Package 'jfa'

July 6, 2021

Title Bayesian and Classical Audit Sampling

Version 0.5.5.1

Date 2021-07-06

Description Implements the audit sampling workflow as discussed in Derks et al. (2019) <doi:10.31234/osf.io/9f6ub>. The package makes it easy for an auditor to plan a statistical sample, select the sample from the population, and evaluate the sample using various methods according to the International Standards on Auditing. Furthermore, the package implements Bayesian equivalents of these methods.

BugReports https://github.com/koenderks/jfa/issues

URL https://koenderks.github.io/jfa/, https://github.com/koenderks/jfa

Imports graphics, stats

Suggests kableExtra, knitr, MUS, rmarkdown, testthat

Language en-US

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

VignetteBuilder knitr

R topics documented:

jfa-package 2
auditBF 4
auditPrior 7
BuildIt 10
evaluation 11
jfa-methods 15
planning 17
report 20
selection 21

Index **23**

jfa-package *jfa — Bayesian and Classical Audit Sampling*

Description

`jfa` is an R package for statistical audit sampling. The package provides functions for planning, performing, evaluating, and reporting an audit sample. Specifically, these functions implement standard audit sampling techniques for calculating sample sizes, selecting items from a population, and evaluating the misstatement from a data sample or from summary statistics. Additionally, the `jfa` package allows the user to create a prior probability distribution to perform Bayesian audit sampling using these functions.

The package and its intended workflow are also implemented with a graphical user interface in the Audit module of JASP, a free and open-source statistical software program.

For documentation on `jfa` itself, including the manual and user guide for the package, worked examples, and other tutorial information visit the package website.

Reference tables

Below you can find several links to reference tables that contain statistical sample sizes, upper limits, and Bayes factors. These tables are created using the `planning()` and `evaluation()` functions provided in the package. See the corresponding help files for more information about these functions and how to replicate this output.

Sample sizes

- Sample sizes based on the binomial distribution
- Sample sizes based on the Poisson distribution
- Sample sizes based on the hypergeometric distribution

Upper limits

- Upper limits based on the binomial distribution
- Upper limits based on the Poisson distribution
- Upper limits based on the hypergeometric distribution

Bayes factors

- Bayes factors based on the beta distribution
- Bayes factors based on the gamma distribution
- Bayes factors based on the beta-binomial distribution

Author(s)

Koen Derks (maintainer, author) <k.derks@nyenrode.nl>

Please use the citation provided by R when citing this package. A BibTex entry is available from `citation("jfa")`.

See Also

Useful links:

- The cheat sheet for a quick overview of the intended workflow.
- The vignettes for worked examples.
- The issue page to submit a bug report or feature request.

Examples

```
# Load the jfa package
library(jfa)

# Load the BuildIt population
data("BuildIt")

############################################
### Example 1: Classical audit sampling ####
############################################

# Stage 1: Planning
stage1 <- planning(materiality = 0.03, expectedError = 0.01,
                   likelihood = "poisson", confidence = 0.95)
summary(stage1)

# Stage 2: Selection
stage2 <- selection(population = BuildIt, sampleSize = stage1,
                    units = "mus", bookValues = "bookValue",
                    algorithm = "interval", intervalStartingPoint = 1)
summary(stage2)

# Stage 3: Execution
sample <- stage2[["sample"]]

# Stage 4: Evaluation
stage4 <- evaluation(materiality = 0.03, method = "stringer",
                     confidence = 0.95, sample = sample,
                     bookValues = "bookValue", auditValues = "auditValue")
summary(stage4)

######################################################################
### Example 2: Bayesian audit sampling using a non-informed prior ####
######################################################################

# Create the prior distribution
prior <- auditPrior(method = "none", likelihood = "poisson")
summary(prior)

# Stage 1: Planning
stage1 <- planning(materiality = 0.03, expectedError = 0.01,
                   likelihood = "poisson", confidence = 0.95, prior = prior)
summary(stage1)

# Stage 2: Selection
stage2 <- selection(population = BuildIt, sampleSize = stage1,
                    units = "mus", bookValues = "bookValue",
                    algorithm = "interval", intervalStartingPoint = 1)
```

```
summary(stage2)

# Stage 3: Execution
sample <- stage2[["sample"]]

# Stage 4: Evaluation
stage4 <- evaluation(materiality = 0.03, confidence = 0.95, sample = sample,
                     bookValues = "bookValue", auditValues = "auditValue",
                     prior = prior)
summary(stage4)

####################################################################
### Example 3: Bayesian audit sampling using an informed prior ####
####################################################################

# Create the prior distribution
prior <- auditPrior(method = "arm", likelihood = "poisson",
                    expectedError = 0.01, materiality = 0.03, cr = 0.6)
summary(prior)

# Stage 1: Planning
stage1 <- planning(materiality = 0.03, expectedError = 0.01,
                   likelihood = "poisson", confidence = 0.95, prior = prior)
summary(stage1)

# Stage 2: Selection
stage2 <- selection(population = BuildIt, sampleSize = stage1,
                    units = "mus", bookValues = "bookValue",
                    algorithm = "interval", intervalStartingPoint = 1)
summary(stage2)

# Stage 3: Execution
sample <- stage2[["sample"]]

# Stage 4: Evaluation
stage4 <- evaluation(materiality = 0.03, confidence = 0.95, sample = sample,
                     bookValues = "bookValue", auditValues = "auditValue",
                     prior = prior)
summary(stage4)
```

