Package: smoothtail (via r-universe)

September 2, 2024

Type Package

Title Smooth Estimation of GPD Shape Parameter

Version 2.0.5

Date 2016-07-12

Author Kaspar Ru{f}{i}bach <kaspar.rufibach@gmail.com> and Samuel Mueller <samuel.mueller@sydney.edu.au>

Maintainer Kaspar Rufibach <kaspar.rufibach@gmail.com>

Depends logcondens (>= 2.0.0)

Imports stats

Description Given independent and identically distributed observations X(1), ..., X(n) from a Generalized Pareto distribution with shape parameter gamma in [-1,0], offers several estimates to compute estimates of gamma. The estimates are based on the principle of replacing the order statistics by quantiles of a distribution function based on a log--concave density function. This procedure is justified by the fact that the GPD density is log--concave for gamma in [-1,0].

License GPL (>= 2)

URL http://www.kasparrufibach.ch, www.maths.usyd.edu.au/ut/people?who=S_Mueller

NeedsCompilation no

Date/Publication 2016-07-13 10:19:42

Repository https://numbersman77.r-universe.dev

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

RemoteRef HEAD

RemoteSha 1780588b1c87249018fe7e13a333fffa450fdfd6

Contents

`smoothtail-package` *Smooth Estimation of GPD Shape Parameter*

Description

Given independent and identically distributed observations $X_1 < \ldots < X_n$ from a Generalized Pareto distribution with shape parameter $\gamma \in [-1, 0]$, offers three methods to compute estimates of γ. The estimates are based on the principle of replacing the order statistics $X_{(1)}, \ldots, X_{(n)}$ of the sample by quantiles $\hat{X}_{(1)}, \ldots, \hat{X}_{(n)}$ of the distribution function $\hat{F}_n$ based on the log–concave density estimator $\hat{f}_n$. This procedure is justified by the fact that the GPD density is log–concave for $\gamma \in [-1, 0]$.

Details

Package:	smoothtail
Type:	Package
Version:	2.0.5
Date:	2016-07-12
License:	GPL (>=2)

Use this package to estimate the shape parameter γ of a Generalized Pareto Distribution (GPD). In extreme value theory, γ is denoted tail index. We offer three new estimators, all based on the fact that the density function of the GPD is log–concave if $\gamma \in [-1, 0]$, see Mueller and Rufibach (2009). The functions for estimation of the tail index are:

`pickands`
`falk`
`falkMVUE`
`generalizedPick`

This package depends on the package **logcondens** for estimation of a log–concave density: all the above functions take as first argument a `dlc` object as generated by `logConDens` in **logcondens**.

Additionally, functions for density, distribution function, quantile function and random number generation for a GPD with location parameter 0, shape parameter γ and scale parameter σ are provided:

dgpd
pgpd
qgpd
rgpd.

Let us shortly clarify what we mean with log–concave density estimation. Suppose we are given an ordered sample $Y_1 < \ldots < Y_n$ of i.i.d. random variables having density function f, where $f = \exp \varphi$ for a concave function $\varphi : [-\infty, \infty) \rightarrow R$. Following the development in Duembgen and Rufibach (2009), it is then possible to get an estimator $\hat{f}_n = \exp \hat{\varphi}_n$ of f via the maximizer $\hat{\varphi}_n$ of

$$L(\varphi) = \sum_{i=1}^{n} \varphi(Y_i) - \int \exp \varphi(t) dt$$

over all concave functions φ. It turns out that $\hat{\varphi}_n$ is piecewise linear, with knots only at (some of the) observation points. Therefore, the infinite-dimensional optimization problem of finding the function $\hat{\varphi}_n$ boils down to a finite dimensional problem of finding the vector $(\hat{\varphi}_n(Y_1), \ldots, \hat{\varphi}(Y_n))$. How to solve this problem is described in Rufibach (2006, 2007) and in a more general setting in Duembgen, Huesler, and Rufibach (2010). The distribution function based on $\hat{f}_n$ is defined as

$$\hat{F}_n(x) = \int_{Y_1}^{x} \hat{f}_n(t) dt$$

for x a real number. The definition of $\hat{F}_n$ is justified by the fact that $\hat{F}_n(Y_1) = 0$.

