Package 'SHELF'

February 11, 2017

Type Package

Title Tools to Support the Sheffield Elicitation Framework

Version 1.2.3

Date 2017-02-10

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. A graphical interface for the roulette elicitation method is also provided. For multiple experts, a weighted linear pool can be calculated. Also includes functions for eliciting beliefs about population distributions.

License GPL-2 | GPL-3

Depends R (>= 3.3.1)

RoxygenNote 6.0.0

Suggests testthat, knitr, rmarkdown, GGally

Imports ggplot2, grid, shiny, stats, graphics, tidyr, MASS, ggExtra

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

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

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-11 01:16:03

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 or quantile 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 additional quantiles. A graphical interface for the roulette elicitation method is also provided. For multiple experts, a weighted linear pool can be calculated.

Package:	SHELF
Type:	Package
Version:	1.2.3
Date:	2017-02-10
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 judgements using the roulette method
roulette(lower = 0, upper = 100, nbins = 10, gridheight = 10)

## 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)
```

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", "Student-t", "Gamma", "Log normal" "Log Student-t", "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, qp=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.

Usage

```
elicit()
```

Details

Parameter limits determine which distributions can be fitted. Non-negative 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.

As an example, if the elicited judgements are P(X<15)=0.25, P(X<20)=0.5) and P(X<40)=0.75, specify the parameter values as 15,20,40 and the cumulative probabilities as 0.25,0.5,0.75.

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

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"`, `"Student-t"`, `"Gamma"`, `"Log normal"` `"Log Student-t"`, `"Beta"`. 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)
```

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.probs`	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)
```

Arguments

`...`	A list of objects of class `elicitation`. command, one per marginal proportion, separated by commas.
`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 mean 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.

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")

## 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)
```

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.

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).
- `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),
  trans = "identity", pplot = TRUE, fontsize = 18)
```

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.
`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`.
`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 $[k_1, k_2]$. The judgements are made conditional on the population median equalling k_1. 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.
`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)
```

`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", "lognormal" 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` *Calculate fitted probabilities or quantiles 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.

Usage

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

Arguments

`fit`	The output of a `fitdist` command.
`x`	A vector of required cumulative probabilities P(X<=x)
`d`	The distribution fitted to each expert's probabilities. This must either be the same distribution for each expert, or the best fitting distribution for each expert. Options are `"normal"`, `"t"`, `"gamma"`, `"lognormal"`, `"logt"`,`"beta"`, `"best"`.
`w`	A vector of weights to be used in the weighted linear pool.
`q`	A vector of required quantiles

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)
```

---

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", int = FALSE, 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)))
```

Arguments

`fit`	An object of class `elicitation`.
`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)
`int`	Set `int = TRUE` to use interactive plotting (using the shiny package). If plotting for a single expert, the argument `d` is ignored, as distributions can be chosen within the display. If plotting for multiple experts, feedback quantiles are not displayed, and the argument `lp` is ignored, as the option to show a linear pool can be chosen within the display.
`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.

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)

# Use interactive plotting for for expert 2, and show 5th and 95th percentiles
plotfit(myfit, int = T, ex = 2)

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

# Use interactive plotting, giving double weight to expert 1, if a linear pool is displayed
plotfit(myfit,  int = T, lpw = c(2,1))

# Plot a linear pool, giving double weight to expert 1,
# show 5th and 95th percentiles, supress 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)
```

`roulette` *Elicit one set of probabilities using the roulette method.*

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.

Usage

```
roulette(lower = 0, upper = 100, gridheight = 10, nbins = 10)
```

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 `Lo` and `Up`.

Value

A list, with outputs

`v`	upper limits of each bin.
`p`	cumulative probabilities for each upper bin limit.

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:
x <- roulette(lower = 0, upper = 100)
# Then allocate probs to bins and click "Finish"

# To fit distributions and see the results
myfit <- fitdist(vals = x$v, probs = x$p, lower = 0, upper = 100)
plotfit(myfit)

## End(Not run)
```

Index