contour line width.
`lty`	contour line style.
`add`	FLAG: true adds contour onto old plot.
`las`	see `par`.
`lon.0`	Not working apart from default.
`lat.0`	Not working apart from default.
`ni`	Number of grid points along x-axis in new grid.
`nj`	Number of grid points along y-axis in new grid.
`n.nearest`	Number of points to use in re-gridding.
`max.dist`	The maximum inter-point distance used for re-gridding.

Value

Author(s)

R.E. Benestad

Examples

```
x <- retrieve.nc("T2M_p.nc")
a <- map.field(x)
satellite(a)
```

`station.obj` *Make monthly climate station series object.*

Description

Create a station object for use as predictand in empirical downscaling on monthly data. Also see `station.obj.dm`.

Usage

```
station.obj(x,yy,obs.name,unit,ele=NULL, mm=NULL,
                   station=NULL,lat=NULL,lon=NULL,alt=NULL,
                   location="unspecified",wmo.no=NULL,
                   start=NULL,yy0=NULL,country=NULL,ref=NULL)
```

Arguments

`x`	the data: a matrix of 12 columns holding the observations of each calendar month: column 1 holds January values, col 2 holds February, .. col 12 holds December values.
`yy`	A vector holding the year of observation of the same length as each of the 12 columns. Or a vector with the same length as the data if `mm` is given.
`mm`	a vector of months with the same length as the data (optional).
`obs.name`	the name of observation (e.g. "Temperature").
`unit`	the unite of observation (e.g. "deg C").
`ele`	element code (e.g. 101).
`station`	local (national) station number.
`lat`	latitude.
`lon`	longitude.
`alt`	altitude.
`location`	name of location.
`wmo.no`	WMO number of station.
`start`	start of measurements.
`yy0`	first year of record.
`country`	name of country.
`ref`	reference to the data.

Value

a "monthly.station.record"-class object.

val	The monthly values (a 12-column matrix with one column for each year)
station	station number.
yy	The years of observation (vector).
lat,lon	Latitude and longitude of the location.
x.0E65N,y.0E65N	Distance in km from 0E, 65N.
location	Name of location .
wmo.no	WMO number.
start	Start of observatins from this location.
yy0	First year in current record.
ele	Code of theelement.
obs.name	Name of the element.
unit	Unit of the element.
country	The country in which the location is located.
quality	Code/description for data quality.
found	Flag: T-> the data requested was found.
ref	Reference for the data set.

Author(s)

R.E. Benestad

Examples

```
a <- read.table("data/bjornholt.dat",
                  col.names=c("station","year","month","rr",
                    "tam","sam","sdm","uum","pom","tax","tan"))
obs <- station.obj(x=a$rr,yy=a$year,mm=a$month,
                   obs.name="Precipitation",unit="mm",ele=601,
                   lat=60.03,lon=10.41,alt=360,
                   station=a$station[1],location="Bjornholt",
                   country="Norway",ref="met.no Climate data base")
plot(obs,mon=11)
```

---

station.obj.dm *Make daily climate station series object.*

Description

Create a station object for use as predictand in empirical downscaling on monthly data. Also see station.obj.

Usage

```
station.obj.dm(t2m,precip,dd,mm,yy,
                       obs.name=NULL,unit=NULL,ele=NULL,
                       station=NULL,lat=NULL,lon=NULL,alt=NULL,
                       location="unspecified",wmo.no=NULL,
                       start=NULL,yy0=NULL,country=NULL,ref=NULL)
```

Arguments

`t2m`	a vector holding daily mean temperature.
`precip`	a vector holding daily precipitation.
`dd`	a vector holding day of month.
`mm`	a vector holding the month.
`yy`	a vector holding the year.
`obs.name`	the name of observation: eg c("Daily mean temperature","Daily precipitation").
`unit`	the unite of observation: eg c("deg C","mm/day").
`ele`	element code: eg c("tam","rr").
`station`	local (national) station number.
`lat`	latitude.
`lon`	longitude.
`alt`	altitude.
`location`	name of location.
`wmo.no`	WMO number of station.
`start`	start of measurements.
`yy0`	first year of record.
`country`	name of country.
`ref`	reference to the data.

Value

a "daily.station.record"-class object.

t2m	a vector holding daily mean temperature.
precip	a vector holding daily precipitation.
day	a vector holding day of month.
month	a vector holding the month.
year	a vector holding the year.
obs.name	the name of observation: eg c("Daily mean temperature","Daily precipitation").
unit	the unite of observation: eg c("deg C","mm/day").
ele	element code: eg c("tam","rr").
station	local (national) station number.

lat	latitude.
lon	longitude.
alt	altitude.
location	name of location.
wmo.no	WMO number of station.
start	start of measurements.
yy0	first year of record.
country	name of country.
ref	reference to the data.