Author(s)

Kaspar Rufibach (maintainer), <kaspar.rufibach@gmail.com> ,
http://www.kasparrufibach.ch

Samuel Mueller, <samuel.mueller@sydney.edu.au>,
www.maths.usyd.edu.au/ut/people?who=S_Mueller

Kaspar Rufibach acknowledges support by the Swiss National Science Foundation SNF, http://www.snf.ch

References

Duembgen, L. and Rufibach, K. (2009) Maximum likelihood estimation of a log–concave density and its distribution function: basic properties and uniform consistency. *Bernoulli*, **15(1)**, 40–68.

Duembgen, L., Huesler, A. and Rufibach, K. (2010) Active set and EM algorithms for log-concave densities based on complete and censored data. Technical report 61, IMSV, Univ. of Bern, available at http://arxiv.org/abs/0707.4643.

Mueller, S. and Rufibach K. (2009). Smooth tail index estimation. *J. Stat. Comput. Simul.*, **79**, 1155–1167.

Mueller, S. and Rufibach K. (2008). On the max–domain of attraction of distributions with log–concave densities. *Statist. Probab. Lett.*, **78**, 1440–1444.

Rufibach K. (2006) *Log-concave Density Estimation and Bump Hunting for i.i.d. Observations.* PhD Thesis, University of Bern, Switzerland and Georg-August University of Goettingen, Germany,

2006.
Available at http://www.zb.unibe.ch/download/eldiss/06rufibach_k.pdf.

Rufibach, K. (2007) Computing maximum likelihood estimators of a log-concave density function. *J. Stat. Comput. Simul.*, **77**, 561–574.

See Also

Package **logcondens**.

Examples

```
# generate ordered random sample from GPD
set.seed(1977)
n <- 20
gam <- -0.75
x <- rgpd(n, gam)

# compute known endpoint
omega <- -1 / gam

# estimate log-concave density, i.e. generate dlc object
est <- logConDens(x, smoothed = FALSE, print = FALSE, gam = NULL, xs = NULL)

# plot distribution functions
s <- seq(0.01, max(x), by = 0.01)
plot(0, 0, type = 'n', ylim = c(0, 1), xlim = range(c(x, s))); rug(x)
lines(s, pgpd(s, gam), type = 'l', col = 2)
lines(x, 1:n / n, type = 's', col = 3)
lines(x, est$Fhat, type = 'l', col = 4)
legend(1, 0.4, c('true', 'empirical', 'estimated'), col = c(2 : 4), lty = 1)

# compute tail index estimators for all sensible indices k
falk.logcon <- falk(est)
falkMVUE.logcon <- falkMVUE(est, omega)
pick.logcon <- pickands(est)
genPick.logcon <- generalizedPick(est, c = 0.75, gam0 = -1/3)

# plot smoothed and unsmoothed estimators versus number of order statistics
plot(0, 0, type = 'n', xlim = c(0,n), ylim = c(-1, 0.2))
lines(1:n, pick.logcon[, 2], col = 1); lines(1:n, pick.logcon[, 3], col = 1, lty = 2)
lines(1:n, falk.logcon[, 2], col = 2); lines(1:n, falk.logcon[, 3], col = 2, lty = 2)
lines(1:n, falkMVUE.logcon[,2], col = 3); lines(1:n, falkMVUE.logcon[,3], col = 3,
    lty = 2)
lines(1:n, genPick.logcon[, 2], col = 4); lines(1:n, genPick.logcon[, 3], col = 4,
    lty = 2)
abline(h = gam, lty = 3)
legend(11, 0.2, c("Pickands", "Falk", "Falk MVUE", "Generalized Pickands'"),
    lty = 1, col = 1:8)
```

falk *Compute original and smoothed version of Falk's estimator*

Description

Given an ordered sample of either exceedances or upper order statistics which is to be modeled using a GPD, this function provides Falk's estimator of the shape parameter $\gamma \in [-1, 0]$. Precisely,

$$\hat{\gamma}_{\mathrm{Falk}} = \hat{\gamma}_{\mathrm{Falk}}(k, n) = \frac{1}{k-1} \sum_{j=2}^{k} \log\Big(\frac{X_{(n)} - H^{-1}((n-j+1)/n)}{X_{(n)} - H^{-1}((n-k)/n)}\Big), \quad k = 3, \ldots, n-1$$

for H either the empirical or the distribution function based on the log–concave density estimator. Note that for any k, $\hat{\gamma}_{\mathrm{Falk}} : R^n \to (-\infty, 0)$. If $\hat{\gamma}_{\mathrm{Falk}} \not\in [-1, 0)$, then it is likely that the log-concavity assumption is violated.

