Package ‘expert’

February 19, 2015

Type Package

Title Modeling without data using expert opinion

Version 1.0-0

Date 2008-09-01

Author Mathieu Pigeon, Michel Jacques, Vincent Goulet

Maintainer Mathieu Pigeon <mathieu.pigeon.3@ulaval.ca>

Description Expert opinion (or judgment) is a body of techniques to estimate the distribution of a random variable when data is scarce or unavailable. Opinions on the quantiles of the distribution are sought from experts in the field and aggregated into a final estimate. The package supports aggregation by means of the Cooke, Mendel-Sheridan and predefined weights models.

Depends R (>= 2.6.0), stats

License GPL (>= 2)

Encoding latin1

LazyLoad yes

LazyData yes

ZipData yes

NeedsCompilation no

Repository CRAN

Date/Publication 2008-10-02 14:18:55

R topics documented:

cdf *Expert Aggregated Cumulative Distribution Function*

Description

Compute or plot the cumulative distribution function for objects of class "expert".

Usage

```
cdf(x, ...)

## S3 method for class 'cdf'
print(x, digits = getOption("digits") - 2, ...)

## S3 method for class 'cdf'
knots(Fn, ...)

## S3 method for class 'cdf'
plot(x, ..., ylab = "F(x)", verticals = FALSE,
     col.01line = "gray70")
```

Arguments

x	an object of class "expert"; for the methods, an object of class "cdf", typically.
digits	number of significant digits to use, see print.
Fn	an R object inheriting from "cdf".
...	arguments to be passed to subsequent methods, e.g. plot.stepfun for the plot method.
ylab	label for the y axis.
verticals	see plot.stepfun.
col.01line	numeric or character specifying the color of the horizontal lines at y = 0 and 1, see colors.

Details

The function builds the expert aggregated cumulative distribution function corresponding to the results of expert.

The function plot.cdf which implements the plot method for cdf objects, is implemented via a call to plot.stepfun; see its documentation.

Value

For cdf, a function of class "cdf", inheriting from the "function" class.

See Also

expert to create objects of class "expert"; ogive for the linear interpolation; ecdf and stepfun for related documentation.

Examples

```
x <- list(E1 <- list(A1 <- c(0.14, 0.22, 0.28),
                     A2 <- c(130000, 150000, 200000),
                     X <- c(350000, 400000, 525000)),
          E2 <- list(A1 <- c(0.2, 0.3, 0.4),
                     A2 <- c(165000, 205000, 250000),
                     X <- c(550000, 600000, 650000)),
          E3 <- list(A1 <- c(0.2, 0.4, 0.52),
                     A2 <- c(200000, 400000, 500000),
                     X <- c(625000, 700000, 800000)))
probs <- c(0.1, 0.5, 0.9)
true.seed <- c(0.27, 210000)
fit <- expert(x, "cooke", probs, true.seed, 0.03)
Fn <- cdf(fit)
Fn
knots(Fn)           # the group boundaries

Fn(knots(Fn))       # true values of the cdf

plot(Fn)            # graphic
```

---

expert *Modeling of Data Using Expert Opinion*

Description

Compute an aggregated distribution from expert opinion using either of the Cooke, Mendel-Sheridan or predefined weights models.

Usage

```
expert(x, method = c("cooke", "ms", "weights"), probs, true.seed,
       alpha = NULL, w = NULL)

## S3 method for class 'expert'
print(x, ...)

## S3 method for class 'expert'
summary(object, ...)

## S3 method for class 'summary.expert'
print(x, ...)
```

Arguments

`x`	a list giving experts' quantiles for the seed variables and the decision variable. See details below for the exact structure of the object. For the methods: an object of class `"expert"`.
`method`	method to be used to aggregate distributions.
`probs`	vector of probabilities corresponding to the quantiles given by the experts.
`true.seed`	vector of true values for the seed variables.
`alpha`	confidence level in Cooke model. If `NULL` or missing, the function determines the confidence level that maximizes the weight given to the aggregated distribution for the seed variables.
`w`	vector of weights in predefined weights model. If `NULL` or missing, equal weights are given to each expert.
`object`	an object of class `"expert"`
`...`	further arguments to `format` for the `print` and `print.summary` methods; unused for the `summary` method.

