Package: spd (via r-universe)

September 18, 2024

Type Package

Title Semi Parametric Distribution

Version 2.0-1

Date 2015-07-02

Author Alexios Ghalanos <alexios@4dscape.com>

Maintainer Alexios Ghalanos <alexios@4dscape.com>

Description The Semi Parametric Piecewise Distribution blends the Generalized Pareto Distribution for the tails with a kernel based interior.

Collate misc-spd.R classes-spd.R methods-GPD.R methods-spdFit.R methods-spdDistribution.R methods-spdPlots.R

Depends methods

Imports KernSmooth, stats, graphics, utils

LazyLoad yes

URL http://www.unstarched.net, https://bitbucket.org/alexiosg

License GPL

NeedsCompilation no

Date/Publication 2015-07-03 05:44:57

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

RemoteUrl https://github.com/cran/spd

RemoteRef HEAD

RemoteSha 54742d11de2bc884013390e27ca6a8aeba39fd6b

Contents

spd-package *The Semi-Parametric Distribution (spd) package*

Description

The Semi-Parametric Distribution is a piecewise distribution constructed by parametrically modelling the tails of the distribution using an appropriate distribution (e.g. generalized pareto) and the interior by kernel methods. The package implements fit, distribution, density, quantile and random number generation. Currently, only the generalized pareto distribution is implemented for modelling the tails, but the package can easily be extended.

Details

Package:	spd
Type:	Package
Version:	2.0-0
Date:	2013-12-15
License:	GPL
LazyLoad:	yes
Depends:	methods

The main functionality of the package is contained in the `SPD` class, created by calling `spdfit`. Methods for density `dspd`, distribution `pspd`, quantile `qspd` and random number generation `rspd` exist and take 2 main arguments, the input value and the fitted object.
The spd package uses the `"bkde"` function from the package KernSmooth for the kernel interior fit, while for the tail fit borrows from the fExtremes package and implements a locally modified copy of the gpd functionality and methods.

Author(s)

Alec Stephenson for the functions from R\‘s `"evd-package"`,
Alec Stephenson for the functions from R\‘s `"evir-package"`,
Alexander McNeil for the EVIS functions underlying the `"evir-package"`,
Diethelm Wuertz for the functions from R\‘s `"fExtremes-package"`,
M.P.Wand and M.C.Jones for the functions from R\‘s `"KernSmooth-package"`,
Alexios Ghalanos for this package.

References

Carmona, R. and J. Morrisson (2001). Heavy Tails and Copulas with Evanesce, ORFE Tech. Report, Princeton University
Carmona, R. (2001). Statistical Analysis of Financial Data, with an implementation in Splus
Embrechts, P., Klueppelberg, C., Mikosch, T. (1997); *Modelling Extremal Events*, Springer

Examples

```
## Not run:
library(MASS)
x<-SP500/100
fit<-spdfit(x)
show(fit)

## End(Not run)
```

GPDFIT-class *Class: Generalized Pareto Distribution*

Description

Locally implemented and slightly modified class for the generalized pareto distribution (gpd) fit, borrowed from package `fExtremes`. Created on modelling the tails of the data by `spdfit`

Objects from the Class

Objects of this class cannot be created by user as the methods are not exported.

Slots

`call`: Object of class `"call"`

`method`: Object of class `"character"`

`parameter`: Object of class `"list"`

`data`: Object of class `"list"`

`fit`: Object of class `"list"`

`residuals`: Object of class `"numeric"`

`title`: Object of class `"character"`

`description`: Object of class `"character"`

Note

S3 plot method exists which provides for visual inspection of the fit and is called by the higher level S3 plot method of the `SPD` class

Author(s)

Alec Stephenson for the functions from R\'s "evd-package", Alec Stephenson for the functions from R\'s "evir-package", Alexander McNeil for the EVIS functions underlying the "evir-package", Diethelm Wuertz for the functions from R\'s "fExtremes-package", M.P.Wand and M.C.Jones for the functions from R\'s "KernSmooth-package", Alexios Ghalanos for this package.

References

Carmona, R. and J. Morrisson (2001). Heavy Tails and Copulas with Evanesce, ORFE Tech. Report, Princeton University
Carmona, R. (2001). Statistical Analysis of Financial Data, with an implementation in Splus
Embrechts, P., Klueppelberg, C., Mikosch, T. (1997); *Modelling Extremal Events*, Springer

`GPDTAILS-class` *Class: GPDTAILS*

Description

Class: SPD with Generalized Pareto Distribution (GPD) Tails

Objects from the Class

Objects can be created by calling `spdfit`. The main implemented class of the spd package holding the details of the fitted object with gpd tails.

Slots

`call`: ...

`method`: the gpd fitting method

`kernel`: the kernel type

`data`: the original dataset

`threshold`: the upper and lower thresholds fitted

`ptails`: the upper and lower cutoff points.

`fit`: the upper and lower gpd fit and the interior kernel fit objects.

`title`: optional title of project

`description`: optional description

Extends

Class `SPD`, directly.

Author(s)

Alexios Ghalanos

References

Carmona, R. and J. Morrisson (2001). Heavy Tails and Copulas with Evanesce, ORFE Tech. Report, Princeton University
Carmona, R. (2001). Statistical Analysis of Financial Data, with an implementation in Splus

plot-methods *Method: Plotting (S4) for implemented S4 classes*

Description

Locally implemented and modified methods for plotting the fit of the GPDFIT object (taken from package fExtremes), and the overall fit of the GPDTAILS object.

Usage

```
plot(x,y,...)
```

Arguments

x An object of class GPDFIT or GPDTAILS.

y missing

... [which] -
Valid arguments are “ask” for interactive plotting, “all” to plot all or a number from 1 to 6 indicating the desired charts to plot.

