Package 'sn'

November 11, 2015

Version 1.3-0

Date 2015-11-11

Title The Skew-Normal and Skew-t Distributions

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

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

Depends R (>= 2.15.3), methods, stats4

Imports mnormt, numDeriv, stats, grDevices, graphics, utils

Description Build and manipulate probability distributions of the skew-normal
family and some related ones, notably the skew-t family, and provide related
statistical methods for data fitting and 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

Repository CRAN

Date/Publication 2015-11-11 10:01:17

R topics documented:

sn-package   *Package* **sn***: overview*

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) family, and to apply connected statistical methods for data fitting and diagnostics, in the univariate and the multivariate case.

Development and basic facts

The first version of the package was written in 1997 (on CRAN since 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'. 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.

Broadly speaking, the available tools can be divided in two groups: the probability section and the statistics section. For a quick start, one could look at their key functions, makeSECdistr and selm,

respectively, and from here explore the rest. In the probability section, one finds also functions dsn, dst, dmsn and others alike; these functions existed also in 'version 0' and their working is still very much the same (not necessarily so their code).

The first version of 'version 1' series (that is, 1.0-0) has appeared at the same time when the companion book by Azzalini and Capitanio (2014) was published. Although the two projects are formally separate, they adopt the same notation, terminology and logical frame. This matching and the numerous references in the software documentation to specific sections of the book for background information should facilitate familiarizing with these tools.

Information on additional and on more recent change to the package is provided in NEWS file, accessible from the package documentation index page.

A word of explanation is appropriate about the numerous references to Azzalini and Capitanio (2014) in the documentation of the package. The reason why the documentation often refers to the monograph rather than to the original research papers is because the book provides a relatively informal summary of material which has been elaborated in a number of technical papers, sometimes very technical or with information on the point of interest mixed with other material. In other words, the motivation behind this policy is readibility, 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 quoted sections of the book include bibliographic notes which refer back to the original sources.

Backward Compatibility of 'version 1.x-y'

There is a partial backward compatibility of 'version 1.x-y' versus 'version 0-4.18'. 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) and dp.to.cp, which has been replaced by the more general function dp2cp.

Finally, some functions are not there any longer, with no similarly-working functions in the new version. The more prominent set of cases is represented by the functions for computing profile log-likelihoods. There is a long-term plan to re-instate similar facilities, possibly in a more flexible form, but not in the near future.

Requirements

R version 2.15-3 or higher, plus packages **mnormt**, **numDeriv**, **stats4** in addition to standard packages (**methods**, **graphics**, 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 additional 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

This package and its documentation are usable under the terms of the "GNU General Public License" version 3 or version 2, as you prefer; a copy of them is available from http://www. R-project.org/Licenses/.

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.

---

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

Description

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

Usage

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

Arguments

object	an object of class SECdistrMv, as created by makeSECdistr 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 of 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 is has dimension 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](#), [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	a factor with levels: `female`, `male`
[,2]	sport	a factor with 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)
```

---

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=)
```

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 http://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, dmnorm, 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=)
```

Function pmst requires [dmt](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 and 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 2nd and 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 (NAs) are allowed
xi	vector of location parameters.
omega	vector of scale parameters; must be positive.

alpha	vector of slant parameters; `+/- Inf` is allowed. With `psn` and `qsn`, it must be of length 1 if `engine="T.Owen"`.
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	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.

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, ...)
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	sample size
log	logical; if TRUE, densities are given as log-densities
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 4 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, which depends on the other arguments.
...	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. 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. Of these, Method 1 and 4 are only suitable for integer values of nu. Method 4 becomes progressively less efficient as nu increases, because its value corresponds to 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, length(x). 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, 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 http://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 term 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 distribution

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)
```

---

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 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)
```

---

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 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.

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.

Among the admissible families, the ESN distribution is not, strictly speaking, of SEC type, but it is nevertheless included because of its strong connection.

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"))
```

---

matrix-op   *vech and other matrix operators*

Description

vech and other matrix operators

Usage

```
vech(A)
vech2mat(v)
duplicationMatrix(n)
```

Arguments

A	a (symmetric) square 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

Value

a vector in case of `vech`, 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, `vect2mat(v)` performs the inverse operation and returns `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).

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)
```

---

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 spe-
              cific 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

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"))
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")
```

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

Description

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 super-imposed 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 http://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 http://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)
```

---

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, level,
   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 parameters must be log-transformed, if case any of them occurs in `param.name`. This applied to `omega` and `nu` in the DP parameter set and to `s.d.` and `gamma2` in the CP parameter set.

level   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

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

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. 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 more information on this point, 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 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 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

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.

- 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)
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, 0.5, by=0.1))

# reduce grid points to save 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(11,16) )
```

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 an 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.