Details

Expert opinion is given by means of quantiles for k seed variables and one decision variable. Results for seed variables are compared to the true values and used to determine the influence of each expert on the aggregated distribution. The three methods supported are different ways to aggregate the information provided by the experts in one final distribution.

The aggregated distribution in the `"cooke"` method is a convex combination of the quantiles, with weights obtained from the calibration phase. The `"weights"` method is similar, but weights are provided in argument to the function.

In the `"ms"` (Mendel-Sheridan) method, the probabilities associated with each quantile are adjusted by a bayesian procedure to reflect results of the calibration phase.

Object `x` is a list of lists, one for each expert. The latter contains $k + 1$ vectors of quantiles, one for each seed variable and one for the decision variable (in this order).

If `x` does not contain the 0th and/or the 100th quantile, they are determined by removing and adding 10% of the smallest interval containing all quantiles given by the experts to the bounds of this interval. Note also that only the Mendel-Sheridan model allows non-finite lower and upper bounds.

Value

Function `expert` computes the aggregated distribution using the model specified in `model`. The value returned is an object of class `"expert"`.

An object of class `"expert"` is a list containing at least the following components:

`breaks`	vector of knots of the aggregated distribution.
`probs`	vector of probabilities of the aggregated distribution.
`nexp`	number of experts in the model.
`nseed`	number of seed variables in the model.
`quantiles`	vector of probabilities corresponding to the quantiles given by the experts.

In addition, for `method = "cooke"`, a component `alpha` containing the confidence level: either the value given in argument to the function or the optimized value.

There are methods available to represent (`print`), plot (`plot`), compute quantiles (`quantile`), summarize (`summary`) and compute the mean (`mean`) of `"expert"` objects.

References

Cooke, R. (1991), *Expert in Uncertainty*, Oxford University Press.

Mendel, M. and Sheridan, T. (1989), Filtering information from human experts, *IEEE Transactions on Systems, Man and Cybernetics*, **36**, 6–16.

Pigeon, M. (2008), *Utilisation d'avis d'experts en actuariat*, M.Sc. thesis, Université Laval.

Examples

```
## An example with three experts (E1, E2, E3), two seed variables
## (A1, A2) and three quantiles (10th, 50th and 90th).
x <- list(E1 <- list(A1 <- c(0.14, 0.22, 0.28),
                     A2 <- c(130000, 150000, 200000),
                     X <- c(350000, 400000, 525000)),
          E2 <- list(A1 <- c(0.2, 0.3, 0.4),
                     A2 <- c(165000, 205000, 250000),
                     X <- c(550000, 600000, 650000)),
          E3 <- list(A1 <- c(0.2, 0.4, 0.52),
                     A2 <- c(200000, 400000, 500000),
                     X <- c(625000, 700000, 800000)))
probs <- c(0.1, 0.5, 0.9)
true.seed <- c(0.27, 210000)

## Cooke model
expert(x, "cooke", probs, true.seed, alpha = 0.03) # fixed alpha
expert(x, "cooke", probs, true.seed)               # optimized alpha

## Mendel-Sheridan model
fit <- expert(x, "ms", probs, true.seed)
fit # print method
summary(fit)                   # more information

## Predefined weights model
expert(x, "weights", probs, true.seed)             # equal weights
expert(x, "weights", probs, true.seed, w = c(0.25, 0.5, 0.25))
```

`hist.expert` *Histogram of the Expert Aggregated Distribution*

Description

This method for the generic function `hist` is mainly useful to plot the histogram of objects of class `"expert"`. If `plot = FALSE`, the resulting object of class `"histogram"` is returned for compatibility with `hist.default`, but does not contain much information not already in `x`.