Examples

```
## Not run:
library(MASS)
x<-SP500/100
fit<-spdfit(x)
plot(fit,which=1)
# this in fact exctracts the GPDFIT object (from GPDTAILS) for which plot
# methods exist.
plot(fit,which=3)

## End(Not run)
```

SPD-class *Class: Semi-Parametric Distribution*

Description

Virtual Class for holding the tail fit and kernel interior objects.

Methods

dspd `signature(x = "numeric", fit = "SPD", linear = "logical")`: density function

pspd `signature(q = "numeric", fit = "SPD", linear = "logical")`: distribution function

qspd `signature(p = "numeric", fit = "SPD", linear = "logical")`: quantile function

rspd `signature(n = "numeric", fit = "SPD", linear = "logical")`: random number generation function

show `signature(object = "SPD")`: show method

Objects from the Class

A virtual Class: No objects may be created from it.

Author(s)

Alexios Ghalanos

References

Carmona, R. and J. Morrisson (2001). Heavy Tails and Copulas with Evanesce, ORFE Tech. Report, Princeton University

Carmona, R. (2001). Statistical Analysis of Financial Data, with an implementation in Splus

Examples

```
showClass("SPD")
```

spd-methods *Method: Semi-Parametric Distribution*

Description

Density, Distribution, Quantile and Random Number Generation methods for the Semi-Parametric Distribution.

Usage

```
dspd(x, fit, linear)
pspd(q, fit, linear)
qspd(p, fit, linear)
rspd(n, fit, linear)
```

Arguments

`n` [rspd] -
the number of random deviates to be generated from fitted distribution.

`p` a vector of probability levels, the desired probability for the quantile estimate (e.g. 0.99 for the 99th percentile).

`x, q` [pspd,dspd] -
a numeric vector of quantiles.

`fit` [all] -
the object of class SPDFIT created by calling the spdfit function.

`linear` [all] -
logical, if TRUE interior smoothing function uses linear interpolation rather than constant.

Value

All values are numeric vectors:
`d*` returns the density (pdf),
`p*` returns the probability (cdf),
`q*` returns the quantiles (inverse cdf), and
`r*` generates random deviates.

Note

The density is computed using the generalized pareto distribution in the tails, while for the middle, the density is computed by using a smooth gradient approach. Interpolation is used to splice together the ends with the middle portion, providing for an approximate piecewise constant density function. As such, caution should be used when interpreting results obtained by use of this function.

Author(s)

Alec Stephenson for the functions from R\‘s `evd` package,
Alec Stephenson for the functions from R\‘s `evir` package,
Alexander McNeil for the EVIS functions underlying the `evir` package,
Diethelm Wuetrz for the `fExtremes` Implementation of the gpd,
Alexios Ghalanos for the SPD Implementation,

References

Embrechts, P., Klueppelberg, C., Mikosch, T. (1997); *Modelling Extremal Events*, Springer.
Carmona, R. (2004);*Statistical Anlaysis of Financial Data in Splus*, Springer.

Examples

```
## Not run:
library(MASS)
x = SP500/100
fit=spdfit(x, upper=0.9, lower=0.1)
## rspd  -
   par(mfrow = c(2, 2), cex = 0.7)
   r = rspd(n = 1000, fit)
   hist(r, n = 100, probability = TRUE, xlab = "r",
   col = "steelblue", border = "white",main = "Density")
   box()
## dspd -
   # Plot empirical density and compare with true density:
   r = rspd(n = 1000, fit)
   hist(r, n = 100, probability = TRUE, xlab = "r",
   col = "steelblue", border = "white",main = "Density")
   box()
   x = seq(-0.3, 0.3, length.out = 1000)
   lines(x, dspd(x, fit), col = "darkorange",lwd=2)

## pspd -
   # Plot df and compare with true df:
   plot(sort(r), (1:length(r)/length(r)),
   ylim = c(0, 1), pch = 19,
   cex = 0.5, ylab = "p", xlab = "q", main = "CDF")
   grid()
   q = seq(-0.3, 0.3, length.out = 1000)
   lines(q, pspd(q, fit), col = "darkorange",lwd=2)

## End(Not run)
```

spdfit-methods *Method: Fitting the Semi-Parametric Distribution*

Description

The semi-parametric distribution fitting method.

Usage

```
spdfit(data, upper = 0.9, lower = 0.1, tailfit="GPD", type = c("mle", "pwm"),
kernelfit = c("normal", "box", "epanech", "biweight", "triweight"),
information = c("observed", "expected"), title = NULL, description = NULL, ...)
```

Arguments

`data`	An object coercible to a `matrix`.
`upper`	Upper tail cutoff for fitting the generalized pareto or other distribution.

`lower` Lower tail cutoff for fitting the generalized pareto or other distribution.

`tailfit` Distribution to Use for fitting the tails.

`type` A character string selecting the desired estimation method, either "mle" for the maximum likelihood method or "pwm" for the probability weighted moment method. By default, the first will be selected.

`kernelfit` Type of kernel to fit to the interior of the distribution.

`information` Whether tail distribution standard errors should be calculated with "observed" or "expected" information. This only applies to the maximum likelihood method; for the probability-weighted moments method "expected" information is used if possible.

`title` A character string which allows for a project title.

`description` A character string which allows for a brief description.

`...` Control parameters and plot parameters optionally passed to the optimization and/or plot function. Parameters for the optimization function are passed to components of the control argument of optim.

Value

Returns an object of class `SPD`.

Examples

```
## Not run:
library(MASS)
x<-SP500/100
fit<-spdfit(x)
show(fit)
#plot(fit,which=”all”)

## End(Not run)
```

Index