Package 'sn'

July 23, 2025

Version 2.1.1

Date 2023-04-04

Title The Skew-Normal and Related Distributions Such as the Skew-t and the SUN

Maintainer Adelchi Azzalini <adelchi.azzalini@unipd.it>

Depends R (>= 3.0.0), methods, stats4

Imports mnormt (>= 2.0.0), numDeriv, utils, quantreg

Suggests R.rsp

VignetteBuilder R.rsp

Description Build and manipulate probability distributions of the skew-normal family and some related ones, notably the skew-t and the SUN families. For the skew-normal and the skew-t distributions, statistical methods are provided for data fitting and model diagnostics, in the univariate and the multivariate case.

License GPL-2 | GPL-3

URL http://azzalini.stat.unipd.it/SN/

Encoding UTF-8

NeedsCompilation no

Author Adelchi Azzalini [aut, cre] (ORCID: <https://orcid.org/0000-0002-7583-1269>)

Repository CRAN

Date/Publication 2023-04-04 18:10:02 UTC

Contents

---

sn-package *Package* **sn***: overview, background and history*

Description

The **sn** package provides facilities to define and manipulate probability distributions of the skew-normal (SN) family and some related ones, notably the skew-t (ST) and the unified skew-normal (SUN) families. For a number of these families, statistical methods are provided, to perform data fitting and model diagnostics, in the univariate and the multivariate case.

Overview of the package structure and commands

A separatate document is entirely dedicated to the presentation of the package structure and its basic functions; see the `package overview`.

Background information and references

The package adopts the terminology, notation and general framework of the monograph by Azzalini and Capitanio (2014). This matching constitutes a reason for the numerous references to the book in the documentation of the package.

An additional reason for referring to that monograph instead of the original research papers is that the book provides a relatively not-so-formal account of material which has been elaborated in a number of publications, sometimes very technical, or re-elabotated over a few papers or possibly mixing the information of key interest with other material. In other words, the motivation behind this policy is readability, not indulgence in self-citation.

When one or a few original sources appeared to deliver the required information in a compact and accessible form, they have been cited directly. In any case, the cited sections of the book include bibliographic notes which refer back to the original sources.

A bit of history

The first version of the package was written in 1997, and it was uploaded on CRAN in 1998. Subsequent versions have evolved gradually up to version 0.4-18 in May 2013.

In January 2014, version 1.0-0 has been uploaded to CRAN. This represented a substantial re-writing of the earlier 'version 0.x', developed in broad connection with the book by Azzalini and Capitanio (2014). Differences between the 'version 0' and the 'version 1' series are radical; they concern the core computational and graphical part as well as the user interface. Since version 1.0-0, the S4 protocol for classes and methods has been adopted.

After various versions 1.x-y, version 2.0.0 has appeared in March 2021, providing support for the SUN distribution.

Additional information on the evolution of the package is provided in NEWS file, accessible from the package documentation index page.

Backward compatibility versus 'version 0.4-18'

There is a partial backward compatibility of newer version versus 'version 0-4.18' of the package. Some functions of the older version would work as before with virtually no change; a wider set arguments is now allowed. Functions dsn, dst, dmsn and alike fall in this category: in some cases, the names of the arguments have been altered, but they work as before if called with unnamed arguments; similar cases are msn.mle, sn.cumulants and T.Owen. Notice, however, that msn.mle and other fitting functions have effectively been subsumed into the more comprehensive fitting function selm.

A second group of functions will work with little or even minimal changes. Specific examples are functions sn.mle and st.mle which have become sn.mple and st.mple, with some additional arguments (again, one can achieve the same result via selm). Another example is constitude by the group of functions dp.to.cp, cp.to.dp and st.cumulants.inversion, which have been replaced by the more general functions dp2cp and cp2dp; one only needs to pay attention to conversion from 3rd and 4th order cumulants to their standardized form in connection with the replacement of st.cumulants.inversion.

Finally, some functions are not there any longer, with no similarly-working functions in the new version. This is the case of sn.mle.grouped and st.mle.grouped for maximum likelihood estimation from grouped data, that is, data recorded as intervals and corresponding frequencies.

Requirements

R version 2.15-3 or higher, plus packages **mnormt**, **numDeriv** and **quantreg**, in addition to standard packages (**methods**, **graphics**, **stats4**, etc.)

Version

The command citation("sn") indicates, among other information, the running version of the package. The most recent version of the package can be obtained from the web page: http://azzalini.stat.unipd.it/SN/ which also provides related material.

From the above-indicated web page, one can also obtain the package 'sn0' which is essentially the last 'version 0' (that is, 0.4-18) with suitable renaming of certain ingredients. This allows to have both the current and the old package installed at the same time.

Author

Adelchi Azzalini. Please send comments, error reports *et cetera* to the author, whose web page is http://azzalini.stat.unipd.it/.

Licence

While the software is freely usable, it would be appreciated if a reference is inserted in publications or other work which makes use of it. For the appropriate way of referencing it, see the command `citation("sn")`.

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

package-overview

`affineTransSECdistr` *Affine transformations and marginals of a skew-elliptical distribution*

Description

Given a multivariate random variable Y with skew-elliptical (SEC) distribution, compute the distribution of a (possibly multivariate) marginal or the distribution of an affine transformation $a + A^\top Y$.

Usage

```
affineTransSECdistr(object, a, A, name, compNames, drop=TRUE)
marginalSECdistr(object, comp, name, drop=TRUE)
```

Arguments

`object`	an object of class `SECdistrMv` which identifies the source random variable, as created by `makeSECdistr` or by `extractSECdistr` or by a previous call to these functions
`a`	a numeric vector with the length `ncol(A)`.
`A`	a full-rank matrix with `nrow(A)` equal to the dimensionality `d` of the random variable identified by `object`.
`name`	an optional character string representing the name of the outcome distribution; if missing, one such string is constructed.
`compNames`	an optional vector of length `ncol(A)` of character strings with the names of the components of the outcome distribution; if missing, one such vector is constructed.
`drop`	a logical flag (default value: `TRUE`), operating only if the returned object has dimension `d=1`, in which case it indicates whether this object must be of class `SECdistrUv`.
`comp`	a vector formed by a subset of `1:d` which indicates which components must be extracted from `object`, on denoting by `d` its dimensionality.

Value

If `object` defines the distribution of a SEC random variable Y, `affineTransSECdistr` computes the distribution of $a + A'Y$ and `marginalSECdistr` computes the marginal distribution of the `comp` components. In both cases the returned object is of class `SECdistrMv`, except when `drop=TRUE` operates, leading to an object of class `SECdistrUv`.

Background

These functions implement formulae given in Sections 5.1.4, 5.1.6 and 6.2.2 of the reference below.

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

`makeSECdistr`, `extractSECdistr`, `SECdistrMv-class`

Examples

```
dp3 <- list(xi=1:3, Omega=toeplitz(1/(1:3)), alpha=c(3,-1,2), nu=5)
st3 <- makeSECdistr(dp3, family="ST", name="ST3", compNames=c("U", "V", "W"))
A <- matrix(c(1,-1,1, 3,0,-2), 3, 2)
new.st <- affineTransSECdistr(st3, a=c(-3,0), A=A)
#
st2 <- marginalSECdistr(st3, comp=c(3,1), name="2D marginal of ST3")
```

---

ais *Australian Institute of Sport data*

Description

Data on 102 male and 100 female athletes collected at the Australian Institute of Sport, courtesy of Richard Telford and Ross Cunningham.

Usage

```
data(ais)
```

Format

A data frame with 202 observations on the following 13 variables.

[,1]	`sex`	categorical, levels: `female`, `male`
[,2]	`sport`	categorical, levels: `B_Ball`, `Field`, `Gym`, `Netball`, `Row`, `Swim`, `T_400m`, `Tennis`, `T_Sprnt`, `W_Polo`
[,3]	`RCC`	red cell count (numeric)

[,4]	WCC	white cell count (numeric)
[,5]	Hc	Hematocrit (numeric)
[,6]	Hg	Hemoglobin (numeric)
[,7]	Fe	plasma ferritin concentration (numeric)
[,8]	BMI	body mass index, weight/(height)^2 (numeric)
[,9]	SSF	sum of skin folds (numeric)
[,10]	Bfat	body fat percentage (numeric)
[,11]	LBM	lean body mass (numeric)
[,12]	Ht	height, cm (numeric)
[,13]	Wt	weight, kg (numeric)

Details

The data have been made publicly available in connection with the book by Cook and Weisberg (1994).

References

Cook and Weisberg (1994), *An Introduction to Regression Graphics*. John Wiley & Sons, New York.

Examples

```
data(ais, package="sn")
pairs(ais[,c(3:4,10:13)], col=as.numeric(ais[,1]), main = "AIS data")
```

---

barolo *Price of Barolo wine*

Description

A data frame with prices of bottles of Barolo wine and some auxiliary variables

Usage

```
data(barolo)
```

Format

A data frame with 307 observations on five variables, as follows:

reseller	reseller code (factor with levels A, B, C, D)
vintage	vintage year (numeric)
volume	content volume in centilitres (numeric)
price	price in Euro (numeric)
age	age in 2010 (numeric)

For six items, vintage is NA's and so also age. Three items have a non-standard volume of 50 cl.

Details

The data have been obtained in July 2010 from the websites of four Italian wine resellers, selecting only quotations of Barolo wine, which is produced in the Piedmont region of Italy. The price does not include the delivery charge.

The data have been presented in Section 4.3.2 of the reference below, where a subset of them has been used for illustrative purposes. This subset refers to reseller "A" and bottles of 75cl.

Source

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

Examples

```
data(barolo)
attach(barolo)
f <- cut(age, c(0, 5,  6, 8, 11,  30))
table(volume, f)
plot(volume, price, col=as.numeric(f), pch=as.character(reseller))
legend(400, 990, col=1:5, lty=1, title="age class",
       legend=c("4-5", "6", "7-8", "9-11", "12-30"))
#
A75 <- (reseller=="A" & volume==75)
hist(log(price[A75],10), col="gray85")
# see Figure 4.7 of the source
```

---

coef.selm *Coefficients of objects created by* selm

Description

coef method for classes "selm" and "mselm".

Usage

```
## S4 method for signature 'selm'
coef(object, param.type = "CP", ...)
## S4 method for signature 'mselm'
coef(object, param.type = "CP", vector=TRUE, ...)
```

Arguments

object	an object of class "selm" or "mselm" as created by a call to function selm.
param.type	a character string which indicates the required type of parameter type; possible values are "CP" (default), "DP", "pseudo-CP" and their equivalent lower-case expressions.
vector	a logical value (default is TRUE) which selects a vector or a list format of the retuned value
...	not used, included for compatibility with the generic method

Value

a numeric vector or a list (the latter only for `mselm-class` objects if `vector=FALSE`)

Note

The possible options of `param.type` are described in the documentation of `dp2cp`; their corresponding outcomes differ by an additive constant only. With the `"CP"` option (that is, the 'centred parametrization'), the residuals are centred around 0, at least approximately; this is a reason for setting `"CP"` as the default option. For more information, see the 'Note' in the documentation of `summary.selm`.

Author(s)

Adelchi Azzalini

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

`dp2cp`, `summary.selm`, `selm` function, `selm-class`

Examples

```
data(wines, package="sn")
m5 <- selm(acidity ~ phenols + wine, family="SN", data=wines)
coef(m5)
coef(m5, "dp")
#
m12 <- selm(cbind(acidity, alcohol) ~ phenols + wine,  family="SN", data=wines)
coef(m12)
coef(m12, "DP", vector=FALSE)
```

---

conditionalSECdistr *Skew-normal conditional distribution*

Description

For a multivariate (extended) skew-normal distribution, compute its conditional distribution for given values of some of its components.

Usage

```
conditionalSECdistr(object, fixed.comp, fixed.values, name, drop = TRUE)
```

Arguments

object	an object of class SECdistrMv with family="SN" or family="ESN".
fixed.comp	a vector containing a subset of 1:d which selects the components whose values are to be fixed, if d denotes the dimensionality of the distribution.
fixed.values	a numeric vector of values taken on by the components fixed.comp; it must be of the same length of fixed.comp.
name	an optional character string with the name of the outcome distribution; if missing, one such string is constructed.
drop	logical (default=TRUE), to indicate whether the returned object must be of class SECdistrUv when length(fixed.comp)+1=d.

Details

For background information, see Section 5.3.2 of the reference below.

Value

an object of class SECdistrMv, except in the case when drop=TRUE operates, leading to an object of class SECdistrUv-class.

References

Azzalini, A. and Capitanio, A. (2014). *The Skew-normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

makeSECdistr, SECdistrMv-class, affineTransSECdistr

Examples

```
Omega <- diag(3) + outer(1:3,1:3)
sn <- makeSECdistr(dp=list(xi=rep(0,3), Omega=Omega, alpha=1:3), family="SN")
esn <- conditionalSECdistr(sn, fixed.comp=2, fixed.values=1.5)
show(esn)
```

---

confint.selm *Confidence intervals for parameters of a selm-class object*

Description

Computes confidence intervals for parameters in a selm-class object produces by selm fit when the response variable is univariate.

Usage

```
  ## S3 method for class 'selm'
confint(object, parm, level=0.95, param.type, tol=1e-3, ...)
```

Arguments

object	an object of class selm as produced by a call to function selm with univariate response.
parm	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. If missing, all parameters are considered.
level	the confidence level required (default value is 0.95).
param.type	a character string with the required parameterization; it must be either "CP" or "DP" or "pseudo-CP", or possibly their equivalent lowercase.
tol	the desired accuracy (convergence tolerance); this is a parameter passed to uniroot for computing the roots of the likelihood-based confidence interval for alpha.
...	not used, only there for compatibility reasons.

Details

A description of the methodology underlying confint.selm is provided in the technical note of Azzalini (2016). That document also explains why in certain cases an interval is not constructed and NA's are returned as endpoint.

Value

An invisible list whose components, described below, are partly different in the one- and the two-parameter cases.

call	the calling statement
<param1>	values of the first parameter
<param2>	values of the second parameter (in a two-parameter case)
logLik	numeric vector or matrix of the profile log-likelihood values
confint	in the one-parameter case, the confidence interval
level	in the one-parameter case, the confidence level
deviance.contour	in the two-parameter case, a list of lists whose elements identify each curve of the contour plot

Author(s)

Adelchi Azzalini

References

Azzalini, A. (2016). Derivation of various types of intervals from a selm object. Technical note distributed with the documentation of the R package sn in file selm-intervals.pdf within section 'User guide, package vignettes and other documentation'.

See Also

selm, summary.selm, profile.selm,

makeSECdistr for the CP/DP parameterizations,

uniroot for its tol argument

Examples

```
data(ais)
m1 <-  selm(log(Fe) ~ BMI + LBM, family = "sn", data = ais)
intervCP <- confint(m1)
intervDP <- confint(m1, param.type="DP")
confint(m1, parm=2:3)
confint(m1, parm=c("omega", "alpha"), param.type="DP")
```

---

convertCSN2SUNpar *Conversion of* CSN *parameters to* SUN *parameters*

Description

The parameter set of a Closed Skew-Normal (CSN) distribution is converted into the parameter set of the equivalent Unified Skew-Normal (SUN) distribution.

Usage

```
convertCSN2SUNpar(mu, Sigma, D, nu, Delta)
```

Arguments

mu	a numeric vector of length p, say.
Sigma	a positive definite variance matrix of size c(p,p).
D	an arbitrary numeric matrix of size say c(q, p), say.
nu	a numeric vector of length q.
Delta	a positive definite variance matrix of size c(q,q).

Details

The arguments of the function match the parameters $(\mu, \Sigma, D, \nu, \Delta)$ of the CSN distribution presented by González-Farías *et alii* (2004a, 2004b). These parameters are converted into those of the equivalent SUN distribution, which is unique. The converse operation, that is, mapping parameters from the SUN to the CSN family, is not handled here. Its solution would be non-unique, because the CSN family is over-parameterized.

Note that, having retained the exact notation of the above-quoted papers, there is a `Delta` argument which must not be confused with one of the arguments for the SUN distribution in `SUNdistr-base`. The coincidence of these names is entirely accidental.

The CSN parameters must only satisfy the requirements that Σ and Δ are symmetric positive definite matrices. Since these conditions are somewhat simpler to check than those for the SUN parameters, as indicated in `SUNdistr-base`, this function may provide a simple option for the specification of a CSN/SUN distribution.

The parameter list `dp` produced by this function can be used as an input for the functions in `SUNdistr-base` or for `makeSUNdistr`.

Value

a list representing the `dp` parameter set of the corresponding SUN distribution

Author(s)

Adelchi Azzalini

References

González-Farías, G., Domínguez-Molina, J. A., & Gupta, A. K. (2004a). Additive properties of skew normal random vectors. *J. Statist. Plann. Inference* **126**, 521-534.

González-Farías, G., Domínguez-Molina, J. A., & Gupta, A. K. (2004b). The closed skew-normal distribution. In M. G. Genton (Ed.), *Skew-elliptical Distributions and Their Applications: a Journey Beyond Normality*, Chapter 2, (pp. 25–42). Chapman & Hall/CRC.

See Also

`SUNdistr-base`, `makeSUNdistr`

Examples

```
p <- 3
q <- 2
mu <- 1:p
Sigma <- toeplitz(1/(1:p))
D <- matrix(sqrt(1:(p*q)), q, p)
nu <- 1/(1:q)
Delta <- diag(q) + outer(rep(1,q), rep(1,q))
dp <- convertCSN2SUNpar(mu, Sigma, D, nu, Delta)
```

convertSN2SUNdistr *Convert a SN distribution into a SUN*

Description

An object of SECdistrMv-class or SECdistrUv-class representing a SN or ESN distribution is converted into a SUNdistr-class object representing the same distribution.

Usage

```
convertSN2SUNdistr(object, HcompNames = "h", silent = FALSE)
```

Arguments

object	an object of SECdistrMv-class with family of type SN or ESN.
HcompNames	an optional character string for the hidden component
silent	a logical value which controls the behaviour if the supplied object is not suitable. If silent = FALSE (detault value) an error message is generated, otherwise a NULL is silently returned.

Value

an object of SUNdistr-class

Author(s)

Adelchi Azzalini

See Also

SUNdistr-class, SECdistrMv-class, SECdistrUv-class

Examples

```
esn <- makeSECdistr(dp=c(0, 1, 2, 0.5), family="ESN")
sun <- convertSN2SUNdistr(esn)
mean(sun) - mean(esn)
vcov(sun) - sd(esn)^2
#
dp0 <- list(xi=1:2, Omega=diag(3:4), alpha=c(3, -5))
f10 <- makeSECdistr(dp=dp0, family="SN", name="SN-2d", compNames=c("u1", "u2"))
sun10 <- convertSN2SUNdistr(f10)
mean(sun10) - mean(f10)
vcov(sun10) - vcov(f10)
```

dmsn *Multivariate skew-normal distribution*

Description

Probability density function, distribution function and random number generation for the multivariate skew-normal (SN) distribution.

Usage

```
dmsn(x, xi=rep(0,length(alpha)), Omega, alpha, tau=0, dp=NULL, log=FALSE)
pmsn(x, xi=rep(0,length(alpha)), Omega, alpha,  tau=0, dp=NULL, ...)
rmsn(n=1, xi=rep(0,length(alpha)), Omega, alpha,  tau=0, dp=NULL)
```

Arguments

x	either a vector of length d, where d=length(alpha), or a matrix with d columns, giving the coordinates of the point(s) where the density or the distribution function must be evaluated.
xi	a numeric vector of length d representing the location parameter of the distribution; see 'Background'. In a call to dmsn and pmsn, xi can be a matrix, whose rows represent a set of location parameters; in this case, its dimensions must match those of x.
Omega	a symmetric positive-definite matrix of dimension (d,d); see 'Background'.
alpha	a numeric vector which regulates the slant of the density; see 'Background'. Inf values in alpha are not allowed.
tau	a single value representing the 'hidden mean' parameter of the ESN distribution; tau=0 (default) corresponds to a SN distribution.
dp	a list with three elements, corresponding to xi, Omega and alpha described above; default value FALSE. If dp is assigned, individual parameters must not be specified.
n	a numeric value which represents the number of random vectors to be drawn.
log	logical (default value: FALSE); if TRUE, log-densities are returned.
...	additional parameters passed to pmnorm.

Details

Typical usages are

```
dmsn(x, xi=rep(0,length(alpha)), Omega, alpha, log=FALSE)
dmsn(x, dp=, log=FALSE)
pmsn(x, xi=rep(0,length(alpha)), Omega, alpha, ...)
pmsn(x, dp=)
rmsn(n=1, xi=rep(0,length(alpha)), Omega, alpha)
rmsn(n=1, dp=)
```

For efficiency reasons, rmsn makes very limited checks on the validity of the arguments. For instance, failure to positive definiteness of Omega would not be detected, and an uncontrolled crash occurs. Function pmsn makes use of pmnorm from package **mnormt**; the accuracy of its computation can be controlled via ...

Value

A vector of density values (dmsn) or of probabilities (pmsn) or a matrix of random points (rmsn).

Background

The multivariate skew-normal distribution is discussed by Azzalini and Dalla Valle (1996). The (Omega,alpha) parametrization adopted here is the one of Azzalini and Capitanio (1999). Chapter 5 of Azzalini and Capitanio (2014) provides an extensive account, including subsequent developments.

Notice that the location vector xi does not represent the mean vector of the distribution. Similarly, Omega is not *the* covariance matrix of the distribution, although it is *a* covariance matrix. Finally, the components of alpha are not equal to the slant parameters of the marginal distributions; to fix the marginal parameters at prescribed values, it is convenient to start from the OP parameterization, as illustrated in the 'Examples' below. Another option is to start from the CP parameterization, but notice that, at variance from the OP, not all CP sets are invertible to lend a DP set.

References

Azzalini, A. and Capitanio, A. (1999). Statistical applications of the multivariate skew normal distribution. *J.Roy.Statist.Soc. B* **61**, 579–602. Full-length version available at https://arXiv.org/abs/0911.2093

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

Azzalini, A. and Dalla Valle, A. (1996). The multivariate skew-normal distribution. *Biometrika* **83**, 715–726.

See Also

dsn, dmst, pmnorm, op2dp, cp2dp

Examples

```
x <- seq(-3,3,length=15)
xi <- c(0.5, -1)
Omega <- diag(2)
Omega[2,1] <- Omega[1,2] <- 0.5
alpha <- c(2,-6)
pdf <- dmsn(cbind(x, 2*x-1), xi, Omega, alpha)
cdf <- pmsn(cbind(x, 2*x-1), xi, Omega, alpha)
p1 <- pmsn(c(2,1), xi, Omega, alpha)
p2 <- pmsn(c(2,1), xi, Omega, alpha, abseps=1e-12, maxpts=10000)
#
rnd <- rmsn(10, xi, Omega, alpha)
#
```

```
# use OP parameters to fix marginal shapes at given lambda values:
op <- list(xi=c(0,1), Psi=matrix(c(2,2,2,3), 2, 2), lambda=c(5, -2))
rnd <- rmsn(10, dp=op2dp(op,"SN"))
#
# use CP parameters to fix mean vector, variance matrix and marginal skewness:
cp <- list(mean=c(0,0), var.cov=matrix(c(3,2,2,3)/3, 2, 2), gamma1=c(0.8, 0.4))
dp <- cp2dp(cp, "SN")
rnd <- rmsn(5, dp=dp)
```

---

dmst *Multivariate skew-t distribution and skew-Cauchy distribution*

Description

Probability density function, distribution function and random number generation for the multivariate skew-t (ST) and skew-Cauchy (SC) distributions.

