Package ‘mobirep’

April 22, 2021

Title Models Bivariate Dependence and Produces Bivariate Return Periods

Version 0.2.3

Description Models the dependence between two variables in the extremes, identifies most relevant models among six models: the conditional extremes model, the Jt-KDE model and four copulae (Gumbel, Galambos, Normal, FGM). Bivariate return periods for the six models and bivariate level curves can be created.
Methods used in the package are described in the following reference:
Tilloy, Malamud, Winter and Joly-Laugel (2020) <doi:10.5194/nhess-20-2091-2020>
Supporting references for the conditional extremes model,
Jt-KDE model and for copula modelling are the following:
Heffernan and Tawn (2004) <doi:10.1111/j.1467-9868.2004.02050.x>
Cooley, Thibaud, Castillo and Wehner (2019) <doi:10.1007/s10687-019-00348-0>
Nelsen (2006) <doi:10.1007/0-387-28678-0>.

License GPL (>= 3)

Encoding UTF-8

LazyData True

RoxygenNote 7.1.1

Depends R (>= 3.5.0), texmex

Imports stats, copBasic, grDevices, ks, lattice, SpatialExtremes, zoo, copula, ggplot2, viridis,

Suggests knitr, testthat, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Alois Tilloy [aut, cre] (<https://orcid.org/0000-0002-5881-0642>)

Maintainer Alois Tilloy <alois.tilloy@kcl.ac.uk>

Repository CRAN

Date/Publication 2021-04-22 14:00:09 UTC

R topics documented:

`AnalogSel` *Identifies synthetic datasets analig to input data*

Description

Automatically identifies analog datasets from the 60 datasets created in Tilloy et al.(2020). The function helps the user to select relevant bivariate model among the six models discussed in Tilloy et al. (2020) See https://nhess.copernicus.org/articles/20/2091/2020/nhess-20-2091-2020.html for more detail.

Usage

```
AnalogSel(u2)
```

Arguments

`u2`	Two column data frame

Value

No return value, called for side effects

Examples

```
data(porto)
AnalogSel(fire01meantemp)
```

`Bv.LT.Dep` *Fits bivariate joint tail model*

Description

Take data on uniform margins and fit the Ledford and Tawn (1997) joint tail model. Also contains the method where additional information from values that are extreme in at most one variable is used.

Usage

```
Bv.LT.Dep(
  data,
  mod.thresh.u,
  crit.lev.u,
  sig.lev = 0.05,
  ci.meth = "se",
  marg.inf = F
)
```

Arguments

`data`	Two column data frame with values given on uniform margins
`mod.thresh.u`	Modelling threshold on uniform margin
`crit.lev.u`	Critical level on uniform margin
`sig.lev`	Significance level for confidence intervals (`default` 0.05 for 95% confidence interval)
`ci.meth`	Method to use to obtain confidence intervals, `'se'` for standard error confidence intervals and `'pl'` for profile likelihood confidence intervals
`marg.inf`	Is additional marginal information from points that are extreme in only one variable used? `FALSE` gives Ledford and Tawn (1997) result, `TRUE` gives results using Section 2.4.2 from Hugo Winter's thesis.

Value

Estimates of two dependence parameters with confidence intervals:

- threshold dependent extremal dependence measure
- threshold dependent coefficient of tail dependence

Cond.mod.ap *Fits the bivariate conditional extremes model*

Description

Fits the bivariate conditional extremes model (from Heffernan and Tawn (2004) and texmex R package) and provides estimates of a conditional or joint exceedance level curve with a probability corresponding to 'pobj'. Also provides estimates of dependence measures.

Usage

```
Cond.mod.ap(
  u2,
  tr1,
  tr2,
  tsim,
  num.sim,
  pobj,
  interh = "comb",
  mar1,
  mar2,
  px,
  py
)
```

Arguments

u2	Two column data frame
tr1	extreme threshold for first variable
tr2	extreme threshold for second variable
tsim	Prediction quantile. The quantile of the conditioning variable above which it will be simulated for importance sampling based prediction (from texmex)
num.sim	The number of simulated observations to be generated for prediction (from texmex)
pobj	objective joint return period modelled with the conditional extremes model
interh	type of hazard interrelation 'comb' for compound (joint exceedance probability) and 'casc' for cascade (conditional porbability)
mar1	Values of the first margin
mar2	Values of the second margin
px	Uniform values of the first margin with a mixed distribution (empirical below and gpd above a threshold)
py	Uniform values of the second margin with a mixed distribution (empirical below and gpd above a threshold)

Value

a list containing the following:

- jline - data frame of the objective level curve with the selected return period 'pobj'
- onlysim - data frame of simulated extreme data for the two variables
- etaHT - threshold dependent extremal dependence measure
- chiHT - threshold dependent coefficient of tail dependence

References

Tilloy, A., Malamud, B.D., Winter, H. and Joly-Laugel, A., 2020. Evaluating the efficacy of bivariate extreme modelling approaches for multi-hazard scenarios. Natural Hazards and Earth System Sciences, 20(8), pp.2091-2117.