Usage

```
falk(est, ks = NA)
```

Arguments

`est`	Log-concave density estimate based on the sample as output by `logConDens` (a `dlc` object).
`ks`	Indices k at which Falk's estimate should be computed. If set to `NA` defaults to $3, \ldots, n-1$.

Value

n x 3 matrix with columns: indices k, Falk's estimator based on the log-concave density estimate, and the ordinary Falk's estimator based on the order statistics.

Author(s)

Kaspar Rufibach (maintainer), <kaspar.rufibach@gmail.com>,
http://www.kasparrufibach.ch

Samuel Mueller, <samuel.mueller@sydney.edu.au>,
www.maths.usyd.edu.au/ut/people?who=S_Mueller

Kaspar Rufibach acknowledges support by the Swiss National Science Foundation SNF, http://www.snf.ch

References

Mueller, S. and Rufibach K. (2009). Smooth tail index estimation. *J. Stat. Comput. Simul.*, **79**, 1155–1167.

Falk, M. (1995). Some best parameter estimates for distributions with finite endpoint. *Statistics*, **27**, 115–125.

See Also

Other approaches to estimate γ based on the fact that the density is log–concave, thus $\gamma \in [-1, 0]$, are available as the functions pickands, falkMVUE, generalizedPick.

Examples

```
# generate ordered random sample from GPD
set.seed(1977)
n <- 20
gam <- -0.75
x <- rgpd(n, gam)

## generate dlc object
est <- logConDens(x, smoothed = FALSE, print = FALSE, gam = NULL, xs = NULL)

# compute tail index estimator
falk(est)
```

---

falkMVUE *Compute original and smoothed version of Falk's estimator for a known endpoint*

Description

Given an ordered sample of either exceedances or upper order statistics which is to be modeled using a GPD with distribution function F, this function provides Falk's estimator of the shape parameter $\gamma \in [-1, 0]$ if the endpoint

$$\omega(F) = \sup\{x \, : \, F(x) < 1\}$$

of F is known. Precisely,

$$\hat{\gamma}_{\rm MVUE} = \hat{\gamma}_{\rm MVUE}(k,n) = \frac{1}{k}\sum_{j=1}^{k}\log\Big(\frac{\omega(F) - H^{-1}((n-j+1)/n)}{\omega(F) - H^{-1}((n-k)/n)}\Big), \quad k = 2, \ldots, n-1$$

for H either the empirical or the distribution function based on the log–concave density estimator. Note that for any k, $\hat{\gamma}_{\rm MVUE} \, : \, R^n \to (-\infty, 0)$. If $\hat{\gamma}_{\rm MVUE} \not\in [-1, 0)$, then it is likely that the log-concavity assumption is violated.

Usage

```
falkMVUE(est, omega, ks = NA)
```

Arguments

`est`	Log-concave density estimate based on the sample as output by `logConDens` (a `dlc` object).
`omega`	Known endpoint. Make sure that $\omega \geq X_{(n)}$.
`ks`	Indices k at which Falk's estimate should be computed. If set to `NA` defaults to $2, \ldots, n-1$.

Value

n x 3 matrix with columns: indices k, Falk's MVUE estimator using the log-concave density estimate, and the ordinary Falk MVUE estimator based on the order statistics.

Author(s)

Kaspar Rufibach (maintainer), <kaspar.rufibach@gmail.com>,
http://www.kasparrufibach.ch

Samuel Mueller, <samuel.mueller@sydney.edu.au>,
www.maths.usyd.edu.au/ut/people?who=S_Mueller

Kaspar Rufibach acknowledges support by the Swiss National Science Foundation SNF, http://www.snf.ch

References

Mueller, S. and Rufibach K. (2009). Smooth tail index estimation. *J. Stat. Comput. Simul.*, **79**, 1155–1167.

Falk, M. (1994). Extreme quantile estimation in δ-neighborhoods of generalized Pareto distributions. *Statistics and Probability Letters*, **20**, 9–21.

Falk, M. (1995). Some best parameter estimates for distributions with finite endpoint. *Statistics*, **27**, 115–125.