Usage

```
dmst(x, xi=rep(0,length(alpha)), Omega, alpha, nu=Inf, dp=NULL, log=FALSE)
pmst(x, xi=rep(0,length(alpha)), Omega, alpha, nu=Inf, dp=NULL, ...)
rmst(n=1, xi=rep(0,length(alpha)), Omega, alpha, nu=Inf, dp=NULL)
dmsc(x, xi=rep(0,length(alpha)), Omega, alpha, dp=NULL, log=FALSE)
pmsc(x, xi=rep(0,length(alpha)), Omega, alpha, dp=NULL, ...)
rmsc(n=1, xi=rep(0,length(alpha)), Omega, alpha, dp=NULL)
```

Arguments

x	for dmst and dmsc, this is either a vector of length d, where d=length(alpha), or a matrix with d columns, representing the coordinates of the point(s) where the density must be avaluated; for pmst and pmsc, only a vector of length d is allowed.
xi	a numeric vector of length d representing the location parameter of the distribution; see 'Background'. In a call to dmst or dmsc, xi can be a matrix, whose rows represent a set of location parameters; in this case, its dimensions must match those of x.
Omega	a symmetric positive-definite matrix of dimension (d,d); see Section 'Background'.
alpha	a numeric vector of length d which regulates the slant of the density; see Section 'Background'. Inf values in alpha are not allowed.
nu	a positive value representing the degrees of freedom of ST distribution; does not need to be integer. Default value is nu=Inf which corresponds to the multivariate skew-normal distribution.
dp	a list with three elements named xi, Omega, alpha and nu, containing quantities as described above. If dp is specified, this prevents specification of the individual parameters.

n	a numeric value which represents the number of random vectors to be drawn; default value is 1.
log	logical (default value: FALSE); if TRUE, log-densities are returned.
...	additional parameters passed to pmt.

Details

Typical usages are

```
dmst(x, xi=rep(0,length(alpha)), Omega, alpha, nu=Inf, log=FALSE)
dmst(x, dp=, log=FALSE)
pmst(x, xi=rep(0,length(alpha)), Omega, alpha, nu=Inf, ...)
pmst(x, dp=, ...)
rmst(n=1, xi=rep(0,length(alpha)), Omega, alpha, nu=Inf)
rmst(n=1, dp=)
dmsc(x, xi=rep(0,length(alpha)), Omega, alpha, log=FALSE)
dmsc(x, dp=, log=FALSE)
pmsc(x, xi=rep(0,length(alpha)), Omega, alpha, ...)
pmsc(x, dp=, ...)
rmsc(n=1, xi=rep(0,length(alpha)), Omega, alpha)
rmsc(n=1, dp=)
```

For efficiency reasons, rmst, rmsc make very limited checks on the validity of the arguments. For instance, failure to positive definiteness of Omega would not be detected, and an uncontrolled crash occurs. Function pmst requires dmt from package **mnormt**; the accuracy of its computation can be controlled via argument ....

Value

A vector of density values (dmst and dmsc) or a single probability (pmst and pmsc) or a matrix of random points (rmst and rmsc).

Background

The family of multivariate ST distributions is an extension of the multivariate Student's t family, via the introduction of a alpha parameter which regulates asymmetry; when alpha=0, the skew-t distribution reduces to the commonly used form of multivariate Student's t. Further, location is regulated by xi and scale by Omega, when its diagonal terms are not all 1's. When nu=Inf the distribution reduces to the multivariate skew-normal one; see dmsn. Notice that the location vector xi does not represent the mean vector of the distribution (which in fact may not even exist if nu <= 1), and similarly Omega is not *the* covariance matrix of the distribution, although it is *a* covariance matrix. For additional information, see Section 6.2 of the reference below.

The family of multivariate SC distributions is the subset of the ST family, obtained when nu=1. While in the univariate case there are specialized functions for the SC distribution, dmsc, pmsc and rmsc simply make a call to dmst,pmst, rmst with argument nu set equal to 1.

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monograph series.

See Also

dst, dsc, dmsn, dmt, makeSECdistr

Examples

```
x <- seq(-4,4,length=15)
xi <- c(0.5, -1)
Omega <- diag(2)
Omega[2,1] <- Omega[1,2] <- 0.5
alpha <- c(2,2)
pdf <- dmst(cbind(x,2*x-1), xi, Omega, alpha, 5)
rnd <- rmst(10,  xi, Omega, alpha, 6)
p1 <- pmst(c(2,1), xi, Omega, alpha, nu=5)
p2 <- pmst(c(2,1), xi, Omega, alpha, nu=5, abseps=1e-12, maxpts=10000)
```

---

dp2cp *Conversion between parametrizations of a skew-elliptical distribution*

Description

Convert direct parameters (DP) to centred parameters (CP) of a skew-elliptical distribution and *vice versa*.

Usage

```
dp2cp(dp, family, object = NULL, cp.type = "proper", upto = NULL)
cp2dp(cp, family)
dp2op(dp, family)
op2dp(op, family)
```

Arguments

dp	a vector (in the univariate case) or a list (in the multivariate case) as described in makeSECdistr; see 'Background and Details' for an extented form of usage.
cp	a vector or a list, in agreement with dp as for type and dimension.
op	a vector or a list, in agreement with dp as for type and dimension.
family	a characther string with the family acronym, as described in makeSECdistr, except that family "ESN" is not implemented.
object	optionally, an S4 object of class SECdistrUv or SECdistrMv, as produced by makeSECdistr (default value: NULL). If this argument is not NULL, then family and dp must not be set.
cp.type	character string, which has effect only if family="ST" or "SC", otherwise a warning message is generated. Possible values are "proper", "pseudo", "auto", which correspond to the CP parameter set, their 'pseudo-CP' version and an automatic selection based on nu>4, where nu represents the degrees of freedom of the ST distribution.
upto	numeric value (in 1:length(dp), default=NULL) to select how many CP components are computed. Default value upto=NULL is equivalent to length(dp).

Value

For dp2cp, a matching vector (in the univariate case) or a list (in the multivariate case) of cp parameters. For cp2dp and op2dp, a similar object of dp parameters, provided the set of input parameters is in the admissible region. For dp2op, a similar set of op parameters.

Background

For a description of the DP parameters, see Section 'Details' of makeSECdistr. The CP form of parameterization is cumulant-based. For a univariate distribution, the CP components are the mean value (first cumulant), the standard deviation (square root of the 2nd cumulant), the coefficient of skewness (3rd standardized cumulant) and, for the ST, the coefficient of excess kurtosis (4th standardized cumulant). For a multivariate distribution, there exists an extension based on the same logic; its components represent the vector mean value, the variance matrix, the vector of marginal coefficients of skewness and, only for the ST, the Mardia's coefficient of excess kurtosis. The pseudo-CP variant provides an 'approximate form' of CP when not all required cumulants exist; however, this parameter set is not uniquely invertible to DP. The names of pseudo-CP components printed in summary output are composed by adding a ~ after the usual component name; for example, the first one is denoted mean~.

Additional information is provided by Azzalini and Capitanio (2014). Specifically, their Section 3.1.4 presents CP in the univariate SN case, Section 4.3.4 CP for the ST case and the 'pseudo-CP' version. Section 5.2.3 presents the multivariate extension for the SN distribution, Section 6.2.5 for the multivariate ST case. For a more detailed discussion, see Arellano-Valle & Azzalini (2013).

The OP parameterization is very similar to DP, from which it differs only for the components which regulate dispersion (or scatter) and slant. Its relevance lies essentially in the multivariate case, where the components of the slant parameter can be interpreted component-wise and remain unaffected if marginalization with respect to some other components is performed. In the multivariate SN case, the components of OP, denoted ξ, Ψ, λ, are associated to the expression of the density function (5.30) of Azzalini & Capitanio (2014); see pp.128–131 for more information. In the univariate case, the slant component of DP and the one of OP coincide, that is, $\alpha = \lambda$, Parameter ξ and other parameters which may exist with other families remain the same of the DP set. The term OP stands for 'original parameterization' since this is, up to a negligible difference, the parameterization adopted by Azzalini & Dalla Valle (1996).

Details

While any choice of the components of DP or OP is admissible, this is not true for CP. An implication is that a call to cp2dp may fail with an error message "non-admissible CP" for certain input values. The most extreme case is represented by the SC family, for which CP never exists; hence it makes to sense to call cp2dp with family="SC".

It is possible to call the functions with dp or cp having more components than those expected for a given family as described above and in makeSECdistr. In the univariate case, this means that dp or cp can be vectors of longer length than indicated earlier. This occurrence is interpreted in the sense that the additional components after the first one are regarded as regression coefficients of a selm model, and they are transferred unchanged to the matching components of the transformed parameter set; the motivation is given in Section 3.1.4 of Azzalini and Capitanio (2014). In the multivariate case, dp[[1]] and cp[[1]] can be matrices instead of vectors; the rows beyond the first one are transferred unchanged to cp[[1]] and dp[[1]], respectively.

References

Arellano-Valle, R. B. and Azzalini, A. (2013, available on-line 12 June 2011). The centred parameterization and related quantities of the skew-*t* distribution. *J. Multiv. Analysis* **113**, 73-90.

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

Azzalini, A. and Dalla Valle, A. (1996). The multivariate skew-normal distribution. *Biometrika* **83**, 715–726.

See Also

`makeSECdistr`, `summary.SECdistr`, `sn.cumulants`,

the 'Note' at `summary.selm` for the reason why CP is the default parameterization in that function and in related ones,

the 'Examples' at `rmsn` for use of the CP parameterization

Examples

```
# univariate case
cp <- dp2cp(c(1, 2222, 3333, 2, 3), "SN")
dp <- cp2dp(cp, "SN")
# notice that the 2nd and the 3rd component remain unchanged
#
# multivariate case
dp3 <- list(xi=1:3, Omega=toeplitz(1/(1:3)), alpha=c(-3, 8, 5), nu=6)
cp3 <- dp2cp(dp3, "ST")
dp3.back <- cp2dp(cp3, "ST")
#
op3 <- dp2op(dp3, "ST")
dp3back <- op2dp(op3,"ST")
```

---

dsc *Skew-Cauchy Distribution*

Description

Density function, distribution function, quantiles and random number generation for the skew-Cauchy (SC) distribution.

Usage

```
dsc(x, xi = 0, omega = 1, alpha = 0, dp = NULL, log = FALSE)
psc(x, xi = 0, omega = 1, alpha = 0, dp = NULL)
qsc(p, xi = 0, omega = 1, alpha = 0, dp = NULL)
rsc(n = 1, xi = 0, omega = 1, alpha = 0, dp = NULL)
```

Arguments

x	vector of quantiles. Missing values (NAs) and Inf's are allowed.
p	vector of probabilities. Missing values (NAs) are allowed.
xi	vector of location parameters.
omega	vector of (positive) scale parameters.
alpha	vector of slant parameters.
dp	a vector of length 3 whose elements represent the parameters described above. If dp is specified, the individual parameters cannot be set.
n	sample size.
log	logical flag used in dsc (default FALSE). When TRUE, the logarithm of the density values is returned.

Value

density (dsc), probability (psc), quantile (qsc) or random sample (rsc) from the skew-Cauchy distribution with given xi, omega and alpha parameters or from the extended skew-normal if tau!=0

Details

Typical usages are

```
dsc(x, xi=0, omega=1, alpha=0, log=FALSE)
dsc(x, dp=, log=FALSE)
psc(x, xi=0, omega=1, alpha=0)
psc(x, dp= )
qsc(p, xi=0, omega=1, alpha=0)
qsc(x, dp=)
rsc(n=1, xi=0, omega=1, alpha=0)
rsc(x, dp=)
```

Background

The skew-Cauchy distribution can be thought as a skew-t with tail-weight parameter nu=1. In this case, closed-form expressions of the distribution function and the quantile function have been obtained by Behboodian *et al.* (2006). The key facts are summarized in Complement 4.2 of Azzalini and Capitanio (2014). A multivariate version of the distribution exists.

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-normal and Related Families*. Cambridge University Press, IMS Monographs series.

Behboodian, J., Jamalizadeh, A., and Balakrishnan, N. (2006). A new class of skew-Cauchy distributions. *Statist. Probab. Lett.* **76**, 1488–1493.

See Also

dst, dmsc

Examples

```
pdf <- dsc(seq(-5,5,by=0.1), alpha=3)
cdf <- psc(seq(-5,5,by=0.1), alpha=3)
q <- qsc(seq(0.1,0.9,by=0.1), alpha=-2)
p <- psc(q, alpha=-2)
rn <- rsc(100, 5, 2, 5)
```

---

dsn *Skew-Normal Distribution*

Description

Density function, distribution function, quantiles and random number generation for the skew-normal (SN) and the extended skew-normal (ESN) distribution.

Usage

```
dsn(x, xi=0, omega=1, alpha=0, tau=0, dp=NULL, log=FALSE)
psn(x, xi=0, omega=1, alpha=0, tau=0, dp=NULL, engine, ...)
qsn(p, xi=0, omega=1, alpha=0, tau=0, dp=NULL, tol=1e-8, solver="NR", ...)
rsn(n=1, xi=0, omega=1, alpha=0, tau=0,  dp=NULL)
```

Arguments

x	vector of quantiles. Missing values (NA's) and Inf's are allowed.
p	vector of probabilities. Missing values (NA's) are allowed
xi	vector of location parameters.
omega	vector of scale parameters; must be positive.
alpha	vector of slant parameter(s); +/- Inf is allowed. For psn, it must be of length 1 if engine="T.Owen". For qsn, it must be of length 1.
tau	a single value representing the 'hidden mean' parameter of the ESN distribution; tau=0 (default) corresponds to a SN distribution.
dp	a vector of length 3 (in the SN case) or 4 (in the ESN case), whose components represent the individual parameters described above. If dp is specified, the individual parameters cannot be set.
n	a positive integer representing the sample size.
tol	a scalar value which regulates the accuracy of the result of qsn, measured on the probability scale.
log	logical flag used in dsn (default FALSE). When TRUE, the logarithm of the density values is returned.
engine	a character string which selects the computing engine; this is either "T.Owen" or "biv.nt.prob", the latter from package mnormt. If tau != 0 or length(alpha)>1, "biv.nt.prob" must be used. If this argument is missing, a default selection rule is applied.

solver	a character string which selects the numerical method used for solving the quantile equation; possible options are "NR" (default) and "RFB", described in the 'Details' section.
...	additional parameters passed to T.Owen

Value

density (dsn), probability (psn), quantile (qsn) or random sample (rsn) from the skew-normal distribution with given xi, omega and alpha parameters or from the extended skew-normal if tau!=0

Details

Typical usages are

```
dsn(x, xi=0, omega=1, alpha=0, log=FALSE)
dsn(x, dp=, log=FALSE)
psn(x, xi=0, omega=1, alpha=0,  ...)
psn(x, dp=,  ...)
qsn(p, xi=0, omega=1, alpha=0, tol=1e-8, ...)
qsn(x, dp=, ...)
rsn(n=1, xi=0, omega=1, alpha=0)
rsn(x, dp=)
```

psn and qsn make use of function T.Owen or biv.nt.prob

In qsn, the choice solver="NR" selects the Newton-Raphson method for solving the quantile equation, while option solver="RFB" alternates a step of *regula falsi* with one of bisection. The "NR" method is generally more efficient, but "RFB" is occasionally required in some problematic cases.

In version 1.6-2, the random number generation method for rsn has changed; the so-called transformation method (also referred to as the 'additive representation') has been adopted for all values of tau. Also, the code has been modified so that there is this form of consistency: provided set.seed() is reset similarly before calls, code like rsn(5, dp=1:3) and rsn(10, dp=1:3), for instance, will start with the same initial values in the longer sequence as in the shorter sequence.

Background

The family of skew-normal distributions is an extension of the normal family, via the introdution of a alpha parameter which regulates asymmetry; when alpha=0, the skew-normal distribution reduces to the normal one. The density function of the SN distribution in the 'normalized' case having xi=0 and omega=1 is $2\phi(x)\Phi(\alpha x)$, if ϕ and Φ denote the standard normal density and distribution function. An early discussion of the skew-normal distribution is given by Azzalini (1985); see Section 3.3 for the ESN variant, up to a slight difference in the parameterization.

An updated exposition is provided in Chapter 2 of Azzalini and Capitanio (2014); the ESN variant is presented Section 2.2. See Section 2.3 for an historical account. A multivariate version of the distribution is examined in Chapter 5.

References

Azzalini, A. (1985). A class of distributions which includes the normal ones. *Scand. J. Statist.* **12**, 171-178.

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

Functions used by psn: `T.Owen`, `biv.nt.prob`

Related distributions: `dmsn`, `dst`, `dmst`

Examples

```
pdf <- dsn(seq(-3, 3, by=0.1), alpha=3)
cdf <- psn(seq(-3, 3, by=0.1), alpha=3)
q <- qsn(seq(0.1, 0.9, by=0.1), alpha=-2)
r <- rsn(100, 5, 2, 5)
qsn(1/10^(1:4), 0, 1, 5, 3, solver="RFB")
```

---

dst *Skew-t Distribution*

Description

Density function, distribution function, quantiles and random number generation for the skew-t (ST) distribution.

Usage

```
dst(x, xi=0, omega=1, alpha=0, nu=Inf, dp=NULL, log=FALSE)
pst(x, xi=0, omega=1, alpha=0, nu=Inf, dp=NULL, method=0, lower.tail=TRUE,
    log.p=FALSE, ...)
qst(p, xi=0, omega=1, alpha=0, nu=Inf, tol=1e-08, dp=NULL, method=0, ...)
rst(n=1, xi=0, omega=1, alpha=0, nu=Inf, dp=NULL)
```

Arguments

x	vector of quantiles. Missing values (NAs) are allowed.
p	vector of probabililities.
xi	vector of location parameters.
omega	vector of scale parameters; must be positive.
alpha	vector of slant parameters. With pst and qst, it must be of length 1.
nu	a single positive value representing the degrees of freedom; it can be non-integer. Default value is nu=Inf which corresponds to the skew-normal distribution.

dp	a vector of length 4, whose elements represent location, scale (positive), slant and degrees of freedom, respectively. If dp is specified, the individual parameters cannot be set.
n	a positive integer representing the sample size.
log, log.p	logical; if TRUE, densities are given as log-densities and probabilities p are given as log(p)
tol	a scalar value which regulates the accuracy of the result of qsn, measured on the probability scale.
method	an integer value between 0 and 5 which selects the computing method; see 'Details' below for the meaning of these values. If method=0 (default value), an automatic choice is made among the four actual computing methods, depending on the other arguments.
lower.tail	logical; if TRUE (default), probabilities are $P\{X \leq x\}$, otherwise $P\{X \geq x\}$
...	additional parameters passed to integrate or pmst.

Value

Density (dst), probability (pst), quantiles (qst) and random sample (rst) from the skew-t distribution with given xi, omega, alpha and nu parameters.

Details

Typical usages are

```
dst(x, xi=0, omega=1, alpha=0, nu=Inf, log=FALSE)
dst(x, dp=, log=FALSE)
pst(x, xi=0, omega=1, alpha=0, nu=Inf, method=0, ...)
pst(x, dp=, log=FALSE)
qst(p, xi=0, omega=1, alpha=0, nu=Inf, tol=1e-8,  method=0, ...)
qst(x, dp=, log=FALSE)
rst(n=1, xi=0, omega=1, alpha=0, nu=Inf)
rst(x, dp=, log=FALSE)
```

Background

The family of skew-t distributions is an extension of the Student's t family, via the introduction of a alpha parameter which regulates skewness; when alpha=0, the skew-t distribution reduces to the usual Student's t distribution. When nu=Inf, it reduces to the skew-normal distribution. When nu=1, it reduces to a form of skew-Cauchy distribution. See Chapter 4 of Azzalini & Capitanio (2014) for additional information. A multivariate version of the distribution exists; see dmst.

Details

For evaluation of pst, and so indirectly of qst, four different methods are employed. In all the cases, the actual computations are performed for the normalized values z=(x-xi)/omega). Method 1 consists in using pmst with dimension d=1. Method 2 applies integrate to the density function dst. Method 3 again uses integrate too but with a different integrand, as given in Section 4.2 of Azzalini & Capitanio (2003, full version of the paper). Method 4 consists in the recursive

procedure of Jamalizadeh, Khosravi and Balakrishnan (2009), which is recalled in Complement 4.3 on Azzalini & Capitanio (2014); the recursion over nu starts from the explicit expression for nu=1 given by psc. Method 5 is targeted to tail probabilities only, and it returns NAs for non-extreme x values (those with abs(z)<=20); it is based on expressions given in Complement 4.4 of Azzalini and Capitanio (2014). Method 1 and 4 are only suitable for integer values of nu. Method 4 becomes progressively less efficient as nu increases, because the value of nu determines the number of nested calls, but the decay of efficiency is slower for larger values of length(x). If the default argument value method=0 is retained, an automatic choice among the above four methods is made, which depends on the values of nu, alpha, z. The numerical accuracy of methods 1, 2 and 3 can be regulated via the ... argument, while method 4 is conceptually exact, up to machine precision.

If qst is called with nu>1e4, the computation is transferred to qsn.

References

Azzalini, A. and Capitanio, A. (2003). Distributions generated by perturbation of symmetry with emphasis on a multivariate skew-*t* distribution. *J.Roy. Statist. Soc. B* **65**, 367–389. Full version of the paper at https://arXiv.org/abs/0911.2342.

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-normal and Related Families*. Cambridge University Press, IMS Monographs series.

Jamalizadeh, A., Khosravi, M., and Balakrishnan, N. (2009). Recurrence relations for distributions of a skew-*t* and a linear combination of order statistics from a bivariate-*t*. *Comp. Statist. Data An.* **53**, 847–852.

See Also

dmst, dsn, dsc

Examples

```
pdf <- dst(seq(-4, 4, by=0.1), alpha=3, nu=5)
rnd <- rst(100, 5, 2, -5, 8)
q <- qst(c(0.25, 0.50, 0.75), alpha=3, nu=5)
pst(q, alpha=3, nu=5)  # must give back c(0.25, 0.50, 0.75)
#
p1 <- pst(x=seq(-3,3, by=1), dp=c(0,1,pi, 3.5))
p2 <- pst(x=seq(-3,3, by=1), dp=c(0,1,pi, 3.5), method=2, rel.tol=1e-9)
```

---

extractSECdistr *Extract the SEC error distribution from an object created by* selm

Description

Given an object created by a call to selm, the function delivers the SEC distribution representing the stochastic term of the fitted model

Usage

```
extractSECdistr(object, name, compNames)
```

Arguments

object	an object of class selm or mselm, as created by selm.
name	an optional character string representing the name of the outcome distribution; if missing, a string is constructed from the object ingredients.
compNames	in the multivariate case, an optional vector of character strings with the names of the components of the error distribution; if missing, one such vector is constructed from the object ingredients.

Value

An object of class SECdistrMv or SECdistrUv, depending of the class of object.

Details

When the formula of the fitted model includes only the constant 1, the returned object represents the fitted SEC distribution. If the formula includes additional terms, the linear predictor is eliminated and the returned object corresponds to the error term of the model; hence the location parameter xi in the DP parameterization is set to zero.

The returned object can be submitted to tools available for objects created by makeSECdistr, such as summary.SECdistr, conditionalSECdistr and and so on.

See Also

selm, makeSECdistr

Examples

```
data(ais)
m2 <- selm(log(Fe) ~ 1, family="ST", data=ais, fixed=list(nu=8))
f2 <- extractSECdistr(m2)
show(f2)
#
m4 <- selm(cbind(BMI, LBM) ~ 1, family="SN", data=ais)
f4 <- extractSECdistr(m4)
mean(f4)
vcov(f4)
```

---

fitdistr.grouped *Maximum-likelihood fitting of a univariate distribution from grouped data*

Description

Maximum-likelihood fitting of a univariate distribution when the data are represented by a set of frequencies pertaining to given set of contiguous intervals.