auditBF *Function to compute Bayes factors for audit sampling*

Description

This function computes Bayes factors for audit sampling from summary statistics of an audit sample. By default, the Bayes factor is computed using an impartial prior distribution on the misstatement (Derks et al., 2021). However, the arguments `nPrior` and `kPrior` can be used to specify an alternative prior distribution (Derks et al., 2021).

For more details on how to use this function, see the package vignette: `vignette('jfa',package = 'jfa')`

Usage

```
auditBF(materiality, n, k, expectedError = 0, likelihood = 'binomial',
        nPrior = NULL, kPrior = NULL, N = NULL, log = FALSE)
```

Arguments

`materiality`	a numeric value between 0 and 1 specifying the performance materiality (i.e., the maximum upper limit) as a fraction of the total population size. Can be `NULL` for some methods.
`n`	an integer larger than 0 specifying the number of items in the sample.
`k`	a number larger than zero specifying the observed proportional error (i.e., sum of taints) in the sample.
`expectedError`	a numeric value between 0 and 1 specifying the expected errors in the sample relative to the total sample size, or a numeric value (>= 1) that represents the sum of expected errors in the sample. It is advised to set this value conservatively to minimize the probability of the observed errors exceeding the expected errors, which would imply that insufficient work has been done in the end.
`likelihood`	a character specifying the likelihood assumed when updating the prior distribution. This can be either `binomial` for the binomial likelihood and beta prior distribution, `poisson` for the Poisson likelihood and gamma prior distribution, or `hypergeometric` for the hypergeometric likelihood and beta-binomial prior distribution. See the details section for more information about the available likelihoods.
`nPrior`	numeric value larger than, or equal to, 0 specifying the sample size of the sample equivalent to the prior information.
`kPrior`	a numeric value larger than, or equal to, 0 specifying the sum of errors in the sample equivalent to the prior information.
`N`	an integer larger than 0 specifying the total population size. Only required when `likelihood = 'hypergeometric'`.
`log`	logical; if TRUE, the Bayes factor is given as log(bf).

Details

The Bayes Factor BF_{-+} quantifies how much more likely the data are to be observed under H_{-} : $\theta < \theta_{max}$ than under $H_{+} : \theta > \theta_{max}$. Therefore, BF_{-+} can be interpreted as the relative support in the observed data for H_{-} versus H_{+}. If BF_{-+} is 1, there is no preference for either H_{-} or H_{+}. If BF_{-+} is larger than 1, H_{-} is preferred. If BF_{-+} is between 0 and 1, H_{+} is preferred.

This section elaborates on the available likelihoods and corresponding prior distributions for the `likelihood` argument.

- `binomial`: The binomial likelihood is often used as a likelihood for attributes sampling *with* replacement. The likelihood function is defined as:

  $$p(x) = \binom{n}{k} p^k (1-p)^{n-k}$$

  The conjugate *beta(α, β)* prior has probability density function:

  $$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

- `poisson`: The Poisson likelihood is often used as a likelihood for monetary unit sampling (MUS). The likelihood function is defined as:

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

The conjugate *gamma(α, β)* prior has probability density function:

$$f(x; \alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}$$

- `hypergeometric`: The hypergeometric likelihood is used as a likelihood for sampling *without* replacement. The likelihood function is defined as:

$$p(x = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

The conjugate *beta-binomial(α, β)* prior (Dyer and Pierce, 1993) has probability density function:

$$f(k|n, \alpha, \beta) = \binom{n}{k} \frac{B(k + \alpha, n - k + \beta)}{B(\alpha, \beta)}$$

Value

A value for the Bayes factor in favor of the hypothesis of tolerable misstatement against the hypothesis of intolerable misstatement.