Usage

```
## S3 method for class 'expert'
hist(x, freq = NULL, probability = !freq,
     density = NULL, angle = 45, col = NULL, border = NULL,
     main = paste("Histogram of" , xname),
     xlim = NULL, ylim = NULL, xlab = "x", ylab = expression(f(x)),
     axes = TRUE, plot = TRUE, labels = FALSE, ...)
```

Arguments

`x`	an object of class `"expert"`
`freq`	logical; if `TRUE`, the histogram graphic is a representation of frequencies, the `counts` component of the result; if `FALSE`, probability densities, component `density`, are plotted (so that the histogram has a total area of one). Defaults to `TRUE` *iff* group boundaries are equidistant (and `probability` is not specified).
`probability`	an *alias* for `!freq`, for S compatibility.
`density`	the density of shading lines, in lines per inch. The default value of `NULL` means that no shading lines are drawn. Non-positive values of `density` also inhibit the drawing of shading lines.
`angle`	the slope of shading lines, given as an angle in degrees (counter-clockwise).
`col`	a colour to be used to fill the bars. The default of `NULL` yields unfilled bars.
`border`	the color of the border around the bars. The default is to use the standard foreground color.
`main, xlab, ylab`	these arguments to `title` have useful defaults here.
`xlim, ylim`	the range of x and y values with sensible defaults. Note that `xlim` is *not* used to define the histogram (breaks), but only for plotting (when `plot = TRUE`).
`axes`	logical. If `TRUE` (default), axes are draw if the plot is drawn.
`plot`	logical. If `TRUE` (default), a histogram is plotted, otherwise a list of breaks and counts is returned.
`labels`	logical or character. Additionally draw labels on top of bars, if not `FALSE`; see `plot.histogram`.
`...`	further graphical parameters passed to `plot.histogram` and their to `title` and `axis` (if `plot=TRUE`).

Value

An object of class `"histogram"` which is a list with components:

`breaks`	the $r + 1$ group boundaries.
`counts`	r integers; the frequency within each group.
`density`	the relative frequencies within each group n_j/n, where n_j = `counts[j]`.
`intensities`	same as `density`. Deprecated, but retained for compatibility.
`mids`	the r group midpoints.
`xname`	a character string with the actual `x` argument name.
`equidist`	logical, indicating if the distances between `breaks` are all the same.

Note

The resulting value does *not* depend on the values of the arguments `freq` (or `probability`) or `plot`. This is intentionally different from S.

References

Klugman, S. A., Panjer, H. H. and Willmot, G. E. (1998), *Loss Models, From Data to Decisions*, Wiley.

See Also

`hist` and `hist.default` for histograms of individual data and fancy examples.

Examples

```
x <- list(E1 <- list(A1 <- c(0.14, 0.22, 0.28),
                     A2 <- c(130000, 150000, 200000),
                     X <- c(350000, 400000, 525000)),
          E2 <- list(A1 <- c(0.2, 0.3, 0.4),
                     A2 <- c(165000, 205000, 250000),
                     X <- c(550000, 600000, 650000)),
          E3 <- list(A1 <- c(0.2, 0.4, 0.52),
                     A2 <- c(200000, 400000, 500000),
                     X <- c(625000, 700000, 800000)))
probs <- c(0.1, 0.5, 0.9)
true.seed <- c(0.27, 210000)
fit <- expert(x, "cooke", probs, true.seed, 0.03)
hist(fit)
```

`mean.expert` *Arithmetic Mean of the Expert Aggregated Distribution*

Description

Mean of objects of class `"expert"`.

Usage

```
## S3 method for class 'expert'
mean(x, ...)
```

Arguments

`x`	an object of class `"expert"`.
`...`	further arguments passed to or from other methods.

Details

The mean of a distribution with probabilities $p_1, \dots, p_r$ on intervals defined by the boundaries $c_0, \dots, c_r$ is

$$\sum_{j=1}^{r} \frac{c_{j-1} + c_j}{2}\, p_j.$$

Value

A numeric value.

References

Klugman, S. A., Panjer, H. H. and Willmot, G. E. (1998), *Loss Models, From Data to Decisions*, Wiley.

See Also

`expert` to create objects of class `"expert"`

Examples

```
x <- list(E1 <- list(A1 <- c(0.14, 0.22, 0.28),
                     A2 <- c(130000, 150000, 200000),
                     X <- c(350000, 400000, 525000)),
          E2 <- list(A1 <- c(0.2, 0.3, 0.4),
                     A2 <- c(165000, 205000, 250000),
                     X <- c(550000, 600000, 650000)),
          E3 <- list(A1 <- c(0.2, 0.4, 0.52),
                     A2 <- c(200000, 400000, 500000),
                     X <- c(625000, 700000, 800000)))
probs <- c(0.1, 0.5, 0.9)
true.seed <- c(0.27, 210000)
fit <- expert(x, "cooke", probs, true.seed, 0.03)
mean(fit)
```

`ogive` *Ogive of the Expert Aggregated Distribution*

Description

Compute a smoothed empirical distribution function for objects of class `"expert"`.

