Package ‘stable’

February 4, 2019

Version 1.1.4

Title Probability Functions and Generalized Regression Models for Stable Distributions

Imports stabledist

Depends R (>= 1.4), rmutil

Description Density, distribution, quantile and hazard functions of a stable variate; generalized regression models for the parameters of a stable distribution. See the README for how to make equivalent calls to those of 'stabledist'. See github for Journal article.

License GPL (>= 2)

URL http://www.commanster.eu/rcode.html

BugReports https://github.com/swihart/stable/issues

Encoding UTF-8

LazyData true

LazyLoad true

RoxygenNote 6.1.0

NeedsCompilation yes

Author Bruce Swihart [cre, aut],
Jim Lindsey [aut] (Jim created this package, Bruce is maintaining the CRAN version),
Philippe Lambert [aut]

Maintainer Bruce Swihart <bruce.swihart@gmail.com>

Repository CRAN

Date/Publication 2019-02-04 19:20:03 UTC

R topics documented:

`dstable` *Stable Distribution*

Description

These functions provide information about the stable distribution with the location, the dispersion, the skewness and the tail thickness respectively modelled by the parameters `loc`, `disp`, `skew` and `tail`.

Usage

```
dstable(x, loc = 0, disp = 1/sqrt(2), skew = 0, tail = 2,
  npt = 501, up = 10, eps = 1e-06, integration = "Romberg")

pstable(q, loc = 0, disp = 1/sqrt(2), skew = 0, tail = 2,
  eps = 1e-06)

qstable(p, loc = 0, disp = 1/sqrt(2), skew = 0, tail = 2,
  eps = 1e-06)

rstable(n = 1, loc = 0, disp = 1/sqrt(2), skew = 0, tail = 2,
  eps = 1e-06)

hstable(x, loc = 0, disp = 1/sqrt(2), skew = 0, tail = 2,
  eps = 1e-06)
```

Arguments

`x, q`	vector of quantiles.
`loc`	vector of (real) location parameters.
`disp`	vector of (positive) dispersion parameters.
`skew`	vector of skewness parameters (in [-1,1]).
`tail`	vector of parameters (in [0,2]) related to the tail thickness.
`npt, up, integration`	As detailed herein – only available when using `dstable`.
`eps`	scalar giving the required precision in computation.
`p`	vector of probabilites.
`n`	number of observations.

Details

`dstable`, `pstable`, `qstable` and `hstable` compute the density, the distribution, the quantile and the hazard functions of a stable variate. `rstable` generates random deviates with the prescribed stable distribution.

`loc` is a location parameter in the same way as the mean in the normal distribution: it can take any real value.

`disp` is a dispersion parameter in the same way as the standard deviation in the normal distribution: it can take any positive value.

`skew` is a skewness parameter: it can take any value in $(-1, 1)$. The distribution is right-skewed, symmetric and left-skewed when `skew` is negative, null or positive respectively.

`tail` is a tail parameter (often named the characteristic exponent): it can take any value in $(0, 2)$ (with `tail=1` and `tail=2` yielding the Cauchy and the normal distributions respectively when symmetry holds).

If `loc`, `disp`, `skew`, or `tail` are not specified they assume the default values of 0, $1/sqrt(2)$, 0 and 2 respectively. This corresponds to a normal variate with mean$= 0$ and variance$= 1/2disp^2$.

The stable characteristic function is given by

$$greekphi(t) = ilocat - disp|t|^{tail}[1 + iskewsign(t)greekomega(t, tail)]$$

where

$$greekomega(t, tail) = \frac{2}{\pi}LOG(ABS(t))$$

when `tail=1`, and

$$greekomega(t, tail) = tan(\frac{\pi tail}{2})$$

otherwise.

The characteristic function is inverted using Fourier's transform to obtain the corresponding stable density. This inversion requires the numerical evaluation of an integral from 0 to ∞. Two algorithms are proposed for this. The default is Romberg's method (`integration`="Romberg") which is used to evaluate the integral with an error bounded by `eps`. The alternative method is Simpson's integration (`integration`="Simpson"): it approximates the integral from 0 to ∞ by an integral from 0 to `up` with `npt` points subdividing (O, up). These three extra arguments – `integration`, `up` and `npt` – are only available when using `dstable`. The other functions are all based on Romberg's algorithm.