Usage

```
fitdistr.grouped(breaks, counts, family, weights, trace = FALSE, wpar = NULL)
```

Arguments

breaks	A numeric vector of strictly increasing values which identify a set of contiguous intervals on the real line. See 'Details' for additional information.
counts	A vector of non-negative integers representing the number of observations falling in the intervals specified by breaks; it is then required that length(counts)+1=length(breaks).
family	A character string specifying the parametric family of distributions to be used for fitted. Admissible names are: "normal", "logistic", "t", "Cauchy", "SN", "ST", "SC", "gamma", "Weibull"; the names "gaussian" and "Gaussian" are also allowed, and are converted to "normal".
weights	An alias for counts, allowed for analogy with the selm function.
trace	A logical value which indicates whether intermediate evaluations of the optimization process are printed (default: FALSE).
wpar	An optional vector with initial values of the 'working parameters' for starting the maximization of the log-likelihood function; see 'Details' for their description.

Details

The original motivation of this function was fitting a univariate SN, ST or SC distribution from grouped data; its scope was later extended to include some other continuous distributions. The adopted name of the function reflects the broad similarity of its purpose with the one of fitdistr, but there are substantial differences in the actual working of the two functions.

The parameter set of a given family is the same as appearing in the corresponding d<basename> function, with the exception of the "t" distribution, for which a location and a scale parameter are included, besides df.

The range of breaks does not need to span the whole support of the chosen family of distributions, that is, (0, Inf) for "Weibull" and "gamma" families, (-Inf, Inf) for the other families. In fact, for the purpose of post-fitting plotting, an infinite range(breaks) represents a complication, requiring an *ad hoc* handling; so it is sensible to avoid it. However, at the maximum-likelihood fitting stage, the full support of the probability distribution is considered, with the following implications. If max(breaks)=xR, say, and xR<Inf, then an additional interval (xR, Inf) is introduced, with value counts=0 assigned. A similar action is taken at the lower end: if min(breaks)=xL is larger than the infimum of the support of the distribution, an extra 0-counts interval is introduced as (0, xL) or (-Inf, xL), depending on the support of the family.

Maximum likelihood fitting is obtained by maximizing the pertaining multinomial log-likelihood using the optim function with method="Nelder-Mead". For numerical convenience, the numerical search is performed using 'working parameters' in place of the original ones, with reverse conversion at the end. The working parameters coincide with the original distribution parameters when they have unbounded range, while they are log-transformed in case of intrinsically positive parameters. This transformation applies to the parameters of the positive-valued distributions ("gamma" and "Weibull"), all scale parameters and df of the "t" distribution.

Value

An object of class `fitdistr.grouped`, whose components are described in `fitdistr.grouped-class`.

Author(s)

Adelchi Azzalini

See Also

For methods pertaining to this class of objects, see `fitdistr.grouped-class` and `plot.fitdistr.grouped`; see also `dsn`, `dst`, `dsc`, `Distributions`, `dmultinom`; see also `selm` for ungrouped data fitting and an example elaborated on below.

Examples

```
data(barolo)
attach(barolo)
A75 <- (reseller=="A" & volume==75)
logPrice <- log(price[A75], 10) # used in documentation of 'selm'; see its fitting
detach(barolo)
breaks<- seq(1, 3, by=0.25)
f <- cut(logPrice, breaks = breaks)
counts <- tabulate(f, length(levels(f)))
logPrice.grouped <- fitdistr.grouped(breaks, counts, family='ST')
summary(logPrice.grouped) # compare this fit with the ungrouped data fitting
```

---

fitdistr.grouped-class

Methods for objects of class created by `fitdistr.grouped`

Description

A successful call to function `fitdistr.grouped` creates an object of class, also named `fitdistr.grouped`, for which a set of methods exist. The structure of an object of this class is described in section 'Object components'.

Usage

```
  ## S3 method for class 'fitdistr.grouped'
logLik(object, ...)
  ## S3 method for class 'fitdistr.grouped'
coef(object, ...)
  ## S3 method for class 'fitdistr.grouped'
vcov(object, ...)
  ## S3 method for class 'fitdistr.grouped'
print(x, ...)
  ## S3 method for class 'fitdistr.grouped'
summary(object, cor=FALSE, ...)
```

```
  ## S3 method for class 'fitdistr.grouped'
fitted(object, full=FALSE, ...)
```

Arguments

x, object	an object of class fitdistr.grouped as created by a call to the function with this name.
cor	logical (default=FALSE); is the correlation matrix required?
full	logical (default=FALSE); must the vector of fitted frequencies include the boundary classes, when they are added to cover the full support of the fitted distribution?
...	further arguments passed to or from other methods.

Object components

Components of an object of class fitdistr.grouped:

call the matched call

family the selected family of distributions

logL the achieved maximum log-likelihood

param a vector of estimated parameters

vcov the approximate variance-covariance matrix of the estimates

input a list with the input quantities and some derived ones

opt a list as returned by optim

Author(s)

Adelchi Azzalini

References

Possolo, A., Merkatas, C. and Bodnar, O. (2019). Asymmetrical uncertainties. *Metrologia* 56, 045009.

See Also

the function fitdistr.grouped, the plotting method plot.fitdistr.grouped

Examples

```
data(barolo)
attach(barolo)
A75 <- (reseller=="A" & volume==75)
logPrice <- log(price[A75], 10) # as used in selm documentation; see its fitting
detach(barolo)
breaks <- seq(1, 3, by=0.25)
f <- cut(logPrice, breaks = breaks)
counts <- tabulate(f, length(levels(f)))
```

```
fit.logPrice.gr <- fitdistr.grouped(breaks, counts, family='ST')
summary(fit.logPrice.gr) # compare this fit with the ungrouped data fitting
print(fit.logPrice.gr)
coef(fit.logPrice.gr)
vcov(fit.logPrice.gr)
data.frame(intervals=levels(f), counts, fitted=format(fitted(fit.logPrice.gr)))
full.intervals <- c("(-Inf, 1]", levels(f), "(3, Inf)")
data.frame("full-range intervals" = full.intervals,
       "full-range counts" = c(0, counts, 0),
      "full-range fit" = fitted(fit.logPrice.gr, full=TRUE))
sum(counts) - sum(fitted(fit.logPrice.gr))
sum(counts) - sum(fitted(fit.logPrice.gr, full=TRUE))  # must be "nearly 0"
#---
# Use first entry in Table 3 of Possolo et al. (2019) and do similar fitting
# to the *probability* values, not observation counts
afcrc59 <- 1.141
breaks <- c(-Inf, afcrc59 - 0.033, afcrc59, afcrc59 + 0.037, Inf)
prob <-  c(0.16, 0.50, 0.84)
cum.percent <- c(0, prob, 1)*100
fitSN <- fitdistr.grouped(breaks, counts=diff(cum.percent), family="SN")
print(coef(fitSN))
print(rbind(target=c(prob, 1)*100, fitted=cumsum(fitted(fitSN))), digits=5)
# Note: given the nature of these data (i.e. probabilities, not counts),
#       there is no point to use vcov, logLik and summary on the fitted object.
```

`fournum` *Four-number summary of a numeric vector*

Description

Returns a quantile-based four-number summary of the input data

Usage

```
fournum(x, na.rm = TRUE, ...)
```

Arguments

`x`	a numeric vector, maybe including `NA`s and `+/-Inf`'s. At least 8 not-`NA` values are required. It works with objects which can be coerced to vector.
`na.rm`	logical; if `TRUE`, all `NA` and `NaN`s are dropped, before the statistics are computed.
`...`	optional arguments passed to `quantile`

Details

Function `quantile` is used to compute 7 octiles of `x`, that is, quantiles of level `(1:7)/8`, denoted `oct[1:7]`, and derive four summary quantities:

1. the median, which corresponds to `oct[4]`,

2. the '(coefficient of) quartile deviation' or semi-interquantile range: `(oct[6] - oct[2])/2`;
3. the Galton-Bowley measure of asymmetry, that is, skewness: `(oct[6] - 2 * oct[4] + oct[2])/(oct[6] - oct[2])`;
4. the Moors measure of kurtosis: `(oct[7] - oct[5] + oct[3] - oct[1])/(oct[6] - oct[2])`

The term 'coefficient of quartile deviation' is adopted from the Encyclopedia of Statistical Sciences; see the references below. What is called Galton-Bowley measure here is often named 'Bowley's measure', but some sources attribute it to Francis Galton. For the Moors measure, see Moors (1988).

Value

a vector of length four containing the median, the quartile deviation, the Galton-Bowley measure and the Moors measure. However, if `x` does not contain at least 8 values (after removing `NA`s), `rep(NA,4)` is returned.

Note

Computation of octiles makes real sense only if `length(x)` is substantially larger than 8.

Author(s)

Adelchi Azzalini

References

'Quartile deviation, coefficient of', in: *Encyclopedia of Statistical Sciences*, 2nd edition (2006). Editors: Samuel Kotz (Editor-in-Chief), Campbell B. Read, N. Balakrishnan, Brani Vidakovic. Volume 10, p.6743.

'Skewness, Bowleys's measures of', in: *Encyclopedia of Statistical Sciences*, 2nd edition (2006). Editors: Samuel Kotz (Editor-in-Chief), Campbell B. Read, N. Balakrishnan, Brani Vidakovic. Volume 12, p.7771-7773.

Moors, J.J.A. (1988). A quantile alternative for kurtosis. *Source: Journal of the Royal Statistical Society. Series D (The Statistician)*, Vol. 37, pp. 25-32

See Also

`quantile`, `fivenum`, `IQR`

Examples

```
fournum(datasets::rivers)
```

frontier *Simulated sample from a skew-normal distribution*

Description

A sample simulated from the SN(0,1,5) distribution with sample coefficient of skewness inside the admissible range (-0.9952719, 0.9952719) for the skew-normal family but maximum likelihood estimate on the frontier of the parameter space.

Usage

```
data(frontier)
```

Format

A numeric vector of length 50.

Source

Generated by a run of `rsn(50, 0, 1, 5)`.

Examples

```
data(frontier, package="sn")
fit <- selm(frontier ~ 1)
plot(fit, which=2)
#
fit.p <- selm(frontier ~ 1, method="MPLE")
plot(fit.p, which=2)
```

galton_moors2alpha_nu *Mapping of the (Galton-Bowley, Moors) measures to the (alpha, nu) parameters of a ST distribution*

Description

Given a pair of (Galton-Bowley, Moors) measures of skewness and kurtosis for a given sample, `galton_moors2alpha_nu` delivers values ($alpha$, nu) such that a skew-*t* (ST) distribution with these slant and tail-weight parameter has its (Galton-Bowley, Moors) measures equal to the input values. Its simplified version `galton2alpha` uses only a Galton-Bowley measure to deliver a $alpha$ value, assuming a SN distribution. These functions are mainly intended for internal package usage.

Usage

```
galton_moors2alpha_nu(galton, moors, quick = TRUE, move.in = TRUE, verbose = 0,
  abstol = 1e-04)
galton2alpha(galton, move.in = TRUE)
```

Arguments

galton	a numeric value, representing a Galton-Bowley measure
moors	a numeric value, representing a Moors measure
quick	a logical value; if TRUE, a quick mapping is performed
move.in	if the input values (galton, moors) are outside the feasible ST region, a suitable point within the feasible area is returned
verbose	a numeric value which regulates the amount of printed detail
abstol	the tolerance value of the mapping, only relevant is quick=FALSE

Details

For background information about the Galton-Bowley's and the Moors measures, see the documentation of fournum. The working of the mapping by described in Azzalini and Salehi (2020).

Value

for galton_moors2alpha_nu, named vector of length two, with one or more descriptive attributes; for galton2alpha, a single alpha value.

Note

These functions are mainly intended for internal package usage. Specifically they are used by st.prelimFit.

Author(s)

Adelchi Azzalini

References

Azzalini, A. and Salehi, M. (2020). Some computational aspects of maximum likelihood estimation of the skew-*t* distribution. In: *Computational and Methodological Statistics and Biostatistics*, edited by Andriëtte Bekker, Ding-Geng Chen and Johannes T. Ferreira. Springer. DOI: 10.1007/978-3-030-42196-0

See Also

fournum, st.prelimFit

Examples

```
galton_moors2alpha_nu(0.5, 3, quick=FALSE)  # input in the feasible area
galton_moors2alpha_nu(0.5, 3)    # very similar output, much more quickly
galton_moors2alpha_nu(0.5, 0.5)  # input outside the feasible area
```

`makeSECdistr` *Build a skew-elliptically contoured distribution*

Description

Build an object which identifies a skew-elliptically contoured distribution (SEC), in the univariate and in the multivariate case. The term 'skew-elliptical distribution' is a synonym of SEC distribution.

Usage

```
makeSECdistr(dp, family, name, compNames)
```

Arguments

`dp`	a numeric vector (in the univariate case) or a list (in the multivariate case) of parameters which identify the specific distribution within the named `family`. See 'Details' for their expected structure.
`family`	a character string which identifies the parametric family; currently, possible values are: `"SN"`, `"ESN"`, `"ST"`, `"SC"`. See 'Details' for additional information.
`name`	an optional character string with the name of the distribution. If missing, one is created.
`compNames`	in the multivariate case, an optional vector of character strings with the names of the component variables; its length must be equal to the dimensionality of the distribution being generated. If missing and the first component of `dp` is a named vector, its names are used as `compNames`; otherwise the components are named `"V1"`, `"V2"`, ...

Details

If `dp` is a numeric vector, a univariate distribution is built. Alternatively, if `dp` is a list, a multivariate distribution is built. In both cases, the required number of components of `dp` depends on `family`: it must be 3 for `"SN"` and `"SC"`; it must be 4 for `"ESN"` and `"ST"`.

In the univariate case, the first three components of `dp` represent what for the specific distributions are denoted `xi` (location), `omega` (scale, positive) and `alpha` (slant); see functions `dsn`, `dst`, `dsc` for their description. The fourth component, when it exists, represents either `tau` (hidden variable mean) for `"ESN"` or `nu` (degrees of freedom) for `"ST"`. The names of the individual parameters are attached to the components of `dp` in the returned object.

In the multivariate case, `dp` is a list with components having similar role as in the univariate case, but `xi=dp[[1]]` and `alpha=dp[[3]]` are now vectors and the scale parameter `Omega=dp[[2]]` is a symmetric positive-definite matrix. For a multivariate distribution of dimension 1 (which can be created, although a warning message is issued), `Omega` corresponds to the square of `omega` in the univariate case. Vectors `xi` and `alpha` must be of length `ncol(Omega)`. See also functions `dmsn`, `dmst` and `dmsc`. The fourth component, when it exists, is a scalar with the same role as in the univariate case.

In the univariate case `alpha=Inf` is allowed, but in the multivariate case all components of the vector `alpha` must be finite.

An object built by this function operates according to the S4 protocol.

Value

In the univariate case, an object of class `SECdistrUv`; in the multivariate case, an object of class `SECdistrMv`. See `SECdistrUv-class` and `SECdistrMv-class` for their description.

Background

For background information, see Azzalini and Capitanio (2014), specifically Chapters 2 and 4 for univariate cases, Chapters 5 and 6 for multivariate cases; Section 6.1 provides a general formulation of SEC distributions.

If the slant parameter `alpha` is `0` (or a vector of `0`'s, in the multivariate case), the distribution is of classical elliptical type.

The ESN distribution is included here as a members of the SEC class, with a very slight extension of the original definition of this class, since the only difference is the non-zero truncation point of the unobserved component of the `(d+1)`-dimensional EC variable.

Author(s)

Adelchi Azzalini

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

The description of classes `SECdistrUv-class` and `SECdistrMv-class`

`plot.SECdistr` for plotting and `summary.SECdistr` for summaries

Related functions `dsn`, `dst`, `dsc`, `dmsn`, `dmst`, `dp2cp`

Functions `affineTransSECdistr` and `conditionalSECdistr` to manipulate objects of class `SECdistrMv-class`

Function `extractSECdistr` to extract objects of class `SECdistrMv-class` and `SECdistrUv-class` representing the SEC distribution of a `selm` fit

Examples

```
f1 <- makeSECdistr(dp=c(3,2,5), family="SN", name="First-SN")
show(f1)
summary(f1)
plot(f1)
plot(f1, probs=c(0.1, 0.9))
#
f2 <- makeSECdistr(dp=c(3, 5, -4, 8), family="ST", name="First-ST")
f9 <- makeSECdistr(dp=c(5, 1, Inf, 0.5), family="ESN", name="ESN,alpha=Inf")
#
dp0 <- list(xi=1:2, Omega=diag(3:4), alpha=c(3, -5))
f10 <- makeSECdistr(dp=dp0, family="SN", name="SN-2d", compNames=c("u1", "u2"))
```

```
#
dp1 <- list(xi=1:2, Omega=diag(1:2)+outer(c(3,3),c(2,2)), alpha=c(-3, 5), nu=6)
f11 <- makeSECdistr(dp=dp1, family="ST", name="ST-2d", compNames=c("t1", "t2"))
```

`makeSUNdistr` *Build an object representing a SUN distribution*

Description

Build an object which identifies a Unified Skew-Normal distribution (SUN) within this parametric family. The SUN family is essentially equivalent to some other parametric families examined in the literature, notably the Closed Skew-Normal.

Usage

```
makeSUNdistr(dp, name, compNames, HcompNames, drop = TRUE)
```

Arguments

`dp`	a list of parameters as described at `SUNdistr-base`.
`name`	an optional character string with the name of the distribution. If missing, one is created.
`compNames`	an optional vector of character strings with the names of the component variables; its length must be equal to the dimensionality `d` of the distribution being generated. If missing, the components are named `"V1"`, `"V2"`, ...
`HcompNames`	an optional vector of character strings with the names of the hidden component variables; its length must be equal to the dimensionality component `m` described in the 'Details'. If missing, the components are named `"H1"`, `"H2"`, ...
`drop`	a logical value (default: `TRUE`) relevant only in the case `m=1`. When both `m=1` and `drop=TRUE`, the returned object is of class either `SECdistrUv` or `SECdistrMv`, depending on the value of `d`, and family `"SN"` or `"ESN"`, depending on the `dp` ingredients.

Details

The argument `dp` is a list, whose components are described at `SUNdistr-base`; see especially the 'Details' there. In this respect, there is no difference between the univariate and the univariate case, differently from the similar command `makeSECdistr`.

If the arguments `name`, `compNames` and `HcompNames` are missing, they are composed from the supplied arguments.

A `SUNdistr-class` object operates according to the S4 protocol.

Value

An object of `SUNdistr-class`

Note

The present structure and user interface of this function, and of other ones related to the SUN distribution, must be considered experimental, and they might possibly change in the future.

Author(s)

Adelchi Azzalini

See Also

Basic information on the SUN distribution SUNdistr-base, the description of the class SUNdistr-class,

Related methods: show.SUNdistr for displaying the object constituents, plot.SUNdistr for plotting, mean.SUNdistr for the mean value, vcov.SUNdistr for the variance matrix, summary.SUNdistr for various summary quantities

Functions SUNdistr-op manipulate objects created by this function, producing new SUNdistr-class objects

Examples

```
xi <- c(1, 0, -1)
Omega <- matrix(c(2,1,1, 1,3,1, 1,1,4), 3, 3)
Delta <- matrix(c(0.72,0.20, 0.51,0.42, 0.88, 0.94), 3, 2, byrow=TRUE)
Gamma <- matrix(c(1, 0.8, 0.8, 1), 2, 2)
dp3 <- list(xi=xi, Omega=Omega, Delta=Delta, tau=c(-0.5, 0), Gamma=Gamma)
sun3 <- makeSUNdistr(dp=dp3, name="SUN3", compNames=c("x", "w", "z"))
show(sun3)
```

---

matrix-op *vech, tr and other matrix operators*

Description

vech and other matrix operators

Usage

```
vech(A)
vech2mat(v)
duplicationMatrix(n)
tr(x)
blockDiag(...)
```

Arguments

A	a (symmetric) square numeric matrix.
v	a numeric vector such that length(v)=n*(n+1)/2 for some positive integer n.
n	a positive integer number; default is n=1.
x	a square numeric matrix.
...	an abitrary numer of matrices or objects coercible into matrices.

Value

a vector in case of vech, a scalar in case of tr, otherwise a matrix.

Details

For a square matrix A, vech(A) returns the vector formed by the lower triangular portion of the matrix, including the diagonal; usually, this only makes sense for a symmetric matrix of numeric values. If v=vech(M) where M is a symmetric numeric matrix, vech2mat(v) performs the inverse operation and returns the original matrix M; this explain the requirement on length(v).

For a positive integer n, D=duplicationMatrix(n) is a matrix of dimension (n^2, n*(n+1)/2) such that D %*% vech(M) returns the vec-form of a symmetric matrix M of order n, that is, the vector which stacks the columns of M; for more information, see Section 3.8 of Magnus and Neudecker (1988).

For a square numeric matrix x, tr(x) returns its trace.

blockDiag(...) creates a block-diagonal matrix from a set of matrices or objects coercible into matrices. Generally, this is useful only for numeric objects.

Author

Adelchi Azzalini; the original Octave code of duplicationMatrix is by Kurt Hornik.

References

Magnus, Jan R. and Neudecker, Heinz (1988). *Matrix differential calculus with application in statistics and econometrics*. Wiley series in probability and statistics.

Examples

```
M <- toeplitz(1:4)
v <- vech(M)
vech2mat(v) - M
D <- duplicationMatrix(ncol(M))
# D %*% vech(M) - as.vector(M), must be a one-column matrix of 0s
tr(outer(1:4,2:5))
blockDiag(M[1:2,], 1:2, diag(5:6))
```

modeSECdistr *The mode of a skew-elliptically contoured (*SEC*) distribution*

Description

Compute compute the mode of a univariate or multivariate SEC distribution.

Usage

```
modeSECdistr(dp, family, object=NULL)
```

Arguments

dp	a numeric vector (in the univariate case, for class SECdistrUv) or a list (in the multivariate case, , for class SECdistrUv) of parameters which identify the specific distribution within the named family.
family	a character string which identifies the parametric family among those admissible for classes SECdistrUv or SECdistrMv.
object	an object of class SECdistrUv or SECdistrMv as created by makeSECdistr or extractSECdistr; if this argument is used, arguments dp and family must not be set, and *vice versa*.

Value

a numeric vector

Background

The mode is obtained through numerical maximization. In the multivariate case, the problem is reduced to a one-dimensional search using Propositions 5.14 and 6.2 of the reference below.

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

makeSECdistr and extractSECdistr for additional information and for constructing a suitable object,

SECdistrUv-class and SECdistrMv-class for methods mean and vcov which compute the mean (vector) and the variance (matrix) of the object distribution

Examples

```
dp3 <- list(xi=1:3, Omega=toeplitz(1/(1:3)), alpha=c(3,-1,2), nu=5)
st3 <- makeSECdistr(dp3, family="ST", name="ST3", compNames=c("U", "V", "W"))
mode1 <- modeSECdistr(dp3, "ST")
mode2 <- modeSECdistr(object=st3) # the same of mode1
```

`overview-sn` *Package* **sn***: overview of the structure and main commands*

Description

The package provides facilities to build and manipulate probability distributions of the skew-normal (SN) and some related families, notably the skew-t (ST) and the 'unified skew-normal' (SUN) families. For the SN, ST and skew-Cauchy (SC) families, statistical methods are made available for data fitting and model diagnostics, in the univariate and the multivariate case.

Two main sides

The package comprises two main sides: one side provides facilities for the pertaining probability distributions; the other one deals with related statistical methods. Underlying formulation, parameterizations of distributions and terminology are in agreement with the monograph of Azzalini and Capitanio (2014).

