Package 'SHELF'

June 18, 2021

Type Package

Title Tools to Support the Sheffield Elicitation Framework

Version 1.8.0

Date 2021-06-18

Author Jeremy Oakley

Maintainer Jeremy Oakley <j.oakley@sheffield.ac.uk>

Description Implements various methods for eliciting a probability distribution for a single parameter from an expert or a group of experts. The expert provides a small number of probability judgements, corresponding to points on his or her cumulative distribution function. A range of parametric distributions can then be fitted and displayed, with feedback provided in the form of fitted probabilities and percentiles. For multiple experts, a weighted linear pool can be calculated. Also includes functions for eliciting beliefs about population distributions, eliciting multivariate distributions using a Gaussian copula, eliciting a Dirichlet distribution, and eliciting distributions for variance parameters in a random effects meta-analysis model. R Shiny apps for most of the methods are included.

License GPL-2 | GPL-3

Depends R (>= 3.5.0)

RoxygenNote 7.1.1

Suggests testthat, knitr, GGally

Imports ggplot2, grid, shiny, stats, graphics, tidyr, MASS, ggExtra, gridExtra, scales, rmarkdown, grDevices, shinyMatrix, utils, ggridges, Hmisc

URL https://github.com/OakleyJ/SHELF

BugReports https://github.com/OakleyJ/SHELF/issues

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-06-18 16:20:02 UTC

R topics documented:

`SHELF-package` *Tools to Support the Sheffield Elicitation Framework*

Description

Implements various methods for eliciting a probability distribution for a single parameter from an expert or a group of experts. The expert provides a small number of probability judgements, corresponding to points on his or her cumulative distribution function. A range of parametric distributions can then be fitted and displayed, with feedback provided in the form of fitted probabilities and percentiles. For multiple experts, a weighted linear pool can be calculated. Also includes functions

for eliciting beliefs about population distributions, eliciting multivariate distributions using a Gaussian copula, eliciting a Dirichlet distribution, and eliciting distributions for variance parameters in a random effects meta-analysis model. R Shiny apps for most of the methods are included.

Package:	SHELF
Type:	Package
Version:	1.8.0
Date:	2021-06-18
License:	GPL-2 | GPL-3

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

References

The SHELF homepage

Examples

```
## Not run:
## 1) Elicit judgements from two experts individually
# Expert A states P(X<30)=0.25, P(X<40)=0.5, P(X<50)=0.75
# Expert B states P(X<20)=0.25, P(X<25)=0.5, P(X<35)=0.75
# Both experts state 0<X<100.

## 2) Fit distributions to each expert's judgements
v <- matrix(c(30, 40, 50, 20, 25, 35), 3, 2)
p <- c(0.25, 0.5, 0.75)
myfit <- fitdist(vals = v, probs = p, lower = 0, upper = 100)

## 3) Plot the fitted distributions, including a linear pool
plotfit(myfit, lp = T)

## 4) Now elicit a single 'consensus' distribution from the two experts
# Suppose they agree P(X<25)=0.25, P(X<30)=0.5, P(X<40)=0.75
v <-c(25, 30, 40)
p <-c(0.25, 0.5, 0.75)
myfit <- fitdist(vals = v, probs = p, lower = 0, upper = 100)

## 5) Plot the fitted density, and report some feedback, such as the
# fitted 5th and 95th percentiles
plotfit(myfit, ql = 0.05, qu = 0.95)
feedback(myfit, quantiles = c(0.05, 0.95))

## Can also use interactive plotting
v <- matrix(c(30, 40, 50, 20, 25, 35), 3, 2)
p <- c(0.25, 0.5, 0.75)
myfit <- fitdist(vals = v, probs = p, lower = 0, upper = 100)
# plot each distribution
plotfit(myfit, int = TRUE)
```

```
## plot the distribution for one expert only
plotfit(myfit, int = TRUE, ex = 1)

## Enter judgements in interactive mode
elicit()

#' ## Enter separate judgements for each expert in interactive mode
elicitMultiple()


## End(Not run)
```

---

cdffeedback *Feedback for the elicited distribution of the population CDF*

Description

Report the median and 100(1-alpha)% credible interval for point on the population CDF

Usage

```
cdffeedback(
  medianfit,
  precisionfit,
  quantiles = c(0.05, 0.95),
  vals = NA,
  alpha = 0.05,
  median.dist = "best",
  precision.dist = "gamma",
  n.rep = 10000
)
```

Arguments

`medianfit`	The output of a fitdist command following elicitation of the expert's beliefs about the population median.
`precisionfit`	The output of a fitprecision command following elicitation of the expert's beliefs about the population precision.
`quantiles`	A vector of quantiles $q_1, \ldots, q_n$ required for feedback
`vals`	A vector of population values $x_1, \ldots, x_n$ required for feedback
`alpha`	The size of the 100(1-alpha)% credible interval
`median.dist`	The fitted distribution for the population median. Can be one of `"normal"`, `"lognormal"` or `"best"`, where `"best"` will select the best fitting out of normal and lognormal.
`precision.dist`	The fitted distribution for the population precision. Can either be `"gamma"` or `"lognormal"`.

`n.rep` The number of randomly sampled CDFs used to estimated the median and credible interval.

Details

Denote the uncertain population CDF by

$$P(X \leq x|\mu, \sigma^2),$$

where μ is the uncertain population median and $\sigma^(- 2)$ is the uncertain population precision. Feedback can be reported in the form of the median and 100(1-alpha)% credible interval for (a) an uncertain probability $P(X \leq x|\mu, \sigma^2)$, where x is a specified population value and (b) an uncertain quantile x_q defined by $P(X \leq x_q|\mu, \sigma^2) = q$, where q is a specified population probability.

Value

Fitted median and 100(1-alpha)% credible interval for population quantiles and probabilities.

`$quantiles` Each row gives the fitted median and 100(1-alpha)% credible interval for each uncertain population quantile specified in `quantiles`: the fitted median and 100(1-alpha)% credible interval for the value of x_{q_i} where $P(X \leq x_{q_i}|\mu, \sigma^2) = q_i$.

`$probs` Each row gives the fitted median and 100(1-alpha)% credible interval for each uncertain population probability specified in `probs`: the fitted median and 100(1-alpha)% credible interval for the value of $P(X \leq x_i|\mu, \sigma^2)$.