Functions

- `dstable`: density
- `pstable`: cdf
- `qstable`: quantiles
- `rstable`: random deviates
- `hstable`: hazard

Author(s)

Philippe Lambert (Catholic University of Louvain, Belgium, <phlambert@stat.ucl.ac.be>)

Jim Lindsey

References

Lambert, P. and Lindsey, J.K. (1999) Analysing financial returns using regression models based on non-symmetric stable distributions. Applied Statistics, 48, 409-424.

See Also

stablereg to fit generalized nonlinear regression models for the stable distribution parameters.

Examples

```
par(mfrow=c(2,2))
x <- seq(-5,5,by=0.1)

# Influence of loc (location)
plot(x,dstable(x,loc=-2,disp=1/sqrt(2),skew=-0.8,tail=1.5),
  type="l",ylab="",main="Varying LOCation")
lines(x,dstable(x,loc=0,disp=1/sqrt(2),skew=-0.8,tail=1.5))
lines(x,dstable(x,loc=2,disp=1/sqrt(2),skew=-0.8,tail=1.5))

# Influence of disp (dispersion)
plot(x,dstable(x,loc=0,disp=0.5,skew=0,tail=1.5),
  type="l",ylab="",main="Varying DISPersion")
lines(x,dstable(x,loc=0,disp=1/sqrt(2),skew=0,tail=1.5))
lines(x,dstable(x,loc=0,disp=0.9,skew=0,tail=1.5))

# Influence of skew (skewness)
plot(x,dstable(x,loc=0,disp=1/sqrt(2),skew=-0.8,tail=1.5),
  type="l",ylab="",main="Varying SKEWness")
lines(x,dstable(x,loc=0,disp=1/sqrt(2),skew=0,tail=1.5))
lines(x,dstable(x,loc=0,disp=1/sqrt(2),skew=0.8,tail=1.5))

# Influence of tail (tail)
plot(x,dstable(x,loc=0,disp=1/sqrt(2),skew=0,tail=0.8),
  type="l",ylab="",main="Varying TAIL thickness")
lines(x,dstable(x,loc=0,disp=1/sqrt(2),skew=0,tail=1.5))
lines(x,dstable(x,loc=0,disp=1/sqrt(2),skew=0,tail=2))

stabledist::dstable(x=1, 1, 0)
```

Links *Links*

Description

Link and inverse functions for use in stablereg

Usage

```
loc_g(x)

loc_h(x)

disp_g(x)

disp_h(x)

skew_g(x)

skew_h(x)

tail_g(x)

tail_h(x)
```

Arguments

x	the function argument

Parameter_Conversion *Easy conversion of parameters between stabledist and stable*

Description

sd2s has stabledist parameter inputs and returns stable parameters. s2sd has stable parameter inputs and returns stabledist parameters.

Usage

```
sd2s(alpha, beta, gamma, delta, pm = 1)

s2sd(tail, skew, disp, loc, pm = 1)
```

Arguments

alpha	the stabledist 'alpha'
beta	the stabledist 'beta'
gamma	the stabledist 'gamma'
delta	the stabledist 'delta'
pm	default 1; currently only value supported. the stabledist parameterization 'pm'
tail	the stable 'tail' analogous to 'alpha'
skew	the stable 'skew' analogous to 'beta'
disp	the stable 'disp' analogous to 'gamma'
loc	the stable 'loc' analogous to 'delta'

Details

This is a generic function: methods can be defined for it directly or via the Summary group generic. For this to work properly, the arguments ... should be unnamed, and dispatch is on the first argument.

Value

What you need. See examples.

Examples

```
q <- -1
# nolan pm=1 parameters:
a <-  1.3
b <-  -0.4
c <-  2
d <-  0.75
s <- sd2s(alpha=a, beta=b, gamma=c, delta=d)
stable::pstable(q, tail = s$tail, skew=s$skew, disp = s$disp, loc  = s$loc)
stabledist::pstable(q, alpha=a, beta=b , gamma=c , delta=d, pm=1)
sd <- s2sd(tail = s$tail, skew=s$skew, disp = s$disp, loc  = s$loc)
stabledist::pstable(q, alpha=sd$alpha, beta=sd$beta , gamma=sd$gamma , delta=sd$delta, pm=1)
```

`Parameter_Conversion_Nolan_pm1_pm0`

Easy conversion of parameters between stabledist and stable

Description

`pm0_to_pm1` has stabledist parameter inputs for pm=0 and returns pm=1 equivalent parameterization. `pm1_to_pm0` has stabledist parameter inputs for pm=1 and returns pm=0 equivalent parameterization.