Probability side

There are two layers of support for the probability distributions of interest. At the basic level, there exist functions which follow the classical R scheme for distributions. In addition, there exists facilities to build an object which incapsulates a probability distribution and then certain operations can be be performed on such an object; these probability objects operate according to the S4 protocol. The two schemes are described next.

Classical R scheme The following functions work similary to `{d,p,q,r}norm` and other R functions for probability distributions:

- skew-normal (SN): functions `{d,p,q,r}sn` for the univariate case, functions `{d,p,r}msn` for the multivariate case, where in both cases the 'Extended skew-normal' (ESN) variant form is included;
- skew-t (ST): functions `{d,p,q,r}st` for the univariate case, functions `{d,p,r}mst` for the multivariate case;
- skew-Cauchy (SC): functions `{d,p,q,r}sc` for the univariate case, functions `{d,p,r}msc` for the multivariate case.

In addition to the usual specification of their parameters as a sequence of individual components, a parameter set can be specified as a single `dp` entity, namely a vector in the univariate case, a list in the multivariate case; `dp` stands for 'Direct Parameters' (DP).

Another set of parameters is in use, denoted Centred Parameters (CP), which are more convenient for interpretability, since they correspond to familiar quantifies, such as the mean and the standard deviation. Conversion from the `dp` parameter set to the corresponding CP parameters

can be accomplished using the function dp2cp, while function cp2dp performs the inverse transformation.

The SUN family is mainly targeted to the multivariate context, and this is reflected in the organization of the pertaining functions, although univariate SUN distributions are supported. Density, distribution function and random numbers are handled by {d,p,r}sun. Mean value, variance matrix and Mardia's measures of multivariate skewness and kurtosis are computed by sun{Mean,Vcov,Mardia}.

In addition, one can introduce a user-specified density function using dSymmModulated and dmSymmModulated, in the univariate and the multivariate case, respectively. These densities are of the 'symmetry-modulated' type, also called 'skew-symmetric', where one can specify the base density and the modulation factor with high degree of flexibility. Random numbers can be sampled using the corresponding functions rSymmModulated and rmSymmModulated. In the bivariate case, a dedicated plotting function exists.

Probability distribution objects: SEC families Function makeSECdistr can be used to build a 'SEC distribution' object representing a member of a specified parametric family (among the types SN, ESN, ST, SC) with a given dp parameter set. This object can be used for various operations such as plotting or extraction of moments and other summary quantities. Another way of constructing a SEC distribution object is via extractSECdistr which extracts suitable components of an object produced by function selm to be described below.

Additional operations on these objects are possible in the multivariate case, namely marginalSECdistr and affineTransSECdistr for marginalization and affine trasformations. For the multivariate SN family only (but including ESN), conditionalSECdistr performs a conditioning on the values taken on by some components of the multivariate variable.

Probability distribution objects: the SUN family Function makeSUNdistr can be used to build a SUN distribution object representing a member of the SUN parametric family. This object can be used for various operations such as plotting or extraction of moments and other summary quantities.

Moreover there are several trasformation operations which can be performed on a SUN distribution object, or two such objects in some cases: computing a (multivariate) marginal distribution, a conditional distribution (on given values of some components or on one-sided intervals), an affine trasformation, a convolution (that is, the distribution of the sum of two independent variables), and joining two distributions under assumption of independence.

Statistics side

The main function for data fitting is represented by selm, which allows to specify a linear regression model for the location parameter, similarly to function lm, but assuming a *skew-elliptical* distribution of the random component; this explains the name *selm=(se+lm)*. Allowed types of distributions are SN (but not ESN), ST and SC. The fitted distribution is univariate or multivariate, depending on the nature of the response variable of the posited regression model. The model fitting method is either maximum likelihood or maximum penalized likelihood; the latter option effectively allows the introduction of a prior distribution on the slant parameter of the error distribution, hence leading to a 'maximum a posteriori' estimate.

Once the fitting process has been accomplished, an object of class either *selm* (for univariate response) or *mselm* (for multivariate response) is produced. A number of 'methods' are available for these objects: show, plot, summary, coef, residuals, logLik and others. For univariate *selm*-class objects, univariate and bivariate profile log-likelihood functions can be obtained; a predict

method also exists. These methods are built following the S4 protocol; however, the user must not be concerned with the choice of the adopted protocol (unless this is wished).

The actual fitting process invoked via `selm` is actually performed by a set of lower-level procedures. These are accessible for direct call, if so wished, typically for improved efficiency, at the expense of a little additional programming effort. Similarly, functions to compute the Fisher information matrix are available, in the expected and the observed form (with some restrictions depending on the selected distribution).

The `extractSECdistr` function extracts the fitted SEC distribution from *selm*-class and *mselm*-class objects, hence providing a bridge with the probability side of the package.

The facilities for statistical work do not support the SUN family.

Additional material

Additional material is available in the section 'User guides, package vignettes and other documentation' accessible from the front page of the documentation. See especially the document `pkg_sn-intro.pdf`

Author

Adelchi Azzalini. Please send comments, error reports *et cetera* to the author, whose web page is http://azzalini.stat.unipd.it/.

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

`plot.fitdistr.grouped` *Plot an object generated by* `fitdistr.grouped`

Description

The object produced by `fitdistr.grouped` is plotted in the form of an histogram of the input original observed frequencies with superimposed the fitted parametric density function.

Usage

```
## S3 method for class 'fitdistr.grouped'
plot(x, freq = FALSE, col = "grey90",
  border = "grey80",  pdfcol = "blue", main, sub = NULL, xlab, ylab, xlim, ylim,
  axes = TRUE, labels = FALSE, ...)
```

Arguments

x	a link{fitdistr.grouped} object.
freq	logical; if TRUE, the histogram graphic is to present a representation of frequencies; if FALSE (default value), densities are plotted.
col	a colour to be used to fill the bars.
border	the colour of the border around the bars.
pdfcol	the colour of the fitted parametric distribution.
main, sub, xlab, ylab	these arguments to title an be omitted, in which case default values are constructed.
xlim, ylim	the range of x and y values with sensible defaults.
axes	logical, indicating if axes should be drawn.
labels	logical or character. Additionally draw labels on top of bars, if not FALSE; if TRUE, draw the counts or rounded densities; if labels is a character, draw itself.
...	further graphical parameters to title and axis.

Details

The function builds on plot.histogram, but some lesser features have been dropped.

Value

A list with the following components

hist	an object of class histogram which produces the histogram.
x, y	the values used for plotting the fitted parametric distribution.

Author(s)

Adelchi Azzalini

See Also

fitdistr.grouped for generating a suitable object, plot.histogram for the related more general function.

Examples

```
data(barolo)
attach(barolo)
A75 <- (reseller=="A" & volume==75)
logPrice <- log(price[A75],10) # as used in selm documentation; see its fitting
detach(barolo)
breaks<- seq(1, 3, by=0.25)
f <- cut(logPrice, breaks = breaks)
freq <- tabulate(f, length(levels(f)))
logPrice.grouped <- fitdistr.grouped(breaks, freq, family='ST')
plot(logPrice.grouped)
```

plot.SECdistr *Plotting methods for classes* SECdistrUv *and* SECdistrMv

Description

Plotting methods for classes SECdistrUv and SECdistrMv

Usage

```
## S4 method for signature 'SECdistrUv'
plot(x, range, probs, main, npt = 251, ...)

## S4 method for signature 'SECdistrMv'
plot(x, range, probs, npt, landmarks = "auto",
   main, comp, compLabs, data = NULL, data.par = NULL, gap = 0.5, ...)
```

Arguments

x	an object of class SECdistrUv or SECdistrMv.
range	in the univariate case, a vector of length 2 which defines the plotting range; in the multivariate case, a matrix with two rows where each column defines the plotting range of the corresponding component variable. If missing, a sensible choice is made.
probs	a vector of probability values. In the univariate case, the corresponding quantiles are plotted on the horizontal axis; it can be skipped by setting probs=NULL. In the multivariate case, each probability value corresponds to a contour level in each bivariate plot; at least one probability value is required. See 'Details' for further information. Default value: c(0.05, 0.25, 0.5, 0.75, 0.95) in the univariate case, c(0.25, 0.5, 0.75, 0.95) in the multivariate case.
npt	a numeric value or vector (in the univariate and in the multivariate case, respectively) to assign the number of evaluation points of the distribution, on an equally-spaced grid over the range defined above. Default value: 251 in the univariate case, a vector of 101's in the multivariate case.
landmarks	a character string which affects the placement of some landmark values in the multivariate case, that is, the origin, the mode and the mean (or its substitute pseudo-mean), which are all aligned. Possible values: "proper", "pseudo", "auto" (default), "". The option "" prevents plotting of the landmarks. With the other options, the landmarks are plotted, with some variation in the last one: "proper" plots the proper mean value, "pseudo" plots the pseudo-mean, useful when the proper mean does not exists, "auto" plots the proper mean if it exists, otherwise it switches automatically to the pseudo-mean. See dp2cp for more information on pseudo-CP parameters, including pseudo-mean.
main	a character string for main title; if missing, one is built from the available ingredients.
comp	a subset of the vector 1:d, if d denotes the dimensionality of the multivariate distribution.

compLabs	a vector of character strings or expressions used to denote the variables in the plot; if missing, slot(object,"compNames") is used.
data	an optional set of data of matching dimensionity of object to be superimposed to the plot. The default value data=NULL produces no effect. In the univariate case, data are plotted using rug at the top horizontal axis, unless if probs=NULL, in which case plotting is at the bottom axis. In the multivariate case, points are plotted in the form of a scatterplot or matrix of scatterplots; this can be regulated by argument data.par.
data.par	an optional list of graphical parameters used for plotting data in the multivariate case, when data is not NULL. Recognized parameters are: col, pch, cex. If missing, the analogous components of par() are used.
gap	a numeric value which regulates the gap between panels of a multivariate plot when d>2.
...	additional graphical parameters

Value

an invisible list. In the univariate case the list has three components: the input object representing the distribution and two numeric vectors with the coordinates of the plotted density values. In the multivariate case, the first element of the list is the input object representing the distribution and all subsequent list elements are lists with components of the panels comprising the matrix plot; the elements of these sub-lists are: the vectors of x and y coordinates, the names of the variables, the density values at the (x,y) points, a vector of the density levels of the curves appearing in each panel plot, with the corresponding approximate probability content as a vector attribute.

Details

For univariate density plots, probs are used to compute quantiles from the appropriate distribution, and these are superimposed to the plot of the density function, unless probs=NULL. In the multivariate case, each bivariate plot is constructed as a collection of contour curves, one curve for each probability level; consequently, probs cannot be missing or NULL. The level of the density contour lines are chosen so that each curve circumscribes a region with the quoted probability, to a good degree of approssimation; for additional information, see Azzalini and Capitanio (2014), specifically Complement 5.2 and p.179, and references therein.

Methods

signature(x = "SECdistrUv") Plot an object x of class SECdistrUv.

signature(x = "SECdistrMv") Plot an object x of class SECdistrMv.

Author(s)

Adelchi Azzalini

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

makeSECdistr, summary.SECdistr, dp2cp

Examples

```
# d=1
f1 <- makeSECdistr(dp=c(3,2,5), family="SC", name="Univariate Skew-Cauchy")
plot(f1)
plot(f1, range=c(-3,40), probs=NULL, col=4)
#
# d=2
Omega2 <- matrix(c(3, -3, -3, 5), 2, 2)
f2 <-  makeSECdistr(dp=list(c(10,30), Omega=Omega2, alpha=c(-3, 5)),
                family="sn", name="SN-2d", compNames=c("x1","x2"))
plot(f2)
x2 <- rmsn(100, dp=slot(f2,"dp"))
plot(f2, main="Distribution 'f2'", probs=c(0.5,0.9), cex.main=1.5, col=2,
  cex=0.8, compLabs=c(expression(x[1]), expression(log(z[2]-beta^{1/3}))),
  data=x2, data.par=list(col=4, cex=0.6, pch=5))
```

---

plot.selm *Diagnostic plots for* selm *fits*

Description

Diagnostic plots for objects of class selm and mselm generated by a call to function selm

Usage

```
  ## S4 method for signature 'selm'
plot(x, param.type="CP", which = c(1:4), caption,
    panel = if (add.smooth) panel.smooth else points, main = "",
    ask = prod(par("mfcol")) < length(which) && dev.interactive(), ...,
    id.n = 3, labels.id = names(x@residuals.dp),
    cex.id = 0.75, identline = TRUE, add.smooth = getOption("add.smooth"),
    label.pos = c(4, 2), cex.caption = 1)

 ## S4 method for signature 'mselm'
plot(x, param.type="CP",  which, caption,
    panel = if (add.smooth) panel.smooth else points,  main = "",
    ask = prod(par("mfcol")) < length(which) && dev.interactive(), ...,
    id.n = 3, labels.id = names(x@residuals.dp),
    cex.id = 0.75, identline = TRUE, add.smooth = getOption("add.smooth"),
    label.pos = c(4, 2), cex.caption = 1)
```

Arguments

x	an object of class selm or mselm.
param.type	a character string which selects the type of residuals to be used for some of of the plots; possible values are: "CP" (default), "DP", "pseudo-CP". The various type of residuals only differ by an additive term; see 'Details' for more information.
which	if a subset of the plots is required, specify a subset of 1:4; see 'Details' for a description of the plots.
caption	a vector of character strings with captions to appear above the plots.
panel	panel function. The useful alternative to points, panel.smooth can be chosen by add.smooth = TRUE.
main	title to each plot, in addition to the above caption.
ask	logical; if TRUE, the user is asked before each plot.
...	other parameters to be passed through to plotting functions.
id.n	number of points to be labelled in each plot, starting with the most extreme.
labels.id	vector of labels, from which the labels for extreme points will be chosen. NULL uses observation numbers..
cex.id	magnification of point labels.
identline	logical indicating if an identity line should be added to QQ-plot and PP-plot (default: TRUE).
add.smooth	logical indicating if a smoother should be added to most plots; see also panel above.
label.pos	positioning of labels, for the left half and right half of the graph respectively, for plots 1-3.
cex.caption	controls the size of caption.

Details

The meaning of param.type is described in dp2cp. However, for these plot only the first parameter component is relevant, which affects the location of the residuals; the other components are not computed. Moreover, for QQ-plot and PP-plot, DP-residuals are used irrespectively of param.type; see Section 'Background'.

Values which=1 and which=2 have a different effect for object of class "selm" and class "mselm". In the univariate case, which=1 plots the residual values versus the fitted values if p>1, where p denotes the number of covariates including the constant; if p=1, a boxplot of the response is produced. Value which=2 produces an histogram of the residuals with superimposed the fitted curve, when p>1; if p=1, a similar plot is generated using the response variable instead of the residuals. Default value for which is 1:4.

In the multivariate case, which=1 is feasible only if p=1 and it displays the data scatter with superimposed the fitted distribution. Value which=2 produces a similar plot but for residuals instead of data. Default value for codewhich is 2:4 if p>1, otherwise c(1,3,4).

Value which=3 produces a QQ-plot, both in the univariate and in the multivariate case; the difference is that the squares of normalized residuals and suitably defined Mahalanobis distances, respectively, are used in the two cases. Similarly, which=4 produces a PP-plot, working in a similar fashion.

Background

Healy-type graphical diagnostics, in the form of QQ- and PP-plots, for the multivariate normal distribution have been extended to the skew-normal distribution by Azzalini and Capitanio (1999, section 6.1), and subsequently to the skew-t distribution in Azzalini and Capitanio (2003). A brief explanation in the univariate SN case is provided in Section 3.1.1 of Azzalini and Capitanio (2014); see also Section 3.1.6. For the univariate ST case, see p.102 and p.111 of the monograph. The multivariate case is discussed in Section 5.2.1 as for the SN distribution, in Section 6.2.6 as for the ST distribution.

Author(s)

Adelchi Azzalini

References

Azzalini, A. and Capitanio, A. (1999). Statistical applications of the multivariate skew normal distribution. *J.Roy.Statist.Soc. B* **61**, 579-602. Full-length version available at https://arXiv.org/abs/0911.2093

Azzalini, A. and Capitanio, A. (2003). Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t distribution. *J.Roy. Statist. Soc. B* **65**, 367-389. Full-length version available at https://arXiv.org/abs/0911.2342

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

selm, dp2cp

Examples

```
data(wines)
#
m10 <- selm(flavanoids  ~ 1, family="SN", data=wines, subset=(wine=="Barolo"))
plot(m10)
plot(m10, which=c(1,3)) # fig 3.1 and 3.2(a) of Azzalini and Capitanio (2014)
#
m18 <- selm(acidity ~ sugar + nonflavanoids + wine, family="SN", data=wines)
plot(m18)
plot(m18, param.type="DP")
#
m28 <- selm(cbind(acidity, alcohol) ~ sugar + nonflavanoids + wine,
            family="SN", data=wines)
plot(m28, col=4)
#
data(ais)
m30 <-  selm(cbind(RCC, Hg, Fe) ~ 1, family="SN", data=ais)
plot(m30, col=2, which=2)

# multiple plots on the same sheet
par(mfcol=c(2,2))
```

```
plot(m30, which=1:3)
par(mfcol=c(1,1))
```

`plot.SUNdistr-method` *Plotting method for class* `SUNdistr`

Description

Plotting method for class `SUNdistr`

Usage

```
## S4 method for signature 'SUNdistr'
plot(x, range, nlevels = 8, levels, npt,  main, comp, compLabs, gap = 0.5, ...)
```

Arguments

`x`	an object of class `SUNdistr`
`range`	in the univariate case, a vector of length 2 which defines the plotting range; in the multivariate case, a matrix with two rows where each column defines the plotting range of the corresponding component variable. If missing, a sensible choice is made.
`nlevels`	number of contour levels desired **iff** levels is not supplied.
`levels`	numeric vector of levels at which to draw contour lines.
`npt`	a numeric value or vector (in the univariate and in the multivariate case, respectively) to assign the number of evaluation points of the distribution, on an equally-spaced grid over the `range` defined above. Default value: 251 in the univariate case, a vector of 101's in the multivariate case.
`main`	a character string for main title; if missing, one is built from the available ingredients.
`comp`	an optional integer vector representing the subset of the vector `1:d`, if `d` denotes the dimensionality of the distribution.
`compLabs`	a vector of character strings or expressions used to label the variables in the plot; if missing, `slot(object,"compNames")[comp]` is used.
`gap`	a numeric value which regulates the gap between panels of a multivariate plot when `d>2`; default: `0.5`.
`...`	additional graphical parameters

Details

For univariate density plots, `probs` are used to compute quantiles from the appropriate distribution, and these are superimposed to the plot of the density function, unless `probs=NULL`. In the multivariate case, each bivariate plot is constructed as a collection of contour curves, one curve for each probability level; consequently, `probs` cannot be missing or `NULL`. The level of the density contour lines are chosen so that each curve circumscribes a region with the quoted probability, to a good degree of approssimation; for additional information, see Azzalini and Capitanio (2014), specifically Complement 5.2 and p.179, and references therein.

Value

an invisible list. In the univariate case the list has three components: the input object representing the distribution and two numeric vectors with the coordinates of the plotted density values. In the multivariate case, the first element of the list is the input object representing the distribution and all subsequent list elements are lists with components of the panels comprising the matrix plot; the elements of these sub-lists are: the vectors of `x` and `y` coordinates, the names of the variables, the density values at the `(x,y)` points, a vector of the density levels of the curves appearing in each panel plot.

Author(s)

Adelchi Azzalini

See Also

`makeSUNdistr`, `SUNdistr-class`

Examples

```
xi <- c(1, 0, -1)
Omega <- matrix(c(2,1,1, 1,3,1, 1,1,4), 3, 3)
Delta <- matrix(c(0.72,0.20, 0.51,0.42, 0.88, 0.94), 3, 2, byrow=TRUE)
Gamma <- matrix(c(1, 0.8, 0.8, 1), 2, 2)
dp3 <- list(xi=xi, Omega=Omega, Delta=Delta, tau=c(-0.5, 0), Gamma=Gamma)
sun3 <- makeSUNdistr(dp=dp3, name="SUN3", compNames=c("x", "w", "z"))
plot(sun3, npt=rep(51,3))
p <- plot(sun3, comp=2:3, compLabs=c(expression(x[2]), expression(x[3])))
# str(p)
```

---

pprodt2 *The distribution of the product of two jointly normal or* t *variables*

Description

Consider the product $W = X_1 X_2$ from a bivariate random variable (X_1, X_2) having joint normal or Student's *t* distribution, with 0 location and unit scale parameters. Functions are provided for the distribution function of W in the normal and the *t* case, and the quantile function for the *t* case.

Usage

```
pprodn2(x, rho)
pprodt2(x, rho, nu)
qprodt2(p, rho, nu, tol=1e-5, trace=0)
```

Arguments

x	a numeric vector
p	a numeric vector of probabilities
rho	a scalar value representing the correlation (or the matching term in the *t* case when correlation does not exists)
nu	a positive scalar representing the degrees of freedom
tol	the desired accuracy (convergence tolerance), passed to function uniroot
trace	integer number for controlling tracing information, passed on to uniroot

Details

Function pprodt2 implements formulae in Theorem 1 of Wallgren (1980). Corresponding quantiles are obtained by qprodt2 by solving the pertaining non-linear equations with the aid of uniroot, one such equation for each element of p.

Function pprodn2 implements results for the central case in Theorem 1 of Aroian et al. (1978).

Value

a numeric vector

Author(s)

Adelchi Azzalini

References

Aroian, L.A., Taneja, V.S, & Cornwell, L.W. (1978). Mathematical forms of the distribution of the product of two normal variables. *Communications in statistics. Theory and methods*, 7, 165-172

Wallgren, C. M. (1980). The distribution of the product of two correlated *t* variates. *Journal of the American Statistical Association*, 75, 996-1000

See Also

uniroot

Examples

```
p <-  pprodt2(-3:3, 0.5, 8)
qprodt2(p, 0.5, 8)
```

predict.selm *Predict method for selm-class objects*

Description

Predicted values based on a model object produced by selm with univariate response.

Usage

```
  ## S3 method for class 'selm'
predict(object, newdata, param.type = "CP",
           interval = "none", level = 0.95, na.action = na.pass, ...)
```

Arguments

object	an object of class selm as produced by a call to function selm with univariate response.
newdata	an optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
param.type	a character string with the required parameterization; it must be one of "CP", "DP", "pseudo-CP", or possibly their equivalent lowercase.
interval	type of interval calculation among "none", "confidence", "prediction"; it can be abbreviated.
level	tolerance/confidence level (default value is 0.95).
na.action	function determining what should be done with missing values in newdata. The default is to predict NA.
...	not used, only there for compatibility reasons.

Details

Predicted values are obtained by evaluating the regression function in the dataframe newdata (which defaults to model.frame(object)). Setting interval other than "none" produces computation of confidence or prediction (tolerance) intervals at the specified level.

If newdata is omitted the predictions are based on the data used for the fit.

The action taken in case of missing data is regulated by argument na.action, along the lines of function predict.lm.

A detailed description of the methodology underlying predict.selm is available in the technical note of Azzalini (2016).

Value

a vector of predictions (if interval="none") or a matrix of predictions and bounds with column names fit, lwr, and upr, if interval is set.

Author(s)

Adelchi Azzalini

References

Azzalini, A. (2016). Derivation of various types of intervals from a selm object. Technical note distributed with the documentation of the R package sn, in file selm-intervals.pdf within section 'User guide, package vignettes and other documentation'.