Heffernan, J.E. and Tawn, J.A., 2004. A conditional approach for multivariate extreme values (with discussion). Journal of the Royal Statistical Society: Series B (Statistical Methodology), 66(3), pp.497-546.

See Also

mex

Examples

```
# Import data
data(porto)
tr1=0.9
tr2=0.9
fire01meantemp=na.omit(fire01meantemp)
u=fire01meantemp

#Compute uniform margins
marg=Margins.mod(tr1,tr2,u=fire01meantemp)
kk=marg$uvar
pp=marg$uvar_ext
uu=marg$val_ext

upobj=0.001
t.sim=0.98
interh="comb"
 ## Not run:
# Fit conditional extremes model
condexres<-Cond.mod.ap(u2=u,tr1,tr2,tsim=t.sim,num.sim=10000,
pobj=upobj,mar1=uu[,1],mar2=uu[,2],px=pp[,1],py=pp[,2],interh=interh)

 plot(condexres$jline)

## End(Not run)
```

`curve.funct` *Creates extreme level curves for copulae*

Description

The function creates extreme level curves for copulae. The function divides the 2D space into 3 subspaces to model level curves with extreme low probability 'pobj' (<0.001). Two types of level curves can be modeled: conditional or joint exceedance.

Usage

```
curve.funct(
  pxf,
  pyf,
  mar1,
  mar2,
  pos,
  pobje,
  ng = 100,
  inter = "comb",
  coco,
  c1,
  logm = FALSE
)
```

Arguments

`pxf`	Uniform values of the first margin with a mixed distribution (empirical below and gpd above a threshold)
`pyf`	Uniform values of the second margin with a mixed distribution (empirical below and gpd above a threshold)
`mar1`	Values of the first margin
`mar2`	Values of the second margin
`pos`	part of the curve to be modelled 'l' for the left part, 'm' for the middle part and 'r' for the right part
`pobje`	probability of the level curve to be modelled
`ng`	number of points to be interpolated
`inter`	type of hazard interrelation 'comb' for compound (joint exceedance probability) and 'casc' for cascade (conditional porbability)
`coco`	a copula function from the following: GHcop,NORMcop,FGMcop,GLcop
`c1`	the parameter of the copula
`logm`	log tranformation of the margins 'TRUE'or 'FALSE'

Value

two column matrix representing the level curve for a given probability

Examples

```
data(porto)
tr1=0.9
tr2=0.9
fire01meantemp=na.omit(fire01meantemp)
u=fire01meantemp

#Compute uniform margins
marg=Margins.mod(tr1,tr2,u=fire01meantemp)
pp=marg$uvar_ext
uu=marg$val_ext

#Copula parameters
c1=1.5
copu<-copBasic::GHcop

upobj=0.001
interh="comb"

#compute the curve on 3 subdomains
cl1<-curve.funct(pxf=pp[,1],pyf=pp[,2],mar1=uu[,1],mar2=uu[,2],pos="l",
pobje=upobj,ng=100,inter=interh,coco=copu,c1=c1)
cl2<-curve.funct(pxf=pp[,1],pyf=pp[,2],mar1=uu[,1],mar2=uu[,2],pos="m",
pobje=upobj,ng=100,inter=interh,coco=copu,c1=c1)
cl3<-curve.funct(pxf=pp[,1],pyf=pp[,2],mar1=uu[,1],mar2=uu[,2],pos="r",
pobje=upobj,ng=100,inter=interh,coco=copu,c1=c1)

cl<-rbind(cl1,cl2,cl3)
```

---

densi.curv.cop *Computes the density of level curves for copulae models*

Description

Computes the density along the level curve estimate with a copula. Based on the density function of the selected copula.