Examples

```
## Not run:
prfit <- fitprecision(interval = c(60, 70), propvals = c(0.2, 0.4), trans = "log")
medianfit <- fitdist(vals = c(50, 60, 70), probs = c(0.05, 0.5,  0.95), lower = 0)
cdffeedback(medianfit, prfit, quantiles = c(0.01, 0.99),
            vals = c(65, 75), alpha = 0.05, n.rep = 10000)

## End(Not run)
```

`cdfplot` *Plot distribution of CDF*

Description

Plot the elicited pointwise median and credible interval for an uncertain population CDF

Usage

```
cdfplot(
  medianfit,
  precisionfit,
  lower = NA,
  upper = NA,
  ql = 0.025,
  qu = 0.975,
  median.dist = "best",
  precision.dist = "gamma",
  n.rep = 10000,
  n.X = 100,
  fontsize = 18
)
```

Arguments

medianfit	The output of a fitdist command following elicitation of the expert's beliefs about the population median.
precisionfit	The output of a fitdist command following elicitation of the expert's beliefs about the population precision.
lower	lower limit on the x-axis for plotting.
upper	upper limit on the x-axis for plotting.
ql	lower quantile for the plotted pointwise credible interval.
qu	upper quantile for the plotted pointwise credible interval.
median.dist	The fitted distribution for the population median. Can be one of "normal", "lognormal" or "best", where "best" will select the best fitting out of normal and lognormal.
precision.dist	The fitted distribution for the population precision. Can either be "gamma" or "lognormal".
n.rep	The number of randomly sampled CDFs used to estimated the median and credible interval.
n.X	The number of points on the x-axis at which the CDF is evaluated.
fontsize	Font size used in the plots.

Examples

```
## Not run:
prfit <- fitprecision(interval = c(60, 70), propvals = c(0.2, 0.4), trans = "log")
medianfit <- fitdist(vals = c(50, 60, 70), probs = c(0.05, 0.5,  0.95), lower = 0)
cdfplot(medianfit, prfit)

## End(Not run)
```

`compareIntervals` *Plot fitted intervals for each expert*

Description

Following elicitation of distributions from individual experts, plot fitted probability intervals for each expert.

Usage

```
compareIntervals(
  fit,
  interval = 0.95,
  dist = "best",
  fs = 12,
  xlab = "x",
  ylab = "expert"
)
```

Arguments

`fit`	An object of class `elicitation`
`interval`	The probability p for each interval (i.e. the fitted probability for each expert that the displayed interval contains the uncertain quantity will be p)
`dist`	The distribution fitted to each expert's probabilities. Options are `"normal"`, `"t"`, `"gamma"`, `"lognormal"`, `"logt"`,`"beta"`, and `"best"` (for best fitting). Can be a vector if different distributions are desired for each expert.
`fs`	font size used in the plot.
`xlab`	A string or expression giving the x-axis label.
`ylab`	A string or expression giving the y-axis label.

Examples

```
## Not run:
v <- matrix(c(30, 40, 50, 20, 25, 35, 40, 50, 60, 35, 40, 50), 3, 4)
p <- c(0.25, 0.5, 0.75)
myfit <- fitdist(vals = v, probs = p, lower = 0, upper = 100)
compareIntervals(myfit, interval = 0.5)

## End(Not run)
```

`condDirichlet` *Plot conditional distributions from an elicited Dirichlet prior*

Description

Opens up a web browser (using the shiny package), from which you can choose to condition on one of the category probability values, and then display the resulting conditional marginal distributions for the remaining categories

Usage

```
condDirichlet(d)
```

Arguments

`d`	A fitted Dirichlet distribution, produced from a `fitDirichlet` command.

Details

Press Esc in the R console window to exit the elicitation session.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
p1 <- c(0.25, 0.5, 0.75)
v1 <- c(0.5, 0.55, 0.6)
v2 <- c(0.22, 0.3, 0.35)
v3 <- c(0.11, 0.15, 0.2)
myfit1 <- fitdist(v1, p1, 0, 1)
myfit2 <- fitdist(v2, p1, 0, 1)
myfit3 <- fitdist(v3, p1, 0, 1)
d <- fitDirichlet(myfit1, myfit2, myfit3,
                  categories = c("A","B","C"),
                  n.fitted = "opt")
condDirichlet(d)

## End(Not run)
```

copulaSample *Generate correlated samples from elicited marginal distributions using a multivariate normal copula*

Description

Takes elicited marginal distributions and elicited concordance probabilities: pairwise probabilities of two uncertain quantities being greater than their medians, and generates a correlated sample, assuming the elicited marginal distributions and a multivariate normal copula

Usage

```
copulaSample(..., cp, n, d = NULL)
```

Arguments

...	A list of objects of class elicitation. command, one per marginal distribution, separated by commas.
cp	A matrix of pairwise concordance probabilities, with element i,j the elicited probability P(X_i > m_i, X_j > m_j or X_i < m_i, X_j < m_j), where m_i and m_j are the elicited medians of the uncertain quantities X_i and X_j. Only the upper triangular elements in the matrix need to be specified; the remaining elements can be set at 0.
n	The sample size to be generated
d	A vector of distributions to be used for each elicited quantity: a string with elements chosen from "normal","t","gamma","lognormal","logt","beta". The default is to use the best fitting distribution in each case.

Value

A matrix of sampled values, one row per sample.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
p1 <- c(0.25, 0.5, 0.75)
v1 <- c(0.5, 0.55, 0.6)
v2 <- c(0.22, 0.3, 0.35)
v3 <- c(0.11, 0.15, 0.2)
myfit1 <- fitdist(v1, p1, 0, 1)
myfit2 <- fitdist(v2, p1, 0, 1)
myfit3 <- fitdist(v3, p1, 0, 1)
quad.probs <- matrix(0, 3, 3)
```

```
quad.probs[1, 2] <- 0.4
quad.probs[1, 3] <- 0.4
quad.probs[2, 3] <- 0.3
copulaSample(myfit1, myfit2, myfit3, cp=quad.probs, n=100, d=NULL)

## End(Not run)
```

`elicit` *Elicit judgements and fit distributions interactively*

Description

Opens up a web browser (using the shiny package), from which you can specify judgements, fit distributions and plot the fitted density functions with additional feedback. Probabilities can be specified directly, or the roulette elicitation method can be used.

Usage

```
elicit()
```

Details

Click on the "Help" tab for instructions. Click the "Quit" button to exit the app and return the results from the `fitdist` command. Click "Download report" to generate a report of all the fitted distributions.

Value

An object of class `elicitation`, which is returned once the Quit button has been clicked. See `fitdist` for details.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:

elicit()


## End(Not run)
```

`elicitBivariate` *Elicit a bivariate distribution using a Gaussian copula*

Description

Opens up a web browser (using the shiny package), from which you can specify judgements, fit distributions, plot the fitted density functions, and plot samples from the joint distributions. A joint distribution is constructed using a Gaussian copula, whereby the correlation parameter is determined via the elicitation of a concordance probability (a probability that the two uncertain quantities are either both greater than their medians, or both less than their medians.)

Usage

```
elicitBivariate()
```

Details

Click on the "Help" tab for instructions. Click the "Quit" button to exit the app and return the results from the `fitdist` command. Click "Download report" to generate a report of all the fitted distributions for each uncertain quantity, and "Download sample" to generate a csv file with a sample from the joint distribution.

Value

A list, with two objects of class `elicitation`, and the elicited concordance probability. See `fitdist` for details.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:

elicit()


## End(Not run)
```

`elicitConcProb` *Elicit a concordance probability for two uncertain quantities, and plot a joint sample*

Description

Given two elicited marginal distributions, open a browser in which one specifies a quadrant probability P(X_1 > m_1, X_2 > m_2), where m_1 and m_2 are the elicited medians of X_1 and X_2. A joint sample from the distribution of X_1 and X_2 is generated, using the two elicited marginal distributions and a bivariate normal copula.