See Also

selm, summary.selm,

makeSECdistr for the CP/DP parameterizations,

predict.lm for usage of na.action

Examples

```
data(barolo)
attach(barolo)
A75 <- (reseller=="A" & volume==75)
detach(barolo)
m3 <- selm(log(price, 10) ~ age, data=barolo[A75,], family="ST")
```

---

profile.selm *Profile log-likelihood function of selm-class objects*

Description

One- or two-dimensional profile (penalized) log-likelihood function of a selm fit and corresponding confidence interval or regions

Usage

```
  ## S3 method for class 'selm'
profile(fitted, param.type, param.name, param.values, npt,
   opt.control = list(), plot.it = TRUE, log = TRUE, levels,
   trace = FALSE, ...)
```

Arguments

fitted	an object of class selm as produced by a call to function selm with univariate response.
param.type	a character string with the required parameterization; it must be either "CP" or "DP", or possibly their equivalent lowercase.
param.name	either a single character string or a vector of two such terms with the name(s) of the parameter(s) for which the profile log-likelihood is required; these names must match those appearing in summary.selm(object, param.type).

param.values	in the one-parameter case, a numeric vector with the values where the log-likelihood must be evaluated; in the two-parameter case, a list of two such vectors used to build a grid of coordinates of points. Their range must identify an interval or a rectangle which includes the MLE or MPLE obtained by selm. See 'Details' for more information.
npt	in case the vector or any of the vectors of argument param.values has length 2, an equally spaced grid of values is build with length equal to the corresponding component of npt. If the above condition is met but this argument is missing, a default choice is made, namely 51 or (26,26) in the one- or two-parameter case, respectively.
opt.control	an optional list passed as argument control to optim to optimize the log-likelihood; see 'Details' for more information.
plot.it	a logical value; if TRUE (default value), a plot is produced representing the deviance, which is described in 'Details' below. In the one-parameter case, a confidence interval of prescribed level is marked on the plot; in the two-parameter case, the contour curves are labelled with approximate confidence levels. See however for more information.
log	a logical value (default: TRUE) indicating whether the scale and tail-weight parameter (the latter only for the ST family) must be log-transformed, if case any of them occurs in param.name. This applies to omega and nu in the DP parameter set and to s.d. and gamma2 in the CP parameter set.
levels	a single probability value (in the one-parameter case) or a vector of such values (in the two-parameter case) for which the confidence interval or region is requited; see 'Details' for more information.
trace	a logical value (default: FALSE) to activate printing of intermediate outcome of the log-likelihood optimization process
...	optional graphical parameters passed to the plotting functions.

Details

For each chosen point of the parameter(s) to be profiled, the log-likelihood is maximized with respect to the remaining parameters. The optimization process is accomplished using the optim optimization function, with method="BFGS". This step can be regulated by the user via opt.control which is passed to optim as control argument, apart from element fnscale whose use is reserved.

If the original fitted object included a fixed parameter value, this is kept fixed here. If the estimation method was "MPLE", that choice carries on here; in case the penalty function was user-defined, it must still be accessible.

For plotting purposes and also in the numerical output, the deviance function D is used, namely

$$D = 2\,[\max(\log L) - \log L]$$

where L denotes the likelihood.

The range of param.values must enclose the maximum (penalized) likelihood estimates (MLE or MPLE) by an adequate extent such that suitable confidence intervals or regions can be established from standard asymptotic theory. If this condition does not hold, the function still proceeds, but no confidence interval or region is delivered. For the SN family and DP parameterization, the asymptotic theory is actually non-standard near the important point $\alpha = 0$, but the correspondence with

the regular case of the CP parameterization, still allows to derive confidence regions using standard procedures; for additional information, see Section 3.1.6 of Azzalini and Capitanio (2014). When the MLE occurs on the frontier of the parameter space, a message is issued and no confidence interval is produced, while in the two-parameter case the plot is not labelled with probability values, but only with deviance levels.

Value

An invisible list whose components, described below, are partly different in the one- and the two-parameter cases.

`call`	the calling statement
`<param1>`	values of the first parameter
`<param2>`	values of the second parameter (in the two-parameter case)
`logLik`	numeric vector or matrix of the profile log-likelihood values
`confint`	in the one-parameter case, the confidence interval
`level`	in the one-parameter case, the confidence level
`deviance.contour`	in the two-parameter case, a list of lists whose elements identify each curve of the contour plot

Warnings

- This function is experimental and changes in future versions of the package may occur. Users should not rely on the persistence of the same user interface or the same name(s).
- It is a known fact that, in some critical situations, peculiar outcomes are produced.

Author(s)

Adelchi Azzalini

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

`selm`, `summary.selm`,

`makeSECdistr` for the CP/DP parameterizations,

`optim` for its `control` argument

Examples

```
data(ais, package="sn")
m1 <-  selm(log(Fe) ~ BMI + LBM, family = "sn", data = ais)

pll <- profile(m1, "dp", param.name="alpha", param.val=c(-3,2))

profile(m1, "cp", param.name="gamma1", param.val=seq(-0.7, 0.4, by=0.1))

# in the next example, we reduce the grid points to save execution time
pll <- profile(m1, "cp", param.name=c("(Intercept.CP)", "gamma1"),
           param.val = list(c(1.5, 4), c(-0.8, 0.5)), npt=c(11,11) )
```

Qpenalty *Penalty function for log-likelihood of* selm *models*

Description

Penalty function for the log-likelihood of selm models when method="MPLE". Qpenalty is the default function; MPpenalty is an example of a user-defined function effectively corresponding to a prior distributio on alpha.

Usage

```
Qpenalty(alpha_etc, nu = NULL, der = 0)

MPpenalty(alpha, der = 0)
```

Arguments

alpha_etc, alpha	in the univariate case, a single value alpha; in the multivariate case, a two-component list whose first component is the vector alpha, the second one is matrix cov2cor(Omega).
nu	degrees of freedom, only required if selm is called with family="ST".
der	a numeric value in the set 0,1,2 which indicates the required numer of derivatives of the function. In the multivariate case the function will only be called with der equal to 0 or 1.

Details

The penalty is a function of alpha, but its expression may depend on other ingredients, specifically nu and cov2cor(Omega). See 'Details' of selm for additional information.

The penalty mechanism allows to introduce a prior distribution π for α by setting $Q = -\log \pi$, leading to a maximum *a posteriori* estimate in the stated sense.

As a simple illustration of this mechanism, function MPpenalty implements the 'matching prior' distribution for the univariate SN distribution studied by Cabras *et al.* (2012); a brief summary

of the proposal is provided in Section 3.2 of Azzalini and Capitanio (2014). Note that, besides `alpha=+/-Inf`, this choice also penalizes `alpha=0` with `Q=Inf`, effectively removing `alpha=0` from the parameter space.

Starting from the code of function `MPpenalty`, a user should be able to introduce an alternative prior distribution if so desired.

Value

A positive number `Q` representing the penalty, possibly with attributes `attr(Q, "der1")` and `attr(Q, "der2")`, depending onthe input value `der`.

Author(s)

Adelchi Azzalini

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

Cabras, S., Racugno, W., Castellanos, M. E., and Ventura, L. (2012). A matching prior for the shape parameter of the skew-normal distribution. *Scand. J. Statist.* **39**, 236–247.

See Also

`selm` function

Examples

```
data(frontier)
m2 <- selm(frontier ~ 1)  # no penalty
m2a <- selm(frontier ~ 1, method="MPLE") # penalty="Qpenalty" is implied here
m2b <- selm(frontier ~ 1, method="MPLE", penalty="MPpenalty")
```

`residuals.selm` *Residuals and fitted values from* `selm` *fits*

Description

`residuals` and `fitted` methods for classes `"selm"` and `"mselm"`.

Usage

```
## S4 method for signature 'selm'
residuals(object, param.type = "CP", ...)
## S4 method for signature 'mselm'
residuals(object, param.type = "CP", ...)
## S4 method for signature 'selm'
fitted(object, param.type = "CP", ...)
```

```
## S4 method for signature 'mselm'
fitted(object, param.type = "CP", ...)
```

Arguments

object	an object of class "selm" or "mselm" as created by a call to function selm.
param.type	a character string which indicates the required type of parameter type; possible values are "CP" (default), "DP", "pseudo-CP" and their equivalent lower-case expressions.
...	not used, included for compatibility with the generic method.

Value

a numeric vector (for selm-class objects) or a matrix (for mselm-class objects).

Note

The possible options of param.type are described in the documentation of dp2cp; their corresponding outcomes differ by an additive constant only. With the "CP" option (that is, the 'centred parametrization'), the residuals are centred around 0, at least approximately; this is a reason for setting "CP" as the default option. For more information, see the 'Note' in the documentation of summary.selm.

Author(s)

Adelchi Azzalini

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

dp2cp, summary.selm, selm function, selm-class

Examples

```
data(wines, package="sn")
m5 <- selm(acidity ~ phenols + wine, family="SN", data=wines)
residuals(m5)
residuals(m5, "dp")
fitted(m5, "dp")
#
m12 <- selm(cbind(acidity, alcohol) ~ phenols + wine,  family="SN", data=wines)
residuals(m12)
#
# see other examples at function selm
```

sd *Standard deviation*

Description

The sd function from the **stats** is replaced by a new method in order to introduce a separate method to deal with objects of class SECdistrUv. The function sd.default is an alias of the original function sd.

Usage

```
  sd(x, ...)
  ## Default S3 method:
sd(x, na.rm = FALSE, ...)
```

Arguments

x	a numeric vector, matrix or data frame.
na.rm	logical. Should missing values be removed?
...	further arguments passed to or from other methods.

See Also

sd, SECdistrUv

SECdistrMv-class *Class "SECdistrMv"*

Description

A class of objects representing multivariate skew-elliptically contoured (SEC) distributions.

Objects from the Class

Objects can be created by a call to function makeSECdistr, when its argument dp is a list, or by a suitable transformation of some object of this class. They can also obtained from an object generated by selm using the function extractSEDdistr.

Slots

family: a character string which identifies the parametric family; currently, possible values are: "SN", "ESN", "ST", "SC".

dp: a list of parameters; its length depends on the selected family.

name: a character string with the name of the multivariate variable; it can be an empty string.

compNames: a vector of character strings with the names of the component variables.

Methods

show `signature(object = "SECdistrMv-class")`: ...

plot `signature(x = "SECdistrMv-class")`: ...

summary `signature(object = "SECdistrMv-class")`: ...

mean `signature(x = "SECdistrMv")`: ...

vcov `signature(object = "SECdistrMv")`: ...

Note

See `makeSECdistr` for a detailed description of `family` and `dp`.

Note that here methods `mean` and `vcov` are not applied to data or to a fitted model, but to a *probability distribution* instead, of which they provide the mean (vector) value and the variance-covariance matrix. If methods `mean` and `vcov` are applied to a distribution for which the mean or the variance do not exist, a `NULL` value is returned and a warning message is issued.

Author(s)

Adelchi Azzalini

See Also

`SECdistrUv`, `plot,SECdistrMv-method`, `summary,SECdistrMv-method`, `affineTransSECdistr`, `marginalSECdistr`, `extractSECdistr`

Examples

```
dp0 <- list(xi=1:2, Omega=diag(3:4), alpha=c(3, -5))
f10 <- makeSECdistr(dp=dp0, family="SN", name="SN-2D", compNames=c("x", "y"))
show(f10)
plot(f10)
summary(f10)
mean(f10)  # the mean value of the probability distribution
vcov(f10)  # the variance-covariance matrix of the probability distribution
```

SECdistrUv-class *Class* "SECdistrUv"

Description

A class of objects representing univariate skew-elliptically contoured (SEC) distributions.

Objects from the class

Objects can be created by a call to function `makeSECdistr` when its argument `dp` is a vector. They can also obtained from an object generated by `selm` using the function `extractSEDdistr`.

Slots

`family`: a character string which selects the parametric family; currently, possible values are: `"SN"`, `"ESN"`, `"ST"`, `"SC"`.

`dp`: a numeric vector of parameters; its length depends on the selected `family`.

`name`: a character string with name of the distribution.

Methods

show `signature(object = "SECdistrUv")`: ...

plot `signature(x = "SECdistrUv")`: ...

summary `signature(object = "SECdistrUv")`: ...

mean `signature(x = "SECdistrUv")`: ...

sd `signature(object = "SECdistrUv")`: ...

Note

See `makeSECdistr` for a detailed description of `family` and `dp`.

Unlike various other packages, methods `mean` and `sd` here are not targeted to data or to a fitted model, but to a *probability distribution* instead, of which they provide the mean value and the standard deviation. If these methods are applied to a distribution of which the mean or the variance do not exist, a `NULL` value is returned and a warning message is issued.

Author(s)

Adelchi Azzalini

See Also

`SECdistrMv`, `plot,SECdistrUv-method`, `summary,SECdistrUv-method`, `extractSECdistr`

Examples

```
f2 <- makeSECdistr(dp=c(3, 5, -pi, 6), family="ST", name="My first ST")
show(f2)
plot(f2)
plot(f2, probs=c(1,5,9)/10)
plot(f2, range=c(-30,10), probs=NULL, col=2, main=NULL)
summary(f2)
mean(f2)  # the mean value of the probability distribution
sd(f2)  # the standard deviation of the distribution
```

selm *Fitting linear models with skew-elliptical error term*

Description

Function selm fits a linear model with skew-elliptical error term. The term 'skew-elliptical distribution' is an abbreviated equivalent of skew-elliptically contoured (SEC) distribution. The function works for univariate and multivariate response variables.

Usage

```
selm(formula, family = "SN", data, weights, subset, na.action,
  start = NULL, fixed.param = list(), method = "MLE",  penalty=NULL,
  model = TRUE, x = FALSE, y = FALSE, contrasts = NULL, offset, ...)
```

Arguments

formula	an object of class "formula" (or one that can be coerced to that class): a symbolic description of the model to be fitted, using the same syntax used for the similar parameter of e.g. "lm", with the restriction that the constant term must not be removed from the linear predictor.
family	a character string which selects the parametric family of SEC type assumed for the error term. It must be one of "SN" (default), "ST" or "SC", which correspond to the skew-normal, the skew-*t* and the skew-Cauchy family, respectively. See makeSECdistr for more information on these families and the set of SEC distributions; notice that the family "ESN" listed there is not allowed here.
data	an optional data frame containing the variables in the model. If not found in data, the variables are taken from environment(formula), typically the environment from which selm is called.
weights	a numeric vector of weights associated to individual observations. Weights are supposed to represent frequencies, hence must be non-negative integers (not all 0) and length(weights) must equal the number of observations. If not assigned, a vector of all 1's is generated.
subset	an optional vector specifying a subset of observations to be used in the fitting process. It works like the same parameter in lm.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the na.action setting of options. The 'factory-fresh' default is na.omit. Another possible value is NULL, no action.
start	a vector (in the univariate case) or a list (in the multivariate case) of initial DP values for searching the parameter estimates. See 'Details' about a choice of start to be avoided. If start=NULL (default), initial values are selected by the procedure. If family="ST", an additional option exists; see 'Details'.
fixed.param	a list of assignments of parameter values which must be kept fixed in the estimation process. Currently, there only two types of admissible constraint: one is to set alpha=0 to impose a symmetry condition of the distribution; the other is

to set nu=<value>, to fix the degrees of freedom at the named <value> when family="ST", for instance list(nu=3). See 'Details' for additional information.

method — a character string which selects the estimation method to be used for fitting. Currently, two options exist: "MLE" (default) and "MPLE", corresponding to standard maximum likelihood and maximum penalized likelihood estimation, respectively. See 'Details' for additional information.

penalty — a character string which denotes the penalty function to be subtracted to the log-likelihood function, when method="MPLE"; if penalty=NULL (default), a pre-defined function is adopted. See 'Details' for a description of the default penalty function and for the expected format of alternative specifications. When method="MLE", no penalization is applied and this argument has no effect.

model, x, y — logicals. If TRUE, the corresponding components of the fit are returned.

contrasts — an optional list. See the contrasts.arg of model.matrix.default.

offset — this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases. One or more offset terms can be included in the formula instead or as well, and if more than one are specified their sum is used.

... — optional control parameters, as follows.

- trace: a logical value which indicates whether intermediate evaluations of the optimization process are printed (default: FALSE).
- info.type: a character string which indicates the type of Fisher information matrix; possible values are "observed" (default) and "expected". Currently, "expected" is implemented only for the SN family.
- opt.method: a character string which selects the numerical optimization method, among the possible values "nlminb", "Nelder-Mead", "BFGS", "CG", "SANN". If opt.method="nlminb" (default), function nlminb is called, otherwise function optim is called with method equal to opt.method.
- opt.control: a list of control parameters which is passed on either to nlminb or to optim, depending on the chosen opt.method.

Details

By default, selm fits the selected model by maximum likelihood estimation (MLE), making use of some numerical optimization method. Maximization is performed in one parameterization, usually DP, and then the estimates are mapped to other parameter sets, CP and pseudo-CP; see dp2cp for more information on parameterizations. These parameter transformations are carried out trasparently to the user. The observed information matrix is used to obtain the estimated variance matrix of the MLE's and from this the standard errors. Background information on MLE in the context of SEC distributions is provided by Azzalini and Capitanio (2014); see specifically Chapter 3, Sections 4.3, 5.2, 6.2.5–6. For additional information, see the original research work referenced therein as well as the sources quoted below.

Although the density functionof SEC distributions are expressed using DP parameter sets, the methods associated to the objects created by this function communicate, by default, their outcomes in the CP parameter set, or its variant form pseudo-CP when CP does not exist; the 'Note' at summary.selm explains why. A more detailed discussion is provided by Azzalini and Capitanio (1999, Section 5.2)

and Arellano-Valle and Azzalini (2008, Section 4), for the univariate and the multivariate SN case, respectively; an abriged account is available in Sections 3.1.4–6 and 5.2.3 of Azzalini and Capitanio (2014). For the ST case, see Arellano-Valle and Azzalini (2013).

There is a known open issue which affects computation of the information matrix of the multivariate skew-normal distribution when the slant parameter α approaches the null vector; see p.149 of Azzalini and Capitanio (2014). Consequently, if a model with multivariate response is fitted with `family="SN"` and the estimate `alpha` of α is at the origin or neary so, the information matrix and the standard errors are not computed and a warning message is issued. In this unusual circumstance, a simple work-around is to re-fit the model with `family="ST"`, which will work except in remote cases when (i) the estimated degrees of freedom `nu` diverge and (ii) still `alpha` remains at the origin.

The optional argument `fixed.param=list(alpha=0)` imposes the constraint $\alpha = 0$ in the estimation process; in the multivariate case, the expression is interpreted in the sense that all the components of vector α are zero, which implies symmetry of the error distribution, irrespectively of the parameterization subsequently adopted for summaries and diagnostics. When this restriction is selected, the estimation method cannot be set to `"MPLE"`. Under the constraint $\alpha = 0$, if `family="SN"`, the model is fitted similarly to `lm`, except that here MLE is used for estimation of the covariance matrix. If `family="ST"` or `family="SC"`, a symmetric Student's t or Cauchy distribution is adopted.

Under the constraint $\alpha = 0$, the location parameter ξ coincides with the mode and the mean of the distribution, when the latter exists. In addition, when the covariance matrix of a ST distribution exists, it differs from Ω only by a multiplicative factor. Consequently, the summaries of a model of this sort automatically adopt the DP parametrization.

The other possible form of constraint allows to fix the degrees of freedom when `family="ST"`. The two constraints can be combined writing, for instance, `fixed.param=list(alpha=0, nu=6)`. The constraint `nu=1` is equivalent to select `family="SC"`. In practice, an expression of type `fixed.param=list(..)` can be abbreviated to `fixed=list(..)`.

Argument `start` allows to set the initial values, with respect to the DP parameterization, of the numerical optimization. However, there is a specific choice of start to be avoided. When `family="SN"`, do not set the shape parameter `alpha` exactly at 0, as this would blow-up computation of the log-likelihood gradient and the Hessian matrix. This is not due to a software bug, but to a known peculiar behaviour of the log-likelihood function at that specific point. Therefore, in the univariate case for instance, do not set e.g. `start=c(12, 21, 0)`, but set instead something like `start=c(12, 21, 0.01)`. Recall that, if one needs to fit a model forcing 0 asymmetry, typically to compare two log-likelihood functions with/without asymmetry, then the option to use is `fixed.param=list(alpha=0)`.

Since version 1.6.0, a new initialization procedure has been introduced for the case `family="ST"`, which adopts the method proposed by Azzalini & Salehi (2020), implemented in functions `st.prelimFit` and `mst.prelimFit`. Correspondingly, the `start` argument can now be of different type, namely a character with possible values `"M0"`, `"M2"` (detault in the univariate case) and `"M3"` (detault in the multivariate case). The choice `"M0"` selects the older method, in use prior to version 1.6.0. For more information, see Azzalini & Salehi (2020).

In some cases, especially for small sample size, the MLE occurs on the frontier of the parameter space, leading to DP estimates with `abs(alpha)=Inf` or to a similar situation in the multivariate case or in an alternative parameterization. Such outcome is regared by many as unsatisfactory; surely it prevents using the observed information matrix to compute standard errors. This problem motivates the use of maximum penalized likelihood estimation (MPLE), where the regular log-likelihood function $\log\ L$ is penalized by subtracting an amount Q, say, increasingly large as $|\alpha|$

increases. Hence the function which is maximized at the optimization stage is now $\log L - Q$. If `method="MPLE"` and `penalty=NULL`, the default function `Qpenalty` is used, which implements the penalization:

$$Q(\alpha) = c_1 \log(1 + c_2 \alpha_*^2)$$

where c_1 and c_2 are positive constants, which depend on the degrees of freedom `nu` in the `ST` case,

$$\alpha_*^2 = \alpha^\top \bar{\Omega} \alpha$$

and $\bar{\Omega}$ denotes the correlation matrix associated to the scale matrix `Omega` described in connection with `makeSECdistr`. In the univariate case $\bar{\Omega} = 1$, so that $\alpha_*^2 = \alpha^2$. Further information on MPLE and this choice of the penalty function is given in Section 3.1.8 and p.111 of Azzalini and Capitanio (2014); for a more detailed account, see Azzalini and Arellano-Valle (2013) and references therein.

It is possible to change the penalty function, to be declared via the argument `penalty`. For instance, if the calling statement includes `penalty="anotherQ"`, the user must have defined

```
anotherQ <- function(alpha_etc, nu = NULL, der = 0)
```

with the following arguments.

- `alpha_etc`: in the univariate case, a single value `alpha`; in the multivariate case, a two-component list whose first component is the vector `alpha`, the second one is matrix equal to `cov2cor(Omega)`.
- `nu`: degrees of freedom, only relevant if `family="ST"`.
- `der`: a numeric value which indicates the required order of derivation; if `der=0` (default value), only the penalty `Q` needs to be retuned by the function; if `der=1`, `attr(Q, "der1")` must represent the first order derivative of `Q` with respect to `alpha`; if `der=2`, also `attr(Q, "der2")` must be assigned, containing the second derivative (only required in the univariate case).

This function must return a single numeric value, possibly with required attributes when is called with `der>1`. Since **sn** imports functions `grad` and `hessian` from package **numDeriv**, one can rely on them for numerical evaluation of the derivatives, if they are not available in an explicit form.

This penalization scheme allows to introduce a prior distribution π for α by setting $Q = -\log \pi$, leading to a maximum *a posteriori* estimate in the stated sense. See `Qpenalty` for more information and an illustration.

The actual computations are not performed within `selm` which only sets-up ingredients for work of `selm.fit` and other functions further below this one. See `selm.fit` for more information.

Value

an S4 object of class `selm` or `mselm`, depending on whether the response variable of the fitted model is univariate or multivariate; these objects are described in the `selm class`.

Cautionary notes

The first of these notes applies to the stage *preceding* the use of `selm` and related fitting procedures. Before fitting a model of this sort, consider whether you have enough data for this task. In this respect, the passage below taken from p.63 of Azzalini and Capitanio (2014) is relevant.