Usage

```
densi.curv.cop(lines, copi, pxf, pyf, u)
```

Arguments

`lines`	location of the objective level curve for which the density needs to be estimated in the 2D space
`copi`	a copula function with the parameters fitted to the bivariate dataset
`pxf`	uniform values of the 1st margin
`pyf`	uniform values of the 1st margin
`u`	original data

Value

density for each points (couple x,y) along the level curves for copulae

See Also

`dcopula`

Examples

```
data(porto)

tr1=0.9
tr2=0.9
fire01meantemp=na.omit(fire01meantemp)
u=fire01meantemp

#Compute uniform margins
marg=Margins.mod(tr1,tr2,u=fire01meantemp)
kk=marg$uvar
pp=marg$uvar_ext
uu=marg$val_ext

#Copula parameters
c1=1.5
copu<-copBasic::GHcop
upobj=0.001
interh="comb"

#compute the curve on 3 subdomains
cl1<-curve.funct(pxf=pp[,1],pyf=pp[,2],mar1=uu[,1],mar2=uu[,2],pos="l",
pobje=upobj,ng=100,inter=interh,coco=copu,c1=c1)
cl2<-curve.funct(pxf=pp[,1],pyf=pp[,2],mar1=uu[,1],mar2=uu[,2],pos="m",
pobje=upobj,ng=100,inter=interh,coco=copu,c1=c1)
cl3<-curve.funct(pxf=pp[,1],pyf=pp[,2],mar1=uu[,1],mar2=uu[,2],pos="r",
pobje=upobj,ng=100,inter=interh,coco=copu,c1=c1)

cl<-rbind(cl1,cl2,cl3)

# Homogenization of the number of points
cli<-digit.curves.p(start=c(cl[1,1],cl[1,2]), as.matrix(cl), nPoints=98, closed = FALSE)
```

```
# Computes the density along the curve
co=copula::gumbelCopula(c1,dim=2)
cli<-densi.curv.cop(lines=cli,copi=co,pxf=kk[,1],pyf=kk[,2],u=u)
```

`densi.curv.em` *Computes the density of level curves for non-parametric models*

Description

Computes the density along the level curve estimated by JT-KDE or Cond-Ex models. It is based on a kernel density estimation of simulated points for Cond-Ex and extrapolation of the kernel density estimation of the base curve for the joint tail model

Usage

```
densi.curv.em(kdetab, lines, tl, lines2)
```

Arguments

`kdetab`	a table of dimentsion k*k representing the bivariate density of the data estimated with a kernel density estimator
`lines`	location of the objective level curve for which the deesnity needs to be estimated in the 2D space
`tl`	indicator which model's density have been estimated in the kdetab, '`l`' the joint tail model, '`h`' for the conditional extremes model
`lines2`	location of the base level curve (only used when tl=l)

Value

density for each points (couple x,y) along the level curves

Examples

```
  ## Not run:
 data(porto)
 fire01meantemp=na.omit(fire01meantemp)
 u=fire01meantemp
 jt.dens<-kde(u,gridsize = 200)
ltl<-densi.curv.em(kdetab=jt.dens,lines2=ltlo, tl="l", lines=ltl)

## End(Not run)
```

digit.curves.p *Creates homogeneous level curve for every model*

Description

Creates homogeneous level curve for every model (adapted from the function digit.curves of the geomorph R package)

Usage

```
digit.curves.p(start, curve, nPoints, closed = TRUE)
```

Arguments

start	A numeric vector of x,y, coordinates for the landmark defining the start of the curve
curve	A matrix (p x k) of 2D coordinates for a set of ordered points defining a curve
nPoints	Numeric how many semilandmarks to place equidistantly along the curve (not counting beginning and end points)
closed	Logical Whether the curve is closed (TRUE) or open (FALSE)

Value

A matrix of coordinates for nPoints equally spaced semilandmarks sampled along the curve

References

Bookstein, F. J. 1997 Landmark Methods for Forms without Landmarks: Morphometrics of Group Differences in Outline Shape. Medical Image Analysis 1(3):225-243.

Rohlf, F.J., 2015. The tps series of software. Hystrix 26(1):9-12.

See Also

digit.curves

Examples

```
# Curve  creation
x<-seq(0,30, length=200)
y=90-x^2
curve=data.frame(x,y)

# Homogenization of the number of points
ltl<-digit.curves.p(start=curve[1,], curve=as.matrix(curve), nPoints=98, closed = FALSE)
plot(ltl)
```

`evenPts.p` *Spaces out curve points via linear interpolation*

Description

Basic function for spacing out curve points via linear interpolation (adapted from the function digit.curves of the geomorph package). The main different is that curves are normalized to allow an intercomaprison of confidence scores regardless of the input data. used in digit.curves.p

Usage

```
evenPts.p(x, n)
```

Arguments

`x, n`	numeric vectors

Value

A matrix of coordinates for nPoints equally spaced semilandmarks sampled along the curve in a normalized space

---

`fire01meantemp` *Fire and extreme temperature in portugal*

Description

Bivariate dataset of daily mean temperature and number of wildfires in the Porto district (Portugal) for the period 1980 to 2005. Daily mean temperature data from E-OBS (Cornes et al. (2018) <10.1029/2017JD028200> and wildfire data from Pereira et al. (2011) <10.5194/nhess-11-3343-2011>.