Usage

```
pm0_to_pm1(a0, b0, c0, d0)

pm1_to_pm0(a1, b1, c1, d1)
```

Arguments

a0	the stabledist 'alpha' for pm=0 in 'stabledist'
b0	the stabledist 'beta' for pm=0 in 'stabledist'
c0	the stabledist 'gamma' for pm=0 in 'stabledist'
d0	the stabledist 'delta' for pm=0 in 'stabledist'
a1	the stabledist 'alpha' for pm=1 in 'stabledist'

b1	the stabledist 'beta' for pm=1 in 'stabledist'
c1	the stabledist 'gamma' for pm=1 in 'stabledist'
d1	the stabledist 'delta' for pm=1 in 'stabledist'

Value

What you need. See examples.

Examples

```
q <- -1
# nolan pm=1 parameters:
a1 <-  1.3
b1 <-  -0.4
c1 <-  2
d1 <-  0.75
# Convert to nolan pm=0 parameters:
pm0 <- pm1_to_pm0(a1,b1,c1,d1)
a0 <- pm0$a0
b0 <- pm0$b0
c0 <- pm0$c0
d0 <- pm0$d0
# check:
stabledist::pstable(q, alpha=a1, beta=b1 , gamma=c1 , delta=d1, pm=1)
#> [1] 0.1965513
# only change delta=d0 for pm=0
stabledist::pstable(q, alpha=a1, beta=b1 , gamma=c1 , delta=d0, pm=0)
stabledist::pstable(q, alpha=a0, beta=b0 , gamma=c0 , delta=d0, pm=0)
#> [1] 0.1965513
stabledist::dstable(q, alpha=a1, beta=b1 , gamma=c1 , delta=d1, pm=1)
#> [1] 0.0572133
# only change delta=d0 for pm=0
stabledist::dstable(q, alpha=a1, beta=b1 , gamma=c1 , delta=d0, pm=0)
stabledist::dstable(q, alpha=a0, beta=b0 , gamma=c0 , delta=d0, pm=0)
#> [1] 0.0572133
```

---

stable.mode *Mode of a Stable Distribution*

Description

This function gives a reliable approximation to the mode of a stable distribution with location, dispersion, skewness and tail thickness specified by the parameters `loc`, `disp`, `skew` and `tail`. `tail` must be in (1,2).

Usage

```
stable.mode(loc, disp, skew, tail)
```

Arguments

loc	vector of (real) location parameters.
disp	vector of (positive) dispersion parameters.
skew	vector of skewness parameters (in [-1,1]).
tail	vector of parameters (in [1,2]) related to the tail thickness.

Details

`loc` is a location parameter in the same way as the mean in the normal distribution: it can take any real value.

`disp` is a dispersion parameter in the same way as the standard deviation in the normal distribution: it can take any positive value.

`skew` is a skewness parameter: it can take any value in $(-1, 1)$. The distribution is right-skewed, symmetric and left-skewed when `skew` is negative, null or positive respectively.

`tail` is a tail parameter (often named the characteristic exponent): it can take any value in $(0, 2)$ (with `tail=1` and `tail=2` yielding the Cauchy and the normal distributions respectively when symmetry holds).

The simplest empirical formula found to give a satisfactory approximation to the mode for values of `tail` in $(1, 2)$ is

$$loc + disp * a * skew * exp(-b * abs(skew))$$

with

$$a = 1.7665114 + 1.8417675 * tail - 2.2954390 * tail^2 + 0.4666749 * tail^3$$

and

$$b = -0.003142967 + 632.4715 * tail * exp(-7.106035 * tail)$$

.

Value

A list of size 3 giving the mode, a and b.

Author(s)

Philippe Lambert (Catholic University of Louvain, Belgium, <phlambert@stat.ucl.ac.be>) and Jim Lindsey.

References

Lambert, P. and Lindsey, J.K. (1999) Analysing financial returns using regression models based on non-symmetric stable distributions. Applied Statistics, 48, 409-424.

See Also

`stable` for more details on the stable distribution.

`stablereg` to fit generalized linear models for the stable distribution parameters.

Examples

```
x <- seq(-5,5,by=0.1)
plot(x,dstable(x,loc=0,disp=1,skew=-1,tail=1.5),type="l",ylab="f(x)")
xhat <- stable.mode(loc=0,disp=1,skew=-1,tail=1.5)$ytilde
fxhat <- dstable(xhat,loc=0,disp=1,skew=-1,tail=1.5)
lines(c(xhat,xhat),c(0,fxhat),lty="dotted")
```

---

stablereg *Stable Generalized Regression Models*

Description

`stablereg` fits user specified generalized linear and nonlinear regression models based on the stable distribution to (uncensored, right and/or left censored) data. This allows the location, the dispersion, the skewness and the tails of the fitted stable distribution to vary with explanatory variables.