Usage

```
elicitConcProb(fit1, fit2, m1, m2, d = c("best", "best"), n = 10000)
```

Arguments

`fit1`	An elicitation fit produced from the `fitdist` command for the first uncertain quantity X_1.
`fit2`	An elicitation fit produced from the `fitdist` command for the second uncertain quantity X_2.
`m1`	The elicited (or fitted) median of X_1.
`m2`	The elicited (or fitted) median of X_2.
`d`	A vector of distributions to be used for each elicited quantity: a string with elements chosen from `"normal"`,`"t"`,`"gamma"`,`"lognormal"`,`"logt"`,`"beta"`,`"hist"`. The default is to use the best fitting distribution in each case.
`n`	The number of sampled (X_1, X_2) pairs to be plotted.

Value

A matrix of sampled values, one row per sample.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
p1 <- c(0.25, 0.5, 0.75)
v1 <- c(0.5, 0.55, 0.6)
v2 <- c(0.22, 0.3, 0.35)
myfit1 <- fitdist(v1, p1, 0, 1)
myfit2 <- fitdist(v2, p1, 0, 1)
elicitConcProb(myfit1, myfit2, 0.55, 0.3, d=c("beta", "beta"))

## End(Not run)
```

elicitDirichlet *Elicit a Dirichlet distribution interactively*

Description

Opens up a web browser (using the shiny package), from which you can elicit a Dirichlet distribution

Usage

```
elicitDirichlet()
```

Details

Click on the "Help" tab for instructions. Click the "Quit" button to exit the app and return the results from the `fitdist` command. Click "Download report" to generate a report of all the fitted distributions.

Value

The parameters of the fitted Dirichlet distribution, which are returned once the Quit button has been clicked.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:

elicit()


## End(Not run)
```

elicitExtension *Elicitation with the extension method*

Description

Opens up a web browser (using the shiny package), from which you can specify judgements, fit distributions, and produce various plots. Judgements are specified for the distribution of the conditioning variable Y, the median function (median of X given Y), and the distribution of X given that Y takes its median value. Plots are provided for the two elicited distributions, the median function, the conditional distribution of X for any specified Y, and the marginal distribution of X.

Usage

```
elicitExtension()
```

Details

Click the "Quit" button to exit the app and return the results from the `fitdist` command. Click "Download report" to generate a report of all the fitted distributions for each uncertain quantity, and "Download sample" to generate a csv file with a sample from the marginal distribution of X.

Value

A list, with two objects of class `elicitation`. See `fitdist` for details.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:

elicitExtension()


## End(Not run)
```

`elicitHeterogen` *Elicit a prior distribution for a random effects variance parameter*

Description

Opens a shiny app for the roulette elicitation method. The user clicks in the grid to allocate 'probs' to 'bins'. The elicited probability inside each bin is the proportion of probs in each bin. This will fit a distribution to the ratio R of the 'largest' (97.5th percentile) to 'smallest' (2.5th percentile) treatment effect. A distribution for the variance effects variance parameter is inferred from the distribution of R, assuming that the random effects are normally distributed.

Usage

```
elicitHeterogen(
  lower = 1,
  upper = 10,
  gridheight = 10,
  nbins = 9,
  scale.free = TRUE,
  sigma = 1
)
```

Arguments

lower	The lower limit on the x-axis of the roulette grid.
upper	The upper limit on the x-axis of the roulette grid.
gridheight	The maximum number of probs that can be allocated to a single bin.
nbins	The number of equally sized bins drawn between lower and upper.
scale.free	Logical. Default is TRUE for a scale free treatment effect, such as an odds ratio, hazard ratio or relative risk. Set to FALSE for a treatment effect that is scale dependent, or is on the probit scale. An approximation to the treatment effect on the logit scale will be used (assuming a dichotomised response).
sigma	Individual observation standard deviation, required if scale.free is FALSE.

Value

BUGS code for incorporating the prior within a BUGS model. Additionally, a list with outputs

allocation	table of bins, with number of probs allocated to each bin.
Gamma	parameters of the fitted gamma distribution.
Log.normal	parameters of the fitted lognormal distribution.
sumsq	sum of squares of elicited - fitted probabilities for each distribution.
best.fitting	the distribution with the lowest sum of squares.

Note

Regarding the option "spread end probs over empty bins" (unchecked as the default): suppose for example, the leftmost and rightmost non-empty bins are [10,20] and [70,80], and each contain one prob, with 20 probs used in total. If the option is unchecked, it is assumed P(X<20) = P(X>70) = 0.05 and P(X<10) = P(X>80) = 0. If the option is checked, it is assumed P(X<20) = P(X>70) = 0.05 only.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
elicitHeterogen()

## End(Not run)
```

`elicitMixture` *Elicit a mixture distribution using the extension method*

Description

Opens up a web browser (using the shiny package), from which you can specify judgements, fit distributions and plot the fitted density function.

Usage

```
elicitMixture()
```

Details

Click the "Quit" button to exit the app and return the fitted distributions. Click "Download report" to generate a report of all the fitted distributions.

Value

When the Quit button is clicked, a list, with elements

`fit`	an object of class `elicitation`. See `fitdist` for details.
`extensionProbs`	the probability mass function for the extension variable.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:

elicitMixture()


## End(Not run)
```

elicitMultiple *Elicit individual judgements and fit distributions for multiple experts*

Description

Opens up a web browser (using the shiny package), from which you can specify judgements, fit distributions and plot the fitted density functions and a (weighted) linear pool with additional feedback.

Usage

```
elicitMultiple()
```

Details

Click the "Quit" button to exit the app and return the results from the `fitdist` command. Click "Download report" to generate a report of all the fitted distributions.

Value

An object of class `elicitation`, which is returned once the Finish button has been clicked. See `fitdist` for details.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:

elicitMultiple()


## End(Not run)
```

elicitQuartiles *Elicit judgements and fit distributions interactively using the quartile method*

Description

Opens up a web browser (using the shiny package), from which you can specify the quartiles, fit distributions and plot the fitted density functions with additional feedback.

Usage

```
elicitQuartiles()
```

Details

Parameter limits determine which distributions can be fitted. Finite lower limits are needed for the gamma, lognormal and log-t distributions, and both limits must be finite for to fit a beta distribution. If a histogram is fitted without specifying finite limits, endpoints are chosen based on fitting a normal distribution.

Click the Finish button to quit the elicitation session.