See Also

Other approaches to estimate γ based on the fact that the density is log–concave, thus $\gamma \in [-1, 0]$, are available as the functions `pickands`, `falk`, `generalizedPick`.

Examples

```
# generate ordered random sample from GPD
set.seed(1977)
n <- 20
gam <- -0.75
x <- rgpd(n, gam)

## generate dlc object
est <- logConDens(x, smoothed = FALSE, print = FALSE, gam = NULL, xs = NULL)

# compute tail index estimators
omega <- -1 / gam
falkMVUE(est, omega)
```

`generalizedPick` *Compute generalized Pickand's estimator*

Description

Given an ordered sample of either exceedances or upper order statistics which is to be modeled using a GPD with distribution function F, this function provides Segers' estimator of the shape parameter γ, see Segers (2005). Precisely, for $k = \{1, \ldots, n-1\}$, the estimator can be written as

$$\hat{\gamma}^k_{\text{Segers}}(H) = \sum_{j=1}^{k} \Big(\lambda(j/k) - \lambda((j-1)/k)\Big) \log\Big(H^{-1}((n - \lfloor cj \rfloor)/n) - H^{-1}((n-j)/n)\Big)$$

for H either the empirical or the distribution function based on the log–concave density estimator and λ the mixing measure given in Segers (2005), Theorem 4.1, (i). Note that for any k, $\hat{\gamma}^k_{\text{Segers}} : R^n \to (-\infty, \infty)$. If $\hat{\gamma}_{\text{Segers}} \not\in [-1, 0)$, then it is likely that the log-concavity assumption is violated.

Usage

```
generalizedPick(est, c, gam0, ks = NA)
```

Arguments

`est`	Log-concave density estimate based on the sample as output by `logConDens` (a `dlc` object).
`c`	Number in $(0, 1)$, determining the spacings that are used.
`gam0`	Number in $R \setminus 0.5$, specifying the mixing measure.
`ks`	Indices k at which Falk's estimate should be computed. If set to `NA` defaults to $4, \ldots, n$.

Value

n x 3 matrix with columns: indices k, Segers' estimator using the smoothing method, and the ordinary Segers' estimator based on the order statistics.

Author(s)

Kaspar Rufibach (maintainer), `<kaspar.rufibach@gmail.com>`,
http://www.kasparrufibach.ch

Samuel Mueller, `<samuel.mueller@sydney.edu.au>`,
www.maths.usyd.edu.au/ut/people?who=S_Mueller

Kaspar Rufibach acknowledges support by the Swiss National Science Foundation SNF, http://www.snf.ch

References

Mueller, S. and Rufibach K. (2009). Smooth tail index estimation. *J. Stat. Comput. Simul.*, **79**, 1155–1167.

Segers, J. (2005). Generalized Pickands estimators for the extreme value index. *J. Statist. Plann. Inference*, **128**, 381–396.

See Also

Other approaches to estimate γ based on the fact that the density is log–concave, thus $\gamma \in [-1, 0]$, are available as the functions `pickands`, `falk`, `falkMVUE`.

Examples

```
# generate ordered random sample from GPD
set.seed(1977)
n <- 20
gam <- -0.75
x <- rgpd(n, gam)

## generate dlc object
est <- logConDens(x, smoothed = FALSE, print = FALSE, gam = NULL, xs = NULL)

# compute tail index estimators
generalizedPick(est, c = 0.75, gam0 = -1/3)
```

---

gpd — *The Generalized Pareto Distribution*

Description

Density function, distribution function, quantile function and random generation for the generalized Pareto distribution (GPD) with shape parameter γ and scale parameter σ.

Usage

```
dgpd(x, gam, sigma = 1)
pgpd(q, gam, sigma = 1)
qgpd(p, gam, sigma = 1)
rgpd(n, gam, sigma = 1)
```

Arguments

x, q	Vector of quantiles.
p	Vector of probabilities.
n	Number of observations.
gam	Shape parameter, real number.
sigma	Scale parameter, positive real number.

Details

The generalized Pareto distribution function (Pickands, 1975) with shape parameter γ and scale parameter σ is

$$W_{\gamma,\sigma}(x) = 1 - (1 + \gamma x/\sigma)_+^{-1/\gamma}.$$

If $\gamma = 0$, the distribution function is defined by continuity. The density is denoted by $w_{\gamma,\sigma}$.