Usage

```
stablereg(y = NULL, loc = 0, disp = 1, skew = 0, tail = 1.5,
  oloc = TRUE, odisp = TRUE, oskew = TRUE, otail = TRUE,
  noopt = FALSE, iloc = NULL, idisp = NULL, iskew = NULL,
  itail = NULL, loc_h = NULL, disp_h = NULL, skew_h = NULL,
  tail_h = NULL, weights = 1, exact = FALSE, delta = 1,
  envir = parent.frame(), integration = "Romberg", eps = 1e-06,
  up = 10, npoint = 501, hessian = TRUE, llik.output = FALSE,
  print.level = 0, ndigit = 10, steptol = 1e-05, gradtol = 1e-05,
  fscale = 1, typsize = abs(p0), stepmax = sqrt(p0 %*% p0),
  iterlim = 100)
```

Arguments

`y` The response vector or a `repeated` data object. If the `repeated` data object contains more than one response variable, give that object in `envir` and give the name of the response variable to be used here.

For censored data, two columns with the second being the censoring indicator (1: uncensored, 0: right censored, -1: left censored.)

`loc, loc_h, oloc, iloc`
Describe the regression model fitted for the location parameter of the stable distribution, perhaps after transformation by the link function `loc_g` (set to the identity by default. The inverse link function is denoted by `loc_h`. Note that these functions cannot contain unknown parameters).

Two specifications are possible:

(1) `loc` is a linear or nonlinear language expression beginning with ~ or an R function, describing the regression function for the location parameter (after transformation by `loc_g`, the link function).

iloc is a vector of initial conditions for the parameters in the regression for this parameter.

oloc is a boolean indicating if an optimization of the likelihood has to be carried out on these parameters. If oloc is set to TRUE, a default zero value is considered for the starting values iloc. But if no optimization is desired on the location parameters, i.e. when the likelihood has to be evaluated or optimized at a fixed location, then iloc has to be explicitely specified.

(2) loc is a numeric expression (i.e. a scalar or a vector of the same size as the data vector y, or y[,1] when censoring is considered).

If oloc is set to TRUE, i.e. when an optimization of the likelihood has to be carried out on the location parameter, then the location parameter (after transformation by the link function loc_g) is set to an unknown parameter with initial value equal to iloc[1] or loc[1] when iloc is not specified.

But when oloc is set to FALSE, i.e. when the likelihood has to be evaluated or optimized at a fixed location, then the transformed location is assumed to be equal to loc when it is of the same length as the data vector y (or y[,1] when censoring is considered), and to loc[1] otherwise.

Specification (1) is especially useful in ANOVA-like situations where the location is assumed to change with the levels of some factor variable.

disp, disp_h, odisp, idisp
: describe the regression model for the dispersion parameter of the fitted stable distribution, after transformation by the link function disp_g (set to the log function by default). The inverse link function is denoted by disp_h. Again these functions cannot contain unknown parameters. The same rules as above apply when specifying the generalized regression model for the dispersion parameter.

skew, skew_h, oskew, iskew
: describe the regression model for the skewness parameter of the fitted stable distribution, after transformation by the link function skew_g (set to log{(1 + [.])/(1 - [.])} by default). The inverse link function is denoted by skew_h. Again these functions cannot contain unknown parameters. The same rules as above apply when specifying the generalized regression model for the skewness parameter.

tail, tail_h, otail, itail
: describe the regression model considered for the tail parameter of the fitted stable distribution, after transformation by the link function tail_g (set to log{([.] -1)/(2 - [.])} by default. The inverse link function is denoted by tail_h. Again these functions cannot contain unknown parameters). The same rules as above apply when specifying the generalized regression model for the tail parameter.

noopt
: When set to TRUE, it forces oloc, odisp, oskew and otail to FALSE, whatever the user choice for these last three arguments. It is especially useful when looking for appropriate initial values for the regression model parameters, before undertaking the optimization of the likelihood.

weights
: Weight vector.

exact
: If TRUE, fits the exact likelihood function for continuous data by integration over intervals of observation, i.e. interval censoring.

delta
: Scalar or vector giving the unit of measurement for each response value, set to unity by default. For example, if a response is measured to two decimals,

delta=0.01. If the response is transformed, this must be multiplied by the Jacobian. For example, with a log transformation, delta=1/y. (The delta values for the censored response are ignored.) The transformation cannot contain unknown parameters.