"Before entering technical aspects, it is advisable to underline a qualitative effect of working with a parametric family which effectively is regulated by moments up to the third order. The implication

is that the traditional rule of thumb by which a sample size is small up to 'about $n = 30$', and then starts to become 'large', while sensible for a normal population or other two-parameter distribution, is not really appropriate here. To give an indication of a new threshold is especially difficult, because the value of α also has a role here. Under this *caveat*, numerical experience suggests that 'about $n = 50$' may be a more appropriate guideline in this context."

The above passage referred to the univariate SN context. In the multivariate case, increase the sample size appropriately, especially so with the ST family. This is not to say that one cannot attempt fitting these models with small or moderate sample size. However, one must be aware of the implications and not be surprised if problems appear.

The second cautionary note refers instead to the outcome of a call to `selm` and related function, or the lack of it. The estimates are obtained by numerical optimization methods and, as usual in similar cases, there is no guarantee that the maximum of the objective function is achieved. Consideration of model simplicity and of numerical experience indicate that models with SN error terms generally produce more reliable results compared to those with the ST family. Take into account that models involving a traditional Student's t distribution with unknown degrees of freedom can already be problematic; the presence of the (multivariate) slant parameter α in the ST family cannot make things any simpler. Consequently, care must be exercised, especially so if one works with the (multivariate) ST family. Consider re-fitting a model with different starting values and, in the ST case, building the profile log-likelihood for a range of ν values; function `profile.selm` can be useful here.

Details on the numerical optimization which has produced object `obj` can be extracted with `slot(obj, "opt.method")`; inspection of this component can be useful in problematic cases. # Be aware that occasionally `optim` and `nlminb` declare successful # completion of a regular minimization problem at a point where the Hessian # matrix is not positive-definite.

Author(s)

Adelchi Azzalini

References

Arellano-Valle, R. B., and Azzalini, A. (2008). The centred parametrization for the multivariate skew-normal distribution. *J. Multiv. Anal.* **99**, 1362–1382. Corrigendum: **100** (2009), 816.

Arellano-Valle, R. B., and Azzalini, A. (2013, available online 12 June 2011). The centred parametrization and related quantities for the skew-*t* distribution. *J. Multiv. Anal.* **113**, 73–90.

Azzalini, A. and Capitanio, A. (1999). Statistical applications of the multivariate skew normal distribution. *J.Roy.Statist.Soc. B* **61**, 579–602. Full-length version available at https://arXiv.org/abs/0911.2093

Azzalini, A. and Arellano-Valle, R. B. (2013, available online 30 June 2012). Maximum penalized likelihood estimation for skew-normal and skew-*t* distributions. *J. Stat. Planning & Inference* **143**, 419–433.

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

Azzalini, A. and Salehi, M. (2020). Some computational aspects of maximum likelihood estimation of the skew-*t* distribution. In *Computational and Methodological Statistics and Biostatistics*, edited by A. Bekker, Ding-Geng Chen and Johannes T. Ferreira, pp.3-28. Springer Nature Switzerland.

See Also

- selm-class for classes "selm" and "mselm", summary.selm for summaries, plot.selm for plots, residuals.selm for residuals and fitted values
- the generic functions coef, logLik, vcov, profile, confint, predict
- the underlying function selm.fit and those further down
- the selection of a penalty function of the log-likelihood, such as Qpenalty
- the function extractSECdistr to extract the SEC error distribution from an object returned by selm
- the broad underlying logic and a number of ingredients are like in function lm

Examples

```
data(ais)
m1 <- selm(log(Fe) ~ BMI + LBM, family="SN", data=ais)
print(m1)
summary(m1)
s <- summary(m1, "DP", cov=TRUE, cor=TRUE)
plot(m1)
plot(m1, param.type="DP")
logLik(m1)
coef(m1)
coef(m1, "DP")
var <- vcov(m1)
#
m1a <- selm(log(Fe) ~ BMI + LBM, family="SN", method="MPLE", data=ais)
m1b <- selm(log(Fe) ~ BMI + LBM, family="ST", fixed.param=list(nu=8), data=ais)
#
data(barolo)
attach(barolo)
A75 <- (reseller=="A" & volume==75)
logPrice <- log(price[A75],10)
m <- selm(logPrice ~ 1, family="ST", opt.method="Nelder-Mead")
summary(m)
summary(m, "DP")
plot(m, which=2, col=4, main="Barolo log10(price)")
# cfr Figure 4.7 of Azzalini & Capitanio (2014), p.107
detach(barolo)
#-----
# examples with multivariate response
#
m3 <- selm(cbind(BMI, LBM) ~ WCC + RCC, family="SN", data=ais)
plot(m3, col=2, which=2)
summary(m3, "dp")
coef(m3)
coef(m3, vector=FALSE)
#
data(wines)
m28 <- selm(cbind(chloride, glycerol, magnesium) ~ 1, family="ST", data=wines)
dp28 <- coef(m28, "DP", vector=FALSE)
pcp28 <- coef(m28, "pseudo-CP", vector=FALSE)
```

```
# the next statement takes a little more time than others
plot(m28)

#
m4 <- selm(cbind(alcohol,sugar)~1, family="ST", data=wines)
m5 <- selm(cbind(alcohol,sugar)~1, family="ST", data=wines, fixed=list(alpha=0))
print(1 - pchisq(2*as.numeric(logLik(m4)-logLik(m5)), 2)) # test for symmetry
```

`selm-class` *Classes* `selm` *and* `mselm` *of objects created by function* `selm`

Description

A successful call to function `selm` creates an object of either of these classes, having a structure described in section 'Slots'. A set of methods for these classes of objects exist, listed in section 'Methods'.

Objects from the class

An object can be created by a successful call to function `selm`.

Slots

`call`: the calling statement.

`family`: the parametric family of skew-ellitically contoured distributed (SEC) type.

`logL`: log-likelihood or penalized log-likelihood value achieved at the end of the maximization process.

`method`: estimation method (`"MLE"` or `"MPLE"`).

`param`: estimated parameters, for various parameterizations.

`param.var`: approximate variance matrices of the parameter estimates, for various parameterizations.

`size`: a numeric vector with size of various components.

`fixed.param`: a vector of parameters which have been kept fixed in the fitting process, if any.

`residuals.dp`: residual values, for DP-type parameters.

`fitted.values.dp`: fitted values, for DP-type parameters.

`control`: a list with control parameters.

`input`: a list of selected input values.

`opt.method`: a list with details on the optimization method.

Methods

```
coef        signature(object = "selm"): ...
logLik      signature(object = "selm"): ...
plot        signature(x = "selm"): ...
show        signature(object = "selm"): ...
summary     signature(object = "selm"): ...
residuals   signature(object = "selm"): ...
fitted      signature(object = "selm"): ...
vcov        signature(object = "selm"): ...
weights     signature(object = "selm"): ...
profile     signature(fitted = "selm"): ...
confint     signature(object = "selm"): ...
predict     signature(object = "selm"): ...

coef        signature(object = "mselm"): ...
logLik      signature(object = "mselm"): ...
plot        signature(x = "mselm"): ...
show        signature(object = "mselm"): ...
summary     signature(object = "mselm"): ...
residuals   signature(object = "mselm"): ...
fitted      signature(object = "mselm"): ...
vcov        signature(object = "mselm"): ...
weights     signature(object = "mselm"): ...
```

Note

See dp2cp for a description of possible parameter sets. When logLik is used on an object obtained using the MPLE estimation method, the value reported is actually the *penalized* log-likelihood.

Author(s)

Adelchi Azzalini

See Also

See also selm function, plot.selm, summary.selm, dp2cp

Examples

```
data(ais)
m1 <- selm(log(Fe) ~ BMI + LBM, family="SN", data=ais)
summary(m1)
plot(m1)
logLik(m1)
res <- residuals(m1)
```

```
fv <- fitted(m1)
#
data(wines, package="sn")
m2 <- selm(alcohol ~ malic + phenols, data=wines)
#
m12 <- selm(cbind(acidity, alcohol) ~ phenols + wine, family="SN", data=wines)
coef(m12)
cp <- coef(m12, vector=FALSE)
dp <- coef(m12, "DP", vector=FALSE)
plot(m12)
plot(m12, which=2, col="gray60", pch=20)
```

`selm.fit` *Fitting functions for* `selm` *models*

Description

A call to `selm` activates a call to `selm.fit` and from here to some other function which actually performs the parameter search, among those listed below. These lower-level functions can be called directly for increased efficiency, at the expense of some more programming effort and lack of methods for the returned object.

Usage

```
selm.fit(x, y, family = "SN", start = NULL, w, fixed.param = list(),
   offset = NULL, selm.control=list())

sn.mple(x, y, cp = NULL, w, penalty = NULL, trace = FALSE, opt.method =
   c("nlminb",  "Nelder-Mead", "BFGS", "CG", "SANN"), control = list())

st.mple(x, y, dp = NULL, w, fixed.nu = NULL, symmetr = FALSE, penalty = NULL,
   trace = FALSE, opt.method = c("nlminb", "Nelder-Mead", "BFGS", "CG", "SANN"),
   control = list())

msn.mle(x, y, start = NULL, w, trace = FALSE, opt.method = c("nlminb",
   "Nelder-Mead", "BFGS", "CG", "SANN"), control = list())

msn.mple(x, y, start = NULL, w, trace = FALSE, penalty = NULL,
   opt.method = c("nlminb", "Nelder-Mead", "BFGS", "CG", "SANN"),
   control = list())

mst.mple(x, y, start = NULL, w, fixed.nu = NULL, symmetr=FALSE,
   penalty = NULL, trace = FALSE,
   opt.method = c("nlminb", "Nelder-Mead", "BFGS", "CG", "SANN"),
   control = list())
```

Arguments

x	a full-rank design matrix with the first column of all 1's.
y	a vector or a matrix of response values such that NROW(y)=nrow(x).
family	a character string which selects the parametric family of distributions assumed for the error term of the regression model. It must one of "SN" (default), "ST" or "SC", which correspond to the skew-normal, the skew-*t* and the skew-Cauchy family, respectively. See makeSECdistr for more information on these families and the skew-elliptically contoured (SEC) distributions; notice that family "ESN" is not allowed here.
start, dp, cp	a vector or a list of initial parameter values, depeding whether y is a vector or a matrix. It is assumed that cp is given in the CP parameterization, dp and start in the DP parameterization. For st.mple and mst.mple, see also the paragraph about start in the documentation 'Details' of selm.
w	a vector of non-negative integer weights of length equal to NROW(y); if missing, a vector of all 1's is generated.
fixed.param	a list of assignments of parameter values to be kept fixed during the optimization process. Currently, there is only one such option, namely fixed.param=list(nu='value'), to fix the degrees of freedom at the named 'value' when family="ST", for instance list(nu=3). Setting fixed.param=list(nu=1) is equivalent to select family="SC".
penalty	an optional character string with the name of the penalty function of the log-likelihood; default value NULL corresponds to no penalty.
offset	this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. This should be NULL or a numeric vector of length equal to the number of cases. One or more offset terms can be included in the formula instead or as well, and if more than one are specified their sum is used.
trace	a logical value which regulates printing of successive calls to the target function; default value is FALSE which suppresses printing.
fixed.nu	a positive value to keep fixed the parameter nu of the ST distribution in the optimization process; with default value NULL, nu is estimated like the other parameters.
symmetr	a logical flag indicating whether a contraint of symmetry is imposed on the slant parameter; default is symmetr=FALSE.
opt.method	a character string which selects the optimization method within the set c("nlminb", "Nelder-Mead", "BFGS", "CG", "SANN"); the last four of these are "methods" of function optim.
selm.control	a list whose components regulate the working of selm.fit; see 'Details' for their description;
control	a list of control items passed to the optimization function.

Details

A call to selm produces a call to selm.fit which selects the appropriate function among sn.mple, st.mple, msn.mle, msn.mple, mst.mple, depending on the arguments of the calling statement. In

the adopted scheme for function names, msn refers to a multivariate skew-normal distribution and mst refers to a multivariate skew-t distribution, while mle and mple refers to maximum likelihood and maximum penalized likelihood estimation, respectively. Of these functions, sn.mple works in CP space; the others in the DP space. In all cases, a correspondig mapping to the alternative parameter space is performed before exiting selm.fit, in addition to the selected parameter set.

The components of selm.control are as follows:

- method: the estimation method, "MLE" or "MPLE".
- penalty: a string with the name of the penalty function.
- info.type: a string with the name of the information matrix, "observed" or "expected"; currently fixed at "observed".
- opt.method: a character string which selects the optimization method.
- opt.control: a list of control parameters of opt.method.

Function msn.mle, for MLE estimation of linear models with SN errors, is unchanged from version 0.4-x of the package. Function msn.mple is similar to msn.mle but allows to introduce a penalization of the log-likelihood; when penalty=NULL, a call to msn.mle is more efficient. Functions sn.mple and mst.mple work like sn.mle and mst.mle in version 0.4-x if the argument penalty is not set or it is set to NULL, except that mst.mple does not handle a univariate response (use st.mple for that).

Value

A list whose specific components depend on the named function. Typical components are:

call	the calling statement
dp	vector or list of estimated DP parameters
cp	vector or list of estimated CP parameters
logL	the maximized (penalized) log-likelihood
aux	a list with auxiliary output values, depending on the function
opt.method	a list produced by the numerical opt.method

Background

Computational aspects of maximum likelihood estimation for univariate SN distributions are discussed in Section 3.1.7 of Azzalini and Capitanio (2014). The working of sn.mple follows these lines; maximization is performed in the CP space. All other functions operate on the DP space.

The technique underlying msn.mle is based on a partial analytical maximization, leading implicitly to a form of profile log-likelihood. This scheme is formulated in detail in Section 6.1 of Azzalini and Capitanio (1999) and summarized in Section 5.2.1 of Azzalini and Capitanio (2014). The same procedure is not feasible when one adopts MPLE; hence function msn.mple has to maximize over a larger parameter space.

When the SN family is fitted with the constraint alpha=0, this amounts to adopt a classical linear model with Gaussian distributional assumption. The corresponding MLE's are the same as those produced by lm, except that the denominator the of the MLE variance (matrix) has the 'uncorrected' form. In the multivariate case, the covariance matrix of MLE is computed using expression (10) in Section 15.8 of Magnus and Neudecker (2007).

Maximization of the univariate ST log-likelihood is speeded-up by using the expressions of the gradient given by DiCiccio and Monti (2011), reproduced with inessential variants in Section 4.3.3 of Azzalini and Capitanio (2014).

The working of `mst.mple` is based on a re-parameterization described in Section 5.1 of Azzalini and Capitanio (2003). The expressions of the corresponding log-likelihood derivatives are given in Appendix B of the full version of the paper.

Author(s)

Adelchi Azzalini

References

Azzalini, A. and Capitanio, A. (1999). Statistical applications of the multivariate skew normal distribution. *J.Roy.Statist.Soc. B* **61**, 579–602. Full-length version available at https://arXiv.org/abs/0911.2093

Azzalini, A. and Capitanio, A. (2003). Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t distribution. *J.Roy. Statist. Soc. B* **65**, 367–389. Full-length version available at https://arXiv.org/abs/0911.2342

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

DiCiccio, T. J. and Monti, A. C. (2011). Inferential aspects of the skew t-distribution. *Quaderni di Statistica* **13**, 1–21.

Magnus, J. R. and Neudecker, H. (2007). *Matrix Differential Calculus with Applications in Statistics and Econometrics*, third edition. John Wiley & Sons.

See Also

`selm` for a comprehensive higher level fitting function, `Qpenalty` for specification of a penalty function

Examples

```
data(wines, package="sn")
X <- model.matrix(~ phenols + wine, data=wines)
fit <- msn.mle(x=X, y=cbind(wines$acidity, wines$alcohol), opt.method="BFGS")
fit <- st.mple(x=X, y = wines$acidity, fixed.nu=4, penalty="Qpenalty")
```

---

sn-st.cumulants *Cumulants of univariate skew-normal and skew-t distributions*

Description

Compute cumulants of univariate (extended) skew-normal and skew-t distributions up to a given order.

Usage

```
sn.cumulants(xi=0, omega=1, alpha=0, tau=0, dp=NULL, n=4)
st.cumulants(xi=0, omega=1, alpha=0, nu=Inf, dp=NULL, n=4)
```

Arguments

xi	location parameters (numeric vector).
omega	scale parameters (numeric vector, positive).
alpha	slant parameters (numeric vector).
tau	hidden mean parameter (numeric scalar).
nu	degrees of freedom (numeric scalar, positive); the default value is nu=Inf which corresponds to the skew-normal distribution.
dp	a vector containing the appropriate set of parameters. If dp is not NULL, the individual parameters must not be supplied.
n	maximal order of the cumulants. For st.cumulants and for sn.cumulants with tau!=0 (ESN distribution), it cannot exceed 4.

Value

A vector of length n or a matrix with n columns, in case the input values are vectors.

Background

See Sections 2.1.4, 2.2.3 and 4.3.1 of the reference below

Author(s)

Adelchi Azzalini

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

dsn, dsn

Examples

```
sn.cumulants(omega=2, alpha=c(0, 3, 5, 10), n=5)
sn.cumulants(dp=c(0, 3, -8), n=6)
st.cumulants(dp=c(0, 3, -8, 5), n=6) # only four of them are computed
st.cumulants(dp=c(0, 3, -8, 3))
```

sn-st.info *Expected and observed Fisher information for* SN *and* ST *distributions*

Description

Computes Fisher information for parameters of simple sample having skew-normal (SN) or skew-t (ST) distribution or for a regression model with errors term having such distributions, in the DP and CP parametrizations.

Usage

```
sn.infoUv(dp=NULL, cp=NULL, x=NULL, y, w, penalty=NULL,  norm2.tol=1e-06)

sn.infoMv(dp, x=NULL, y, w, penalty=NULL, norm2.tol=1e-06, at.MLE=TRUE)

st.infoUv(dp = NULL, cp = NULL, x = NULL, y, w, fixed.nu = NULL,
    symmetr = FALSE, penalty = NULL, norm2.tol = 1e-06)

st.infoMv(dp, x = NULL, y, w, fixed.nu = NULL, symmetr = FALSE,
    penalty = NULL, norm2.tol = 1e-06)
```

Arguments

dp, cp	direct or centred parameters, respectively; one of them can be a non-NULL argument. For the univariate SN distribution, sn.infoUv is to be used, and these arguments are vectors. In the multivariate case, sn.infoMv is to be used and these arguments are lists. See dp2cp for their description.
x	an optional matrix which represents the design matrix of a regression model
y	a numeric vector (for sn.infoUv and st.infoUv) or a matrix (for sn.infoMv and st.infoMv) representing the response. In the SN case (sn.infoUv and sn.infoMv), y can be missing, and in this case the expected information matrix is computed; otherwise the observed information is computed. In the ST case (st.infoUv and st.infoMv), y is a required argument, since only the observed information matrix for ST distributions is implemented. See 'Details' for additional information.
w	an optional vector of weights (only meaningful for the observed information, hence if y is missing); if missing, a vector of 1's is generated.
fixed.nu	an optional numeric value which declares a fixed value of the degrees of freedom, nu. If not NULL, the information matrix has a dimension reduced by 1.
symmetr	a logical flag which indicates whether a symmetry condition of the distribution is being imposed; default is symmetr=FALSE.
penalty	a optional character string with the name of the penalty function used in the call to selm; see this function for its description.

norm2.tol — for the observed information case, the Mahalanobis squared distance of the score function from 0 is evaluated; if it exceeds norm2.tol, a warning message is issued, since the 'information matrix' so evaluated may be not positive-definite. See 'Details' for additional information.

at.MLE — a logical flag; if at.MLE=TRUE (default value), it generates a warning if the information matrix is not positive definite or an error if the observed information matrix is not evaluated at a maximum of the log-likelihood function.

Value

a list containing the following components:

dp, cp — one of the two arguments is the one supplied on input; the other one matches the previous one in the alternative parametrization.

type — the type of information matrix: "observed" or "expected".

info.dp, info.cp — matrices of Fisher (observed or expected) information in the two parametrizations.

asyvar.dp, asyvar.cp — inverse matrices of Fisher information in the two parametrizations, when available; See 'Details' for additional information.

aux — a list containing auxiliary elements, depending of the selected function and the type of computation.

Details

In the univariate SN case, when x is not set, then a simple random sample is assumed and a matrix x with a single column of all 1's is constructed; in this case, the supplied vector dp or cp must have length 3. If x is set, then the supplied vector of parameters, dp or cp, must have length ncol(x)+2. In the multivariate case, a direct extension of this scheme applies.

If the observed information matrix is required, dp or cp should represent the maximum likelihood estimates (MLE) for the given y, otherwise the information matrix may fail to be positive-definite and it would be meaningless anyway. Therefore, the squared Mahalobis norm of the score vector is evaluated and compared with norm2.tol. If it exceeds this threshold, this is taken as an indication that the supplied parameter list is not at the MLE and a warning message is issued. The returned list still includes info.dp and info.cp, but in this case these represent merely the matrices of second derivatives; asyvar.dp and asyvar.cp are set to NULL.

Background

The information matrix for the the univariate SN distribution in the two stated parameterizations in discussed in Sections 3.1.3–4 of Azzalini and Capitanio (2014). For the multivariate distribution, Section 5.2.2 of this monograph summarizes briefly the findings of Arellano-Valle and Azzalini (2008).

For ST distributions, only the observed information matrix is provided, at the moment. Computation for the univariate case is based on DiCiccio and Monti (2011). For the multivariate case, the score function is computed using an expression of Arellano-Valle (2010) followed by numerical differentiation.

References

Arellano-Valle, R. B. (2010). The information matrix of the multivariate skew-*t* distribution. *Metron*, **LXVIII**, 371–386.

Arellano-Valle, R. B., and Azzalini, A. (2008). The centred parametrization for the multivariate skew-normal distribution. *J. Multiv. Anal.* **99**, 1362–1382. Corrigendum: **100** (2009), 816.

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

DiCiccio, T. J. and Monti, A. C. (2011). Inferential aspects of the skew t-distribution. *Quaderni di Statistica* **13**, 1–21.

See Also

dsn, dmsn, dp2cp

Examples

```
infoE <- sn.infoUv(dp=c(0,1,5))           # expected information
set.seed(1); rnd <- rsn(100, dp=c(0, 1, 3))
fit <- selm(rnd~1, family="SN")
infoO <- sn.infoUv(cp=coef(fit), y=rnd)   # observed information
#
data(wines)
X <- model.matrix(~ pH + wine, data=wines)
fit <- sn.mple(x=X, y=wines$alcohol)
infoE <- sn.infoUv(cp=fit$cp, x=X)
infoO <- sn.infoUv(cp=fit$cp, x=X, y=wines$alcohol)
```

---

spread.grouped *Spreading grouped data over intervals*

Description

Assuming that counts represents the frequencies of observations falling into intervals identified by breaks, the function returns a vector of values obtained by uniformly spreading each group of data over the pertaining interval.

Usage

```
spread.grouped(breaks, counts, shift = "centre")
```

Arguments

breaks	A numeric vector of strictly increasing finite values which identify a set of contiguous intervals on the real line.
counts	A vector of non-negative integers representing the number of observations falling in the intervals specified by breaks; it is then required that length(counts)+1=length(breaks).

shift	a character string which regulates the positioning of the constructed points within a given interval, with possible values "left", "center" (default choice) and "right", possibly abbreviated.

Value

A numeric vector of length sum(counts) of values within range(breaks).

Author(s)

Adelchi Azzalini

See Also

fitdistr.grouped

Examples

```
breaks <- c(10, 12, 15, 20)
counts <- c(3, 2, 4)
spread.grouped(breaks, counts)
spread.grouped(breaks, counts, "l")
```

---

st.prelimFit — *Compute preliminary estimates for a linear model with ST-distributed error term*

Description

For a univariate or multivariate linear model where the error term is assumed to have skew-*t* (ST) distribution and the location parameter is a linear function of a set of explanatory values, the functions compute preliminary estimates to be used as initial values for a subsequent maximization of the likelihood function. These functions are mainly intended for internal package use.