Value

dgpd gives the values of the density function, pgpd those of the distribution function, and qgpd those of the quantile function of the GPD at $\boldsymbol{x}, \boldsymbol{q}$, and $\boldsymbol{p}$, respectively. rgpd generates n random numbers, returned as an ordered vector.

Author(s)

Kaspar Rufibach, <kaspar.rufibach@gmail.com>,
http://www.kasparrufibach.ch

Samuel Mueller, <samuel.mueller@sydney.edu.au>,
www.maths.usyd.edu.au/ut/people?who=S_Mueller

References

Pickands, J. (1975). Statistical inference using extreme order statistics. *Annals of Statistics*, **3**, 119-131.

See Also

Similar functions are provided in the R-packages **evir** and **evd**.

`lambdaGenPick` *Auxiliary function to compute Segers' estimator*

Description

This function computes

$$\lambda^c_{\delta,\rho}$$

given in Theorem 4.1 of Segers (2005) and is called by generalizedPick. It is not intended to be called by the user.

Author(s)

Kaspar Rufibach (maintainer), <kaspar.rufibach@gmail.com>,
http://www.kasparrufibach.ch

Samuel Mueller, <samuel.mueller@sydney.edu.au>,
www.maths.usyd.edu.au/ut/people?who=S_Mueller

Kaspar Rufibach acknowledges support by the Swiss National Science Foundation SNF, http://www.snf.ch

References

Mueller, S. and Rufibach K. (2009). Smooth tail index estimation. *J. Stat. Comput. Simul.*, **79**, 1155–1167.

Segers, J. (2005). Generalized Pickands estimators for the extreme value index. *J. Statist. Plann. Inference*, **128**, 381–396.

See Also

Called by `generalizedPick`.

`pickands` *Compute original and smoothed version of Pickands' estimator*

Description

Given an ordered sample of either exceedances or upper order statistics which is to be modeled using a GPD, this function provides Pickands' estimator of the shape parameter $\gamma \in [-1, 0]$. Precisely, for $k = 4, \ldots, n$

$$\hat{\gamma}^k_{\mathrm{Pick}} = \frac{1}{\log 2} \log\Big(\frac{H^{-1}((n - r_k(H) + 1)/n) - H^{-1}((n - 2r_k(H) + 1)/n)}{H^{-1}((n - 2r_k(H) + 1)/n) - H^{-1}((n - 4r_k(H) + 1)/n)}\Big)$$

for H either the empirical or the distribution function $\hat{F}_n$ based on the log–concave density estimator and

$$r_k(H) = \lfloor k/4 \rfloor$$

if H is the empirical distribution function and

$$r_k(H) = k/4$$

if $H = \hat{F}_n$.

Usage

```
pickands(est, ks = NA)
```

Arguments

est	Log-concave density estimate based on the sample as output by logConDens (a dlc object).
ks	Indices k at which Falk's estimate should be computed. If set to NA defaults to $4, \ldots, n$.

Value

n x 3 matrix with columns: indices k, Pickands' estimator using the log-concave density estimate, and the ordinary Pickands' estimator based on the order statistics.

Author(s)

Kaspar Rufibach (maintainer), <kaspar.rufibach@gmail.com>,
http://www.kasparrufibach.ch

Samuel Mueller, <samuel.mueller@sydney.edu.au>,
www.maths.usyd.edu.au/ut/people?who=S_Mueller

Kaspar Rufibach acknowledges support by the Swiss National Science Foundation SNF, http://www.snf.ch

References

Mueller, S. and Rufibach K. (2009). Smooth tail index estimation. *J. Stat. Comput. Simul.*, **79**, 1155–1167.

Pickands, J. (1975). Statistical inference using extreme order statistics. *Annals of Statistics* **3**, 119–131.

See Also

Other approaches to estimate γ based on the fact that the density is log–concave, thus $\gamma \in [-1, 0]$, are available as the functions falk, falkMVUE, generalizedPick.

Examples

```
# generate ordered random sample from GPD
set.seed(1977)
n <- 20
gam <- -0.75
x <- rgpd(n, gam)

## generate dlc object
est <- logConDens(x, smoothed = FALSE, print = FALSE, gam = NULL, xs = NULL)

# compute tail index estimators
pickands(est)
```

Index