Usage

```
ogive(x, ...)

## S3 method for class 'ogive'
print(x, digits = getOption("digits") - 2, ...)

## S3 method for class 'ogive'
knots(Fn, ...)

## S3 method for class 'ogive'
plot(x, main = NULL, xlab = "x", ylab = "G(x)", ...)
```

Arguments

`x`	an object of class `"expert"`; for the methods, an object of class `"ogive"`, typically.
`digits`	number of significant digits to use, see `print`.
`Fn`	an R object inheriting from `"ogive"`.
`main`	main title.
`xlab, ylab`	labels of x and y axis.
`...`	arguments to be passed to subsequent methods.

Details

The ogive is a linear interpolation of the empirical cumulative distribution function.

The equation of the ogive is

$$G(x) = \frac{(c_j - x)F(c_{j-1}) + (x - c_{j-1})F(c_j)}{c_j - c_{j-1}}$$

for $c_{j-1} < x \leq c_j$ and where $c_0, \dots, c_r$ are the $r + 1$ group boundaries and F is the cumulative distribution function.

Value

For `ogive`, a function of class `"ogive"`, inheriting from the `"function"` class.

References

Klugman, S. A., Panjer, H. H. and Willmot, G. E. (1998), *Loss Models, From Data to Decisions*, Wiley.

See Also

`expert` to create objects of class `"expert"`; `cdf` for the true cumulative distribution function; `approxfun`, which is used to compute the ogive; `stepfun` for related documentation (even though the ogive is not a step function).

Examples

```
x <- list(E1 <- list(A1 <- c(0.14, 0.22, 0.28),
                     A2 <- c(130000, 150000, 200000),
                     X <- c(350000, 400000, 525000)),
          E2 <- list(A1 <- c(0.2, 0.3, 0.4),
                     A2 <- c(165000, 205000, 250000),
                     X <- c(550000, 600000, 650000)),
          E3 <- list(A1 <- c(0.2, 0.4, 0.52),
                     A2 <- c(200000, 400000, 500000),
                     X <- c(625000, 700000, 800000)))
probs <- c(0.1, 0.5, 0.9)
true.seed <- c(0.27, 210000)
fit <- expert(x, "cooke", probs, true.seed, 0.03)
Fn <- ogive(fit)
Fn
knots(Fn)            # the group boundaries

Fn(knots(Fn))        # true values of the empirical cdf
Fn(c(80, 200, 2000)) # linear interpolations

plot(Fn)
```

`quantile.expert` *Quantiles of the Expert Aggregated Distribution*

Description

Quantile for objects of class "expert".

Usage

```
## S3 method for class 'expert'
quantile(x, probs = seq(0, 1, 0.25),
         smooth = FALSE, names = TRUE, ...)
```

Arguments

x	an object of class "expert".
probs	numeric vector of probabilities with values in $[0, 1)$.
smooth	logical; when TRUE and x is a step function, quantiles are linearly interpolated between knots.
names	logical; if true, the result has a names attribute. Set to FALSE for speedup with many probs.
...	further arguments passed to or from other methods.

Details

The quantiles are taken directly from the cumulative distribution function defined in x. Linear interpolation is available for step functions.

Value

A numeric vector, named if names is TRUE.

See Also

expert

Examples

```
x <- list(E1 <- list(A1 <- c(0.14, 0.22, 0.28),
                     A2 <- c(130000, 150000, 200000),
                     X <- c(350000, 400000, 525000)),
          E2 <- list(A1 <- c(0.2, 0.3, 0.4),
                     A2 <- c(165000, 205000, 250000),
                     X <- c(550000, 600000, 650000)),
          E3 <- list(A1 <- c(0.2, 0.4, 0.52),
                     A2 <- c(200000, 400000, 500000),
                     X <- c(625000, 700000, 800000)))
probs <- c(0.1, 0.5, 0.9)
true.seed <- c(0.27, 210000)
fit <- expert(x, "cooke", probs, true.seed, 0.03)
quantile(fit)                          # default probs
quantile(fit, probs = c(0.9, 0.95, 0.99)) # right tail
```

Index