Usage

```
st.prelimFit(x, y, w, quick = TRUE, verbose = 0, max.nu = 30, SN=FALSE)
mst.prelimFit(x, y, w, quick = TRUE, verbose = 0, max.nu = 30, SN=FALSE)
```

Arguments

x	design matrix of numeric values. It may be missing; if present, the first column must contain all 1's.
y	vector of observations of length n, or a matrix with n rows.
w	a vector of non-negative integer weights of length n; if missing, a vector of all 1's is generated.
quick	logical value which regulates whether a very quick estimate is produced (default value TRUE); see 'Details' for additional information.

verbose	an integer value which regulates the amount of messages printed out; default value is 0.
max.nu	threshold for the estimated degrees of freedom
SN	logical value (default value: FALSE); if TRUE, a SN distribution is assumed.

Details

The underlying methodology is the one presented by Azzalini and Salehi (2020). In its essence, it is based on the selection of parameter values achieving the best matching between certain quantile-based summaries of the ST distribution and the corresponding empirical quantities for the sample or, in the presence of explanatory variables, the same quantities computed from the residuals after fitting a median regression.

Argument quick selects whether the above-described matching is performed in a quick or in an accurate way. Since the output values of this function are intended to be only initial values for subsequent likelihood maximization, this explains the default option quick=TRUE. Other possible values are FALSE and NULL; the latter simply sets alpha=0 and nu=10.

Since the methodology hinges on some selected sample quantiles, it can occasionally be spoiled by poor behaviour of these basic quantiles, especially for small or moderate sample sizes. The more visible effect of such situation is a very large value of the estimated degrees of freedom, which then hampers rather than help a subsequent likelihood maximization. It is therefore appropriate to set an upper limit max.nu to this component.

Argument x may be missing. In this case, a one-column matrix with all elements 1 is created.

Value

A call to st.prelimFit returns a list with these components:

dp	a vector of estimates in the DP parameterization
residuals	a vector of residual values
logLik	the corresponding log-likelihood value

A call to mst.prelimFit returns a list with these components:

dp	a list with the estimates in the DP parameterization
shrink.steps	the number of shrinking steps applied to the original estimate of the scale matrix to obtain an admissible matrix
dp.matrix	a numeric matrix formed by the component-wise DP estimates
logLik	the corresponding log-likelihood value

Note

These functions are mainly intended to be called by selm, but they could be of interest for people developing their own procedures.

Author(s)

Adelchi Azzalini

References

Azzalini, A. and Salehi, M. (2020). Some computational aspects of maximum likelihood estimation of the skew-*t* distribution. In: *Computational and Methodological Statistics and Biostatistics*, edited by Andriëtte Bekker, Ding-Geng Chen and Johannes T. Ferreira. Springer. DOI: 10.1007/978-3-030-42196-0

See Also

`selm` and either `dst` or `dmst` for explanation of the DP parameters

Examples

```
data(barolo)
attach(barolo)
A75 <- (reseller=="A" & volume==75)
log.price <- log(price[A75], 10)
prelimFit <- st.prelimFit(y=log.price)
detach(barolo)
#
data(ais)
attach(ais)
prelim32 <- mst.prelimFit(y=cbind(BMI, LBM), x=cbind(1, Ht, Wt))
detach(ais)
```

---

summary.SECdistr *Summary of a* SEC *distribution object*

Description

Produce a summary of an object of class either `"SECdistrUv"` or `"SECdistrMv"`, which refer to a univariate or a multivariate SEC distribution, respectively. Both types of objects can be produced by `makeSECditr`.

Usage

```
## S4 method for signature 'SECdistrUv'
summary(object, cp.type = "auto", probs)

## S4 method for signature 'SECdistrMv'
summary(object, cp.type = "auto")
```

Arguments

`object`	an object of class `"SECdistrUv"` or `"SECdistrMv"`.
`cp.type`	a character string to select the required variance of CP parameterization; possible values are `"proper"`, `"pseudo"`, `"auto"` (default). For a description of these codes, see `dp2cp`.

probs — in the univariate case, a vector of probabilities for which the corresponding quantiles are required. If missing, the vector `c(0.05, 0.25, 0.50, 0.75, 0.95)` is used.

Details

For a description of the DP, CP and pseudo-CP parameter sets included in the returned object, see `dp2cp`.

The `aux` slot of the returned object includes other summary quantities, as described next. In the univariate case, the reported quantile-based measures of skewness and kurtosis refer to the Bowley and Moors measures, respectively; see Groeneveld (2006) and Moors (1988) for their specifications. In the multivariate case, the Mardia's measures of skewness and kurtosis are computed from the expressions given on p.153 and p.178 of Azzalini and Capitanio (2014).

In the univariate case, `delta` is a simple transformation of the slant parameter `alpha`; it takes values in $(-1, 1)$. In the multivariate case, `delta` is a vector with components of similar type; they correspond to the matching terms of the univariate components. The `alpha*` and `delta*` coefficients are univariate comprehensive summary quantities of slant; see pp.132-3 of Azzalini and Capitanio (2014) for their expressions. These quantities play an important role in SEC distributions; for instance, the Mardia's measures of multivariare skewness and kurtosis depend on the vector of slant parameters only via `delta*` or, equivalently, via `alpha*`.

The mode, which is unique for all these distributions, is computed by a numerical line search between the DP location and the CP location (or the pseudo-DP location, when the latter does exists). This line search is univariate also in the multivariate case, using Propositions 5.14 and 6.2 of Azzalini and Capitanio (2014); see also Problem 5.14.

The examples below illustrate how extract various components from `aux` and other slots of the returned object.

Value

A list with the following components:

`family`	name of the family within the SEC class, character
`dp`	DP parameters, a list or a vector
`name`	the name of the distribution, character string
`compNames`	in the multivariate case the names of the components, a character vector
`cp`	CP parameters, a list or a vector
`cp.type`	the name of the selected variant of the CP set
`aux`	a list with auxiliary ingredients (mode, coefficients of skewness and kurtosis, in the parametric and non-parametric variants, and more); see Section 'Details' for more information.

The list items `dp` and `cp` are vectors if `class(object)` is `SECdistrUv` (univariate distribution); they are lists if `class(object)` is `SECdistrMv` (multivariate distribution).

Author(s)

Adelchi Azzalini

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

Moors, I. J. A. (1988). A quantile alternative for kurtosis. *The Statistician* **37**, 25-32.

Groeneveld, R. A. (2006). Skewness, Bowley's measures of. In volume **12**, 7771-3, of *Encyclopedia of Statistical Sciences*, 2nd edition, edited by Kotz et al. Wiley, New York.

See Also

`makeSECdistr` for building a SEC distribution

`extractSECdistr` for extracting a SEC distribution from a `selm` fit

methods `mean` and `sd` for computing the mean and the standard deviation of `SECdistrUv-class` objects, methods `mean` and `vcov` for computing the mean vector and the variance matrix of `SECdistrMv-class` objects

`modeSECdistr` for computing the mode directly

Examples

```
f3 <- makeSECdistr(dp=c(3,2,5), family="SC")
summary(f3)
s <- summary(f3, probs=(1:9)/10)
print(slotNames(s))
print(names(slot(s,"aux")))   # the components of the 'aux' slot
slot(s, "aux")$mode           # the same of modeSECdistr(object=f3)
slot(s, "aux")$q.measures     # quantile-based measures of skewness and kurtosis
#
dp3 <- list(xi=1:3, Omega=toeplitz(1/(1:3)), alpha=c(-3, 8, 5), nu=6)
st3 <- makeSECdistr(dp=dp3, family="ST", name="ST3", compNames=c("U", "V", "W"))
s <- summary(st3)
dp <- slot(s, "dp")           # the same of slot(st3, "dp")
slot(s, "cp")$var.cov         # the same of vcov(st3)
slot(s, "aux")$delta.star     # comprehensive coefficient of shape
slot(s, "aux")$mardia         # Mardia's measures of skewness and kurtosis
#
dp2 <- list(xi=rep(0,2), Omega=matrix(c(2,2,2,4),2,2), alpha=c(3,-5), tau=-1)
esn2 <- makeSECdistr(dp=dp2, family="ESN", name="ESN-2d")
summary(esn2)
```

---

summary.SECdistrMv-class

Classes summary.SECdistrMv *and* summary.SECdistrUv

Description

Summaries of objects of classes SECdistrMv and SECdistrUv

Objects from the Class

Objects can be created by calls of type `summary(object)` when `object` is of class either `"SECdistrMv"` or `"SECdistrUv"`.

Slots

`family`: A character string which represents the parametric family of SEC type

`dp`: Object of class `"list"` or `"vector"` for `"SECdistrMv"` and `"SECdistrUv"`, respectively

`name`: Object of class `"character"` with the name of distribution

`compNames`: For `"SECdistrMv"` objects, a character vector with names of the components of the multivariate distribution

`cp`: Object of class `"list"` or `"vector"` for `"SECdistrMv"` and `"SECdistrUv"`, respectively

`cp.type`: a character string of the CP version

`aux`: A list of auxiliary quantities

Methods

show `signature(object = "summary.SECdistrMv")`: ...

show `signature(object = "summary.SECdistrUv")`: ...

Author(s)

Adelchi Azzalini

See Also

`summary.SECdistrMv`, `summary.SECdistrUv`,

`makeSECdistr`, `dp2cp`

---

summary.selm *Summarizing* selm *fits*

Description

`summary` method for class `"selm"` and `"mselm"`.

Usage

```
## S4 method for signature 'selm'
summary(object, param.type = "CP", cov = FALSE, cor = FALSE)

## S4 method for signature 'mselm'
summary(object, param.type = "CP", cov = FALSE, cor = FALSE)
```

Arguments

object	an object of class "selm" or "mselm" as created by a call to function selm.
param.type	a character string which indicates the required type of parameter type; possible values are "CP" (default), "DP", "pseudo-CP" and their equivalent lower-case expressions.
cov	a logical value, to indicate if an estimate of the variance and covariance matrix of the estimates is required (default: FALSE).
cor	a logical value, to indicate if an estimate of the correlation matrix of the estimates is required (default: FALSE).

Value

An S4 object of class summary.selm with 12 slots.

call:	the calling statement.
family:	the parametric family of skew-ellitically contoured distributed (SEC) type.
logL:	the maximized log-likelihood or penalized log-likelihood value
method:	estimation method ("MLE" or "MPLE")
param.type:	a characer string with the chosen parameter set.
param.table:	table of parameters, std.errors and z-values
fixed.param:	a list of fixed parameter values
resid:	residual values
control:	a list with control parameters
aux:	a list of auxiliary quantities
size:	a numeric vector with various lengths and dimensions
boundary:	a logical value which indicates whether the estimates are on the boundary of the parameter space

Note

There are two reasons why the default choice of param.type is CP. One is the the easier interpretation of cumulant-based quantities such as mean value, standard deviation, coefficient of skewness.

The other reason is more technical and applies only to cases when the estimate of the slant parameter $alpha$ of the SN distribution is close to the origin: standard asymptotic distribution theory of maximum likelihood estimates (MLE's) does not apply in this case and the corresponding standard errors are not trustworthy. The problem is especialy severe at $\alpha = 0$ but to some extent propagates to its vicinity. If $d = 1$, adoption of CP leads to MLE's with regular asymptotic distribution across the parameter space, including $\alpha = 0$. For $d > 1$ and $\alpha = 0$, the problem is still unsolved at the present time, which is the reason why selm issues a warning message when the MLE is in the vicinity of $\alpha = 0$; see 'Details' of selm. For background information, see Sections 3.1.4–6 and 5.2.3 of Azzalini and Capitanio (2014) and references therein.

This problem does not occur with the the SC and the ST distribution (unless its tail-weight parameter nu diverges, that is, when we are effectively approaching the SN case).

Author(s)

Adelchi Azzalini

References

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

`selm` function, `selm` (and `mselm`) class, `plot.selm`, `dp2cp`

Examples

```
data(wines, package="sn")
m5 <- selm(acidity ~ phenols + wine, family="SN", data=wines)
summary(m5)
summary(m5, "dp")
s5 <- summary(m5, "dp", cor=TRUE, cov=TRUE)
dp.cor <- slot(s5, "aux")$param.cor
cov2cor(vcov(m5, "dp")) # the same
#
# m6 <- selm(acidity ~ phenols + wine, family="ST", data=wines) # boundary!?
#
m12 <- selm(cbind(acidity, alcohol) ~ phenols + wine,  family="SN", data=wines)
s12 <- summary(m12)
coef(m12, 'dp')
coef(m12, "dp", vector=FALSE)
#
# see other examples at function selm
```

`summary.SUNdistr` *Summary of a* SUN *distribution object*

Description

Produce a summary of an object of class `"SUNdistr"`

Usage

```
## S4 method for signature 'SUNdistr'
summary(object, ...)
```

Arguments

`object`	an object of class `"SUNdistr"`.
`...`	optional arguments passed to `mom.mtruncnorm` for the regulation of its working.qq

Value

An S4-object with the following slots:

`dp`	the parameters of the distrbution, a list
`name`	the name of the distribution, a character string
`compNames`	the names of the components, a character vector
`HcompNames`	the names of the hidden components, a character vector
`mean`	the mean value, a vector
`var.cov`	the variance-covariance matrix
`gamma1`	the marginal indices of asymmetry, a vector
`cum3`	the third order cumulants, a three-dimensional array
`mardia`	the Mardia's measures of multivariate asymmetry and skewness, a vector of length two

Author(s)

Adelchi Azzalini

References

Arellano-Valle, R. B. and Azzalini, A. (2021). Some properties of the unified skew-normal distribution. *Statistical Papers*, doi:10.1007/s00362021012352 and arXiv:2011.06316

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

`makeSUNdistr` for building a SUN distribution object

methods `mean` and `vcov` for computing the mean vector and the variance matrix of `SUNdistr-class` objects

Examples

```
Omega <- matrix(c(5, 1, 1, 6), 2, 2)
Delta <- matrix(c(0.30, 0.50, 0.50, 0.85), 2, 2, byrow=TRUE)
Gamma <- matrix(c(1, 0.18, 0.18, 1), 2, 2)
tau <- c(0.4, -0.8)
dp2 <- list(x=c(1, 0), Omega=Omega, Delta=Delta, tau=tau, Gamma=Gamma)
sun2 <-  makeSUNdistr(dp=dp2, name="SUN2", compNames=c("u", "v"))
s <- summary(sun2)
```

summary.SUNdistr-class

Class summary.SUNdistr

Description

Summaries of objects of classes SUNdistr

Objects from the Class

Objects can be created by calls of type summary(object) when object is of class "SUNdistr".

Slots

dp: a list of parameters

name the name of the distribution, a character string

compNames the names of the components, a character vector

HcompNames the names of the hidden components, a character vector

mean the mean value, a vector

var.cov the variance-covariance matrix

gamma1 the marginal indices of asymmetry, a vector

cum3 the third order cumulants, a three-dimensional array

mardia the Mardia's measures of multivariate asymmetry and skewness, a vector of length two

Methods

show signature(object = "summary.SUNdistr"): ...

Author(s)

Adelchi Azzalini

See Also

summary.SUNdistr, makeSUNdistr

SUNdistr-base *The Unified Skew-Normal (SUN) probability distribution*

Description

Density, distribution function, random number generation, the mean value, the variance-covariance matrix and the Mardia's measures of multivariate skewness and kurtosis of the SUN probability distribution.

Usage

```
dsun(x, xi, Omega, Delta, tau, Gamma, dp = NULL, log = FALSE, silent=FALSE, ...)
psun(x, xi, Omega, Delta, tau, Gamma, dp = NULL, log = FALSE, silent=FALSE, ...)
rsun(n=1, xi, Omega, Delta, tau, Gamma, dp = NULL, silent=FALSE)
sunMean(xi, Omega, Delta, tau, Gamma, dp = NULL, silent=FALSE, ...)
sunVcov(xi, Omega, Delta, tau, Gamma, dp = NULL, silent=FALSE, ...)
sunMardia(xi, Omega, Delta, tau, Gamma, dp = NULL, silent=FALSE, ...)
```

Arguments

x	either a vector of length d, where d=ncol(Omega), with the coordinates of the point where the density or the distribution function must be evaluated, or alternatively a d-column matrix whose rows represent a set of points.
xi	a numeric vector of length d representing the location parameter of the distribution; see 'Background'. In a call to dsun and psun, xi can be a matrix, whose rows represent a set of location parameters; in this case, its dimensions must match those of x.
Omega	a symmetric positive definite matrix of dimension (d,d); see 'Details'.
Delta	a matrix of size (d,m), where m=length(tau); see 'Details' about its constraints.
tau	a vector of length m, say.
Gamma	a symmetric positive definite matrix of dimension (m,m) with 1's on its main diagonal, that is, a correlation matrix
dp	a list with five elements, representing xi (which must be a vector in this case), Omega, Delta, tau and Gamma, with restrictions indicated in the 'Details'. Its default value is NULL; if dp is assigned, the individual parameters must not be specified.
n	a positive integer value.
log	a logical value (default value: FALSE); if TRUE, log-densities and log-probabilities are returned.
silent	a logical value which indicates the action to take in the case m=1, which could be more convenently handled by functions for the SN/ESN family. If silent=FALSE (default value), a warning message is issued; otherwise this is suppressed.
...	additional tuning arguments passed either to pmnorm (for dsun, psun and sunMean) or to mom.mtruncnorm (for sunVcov and sunMardia); see also 'Details'.

Details

A member of the SUN family of distributions is identified by five parameters, which are described in the 'Background' section. The five parameters can be supplied by combining them in a list, denoted dp, in which case the individual parameters must *not* be supplied. The elements of dp must appear in the above-indicated order and must be named.

The optional arguments in ... passed to pmnorm, which uses ptriv.nt when d=3, biv.nt.prob when d=2 and and sadmvn when d>2. In practice these arguments are effective only if d>3, since for lower dimensions the computations are made to full available precision anyway. A similar fact applies to the ... argument passed to mom.mtruncnorm.

Some numerical inaccuracy is inevitably involved in these computations. In most cases, they are of negligible extent, but they can possibly become more relevant, especially in the computation of higher order moments involved by sunMardia, depending on the dimension d and on the specific parameter values. Consider the 'Warning' section in recintab which is used by mom.mtruncnorm.

The above-described functions operate following the traditional R scheme for probability distributions. Another scheme, coexisting with the classical one, works with SUNdistr-class objects, which represent SUN distributions, by encapsulating their parameters and other characteristics. These objects are created by makeSUNdistr, and various methods exist for them; see SUNdistr-class. Moreover these objects can be manipulated by a number of tools, described in SUNdistr-op, leading to new objects of the same class.

Value

The structure of the returned value depends on the called function, as follows:

dsun, psun	a vector of length nrow(x) representing density or probability values, or their log-transformed values if log=TRUE,
rsun	a matrix of size (n,d), where each row represents a SUN random vectors,
sunMean	a vector of length d representing the mean value,
sunVcov	a matrix of size (d,d) representing the variance-covariance matrix,
sunMardia	a vector of length two with the Mardia's measures of multivariate skewness and kurtosis.

Background

A member of the SUN family is characterized by two dimensionality indices, denoted d and m, and a set of five parameters blocks (vector and matrices, as explained soon). The value d represents the number of observable components; the value m represents the number of latent (or hidden) variables notionally involved in the construction of the distribution. The parameters and their corresponding R variables are as follows:

ξ	xi	a vector of length d,
Ω	Omega	a matrix of size (d, d),
Δ	Delta	a matrix of size (d, m),
τ	tau	a vector of length m,
Γ	Gamma	a matrix of size (m, m),

and must satisfy the following conditions:

1. Ω is a symmetric positive definite matrix;

2. Γ is a symmetric positive definite matrix with 1's on the main diagonal, hence a correlation matrix;
3. if $\bar{\Omega}$ denotes the correlation matrix associated to Ω, the matrix of size $(d+m) \times (d+m)$

$$\begin{pmatrix} \bar{\Omega} & \Delta \\ \Delta^\top & \Gamma \end{pmatrix}$$

must be a positive definite correlation matrix.

The formulation adopted here has arisen as the evolution of earlier constructions, which are recalled very briefly next. A number of extensions of the multivariate skew-normal distributions, all involving a number `m` (with $m \geq 1$) of latent variables (instead of `m=1` like the skew-normal distribution), have been put-forward in close succession in the years 2003-2005. Special attention has been drawn by the 'closed skew-normal (CSN)' distribution developed by González-Farías *et alii* (2004a, 2004b) and the 'fundamental skew-normal (FUSN)' distribution developed by Arellano-Valle and Genton (2005), but other formulations have been considered too.

Arellano Valle and Azzalini (2006) have shown the essential equivalence of these apparently alternative constructions, after appropriate reparameterizations, and underlined the necessity of removing over-parameterizations in some cases, to avoid lack of identifiability. This elaboration has led to the SUN formulation. A relatively less technical account of their development is provided in Section 7.1 of Azzalini and Capitanio (2014), using very slightly modified notation and parameterization, which are the ones adopted here.

Additional results have been presented by Arellano-Valle and Azzalini (2021), such as expressions for the variance matrix and higher order moments, the Mardia's measures of multivariate skewness and kurtosis, which are implemented here. Another result is the conditional distribution when the conditioning event is represented by an orthant.

Note

The present structure and user interface of this function, and of other ones related to the SUN distribution, must be considered experimental, and they might possibly change in the future.

Author(s)

Adelchi Azzalini

References

Arellano-Valle, R. B., and Azzalini, A. (2006). On the unification of families of skew-normal distributions. *Scand. J. Stat.* **33**, 561-574. Corrigendum in **49** (2022), 1418-1419.

Arellano-Valle, R. B. and Azzalini, A. (2021). Some properties of the unified skew-normal distribution. *Statistical Papers* **63**, 461-487, doi:10.1007/s00362021012352; see also arXiv:2011.06316

Arellano-Valle, R. B. and Genton, M. G. (2005). On fundamental skew distributions. *J. Multivariate Anal.* **96**, 93–1116.

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

González-Farías, G., Domínguez-Molina, J. A., & Gupta, A. K. (2004a). Additive properties of skew normal random vectors. *J. Statist. Plann. Inference* **126**, 521-534.

González-Farías, G., Domínguez-Molina, J. A., & Gupta, A. K. (2004b). The closed skew-normal distribution. In M. G. Genton (Ed.), *Skew-elliptical Distributions and Their Applications: a Journey Beyond Normality*, Chapter 2, (pp. 25–42). Chapman & Hall/CRC.

See Also

`makeSUNdistr` to build a SUN distribution object, with related methods in `SUNdistr-class`, and other facilities in `SUNdistr-op`

`convertCSN2SUNpar` to convert a parameter set of the Closed Skew-Normal formulation to the equivalent SUN parameter set

Examples

```
xi <- c(1, 0, -1)
Omega <- matrix(c(2,1,1, 1,3,1, 1,1,4), 3, 3)
Delta <- matrix(c(0.72,0.20, 0.51,0.42, 0.88, 0.94), 3, 2, byrow=TRUE)
Gamma <- matrix(c(1, 0.8, 0.8, 1), 2, 2)
dp3 <- list(xi=xi, Omega=Omega, Delta=Delta, tau=c(-0.5, 0), Gamma=Gamma)
x <- c(0.8, 0.5, -1.1)
f1 <- dsun(x, xi, Omega, Delta, c(-0.5, 0), Gamma) # mode 1
f2 <- dsun(x, dp=dp3)   # mode 2, equivalent to mode 1
set.seed(1)
xm <- rsun(10, dp=dp3)
f3 <- dsun(xm, dp=dp3)
psun(xm, dp=dp3)
sunMean(dp=dp3)
sunVcov(dp=dp3)
sunMardia(dp=dp3)
```

`SUNdistr-class` *Class "SUNdistr" and its methods*

Description

A class of objects representing Unified Skew-Normal (SUN) distributions.