Value

An object of class `elicitation`, which is returned once the Finish button has been clicked. See `fitdist` for details.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:

elicitQuartiles()


## End(Not run)
```

---

elicitTertiles *Elicit judgements and fit distributions interactively using the tertile method*

Description

Opens up a web browser (using the shiny package), from which you can specify the median and tertiles, fit distributions and plot the fitted density functions with additional feedback.

Usage

```
elicitTertiles()
```

Details

Parameter limits determine which distributions can be fitted. Finite lower limits are needed for the gamma, lognormal and log-t distributions, and both limits must be finite for to fit a beta distribution. If a histogram is fitted without specifying finite limits, endpoints are chosen based on fitting a normal distribution.

Click the Finish button to quit the elicitation session.

Value

An object of class `elicitation`, which is returned once the Finish button has been clicked. See `fitdist` for details.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:

elicitTertiles()


## End(Not run)
```

---

feedback *Report quantiles and probabilities from the fitted probability distributions*

Description

Having fitted appropriate distributions to one or more expert's judgements individually using the `fitdist` command, use this command to get quantiles and probabilities from the fitted distributions

Usage

```
feedback(fit, quantiles = NA, values = NA, dist = "best", ex = NA, sf = 3)
```

Arguments

`fit`	An object of class `elicitation`.
`quantiles`	A vector of desired quantiles for feedback. If this argument is left out, the default is to use the same quantiles that were elicited from the experts.
`values`	A vector of desired probabilities; desired values of a for reporting back fitted values of P(X<a). If this argument is left out, the default is to use the same values provided by the experts.

`dist` If `fit` contains judgements from multiple experts, `dist` is distribution to be used for calculating probabilities and quantiles. Options are `"normal"`, `"t"`, `"gamma"`, `"lognormal"`, `"logt"`, `"beta"`, or `"best"`. The default option, `"best"`, uses the best fitting distribution for each expert.

`ex` If `fit` contains judgements from multiple experts, specifying a value for `ex` will select a single expert for feedback. Note that for a single expert, feedback is given for all suitable types of distribution, but for multiple experts, feedback is given for one type of distribution only.

`sf` The number of significant figures to be displayed in the output.

Value

`fitted.quantiles`
Fitted quantiles for each expert

`fitted.probabilities`
Fitted probabilities for each expert

`distributions` The distribution used to calculate fitted probabilities/quantiles for each expert, if feedback is given for multiple experts.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
# Two experts
# Expert 1 states P(X<30)=0.25, P(X<40)=0.5, P(X<50)=0.75
# Expert 2 states P(X<20)=0.25, P(X<25)=0.5, P(X<35)=0.75
# Both experts state 0<X<100.

v <- matrix(c(30, 40, 50, 20, 25, 35), 3, 2)
p <- c(0.25, 0.5, 0.75)
myfit <- fitdist(vals = v, probs = p, lower = 0, upper = 100)

feedback(myfit)

# Feedback P(X<60) and the tertiles
feedback(myfit, values=60, quantiles=c(0.33,0.66))

# Compare fitted tertiles for different distributions, expert 2 only
feedback(myfit, quantiles=c(0.33,0.66), ex=2)

## End(Not run)
```

feedbackDirichlet *Calculate quantiles for the marginal distributions of a Dirichlet distribution*

Description

Given a (elicited) Dirichlet distribution, calculate quantiles for each marginal beta distribution corresponding to the elicited quantiles

Usage

```
feedbackDirichlet(d, quantiles = c(0.1, 0.9), sf = 2)
```

Arguments

d	A vector of parameters of the Dirichlet distribution
quantiles	The desired quantiles for feedback
sf	The number of significant figures displayed

Value

Quantiles for each marginal distribution

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
feedbackDirichlet(d = c(20, 10, 5),
                  quantiles = c(0.1, 0.33, 0.66, 0.9))

## End(Not run)
```

fitDirichlet *Fit a Dirichlet distribution to elicited marginal distributions for proportions*

Description

Takes elicited beta distributions for a set of proportions as inputs, and fits a Dirichlet distribution. The beta parameters are adjusted so that the expectations sum to 1, and then the sum of the Dirichlet parameters is chosen based on the sums of the beta parameters for each elicited marginal

Usage

```
fitDirichlet(
  ...,
  categories = NULL,
  n.fitted = "opt",
  plotBeta = TRUE,
  xlab = "x",
  ylab = expression(f[X](x)),
  fs = 12,
  silent = FALSE
)
```

Arguments

...	Multiple arguments, each an objects of class `elicitation`, one per marginal proportion, separated by commas. The sequence can be specified as a single argument by containing all the `elicitation` objects within a single `list` object.
categories	A vector of strings labelling the marginal proportions.
n.fitted	The method used to determine the sum of the Dirichlet parameters. Use `"opt"` for best fitting, derived by matching standard deviations from the elicited marginals and the fitted Dirichlet; `"min"` for a conservative choice based on the smallest equivalent sample size (sum of the beta parameters) from the elicited marginals; `"med"` for the median of the smallest and largest largest equivalent sample size from the elicited marginals; `"mean"` for the mean of all the equivalent sample sizes from the elicited marginals.
plotBeta	logical. Plot the original elicited marginals and the fitted marginals from the Dirichlet fit.
xlab	x-axis label on the marginal distribution plot.
ylab	y-axis label on the marginal distribution plot.
fs	The font size used in the plot.
silent	Set to `TRUE` to supress printing of results to the console.

Value

The parameters of the fitted Dirichlet distribution.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

References

Zapata-Vazquez, R., O'Hagan, A. and Bastos, L. S. (2014). Eliciting expert judgements about a set of proportions. Journal of Applied Statistics 41, 1919-1933.

Examples

```
## Not run:
p1 <- c(0.25, 0.5, 0.75)
v1 <- c(0.5, 0.55, 0.6)
v2 <- c(0.22, 0.3, 0.35)
v3 <- c(0.11, 0.15, 0.2)
myfit1 <- fitdist(v1, p1, 0, 1)
myfit2 <- fitdist(v2, p1, 0, 1)
myfit3 <- fitdist(v3, p1, 0, 1)
d <- fitDirichlet(myfit1, myfit2, myfit3,
                  categories = c("A","B","C"),
                  n.fitted = "opt")

# Note that this will also work:
d <- fitDirichlet(list(myfit1, myfit2, myfit3),
                  categories = c("A","B","C"),
                  n.fitted = "opt")


## End(Not run)
```

---

fitdist *Fit distributions to elicited probabilities*

Description

Takes elicited probabilities as inputs, and fits parametric distributions using least squares on the cumulative distribution function. If separate judgements from multiple experts are specified, the function will fit one set of distributions per expert.