Author(s)

R.E. Benestad

Examples

```
blindern.raw <-read.table("~/data/stations/blindern_rr_day.dat",header=TRUE)
blindern.raw$rr[blindern.raw$rr < 0] <- NA
yy <- floor(blindern.raw$yyyymmdd/10000)
mm <- floor(blindern.raw$yyyymmdd/100) - 10000*yy
dd <- blindern.raw$yyyymmdd - 100*mm  - 10000*yy
blindern <- station.obj.dm(t2m=rep(NA,length(blindern.raw$rr)),
                           precip=blindern.raw$rr,
                           dd=dd,mm=mm,yy=yy,
                           obs.name=c("T(2m)","recip"),
                           unit=c("deg C","mm/day"),ele=NULL,
                           station=18700,lat=59.95,lon=10.71,alt=94,
                           location="Oslo-Blindern",wmo.no=NULL,
                           start=NULL,yy0=1937,country="Norway",
                           ref="www.met.no")
```

`stationmap` *Plot climate station map.*

Description

Plot a map showing the locations of NACD (`getnacd`) and Nordklim (`getnordklim`) station network (monthly data).

Usage

```
stationmap(ele=101)
```

Arguments

`ele`	The code for element (101 = T(2m), 401=SLP, 601= precip).

Value

Author(s)

R.E. Benestad

Examples

```
stationmap()
```

---

stockholm.t2m *Monthly mean temperature in Oslo.*

Description

A station record of monthly mean temperature Oslo-Blindern.

Usage

```
data(stockholm.t2m)
```

Format

list of "monthly.station.record" class:

val	The monthly values (a 12-column matrix with one column for each year).
station	station number.
yy	The years of observation (vector).
lat,lon	Latitude and longitude of the location.
x.0E65N,y.0E65N	Distance in km from 0E, 65N.
location	Name of location .
wmo.no	WMO number.
start	Start of observatins from this location.
yy0	First year in current record.
ele	Code of theelement.
obs.name	Name of the element.
unit	Unit of the element.
country	The country in which the location is located.
quality	Code/description for data quality.
found	Flag: T - the data requested was found.
ref	Reference for the data set.

Source

The Nordklim data set: http://www.smhi.se/hfa_coord/nordklim/

References

Tuomenvirta et al. (2001), "Nordklim data set 1.0", DNMI KLIMA 08/01, pp. 26;
The Norwegian Meteorological Institute, P.O. Box 43, 0313 Oslo, Norway (http://www.met.no).

Examples

`strip` *String operation functions*

Description

The function strips off trailing space (strips the strings by cutting off at the first space).

Usage

```
strip(string.array
```

Arguments

`string.array` Strings or arrays of strings.

Value

converted strings or arrays of strings.

Author(s)

R.E. Benestad

Examples

```
print(upper.case(c("qwerty  e","asdf  rT")))       #  "QWERTY" "ASDF"
print(lower.case(c("QWERTY","ASDF")))              #  "qwErty" "asdf"
print(strip(c("Hello there!","Oslo"," ","NA ")))   # "Hello" "Oslo"  " "     "NA"
```

`tromsoe.t2m` *Monthly mean temperature in Tromsoe.*

Description

A station record of monthly mean temperature Tromsø.

Usage

```
data(tromsoe.t2m)
```

Format

a <- list of "monthly.station.record" class:

val	The monthly values (a 12-column matrix with one column for each year).
station	station number.
yy	The years of observation (vector).
lat,lon	Latitude and longitude of the location.
x.0E65N,y.0E65N	Distance in km from 0E, 65N.
location	Name of location .
wmo.no	WMO number.
start	Start of observatins from this location.
yy0	First year in current record.
ele	Code of theelement.
obs.name	Name of the element.
unit	Unit of the element.
country	The country in which the location is located.
quality	Code/description for data quality.
found	Flag: T - the data requested was found.
ref	Reference for the data set.

Source

The Nordklim data set: http://www.smhi.se/hfa_coord/nordklim/

References

Tuomenvirta et al. (2001), "Nordklim data set 1.0", DNMI KLIMA 08/01, pp. 26;
The Norwegian Meteorological Institute, P.O. Box 43, 0313 Oslo, Norway (http://www.met.no).

Examples

`upper.case` *convert to UPPER CASE*

Description

Converts characters to UPPER CASE.

Usage

```
upper.case(u.case)
```

Arguments

`u.case`	Strings or arrays of strings.

Value

converted strings or arrays of strings.

Author(s)

R.E. Benestad

Examples

```
print(upper.case(c("qwerty  e","asdf  rT")))      #  "QWERTY" "ASDF"
print(lower.case(c("QWERTY","ASDF")))             #  "qwErty" "asdf"
print(strip(c("Hello there!","Oslo"," ","NA ")))  # "Hello" "Oslo"  " "     "NA"
```

`what.data` *Data information*

Description

The data originally used by the clim.pact package: NACD, Nordklim and NCEP reanalysis and links to their sources.

Usage

```
what.data()
```

Arguments

Author(s)

R.E. Benestad

Examples

```
what.data()
```

Index