envir — Environment in which model formulae are to be interpreted or a data object of class, repeated, tccov, or tvcov; the name of the response variable should be given in y. If y has class repeated, it is used as the environment.

integration, eps, up, npoint — integration indicates which algorithm must be used to evaluate the stable density when the likelihood is computed with exact set to FALSE. See the man page on stable for extra information.

hessian — Arguments controlling the optimization procedure nlm.

llik.output — is TRUE when the likelihood has to be displayed at each iteration of the optimization.

print.level — Arguments controlling the optimization procedure nlm.

ndigit — Arguments controlling the optimization procedure nlm.

steptol — Arguments controlling the optimization procedure nlm.

gradtol — Arguments controlling the optimization procedure nlm.

fscale — Arguments controlling the optimization procedure nlm.

typsize — Arguments controlling the optimization procedure nlm.

stepmax — Arguments controlling the optimization procedure nlm.

iterlim — Arguments controlling the optimization procedure nlm.

Value

A list of class stable is returned. The printed output includes the -log-likelihood, the corresponding AIC, the maximum likelihood estimates, standard errors, and correlations. It also include all the relevant information calculated, including error codes.

Warning

Because of the numerical integrations involved, convergence can be very sensitive to the initial parameter values supplied and to the settings of the arguments controlling nlm. If nlm feeds extreme parameter values in the tails of the distribution to the likelihood function, the integration may hang for a long time.

Author(s)

Philippe Lambert (Catholic University of Louvain, Belgium, <phlambert@stat.ucl.ac.be>) and Jim Lindsey.

References

Lambert, P. and Lindsey, J.K. (1999) Analysing financial returns using regression models based on non-symmetric stable distributions. Applied Statistics 48, 409-424.

See Also

lm, glm, stable and stable.mode.

Examples

```
## Share return over a 50 day period (see reference above)
# shares
y <- c(296,296,300,302,300,304,303,299,293,294,294,293,295,287,288,297,
305,307,307,304,303,304,304,309,309,309,307,306,304,300,296,301,298,
295,295,293,292,297,294,293,306,303,301,303,308,305,302,301,297,299)

# returns
ret <- (y[2:50]-y[1:49])/y[1:49]
# hist(ret, breaks=seq(-0.035,0.045,0.01))

day <- seq(0,0.48,by=0.01) # time measured in days/100
x <- seq(1,length(ret))-1

# Classic stationary normal model tail=2
print(z1 <- stablereg(y = ret, delta = 1/y[1:49],
loc = ~1, disp= ~1, skew = ~1, tail = tail_g(1.9999999),
iloc = 0, idisp = -3, iskew = 0, oskew = FALSE, otail = FALSE))

# Normal model (tail=2) with dispersion=disp_h(b0+b1*day)
print(z2 <- stablereg(y = ret, delta = 1/y[1:49], loc = ~day,
disp = ~1, skew = ~1, tail = tail_g(1.999999), iloc = c(0.003,0),
idisp = -4.5, iskew = 0, oskew = FALSE, otail = FALSE))

# Stable model with loc(ation)=loc_h(b0+b1*day)
print(z3 <- stablereg(y = ret, delta = 1/y[1:49],
loc = ~day, disp = ~1, skew = ~1, tail = ~1,
iloc = c(0.001,-0.004), idisp = -4.8, iskew = 0, itail = 0.6))

# Stable model with disp(ersion)=disp_h(b0+b1*day)
print(z4 <- stablereg(y = ret, delta = 1/y[1:49],
loc = ~1, disp = ~day, skew = ~1, tail = ~1,
iloc = 0.003, idisp = c(-4.8,0), iskew = -0.03, itail = 1.6))

# Stable model with skew(ness)=skew_h(b0+b1*day)
# Evaluation at fixed parameter values (because noopt is set to TRUE)
print(z5 <- stablereg(y = ret, delta = 1/y[1:49],
loc = ~1, disp = ~1, skew = ~day, tail = ~1,
iloc = 5.557e-04, idisp = -4.957, iskew = c(2.811,-2.158),
itail = 1.57, noopt=TRUE))

# Stable model with tail=tail_h(b0+b1*day)
print(z6 <- stablereg(y = ret, delta = 1/y[1:49], loc = ret ~ 1,
disp = ~1, skew = ~1, tail = ~day, iloc = 0.002,
idisp = -4.8, iskew = -2, itail = c(2.4,-4), hessian=FALSE))
```

Index