Usage

```
fitdist(
  vals,
  probs,
  lower = -Inf,
  upper = Inf,
  weights = 1,
  tdf = 3,
  expertnames = NULL,
  excludelogt = FALSE
)
```

Arguments

vals	A vector of elicited values for one expert, or a matrix of elicited values for multiple experts (one column per expert). Note that the an elicited judgement about X should be of the form P(X<= vals[i,j]) = probs[i,j]

probs	A vector of elicited probabilies for one expert, or a matrix of elicited values for multiple experts (one column per expert). A single vector can be used if the probabilities are the same for each expert. For each expert, the smallest elicited probability must be less than 0.4, and the largest elicited probability must be greater than 0.6.
lower	A single lower limit for the uncertain quantity X, or a vector of different lower limits for each expert. Specifying a lower limit will allow the fitting of distributions bounded below.
upper	A single upper limit for the uncertain quantity X, or a vector of different lower limits for each expert. Specifying both a lower limit and an upper limit will allow the fitting of a Beta distribution.
weights	A vector or matrix of weights corresponding to vals if weighted least squares is to be used in the parameter fitting.
tdf	The number of degrees of freedom to be used when fitting a t-distribution.
expertnames	Vector of names to use for each expert.
excludelogt	Set to TRUE to exclude log-t and mirror log-t when identifying best fitting distribution.

Value

An object of class `elicitation`. This is a list containing the elements

Normal	Parameters of the fitted normal distributions.
Student.t	Parameters of the fitted t distributions. Note that (X - location) / scale has a standard t distribution. The degrees of freedom is not fitted; it is specified as an argument to `fitdist`.
Gamma	Parameters of the fitted gamma distributions. Note that E(X - `lower`) = shape / rate.
Log.normal	Parameters of the fitted log normal distributions: the mean and standard deviation of log (X - `lower`).
Log.Student.t	Parameters of the fitted log student t distributions. Note that (log(X- `lower`) - location) / scale has a standard t distribution. The degrees of freedom is not fitted; it is specified as an argument to `fitdist`.
Beta	Parameters of the fitted beta distributions. X is scaled to the interval [0,1] via Y = (X - `lower`)/(`upper` - `lower`), and E(Y) = shape1 / (shape1 + shape2).
mirrorgamma	Parameters of ('mirror') gamma distributions fitted to Y = `upper` - X. Note that E(Y) = shape / rate.
mirrorlognormal	Parameters of ('mirror') log normal distributions fitted to Y = `upper` - X.
mirrorlogt	Parameters of ('mirror') log Student-t distributions fitted to Y = `upper` - X. Note that (log(Y) - location) / scale has a standard t distribution. The degrees of freedom is not fitted; it is specified as an argument to `fitdist`.
ssq	Sum of squared errors for each fitted distribution and expert. Each error is the difference between an elicited cumulative probability and the corresponding fitted cumulative probability.

best.fitting	The best fitting distribution for each expert, determined by the smallest sum of squared errors.
vals	The elicited values used to fit the distributions.
probs	The elicited probabilities used to fit the distributions.
limits	The lower and upper limits specified by each expert (+/- Inf if not specified).

Note

The least squares parameter values are found numerically using the optim command. Starting values for the distribution parameters are chosen based on a simple normal approximation: linear interpolation is used to estimate the 0.4, 0.5 and 0.6 quantiles, and starting parameter values are chosen by setting E(X) equal to the 0.5th quantile, and Var(X) = (0.6 quantile - 0.4 quantile)^2 / 0.25. Note that the arguments lower and upper are not included as elicited values on the cumulative distribution function. To include a judgement such as P(X<=a)=0, the values a and 0 must be included in vals and probs respectively.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
# One expert, with elicited probabilities
# P(X<20)=0.25, P(X<30)=0.5, P(X<50)=0.75
# and X>0.
v <- c(20,30,50)
p <- c(0.25,0.5,0.75)
fitdist(vals=v, probs=p, lower=0)

# Now add a second expert, with elicited probabilities
# P(X<55)=0.25, P(X<60=0.5), P(X<70)=0.75
v <- matrix(c(20,30,50,55,60,70),3,2)
p <- c(0.25,0.5,0.75)
fitdist(vals=v, probs=p, lower=0)

# Two experts, different elicited quantiles and limits.
# Expert A: P(X<50)=0.25, P(X<60=0.5), P(X<65)=0.75, and provides bounds 10<X<100
# Expert B: P(X<40)=0.33, P(X<50=0.5), P(X<60)=0.66, and provides bounds 0<X
v <- matrix(c(50,60,65,40,50,60),3,2)
p <- matrix(c(.25,.5,.75,.33,.5,.66),3,2)
l <- c(10,0)
u <- c(100, Inf)
fitdist(vals=v, probs=p, lower=l, upper=u)

## End(Not run)
```

`fitprecision` *Fit a distribution to judgements about a population precision*

Description

Takes elicited probabilities about proportion of a population lying in a specfied interval as inputs, converts the judgements into probability judgements about the population precision, and fits gamma and lognormal distributions to these judgements using the fitdist function.

Usage

```
fitprecision(
  interval,
  propvals,
  propprobs = c(0.05, 0.95),
  med = interval[1],
  trans = "identity",
  pplot = TRUE,
  tdf = 3,
  fontsize = 12
)
```

Arguments

`interval`	A vector specifying the endpoints of an interval $[k_1, k_2]$.
`propvals`	A vector specifying two values θ_1, θ_2 for the proportion.
`propprobs`	A vector specifying two probabilities p_1, p_2.
`med`	The hypothetical value of the population median.
`trans`	A string variable taking the value `"identity"`, `"log"` or `"logit"` corresponding to whether the population distribution is normal, lognormal or logit-normal respectively.
`pplot`	Plot the population distributions with median set at k_1 and precision fixed at the two elicited quantiles implied by `propvals` and `propprobs`.
`tdf`	Degrees of freedom in the fitted log Student-t distribution.
`fontsize`	Font size used in the plots.

Details

The expert provides a pair of probability judgements

$$P(\theta < \theta_1) = p_1,$$

and

$$P(\theta < \theta_2) = p_2,$$

where θ is the proportion of the population that lies in the interval $[k_1, k_2]$, conditional on the population median taking some hypothetical value (k_1 by default). k_1 can be set to `-Inf`, or k_2 can be set to `Inf`; in either case, the hypothetical median value must be specified. If both k_1 and k_2 are finite, the hypothetical median must be one of the interval endpoints. Note that, unlike the fitdist command, a 'best fitting' distribution is not reported, as the distributions are fitted to two elicited probabilities only.

Value

`Gamma`	Parameters of the fitted gamma distribution. Note that E(precision) = shape / rate.
`Log.normal`	Parameters of the fitted log normal distribution: the mean and standard deviation of log precision.
`Log.Student.t`	Parameters of the fitted log student t distributions. Note that (log(X- `lower`) - location) / scale has a standard t distribution. The degrees of freedom is not fitted: it is specified as an input argument.
`vals`	The elicited values θ_1, θ_2
`probs`	The elicited probabilities p_1, p_2
`limits`	The lower and upper limits specified by each expert (+/- Inf if not specified).
`transform`	Transformation used for a normal population distribution.

Examples

```
## Not run:
fitprecision(interval=c(60, 70), propvals=c(0.2, 0.4), trans = "log")

## End(Not run)
```

---

generateReport *Generate a report to show the fitted distributions*

Description

Renders an Rmarkdown document to display the density function of each fitted distribution, the parameter values, and the R command required to sample from each distribution.

Usage