Usage

```
data(porto)
```

Format

Two column dataframe: #'

- temp2.temperature (mean daily temperature in °C)
- temp2.nb (daily number of wildfires)

References

Tilloy, A., Malamud, B.D., Winter, H. and Joly-Laugel, A., 2020. Evaluating the efficacy of bivariate extreme modelling approaches for multi-hazard scenarios. Natural Hazards and Earth System Sciences, 20(8), pp.2091-2117.

JT.KDE.ap *Fits the bivariate joint tail model with Kernel density estimator*

Description

Fits the bivariate joint tail model with Kernel density estimator (Adapted from Cooley et al. (2019))and provides estimates of a conditional or joint exceedance level curve with a probability corresponding to 'pobj'. Also provides estimates of dependence measures.

Usage

```
JT.KDE.ap(
  u2,
  pbas,
  pobj,
  beta,
  vtau,
  devplot = F,
  kk,
  mar1,
  mar2,
  px,
  py,
  interh = NA
)
```

Arguments

u2	Two column data frame
pbas	joint return period to be modelled with a kde
pobj	objective joint return period modelled with the joint tail model
beta	smoothing parameter for the transition between asymptotic dependent and independent regimes near the axes
vtau	estimate of the rank correlation between the two variables
devplot	additional plots for development (significantly slows the function)
kk	uniform margins of the original data
mar1	Values of the first margin
mar2	Values of the second margin
px	Uniform values of the first margin with a mixed distribution (empirical below and gpd above a threshold)
py	Uniform values of the second margin with a mixed distribution (empirical below and gpd above a threshold)
interh	type of hazard interrelation 'comb' for compound and 'casc' for cascade,

Value

a list containing the following:

- levelcurve - data frame the objective containing level curve with a return level 'pobj'
- wq0ri - matrix of the base level curve with a return level 'pbas'
- etaJT - threshold dependent extremal dependence measure
- chiJT - threshold dependent coefficient of tail dependence

References

Tilloy, A., Malamud, B.D., Winter, H. and Joly-Laugel, A., 2020. Evaluating the efficacy of bivariate extreme modelling approaches for multi-hazard scenarios. Natural Hazards and Earth System Sciences, 20(8), pp.2091-2117.

Cooley, D., Thibaud, E., Castillo, F. and Wehner, M.F., 2019. A nonparametric method for producing isolines of bivariate exceedance probabilities. Extremes, 22(3), pp.373-390.

See Also

chi

Examples

```
# Inport data
data(porto)
tr1=0.9
tr2=0.9
fire01meantemp=na.omit(fire01meantemp)
u=fire01meantemp

# Compute uniform margins
marg=Margins.mod(tr1,tr2,u=fire01meantemp)
kk=marg$uvar
pp=marg$uvar_ext
uu=marg$val_ext
upobj=0.001
vtau=cor.test(x=u[,1],y=u[,2],method="kendall")$estimate
interh="comb"
## Not run:
# Fit JT-KDE model
jtres<-JT.KDE.ap(u2=u,pbas=0.01,pobj=upobj,beta=100,kk=kk,vtau=vtau,
devplot=FALSE,mar1=uu[,1],mar2=uu[,2],px=pp[,1],py=pp[,2],interh=interh)
plot(jtres$levelcurve)

## End(Not run)
```

Margins.mod *Compute uniform margins with Generalized Pareto Distribution above threshold*

Description

Compute uniform margins with Generalized Pareto Distribution above threshold

Usage

```
Margins.mod(tr1, tr2, u)
```

Arguments

`tr1`	extreme threshold for first variable
`tr2`	extreme threshold for second variable
`u`	Two column data frame

Value

a list of containing the following pseudo observations (uniform margins) with a mixed distribution (empirical below and gpd above a threshold)

- uvar - data frame of pseudo observations (uniform margins) of the original data 'u'
- uvar_ext - data frame of pseudo observations (uniform margins) with a mixed distribution (empirical below and gpd above a threshold) and 1000 extrapolated values
- val_est - data frame consiting of mix of original data 'u'and 1000 extrapolated values

Examples

```
data(porto)
#set extreme threshold for both variable
tr1=0.9
tr2=0.9
fire01meantemp=na.omit(fire01meantemp)
u=fire01meantemp
marmod=Margins.mod(tr1,tr2,u=fire01meantemp)
```

Index