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

Multivariate skew-elliptically contoured 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 = "SECdistrUv"):...

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

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

Univariate skew-elliptically contoured 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 extractSECdistr.

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,
  offset, model = TRUE, x = FALSE, y = FALSE,  ...)
```

Arguments

formula   an object of class "[formula](#)" (or one that can be coerced to that class): a sym-
               bolic 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 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 distri-
               butions; notice that 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 envi-
               ronment 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.
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 values for the search of the parameter estimates. If start=NULL (default), initial values are selected by the procedure.
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 likekelihood 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.
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.
model, x, y	logicals. If TRUE, the corresponding components of the fit are returned.
...	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 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.

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 available in Sections 3.1.4–6 and 5.2.3 of Azzalini and Capitanio (2014) and in Section 4 of Arellano-Valle and Azzalini (2008).

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 exists, it differs from Ω only by a multiplicative factor. For this reason, 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(..).

In some cases, especially for small sample size, the MLE occurs on the frontier of the parameter space, leading to DP estimates with 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 $\log \tilde{L}$ 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 $\log \tilde{L} - \tilde{Q}$. 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 depends 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.

Warning

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. Both consideration of model simplicity and 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. Details on the numerical optimization which has produced object obj can be estracted with slot(obj, "opt.method"). 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. Two cases of this sort are presented in the final portion of the examples below.

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.

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

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

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`.
- 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

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")
summary(m)
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)


# example of computation and plot of a (relative twice) profile log-likelihood;
# since it takes some time, set a coarse grid of nu values
nu.vector <- seq(3, 8, by=0.5)
logL <- numeric(length(nu.vector))
for(k in 1:length(nu.vector)) {
   m28.f <- selm(cbind(chloride, glycerol, magnesium) ~ 1, family="ST",
         fixed=list(nu=nu.vector[k]),  data=wines)
   logL[k] <- logLik(m28.f)
   cat(format(c(nu.vector[k], logL[k])), "\n")
}
plot(nu.vector, 2*(logL-max(logL)), type="b")
ok <- which.max(logL)
abline(v=nu.vector[ok], lty=2)
# compare maximum of this curve with MLE of nu in summary(m28, 'dp')


#
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
#

# illustrate final passage of 'Warning' section above:
# the execution of the next selm command is known to produce warning messages
# although the optimizer declares successful convergence
m31 <- selm(cbind(BMI, LBM)~ Ht + Wt, family="ST", data=ais)
# Warning message...
slot(m31, "opt.method")$convergence   # a 0 value indicates success
# the next case is similar
m32 <- selm(cbind(BMI, LBM)~ Ht + Wt, family="ST", data=ais, opt.method="BFGS")
# Warning message...
slot(m32, "opt.method")$convergence
```

---

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"): ...

                          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 meth-
ods for the returned object.

Usage

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

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.
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", "E 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.

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 http://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 <http://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.

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 0 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)

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 the two vectors must be supplied, but not both. 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; if missing, a vector of 1's is generated.

fixed.nu   an optional numeric value which declared 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 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.

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 dp should represent the maximum likelihood estimates (MLE) for the given y, otherwise the information matrix may fail to be positive-definite. Therefore, the squared Mahalobis norm of the score vector is evaluated and compared with norm2.tol. If it exceeds this threshold, it is taken as an indication that the parameter 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 currently. Computation for the univariate case is based on DiCiccio and Monti (2011). For the multivariate case, the score function is computed using 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: vol.\,100 (2009), p.\,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))
infoO <- sn.infoUv(cp=c(0,1,0.8), y=rsn(50, dp=c(0,1,5)))
#
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)
```

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.

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).

DP and CP are vectors if class(object) is SECdistrUv (univariate distribution); they are lists if codeclass(object) is SECdistrMv (multivariate distribution). The examples below show how extract components from aux and other slots.

Author(s)

Adelchi Azzalini

See Also

[makeSECdistr](#) for extracting a SEC distribution from a [selm](#) fit

methods [mean](#) and [vcov](#) for computing the mean (vector) and the variance (matrix) of [SECdistrUv-class](#) and [SECdistrMv-class](#) objects

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", 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 asymmetry 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
```

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 numerical vector. Missing values (NAs) and Inf are allowed.
a	a numerical scalar. Inf is allowed.
jmax	an integer scalar value which regulates the accuracy of the result. See the section Details below for explanation.
cut.point	a scalar value which regulates the behaviour of the algorithm, as explained by the 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 numerical 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](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 variable with 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 (NAs) and Infs 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 integer, it is converted to integer and a warning message is generated. If k<0 or k>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 http://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