```
generateReport(
  fit,
  output_format = "html_document",
  sf = 3,
  expert = 1,
  view = TRUE,
  clean = TRUE
)
```

Arguments

fit	An object of class 'elicitation'.
output_format	the output format for the document. One of "html_document", "pdf_document" (requires LaTeX to be installed), or "word_document" (requires Word to be installed).
sf	number of significant figures to be displayed for the fitted parameters.
expert	if the fit object contains judgements from multiple experts, the single expert's distributions to be displayed.
view	set to TRUE to open the document after it has been compiled.
clean	set to TRUE to clean intermediate files that are created during rendering.

Examples

```
## Not run:
# One expert, with elicited probabilities
# P(X<20)=0.25, P(X<30)=0.5, P(X<50)=0.75
# and X>0.
v <- c(20,30,50)
p <- c(0.25,0.5,0.75)
myfit <- fitdist(vals=v, probs=p, lower=0)

generateReport(myfit)

## End(Not run)
```

---

linearPoolDensity *Obtain points on the density function of a linear pool*

Description

Takes an object of class elicitation, evaluates a (weighted) linear pool, and returns points on the density function at a sequence of values of the elicited parameter

Usage

```
linearPoolDensity(fit, xl = -Inf, xu = Inf, d = "best", lpw = 1, nx = 200)
```

Arguments

fit	An object of class elicitation.
xl	The lower limit in the sequence of parameter values. The default is the 0.001 quantile of the fitted distribution (or the 0.001 quantile of a fitted normal distribution, if a histogram fit is chosen).
xu	The upper limit in the sequence of parameter values. The default is the 0.999 quantile of the fitted distribution (or the 0.999 quantile of a fitted normal distribution, if a histogram fit is chosen).

d	The distribution fitted to each expert's probabilities. Options are "normal", "t", "gamma", "lognormal", "logt","beta", "hist" (for a histogram fit), and "best" (for best fitting)
lpw	A vector of weights to be used in linear pool, if unequal weighting is desired.
nx	The number of points in the sequence from xl to xu.

Value

A list, with elements

x	a sequence of values for the uncertain parameter
fx	the density function of the linear pool, evaluated at each element in x.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
# Two experts
# Expert 1 states P(X<30)=0.25, P(X<40)=0.5, P(X<50)=0.75
# Expert 2 states P(X<20)=0.25, P(X<25)=0.5, P(X<35)=0.75
# Both experts state 0<X<100.

v <- matrix(c(30, 40, 50, 20, 25, 35), 3, 2)
p <- c(0.25, 0.5, 0.75)
myfit <- fitdist(vals = v, probs = p, lower = 0, upper = 100)
linearPoolDensity(myfit)

## End(Not run)
```

---

makeCDFPlot *Plot the elicited cumulative probabilities*

Description

Plots the elicited cumulative probabilities and, optionally, a fitted CDF. Elicited are shown as filled circles, and limits are shown as clear circles.

Usage

```
makeCDFPlot(
  lower,
  v,
  p,
  upper,
```

```
  fontsize = 12,
  fit = NULL,
  dist = NULL,
  showFittedCDF = FALSE,
  showQuantiles = FALSE,
  ql = 0.05,
  qu = 0.95,
  ex = 1,
  sf = 3,
  xaxisLower = lower,
  xaxisUpper = upper,
  xlab = "x",
  ylab = expression(P(X <= x))
)
```

Arguments

lower	lower limit for the uncertain quantity
v	vector of values, for each value x in Pr(X<=x) = p in the set of elicited probabilities
p	vector of probabilities, for each value p in Pr(X<=x) = p in the set of elicited probabilities
upper	upper limit for the uncertain quantity
fontsize	font size to be used in the plot
fit	object of class elicitation
dist	the fitted distribution to be plotted. Options are "normal", "t", "gamma", "lognormal", "logt","beta", "mirrorgamma", "mirrorlognormal", "mirrorlogt" "hist" (for a histogram fit)
showFittedCDF	logical. Should a fitted distribution function be displayed?
showQuantiles	logical. Should quantiles from the fitted distribution function be displayed?
ql	a lower quantile to be displayed.
qu	an upper quantile to be displayed.
ex	if the object fit contains judgements from multiple experts, which (single) expert's judgements to show.
sf	number of significant figures to be displayed.
xaxisLower	lower limit for the x-axis.
xaxisUpper	upper limit for the x-axis.
xlab	x-axis label.
ylab	y-axis label.

Examples

```
## Not run:
vQuartiles <- c(30, 35, 45)
pQuartiles<- c(0.25, 0.5, 0.75)
myfit <- fitdist(vals = vQuartiles, probs = pQuartiles, lower = 0)
makeCDFPlot(lower = 0, v = vQuartiles, p = pQuartiles,
 upper = 100, fit = myfit, dist = "gamma",
 showFittedCDF = TRUE, showQuantiles = TRUE)



## End(Not run)
```

---

pdfplots *Plot fitted population pdfs*

Description

Plot fitted population pdfs at combinations of two different values of the population mean and variance.

Usage

```
pdfplots(
  medianfit,
  precisionfit,
  alpha = 0.05,
  tails = 0.05,
  lower = NA,
  upper = NA,
  n.x = 100,
  d = "best",
  fontsize = 18
)
```

Arguments

medianfit	The output of a fitdist command following elicitation of the expert's beliefs about the population median.
precisionfit	The output of a fitdist command following elicitation of the expert's beliefs about the population precision.
alpha	Value between 0 and 1 to determine choice of means and variances used in plots
tails	Value between 0 and 1 to determine the tail area shown in the pdf plots
lower	lower limit on the x-axis for plotting.

upper	upper limit on the x-axis for plotting.
n.x	The number of points on the x-axis at which the pdf is plotted.
d	The fitted distribution for the population median. Can be one of "normal", "log-normal" or "best", where "best" will select the best fitting out of normal and lognormal.
fontsize	Font size used in the plots.

Details

Four pdfs are plotted, using each combination of the alpha/2 and 1-alpha/2 quantiles of the fitted distributions for the population median and standard deviation

Value

A plot and a list, containing

mu	The two population mean values used in the plots.
sigma	The two population standard deviation values used in the plots.

References

multiplot function obtained from http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/

Examples

```
## Not run:
prfit <- fitprecision(interval = c(60, 70), propvals = c(0.2, 0.4), trans = "log")
medianfit <- fitdist(vals = c(50, 60, 70), probs = c(0.05, 0.5,  0.95), lower = 0)
pdfplots(medianfit, prfit, alpha = 0.01)

## End(Not run)
```

---

plinearpool *Probabilities quantiles and samples from a (weighted) linear pool*

Description

Calculates a linear pool given a set of elicited judgements in a fit object. Then calculates required probabilities or quantiles from the pooled cumulative distribution function, or generates a random sample.