Details

See `SUNdistr-base` for a description of the required structure of dp.

Note that here the methods `mean` and `vcov` are not applied to data or to a fitted model, but to a *probability distribution*, of which they provide the mean (expected) value and the variance-covariance matrix.

The object of this class follow the S4 protocol.

Objects from the class

Objects can be created by a call to the function `makeSUNdistr` or by a suitable transformation of some object of this class.

Slots

dp: a list of parameters of length five, as described in SUNdistr-base

name: a character string with the name of the multivariate variable; it can be an empty string.

compNames: a vector of character strings with the names of the component variables.

HcompNames a vector of character strings with the names of the hidden variables.

Methods

show signature(object = "SUNdistr-class"): ...

plot signature(x = "SUNdistr-class"): ...

summary signature(object = "SUNdistr-class"): ...

mean signature(x = "SUNdistr"): ...

vcov signature(object = "SUNdistr"): ...

Author(s)

Adelchi Azzalini

See Also

plot,SUNdistr-method, summary,SUNdistr-method, affineTransSUNdistr, marginalSUNdistr

convertSN2SUNdistr to convert a SECdistr object with family "SN" or "ESN" to the equivalent SUNdistr-class object

Examples

```
xi <- c(1, 0, -1)
Omega <- matrix(c(2,1,1, 1,3,1, 1,1,4), 3, 3)
Delta <- matrix(c(0.72,0.20, 0.51,0.42, 0.88, 0.94), 3, 2, byrow=TRUE)
Gamma <- matrix(c(1, 0.8, 0.8, 1), 2, 2)
dp3 <- list(xi=xi, Omega=Omega, Delta=Delta, tau=c(-0.5, 0), Gamma=Gamma)
sun3 <- makeSUNdistr(dp=dp3, name="firstSUN", compNames=c("x", "w", "z"))
show(sun3)
plot(sun3)
mean(sun3)  # the mean value of the probability distribution
vcov(sun3)  # the variance-covariance matrix of the probability distribution
summary(sun3)  # a more detailed summary
```

SUNdistr-op *Operations on SUNdistr-class objects*

Description

Given an object of SUNdistr-class, or possibly two such things in some cases, the functions perform various operations, and produce a new object of the same class.

Usage

```
affineTransSUNdistr(object, a, A, name, compNames, HcompNames, drop = TRUE)
conditionalSUNdistr(object, comp, values, eventType = "=", name, drop = TRUE)
convolutionSUNdistr(object1, object2, name, compNames, HcompNames)
joinSUNdistr(object1, object2, name, compNames, HcompNames)
marginalSUNdistr(object, comp, name, drop=TRUE)
```

Arguments

object, object1, object2	objects of class SUNdistr
a	a numeric vector; see 'Details'
A	a numeric matrix; see 'Details'
name	an optional character string with the name of the returned distribution
compNames	an optional vector of character strings with the names of the component variables of the returned distribution
HcompNames	an optional vector of character strings with the names of the hidden variables of the returned distribution
drop	a logical value (default: TRUE) relevant only in the case m=1. When both m=1 and drop=TRUE, the returned object is of class either SECdistrUv or SECdistrMv, depending on the dimension of the returned object, and family "SN" or "ESN", as appropriate.
comp	a vector of integers representing the selected components
values	a numeric vector which identifies the conditioning event
eventType	a single character value which indicates the type of the conditioning event, as described in the 'Details' section; possible values are "=" (default) and ">"

Details

For an object which represents the distribution of a multivariate SUN random variable Y of dimension d, say, a number of operations are possible, producing a new object of the same class. This object could have been created by makeSUNdistr or it could be the outcome from some previous call to one of the functions described here.

The function affineTransSUNdistr computes the distribution of $a + A'Y$, provided A is a full-rank matrix with nrow(A)=d and length(a)=ncol(A). See equation (7.6) of Azzalini & Capitanio (2014).

The function marginalSUNdistr builds a SUN distribution from the components selected by the comp vector.

A conditional distribution can be computed using conditionalSUNdistr for two type of events, selected by eventType. The "=" case corresponds to the event $Y_1 = y_1$ where Y_1 is the subset of components identified by the comp argument, y_1 is vector specified by the values argument and the equality sign must hold for each component. See equation (7.6) of Azzalini & Capitanio (2014).

If conditionalSUNdistr is used with eventType=">", the conditiong refers to the event $Y_1 > y_1$, where the inequality must be interpreted components-wise; see Arellano-Valle & Azzalini (2021) for the underlying mathematical result. If the conditional distribution is required for the reverse inequality condition, "<" say, this is equivalent to consideration of the event $-Y_1 > -y_1$. The corresponding distribution can be obtained in two steps: first a new variable is constructed reversing the sign of the required components using affineTransSUNdistr; then conditionalSUNdistr is applied to this new variable with the ">" condition and values $-y_1$. More complex conditions, where the "<" and ">" signs are mixed for different component varables, can be handled similarly, by introducing a square matrix A for affineTransSUNdistr having an appropriate combination of 1s' and -1's on its main diagonal, and 0's elsewhere, and matching changes of sign to the components of y_1.

Functions convolutionSUNdistr and joinSUNdistr operate under the assumptions that object1 and object2 refer to independent variables. Specifically, convolutionSUNdistr computes the convolution of the two objects (i.e. the distribution of the sum of two independent variables), which must have the same dimension d. Function joinSUNdistr combines two objects into a joint distribution.

If the arguments name, compNames and HcompNames are missing, they are composed from the supplied arguments.

Value

an object of SUNdistr-class

Note

The present structure and user interface of this function, and of other ones related to the SUN distribution, must be considered experimental, and they might possibly change in the future.

Author(s)

Adelchi Azzalini

References

Arellano-Valle, R. B. and Azzalini, A. (2021). Some properties of the unified skew-normal distribution. *Statistical Papers*, doi:10.1007/s00362021012352 and arXiv:2011.06316

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

See Also

SUNdistr-base, makeSUNdistr, SUNdistr-class

Examples

```
xi <- c(1, 0, -1)
Omega <- matrix(c(2,1,1, 1,3,1, 1,1,4), 3, 3)
Delta <- matrix(c(0.72,0.20, 0.51,0.42, 0.88, 0.94), 3, 2, byrow=TRUE)
Gamma <- matrix(c(1, 0.8, 0.8, 1), 2, 2)
dp3 <- list(xi=xi, Omega=Omega, Delta=Delta, tau=c(-0.5, 0), Gamma=Gamma)
sun3 <- makeSUNdistr(dp=dp3, name="SUN3", compNames=c("x", "w", "z"))
#
a <- c(1,-2)
A <- matrix(1:6, 3, 2)
sun2at <- affineTransSUNdistr(sun3, a, A, "SUN2at", compNames=c("at1", "at2"))
sun2m <- marginalSUNdistr(sun3, comp=c(1,3), name="SUN2m")
sun1c <- conditionalSUNdistr(sun3, comp=c(1,3), values=c(1.1, 0.8),
            eventType=">", name="SUN1c", drop=FALSE)
#
Omega <- matrix(c(5, 1, 1, 6), 2, 2)
Delta <- matrix(c(0.30, 0.50, 0.50, 0.85), 2, 2, byrow=TRUE)
Gamma <- matrix(c(1, 0.18, 0.18, 1), 2, 2)
tau <- c(0.4, -0.8)
dp2 <- list(x=c(1, 0), Omega=Omega, Delta=Delta, tau=tau, Gamma=Gamma)
sun2 <-  makeSUNdistr(dp=dp2, name="SUN2", compNames=c("u", "v"))
#
sun2conv <- convolutionSUNdistr(sun2, sun2m, name="SUN2sum")
sun5 <- joinSUNdistr(sun3, sun2)
```

`symm-modulated-distr` *Symmetry-modulated distributions*

Description

Symmetry-modulated distributions, univariate and multivariate, AKA skew-symmetric distributions

Usage

```
dSymmModulated(x, xi=0, omega=1, f0, G0, w, par.f0, par.G0, odd="check",
  log=FALSE, ...)
rSymmModulated(n=1, xi=0, omega=1, f0, G0, w, par.f0, par.G0, odd="check", ...)
dmSymmModulated(x, xi, Omega, f0, G0, w, par.f0, par.G0, odd="check",
  log=FALSE, ...)
rmSymmModulated(n=1, xi, Omega, f0, G0, w, par.f0, par.G0, odd="check", ...)
plot2D.SymmModulated(range, npt=rep(101,2), xi=c(0,0), Omega, f0, G0, w,
  par.f0, par.G0, odd="check", ...)
```

Arguments

`x`	a vector of coordinates where the density must be evaluated; for multivariate densities, evaluated by `dmSymmModulated`, a matrix is also allowed, each row representing a point.

xi	a numeric vector representing the location parameter; if must have length 1 for dSymmModulated and rSymmModulated, length 2 for plot2D.SymmModulated).
omega	a positive value representing the scale parameter.
f0	a character string denoting the symmetric density to be modulated; admissible values for dSymmModulated and dSymmModulated are "beta", "cauchy", "logistic" (or "logis"), "normal" (or "norm"), "t", "uniform"; for the other functions the possible values are "cauchy", "normal" (or "norm"), "t"; the meaning of the names is described in the 'Details' section.
G0	a character string denoting the symmetric distribution used in the modulating factor; admissible values are "beta", "cauchy", "logistic" (or "logis"), "normal" (or "norm"), "t", "uniform", with meaning described in the 'Details' section.
w	the name (*not* as a character string) of a user-defined function which satisfies the condition $w(-z) = -w(z)$ for all z; see the 'Details' section for additional specifications.
par.f0, par.G0	parameters required by f0 and G0, when they are of type "beta" or "t", otherwise ignored.
odd	a character string, with possible values "check" (default), "assume", "force", for regulation of the behaviour about the condition that w is an odd function, as explained in the 'Details' section.
log	logical (default: FALSE); if TRUE, densities are given as log(densities).
n	an integer value (default: n=1) indicating the number of random numbers.
Omega	a symmetric positive-definite matrix which regulates the dependence structure of f0 and so of the final density.
range	a two-column matrix whose column-wise range is taken as the plotting intervals on the coordinated axes forming a bivariate grid of points over which the density is plotted.
npt	a numeric vector with two elements representing the number of equally-spaced points on each axis spanning the range described above; default value is rep(101,2).
...	optional parameters regulating the function w and, for plot2D.SymmModulated only, graphical parameters to be supplied to function contour.

Value

For dSymmModulated, rSymmModulated and dmSymmModulated, a numeric vector; for dmSymmModulated a matrix, unless n=1.

For plot2D.SymmModulated an invisible list containing the x and y coordinates forming the grid over which the density pdf has been evaluated for plotting.

Background

In the univariate case, start from symmetric density function f_0, such that $f_0(z) = f_0(-z)$ for all z, and 'modulate' it in the form

$$f(z) = 2\, f_0(z)\, G_0\{w(z)\}$$

where G_0 is a univariate symmetric (about 0) distribution function and $w(z)$is a real-valued odd function, hence satisfying the condition $w(-z) = -w(z)$; then \$f(z)\$ is a proper density function wich integrates to 1. A subsequent location and scale transformation applied to $f(z)$ delivers the final density. Specifically, if Z denotes a univariate random variable with density $f(z)$, then the computed density pertains to the transformed variable

$$\xi + \omega Z.$$

In the multivariate case, the scheme is similar, with natural adaptation. Density f_0 is now d-dimensional, while G_0 is still univariate. The conditions $f_0(z) = f_0(-z)$ and $w(-z) = -w(z)$ refer to a d-dimensional vector z. Given a $d \times d$ symmetric positive-definite matrix Ω, we extract the the square roots ω of the diagonal element of Ω and correspondingly obtain the scale-free matrix

$$\bar{\Omega} = \mathrm{diag}(\omega)^{-1}\, \Omega\, \mathrm{diag}(\omega)^{-1}$$

which is used to regulate the dependence structure of $f_0(z)$ and so of $f(z)$. If Z is multivariate random variable with density $f(z)$, then the final distribution refers to

$$\xi + \mathrm{diag}(\omega)\, Z$$

where ξ is a d-dimensional vector of location parametes.

This construction was put forward by Azzalini and Capitanio (2003). An essentially equivalent formulation has been presented by Wang et al. (2004). A summary account is available in Section 1.2 of Azzalini and Capitanio (2014); this includes, inter alia, an explanation of why the term 'symmetry-modulated' distributions is preferred to 'skew-symmetric' distributions.

Random number generation is based on expression (1.11a) of Azzalini and Capitanio (2014).

Details

Functions `dSymmModulated` and `rSymmModulated` deal with univariate distributions, for computing densities and generating random rumbers, respectively; `dmSymmModulated` and `rmSymmModulated` act similarly for multivariate distributions. For the bivariate case only, `plot2D.SymmModulated` computes a density over a grid of coordinates and produces a `contour` plot.

The distribution names used in `f0` and `G0` have, in the univariate case, the same meaning as described in the `Distributions` page, with the following exceptions, to achive symmetry about 0: `"uniform"` denotes a uniform distribution over the interval $(-1, 1)$; `"beta"` denotes the a symmetric Beta distribution with support over the interval $(-1, 1)$ and a common value of the shape parameters.

In the multivariate case, the available options `"normal"` and `"t"` for `f0` refer to densities computed by `dmnorm` and `dmt` with 0 location and correlation matrix $\bar{\Omega}$, implied by Ω. Argument `G0` has the same meaning as in the univariate case.

Options `"beta"` and `"t"` for `f0` and `G0` require the specification of a shape parameter, via the arguments `par.f0` and `par.G0`, respectively. For `"beta"` the parameter represents the common value of the shape parameters of `Beta`; for `"t"`, it represents `df` of `TDist` and `dmt`.

Function `w` most be of the form `w <- function(z, ...)` where `...` are optional additional parameters and `z` represents valued of the standardized form of the density; in the univariate case, `x` and `z` are related by `z=(x-xi)/omega` and an analogous fact holds in the multivariate setting. The function must satisfy the condition $w(-z) = -w(z)$. It is assumed that the function is vectorized and,

in the multivariate case, it will be called with z representing a matrix with d columns, if d denotes the dimensionality of the random variable.

Argument odd regulates the behaviour with respect to the condition $w(-z) = -w(z)$. If its value is "assume", the condition is just assumed to hold, and no action is taken. If the value is "check" (deafult), a *limited* check is performed; namely, in case of densities, the check is at 0 and the supplied x points, while for random numbers the check is at 0 and the generated points. The value "force" ensures that the condition is satisfied by actually constructing a modified version of the user-supplied function w, such that the required condition is enforced.

Author(s)

Adelchi Azzalini

References

Arellano-Valle, R. B., Gómez, H. W. and Quintana, F. A. (2004). A new class of skew-normal distributions. *Comm. Stat., Theory & Methods*, **58**, 111-121.

Azzalini, A. and Capitanio, A. (2003). Distributions generated by perturbation of symmetry with emphasis on a multivariate skew-*t* distribution. *J.Roy. Statist. Soc. B* **65**, 367–389. Full version of the paper at https://arXiv.org/abs/0911.2342

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

Wang, J., Boyer, J. and Genton, M. G. (2004). A skew-symmetric representation of multivariate distributions. *Statistica Sinica*, **14**, 1259-1270.

See Also

Distributions, Beta, TDist, dmnorm, dmt, contour

Examples

```
x <- seq(2, 13, length=45)
wLinear <- function(z, lambda) lambda*z
y <- dSymmModulated(x, 5, 2, f0="normal", G0="normal", w=wLinear, lambda=3)
# the same of dsn(x, 5, 2, 3), up to negligible numerical differences
#
wSGN <- function(z, lambda) z*lambda[1]/sqrt(1 + lambda[2]*z^2)
y <- dSymmModulated(x, 5, 2, f0="normal", G0="normal", w=wSGN, lambda=c(3,5))
# SGN distribution of Arellano-Valle et al. (2004)
#
wST <- function(z, lambda, nu) lambda*z*sqrt((nu+1)/(nu+z^2))
y <- rSymmModulated(n=100, 5, 2, f0="t", G0="t", w=wST, par.f0=8, par.G0=9,
        lambda=3, nu=8)
# equivalent to rst(n=100, 5, 2, 3, 8)
#
wTrigs <- function(z, p, q) sin(z * p)/(1 + cos(z * q))
x <- seq(-1, 1, length=51)
y <- dSymmModulated(x, 0, 1, f0="beta", G0="logistic", w=wTrigs, par.f0=2,
        par.G0=NULL, p=5, q=0.5)
plot(x, y, type="l")
```

```
# univariate analogue of the bivariate distribution on pp.372-3 of
# Azzalini & Capitanio (2003)
#
range <- cbind(c(-3,3), c(-3,3))
wMvTrigs <- function(z, p, q) sin(z %*% p)/(1 + cos(z %*% q))
plot2D.SymmModulated(range, xi=c(0,0), Omega=diag(2), f0="normal", G0="normal",
    w=wMvTrigs, par.f0=NULL, par.G0=NULL, p=c(2,3), q=c(1,1), col=4)
# w(.) as in (1.6) of Azzalini & Capitanio (2014, p.4) and plot as in
# bottom-right panel of their Figure 1.1.
```

`T.Owen` *Owen's function*

Description

Evaluates function $T(h, a)$ studied by D.B.Owen

Usage

```
T.Owen(h, a, jmax=50, cut.point=8)
```

Arguments

`h`	a numeric vector. Missing values (`NA`s) and `Inf` are allowed.
`a`	a numeric value. `Inf` is allowed.
`jmax`	an integer scalar value which regulates the accuracy of the result. See Section 'Details' below for explanation.
`cut.point`	a scalar value which regulates the behaviour of the algorithm, as explained in Section 'Details' below (default value: `8`).

Details

If `a>1` and `0<h<=cut.point`, a series expansion is used, truncated after `jmax` terms. If `a>1` and `h>cut.point`, an asymptotic approximation is used. In the other cases, various reflection properties of the function are exploited. See the reference below for more information.

Value

a numeric vector.

Background

The function *T(h,a)* studied by Owen (1956) is useful for the computation of the bivariate normal distribution function and related quantities, including the distribution function of a skew-normal variate; see `psn`. See the reference below for more information on function $T(h, a)$.

Author(s)

Adelchi Azzalini and Francesca Furlan

References

Owen, D. B. (1956). Tables for computing bivariate normal probabilities. *Ann. Math. Statist.* **27**, 1075-1090.

See Also

psn

Examples

```
owen <- T.Owen(1:10, 2)
```

---

wines *Piedmont wines data*

Description

Data refer to chemical properties of 178 specimens of three types of wine produced in the Piedmont region of Italy.

Usage

```
data(wines)
```

Format

A data frame with 178 observations on the following 28 variables.

wine	wine name (categorical, levels: Barbera, Barolo, Grignolino)
alcohol	alcohol percentage (numeric)
sugar	sugar-free extract (numeric)
acidity	fixed acidity (numeric)
tartaric	tartaric acid (numeric)
malic	malic acid (numeric)
uronic	uronic acids (numeric)
pH	pH (numeric)
ash	ash (numeric)
alcal_ash	alcalinity of ash (numeric)
potassium	potassium (numeric)
calcium	calcium (numeric)
magnesium	magnesium (numeric)
phosphate	phosphate (numeric)
cloride	chloride (numeric)

phenols	total phenols (numeric)
flavanoids	flavanoids (numeric)
nonflavanoids	nonflavanoid phenols (numeric)
proanthocyanins	proanthocyanins (numeric)
colour	colour intensity (numeric)
hue	hue (numeric)
OD_dw	OD_{280}/OD_{315} of diluted wines (numeric)
OD_fl	OD_{280}/OD_{315} of flavanoids (numeric)
glycerol	glycerol (numeric)
butanediol	2,3-butanediol (numeric)
nitrogen	total nitrogen (numeric)
proline	proline (numeric)
methanol	methanol (numeric)

Details

The data represent 27 chemical measurements on each of 178 wine specimens belonging to three types of wine produced in the Piedmont region of Italy. The data have been presented and examined by Forina *et al.* (1986) and were freely accessible from the PARVUS web-site until it was active. These data or, more often, a subset of them are now available from various places, including some R packages. The present dataset includes all variables available on the PARVUS repository, which are the variables listed by Forina *et al.* (1986) with the exception of 'Sulphate'. Moreover, it reveals the undocumented fact that the original dataset appears to include also the vintage year; see the final portion of the 'Examples' below.

Source

Forina, M., Lanteri, S. Armanino, C., Casolino, C., Casale, M. and Oliveri, P. V-PARVUS 2008: an extendible package of programs for esplorative data analysis, classification and regression analysis. Dip. Chimica e Tecnologie Farmaceutiche ed Alimentari, Università di Genova, Italia. Web-site (not accessible as of 2014): '`http://www.parvus.unige.it`'

References

Forina M., Armanino C., Castino M. and Ubigli M. (1986). Multivariate data analysis as a discriminating method of the origin of wines. *Vitis* **25**, 189–201.

Examples

```
data(wines)
pairs(wines[,c(2,3,16:18)], col=as.numeric(wines$wine))
#
code <- substr(rownames(wines), 1, 3)
table(wines$wine, code)
#
year <- as.numeric(substr(rownames(wines), 6, 7))
table(wines$wine, year)
# coincides with Table 1(a) of Forina et al. (1986)
```

`zeta` *Function* $\log(2\,\Phi(x))$ *and its derivatives*

Description

The function `log(2*pnorm(x))` and its derivatives, including inverse Mills ratio.

Usage

```
zeta(k, x)
```

Arguments

`k`	an integer number between 0 and 5.
`x`	a numeric vector. Missing values (`NA`s) and `Inf`s are allowed.

Details

For `k` between 0 and 5, the derivative of order `k` of $\log(2\,\Phi(x))$ is evaluated, where $\Phi(x)$ denotes the $N(0,1)$ cumulative distribution function. The derivative of order `k=0` refers to the function itself. If `k` is not an integer within `0, . . . , 5`, `NULL` is returned.

Value

a vector representing the `k`-th order derivative evaluated at `x`.

Background

The computation for `k>1` is reduced to the case `k=1`, making use of expressions given by Azzalini and Capitanio (1999); see especially Section 4 of the full-length version of the paper. The main facts are summarized in Section 2.1.4 of Azzalini and Capitanio (2014).

For numerical stability, the evaluation of `zeta(1,x)` when `x < -50` makes use of the asymptotic expansion (26.2.13) of Abramowitz and Stegun (1964).

`zeta(1,-x)` equals `dnorm(x)/pnorm(-x)` (in principle, apart from the above-mentioned asymptotic expansion), called the *inverse Mills ratio*.

References

Abramowitz, M. and Stegun, I. A., editors (1964). *Handbook of Mathematical Functions*. Dover Publications.

Azzalini, A. and Capitanio, A. (1999). Statistical applications of the multivariate skew normal distribution. *J.Roy.Statist.Soc. B* **61**, 579–602. Full-length version available at https://arXiv.org/abs/0911.2093

Azzalini, A. with the collaboration of Capitanio, A. (2014). *The Skew-Normal and Related Families*. Cambridge University Press, IMS Monographs series.

Examples

```
y <- zeta(2,seq(-20,20,by=0.5))
#
for(k in 0:5) curve(zeta(k,x), from=-1.5, to=5, col = k+2, add = k > 0)
legend(3.5, -0.5, legend=as.character(0:5), col=2:7, lty=1)
```

Index