Usage

```
plinearpool(fit, x, d = "best", w = 1)
qlinearpool(fit, q, d = "best", w = 1)
rlinearpool(fit, n, d = "best", w = 1)
```

Arguments

fit	The output of a fitdist command.
x	A vector of required cumulative probabilities P(X<=x)
d	Scalar or vector of distributions to use for each expert. Options for each vector element are "hist", "normal", "t", "gamma", "lognormal", "logt","beta", "best". If given as a scalar, same choice is used for all experts.
w	A vector of weights to be used in the weighted linear pool.
q	A vector of required quantiles
n	Number of random samples from the linear pool

Details

Quantiles are calculate by first calculating the pooled cumulative distribution function at 100 points, and then using linear interpolation to invert the CDF.

Value

A probability or quantile, calculate from a (weighted) linear pool (arithmetic mean) of the experts' individual fitted probability.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
# Expert 1 states P(X<30)=0.25, P(X<40)=0.5, P(X<50)=0.75
# Expert 2 states P(X<20)=0.25, P(X<25)=0.5, P(X<35)=0.75
# Both experts state 0<X<100.

v <- matrix(c(30, 40, 50, 20, 25, 35), 3, 2)
p <- c(0.25, 0.5, 0.75)
myfit <- fitdist(vals = v, probs = p, lower = 0, upper = 100)

plinearpool(myfit, x=c(20, 50, 80))
qlinearpool(myfit, q=c(0.05, 0.5, 0.95))

# give more weight to first expert
plinearpool(myfit, x=c(20, 50, 80), w=c(0.7, 0.3))

# force the use of gamma distributions for each expert
qlinearpool(myfit, q=c(0.05, 0.5, 0.95), d="gamma")

## End(Not run)
```

`plotConditionalDensities` *Plot density of the target variable, conditional on the extension variable*

Description

Plots kernel density estimates of the target variable, conditional on each of a set of specified values of the extension variable. The plot makes use of the function `ggridges::geom_density_ridges()`, and so uses kernel density estimates rather than the exact conditional density function.

Usage

```
plotConditionalDensities(
  y,
  fitX,
  yCP,
  xMed,
  medianY,
  link = "identity",
  dist = "best",
  N = 1e+05,
  xLimits = NULL,
  fs = 12
)
```

Arguments

`y`	vector of values for the extension variable at which to condition on.
`fitX`	an object of class `elicitation` specifying the c-distribution: the distribution of the target variable, conditional on the extension variable taking its median value.
`yCP`	vector of conditioning points for the extension variable.
`xMed`	vector of medians of the target variable, corresponding to each value of the extension variable in `yCP`.
`medianY`	the median value of the extension variable.
`link`	link in the median function. One of `"identity"`, `"log"` or `"logit"`
`dist`	choice of parametric distribution for the c-distribution. Options are `"normal"`, `"t"`, `"gamma"`, `"lognormal"`, `"logt"`,`"beta"`, `"hist"` (for a histogram fit), and `"best"` (for best fitting).
`N`	sample size used in the kernel density estimate
`xLimits`	x-axis limits
`fs`	font size

Examples

```
## Not run:

myfitX <- fitdist(vals = c(5.5, 9, 14),
 probs = c(0.25, 0.5, 0.75),
 lower = 0)

plotConditionalDensities(y = c(2, 6, 10),
 fitX = myfitX,
 yCP = c(3, 5, 7, 9.5, 13.5),
 xMed = c(2, 6.5, 9, 13, 20),
 medianY = 7,
 link = "log",
 dist = "lognormal",
 xLimits = c(0, 60))


# Example with the logit link

myfitXlogit <- fitdist(vals = c(0.2, 0.25, 0.3),
 probs = c(0.25, 0.5, 0.75),
 lower = 0,
 upper = 1)

 plotConditionalDensities(y = c(2, 6, 10),
  fitX = myfitXlogit,
  yCP = c(2, 4, 6, 8, 10),
  xMed = c(0.1, 0.3, 0.5, 0.7, 0.9),
  medianY = 6,
  link = "logit",
  dist = "beta")


## End(Not run)
```

`plotConditionalMedianFunction`

Plot the conditional median function

Description

Produces a plot of the conditional median function, given a set of conditioning points for the extension variable, a set of corresponding medians of the target variable, given the extension variable, and a choice of link. The identity link is the default, a log link can be used for non-negative target variables, and a logit link can be used for target variables constrained to lie between 0 and 1.

Usage

```
plotConditionalMedianFunction(
  yCP,
  xMed,
  yLimits = NULL,
  link = "identity",
  xlab = "Y",
  ylab = "median of X given Y",
  fs = 12,
  ybreaks = NULL,
  xbreaks = NULL
)
```

Arguments

yCP	vector of conditioning points for the extension variable.
xMed	vector of medians of the target variable, corresponding to each value of the extension variable in yCP.
yLimits	limits for the extension variable, used to set the axis limits in the plot
link	link in the median function. One of "identity", "log" or "logit".
xlab	x-axis label
ylab	y-axis label
fs	font size
ybreaks	tick marks on the y-axis
xbreaks	tick marks on the axis

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
plotConditionalMedianFunction(yCP = c(3, 5, 7, 9.5, 13.5),
 xMed = c(2, 6.5, 9, 13, 20),
 yLimits = c(0, 20),
 link = "log")

plotConditionalMedianFunction(yCP = c(2, 4, 6, 8, 10),
 xMed = c(0.1, 0.3, 0.5, 0.7, 0.9),
 yLimits = c(0, 15),
 link = "logit")


## End(Not run)
```

plotfit *Plot the fitted density function for one or more experts*

Description

Plots the fitted density function for one or more experts. Can also plot a fitted linear pool if more than one expert. If plotting the density function of one expert, or the linear pool only, can also indicated desired lower and upper fitted quantiles.

Usage

```
plotfit(
  fit,
  d = "best",
  xl = -Inf,
  xu = Inf,
  ql = NA,
  qu = NA,
  lp = FALSE,
  ex = NA,
  sf = 3,
  ind = TRUE,
  lpw = 1,
  fs = 12,
  lwd = 1,
  xlab = "x",
  ylab = expression(f[X](x)),
  legend_full = TRUE,
  percentages = FALSE,
  returnPlot = FALSE
)
```

Arguments

fit	An object of class elicitation.
d	The distribution fitted to each expert's probabilities. Options are "normal", "t", "gamma", "lognormal", "logt","beta", "mirrorgamma", "mirrorlognormal", "mirrorlogt" "hist" (for a histogram fit), and "best" (for best fitting)
xl	The lower limit for the x-axis. The default is the 0.001 quantile of the fitted distribution (or the 0.001 quantile of a fitted normal distribution, if a histogram fit is chosen).
xu	The upper limit for the x-axis. The default is the 0.999 quantile of the fitted distribution (or the 0.999 quantile of a fitted normal distribution, if a histogram fit is chosen).
ql	A lower quantile to be indicated on the density function plot. Only displayed when plotting the density function for a single expert.

qu	An upper quantile to be indicated on the density function plot. Only displayed when plotting the density function for a single expert.
lp	For multiple experts, set lp = TRUE to plot a linear pool.
ex	If judgements have been elicited from multiple experts, but a density plot for one expert only is required, the expert to be used in the plot.
sf	The number of significant figures to be displayed for the parameter values.
ind	If plotting a linear pool, set ind = FALSE to suppress plotting of the individual density functions.
lpw	A vector of weights to be used in linear pool, if unequal weighting is desired.
fs	The font size used in the plot.
lwd	The line width used in the plot.
xlab	A string or expression giving the x-axis label.
ylab	A string or expression giving the y-axis label.
legend_full	If plotting a linear pool, set ind = TRUE for each expert to be plotted with a different colour, and ind = FALSE for each expert to be plotted with the same colour, reducing the legend size.
percentages	Set to TRUE to use percentages on the x-axis.
returnPlot	Set to TRUE to return the plot as a ggplot object.

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
# Two experts
# Expert 1 states P(X<30)=0.25, P(X<40)=0.5, P(X<50)=0.75
# Expert 2 states P(X<20)=0.25, P(X<25)=0.5, P(X<35)=0.75
# Both experts state 0<X<100.

v <- matrix(c(30, 40, 50, 20, 25, 35), 3, 2)
p <- c(0.25, 0.5, 0.75)
myfit <- fitdist(vals = v, probs = p, lower = 0, upper = 100)

# Plot both fitted densities, using the best fitted distribution
plotfit(myfit)

# Plot a fitted beta distribution for expert 2, and show 5th and 95th percentiles
plotfit(myfit, d = "beta", ql = 0.05, qu = 0.95, ex = 2)


# Plot a linear pool, giving double weight to expert 1
plotfit(myfit,  lp = T, lpw = c(2,1))
```

```
# Plot a linear pool, giving double weight to expert 1,
# show 5th and 95th percentiles, surpress plotting of individual distributions,
# and force use of Beta distributions
plotfit(myfit, d = "beta",  lp = T, lpw = c(2,1), ql = 0.05, qu = 0.95, ind=FALSE )

## End(Not run)
```

`plotQuartiles` *Plot elicted quartiles, median and plausible range for each expert*

Description

Displays a horizontal bar for each expert, to represent the expert's plausible range. The coloured sections indicate the experts' quartiles: four intervals judged by the expert to be equally likely. The experts' medians are shown as dashed lines.

Usage

```
plotQuartiles(
  vals,
  lower,
  upper,
  fs = 12,
  expertnames = NULL,
  xl = NULL,
  xlabel = "X"
)
```

Arguments

`vals`	a matrix of elicited tertiles and medians: one column per expert, first row is the 25th percentile, 2nd row is the median, last row is the 75th percentile.
`lower`	a vector of lower plausible limits: one per expert
`upper`	a vector of upper plausible limits: one per expert
`fs`	font size to be used in the plot
`expertnames`	vector of experts' names
`xl`	vector of limits for x-axis
`xlabel`	x-axis label

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
l <- c(2, 1, 5, 1)
u <- c(95, 90, 65, 40)
v <- matrix(c(15, 25, 40,
 10, 20, 40,
 10, 15, 25,
 5, 10, 20),
 3, 4)
plotQuartiles(vals = v, lower = l, upper  = u)

## End(Not run)
```

`plotTertiles` *Plot elicted tertiles, median and plausible range for each expert*

Description

Displays a horizontal bar for each expert, to represent the expert's plausible range. The coloured sections indicate the experts' tertiles: three intervals judged by the expert to be equally likely. The experts' medians are shown as dashed lines.

Usage

```
plotTertiles(
  vals,
  lower,
  upper,
  fs = 12,
  percentages = FALSE,
  expertnames = NULL,
  xl = NULL,
  xlabel = "X"
)
```

Arguments

`vals`	a matrix of elicited tertiles and medians: one column per expert, first row is the 33rd percentile, 2nd row is the median, last row is the 66th percentile.
`lower`	a vector of lower plausible limits: one per expert
`upper`	a vector of upper plausible limits: one per expert
`fs`	font size to be used in the plot
`percentages`	set to `TRUE` to use percentages on the x-axis
`expertnames`	vector of experts' names
`xl`	vector of limits for x-axis
`xlabel`	x-axis label

Author(s)

Jeremy Oakley <j.oakley@sheffield.ac.uk>

Examples

```
## Not run:
l <- c(-5, 0, 5, -10)
u <- c(15, 35, 50, 35)
v <- matrix(c(5, 8, 10,
 10, 15, 20,
 15, 18, 25,
 10, 20, 30),
 3, 4)
plotTertiles(vals = v, lower = l, upper  = u)

## End(Not run)
```

---

sampleFit *Sample from the elicited distributions*

Description

Generates a random sample from all distributions specified within an object of class `elicitation`

Usage

```
sampleFit(fit, n, expert = 1)
```

Arguments

`fit`	An object of class elicitation
`n`	The required sample size for each elicitation
`expert`	Specify which expert's distributions to sample from, if multiple experts' judgements have been elicited.

Value

A matrix of sampled values, one column per distribution. Column names are given to label the distributions.

Examples

```
## Not run:
v <- c(20,30,50)
p <- c(0.25,0.5,0.75)
myfit <- fitdist(vals = v, probs = p, lower = 0, upper = 100)
samplefit(myfit, n = 10)

## End(Not run)
```

`sampleMarginalFit` *Sample from the marginal distribution of the target variable*

Description

As part of the Extension Method, this function will generate a random sample from the marginal distribution of the target variable, using a sample from the marginal distribution of the extension variable, the specified c-distribution, and the appropriate judgements used to construct the median model.

Usage

```
sampleMarginalFit(
  fitX,
  sampleY,
  medianY,
  yCP,
  xMed,
  dist = "best",
  link = "identity"
)
```

Arguments

`fitX`	an object of class `elicitation` specifying the c-distribution: the distribution of the target variable, conditional on the extension variable taking its median value.
`sampleY`	a sample from the marginal distribution of the extension variable.
`medianY`	the median value of the extension variable.
`yCP`	vector of conditioning points for the extension variable.
`xMed`	vector of medians of the target variable, corresponding to each value of the extension variable in `yCP`.
`dist`	choice of parametric distribution for the c-distribution. Options are `"normal"`, `"t"`, `"gamma"`, `"lognormal"`, `"logt"`,`"beta"`, `"hist"` (for a histogram fit), and `"best"` (for best fitting).
`link`	link in the median function. One of `"identity"`, `"log"` or `"logit"`

Value

a vector containing a sample from the marginal distribution of the target variable.

Examples

```
## Not run:

myfitX <- fitdist(vals = c(5.5, 9, 14),
 probs = c(0.25, 0.5, 0.75),
```

```
 lower = 0)
ry <- rgamma(10, 5.19, 0.694)
sampleMarginalFit(fitX = myfitX,
 sampleY = ry,
 medianY = 7,
 yCP = c(3, 5, 7, 9.5, 13.5),
 xMed = c(2, 6.5, 9, 13, 20),
 dist = "lognormal",
 link = "log")

## End(Not run)
```

Index