Author(s)

Koen Derks, <k.derks@nyenrode.nl>

References

Derks, K., de Swart, J., van Batenburg, P., Wagenmakers, E.-J., & Wetzels, R. (2021). Priors in a Bayesian audit: How integration of existing information into the prior distribution can improve audit transparency and efficiency. *International Journal of Auditing*, 1-16.

See Also

`auditPrior` `planning` `selection` `evaluation` `report`

Examples

```
# Compute a default Bayes factor from an impartial prior
auditBF(materiality = 0.05, n = 50, k = 1)

# Compute a Bayes factor from a negligible prior
auditBF(materiality = 0.05, n = 50, k = 1, nPrior = 0, kPrior = 0)
```

`auditPrior` *Create a prior distribution for audit sampling*

Description

This function creates a prior distribution with audit information to be used in the `planning()` and `evaluation()` functions via their `prior` argument. The function returns an object of class `jfaPrior` which can be used with associated `summary()` and `plot()` methods.

For more details on how to use this function, see the package vignette: `vignette('jfa',package = 'jfa')`

Usage

```
auditPrior(method = 'none', likelihood = 'binomial', expectedError = 0,
           confidence = 0.95, materiality = NULL, N = NULL,
           ir = 1, cr = 1, ub = NULL, pHmin = NULL, pHplus = NULL,
           sampleN = 0, sampleK = 0, factor = 1)
```

Arguments

`method`	a character specifying the method by which the prior distribution is constructed. Defaults to none which incorporates no existing information. Other options are `arm`, `bram`, `median`, `hypotheses`, `sample`, and `factor`. See the details section for more information about the available methods.
`likelihood`	a character specifying the likelihood assumed when updating the prior distribution. This can be either `binomial` for the binomial likelihood and beta prior distribution, `poisson` for the Poisson likelihood and gamma prior distribution, or `hypergeometric` for the hypergeometric likelihood and beta-binomial prior distribution. See the details section for more information about the available likelihoods.
`expectedError`	a numeric value between 0 and 1 specifying the expected errors in the sample relative to the total sample size, or a numeric value (>= 1) that represents the sum of expected errors in the sample. It is advised to set this value conservatively to minimize the probability of the observed errors exceeding the expected errors, which would imply that insufficient work has been done in the end.
`confidence`	a numeric value between 0 and 1 specifying the confidence level to be used in the planning. Defaults to 0.95 for 95% confidence.
`materiality`	a numeric value between 0 and 1 specifying the performance materiality (i.e., the maximum upper limit) as a fraction of the total population size. Can be `NULL` for some methods.
`N`	an numeric value larger than 0 specifying the total population size. Optional unless `likelihood = 'hypergeometric'`.
`ir`	if `method = 'arm'`, a numeric value between 0 and 1 specifying the inherent risk in the audit risk model. Defaults to 1 for 100% risk.
`cr`	if `method = 'arm'`, a numeric value between 0 and 1 specifying the internal control risk in the audit risk model. Defaults to 1 for 100% risk.
`ub`	if `method = 'bram'`, a numeric value between 0 and 1 specifying the upper bound for the prior distribution as a fraction of the population size.

pHmin if method = 'hypotheses', a numeric value between 0 and 1 specifying the prior probability of the hypothesis $\theta <$ materiality.

pHplus if method = 'hypotheses', a numeric value between 0 and 1 specifying the prior probability of the hypothesis $\theta >$ materiality.

sampleN if method = 'sample' or method = 'factor', an integer larger than, or equal to, 0 specifying the sample size of the sample equivalent to the prior information.

sampleK if method = 'sample' or method = 'factor', a numeric value larger than, or equal to, 0 specifying the sum of errors in the sample equivalent to the prior information.

factor if method = 'factor', a numeric value between 0 and 1 specifying the weighting factor for the results of the sample equivalent to the prior information.

Details

This section elaborates on the available options for the method argument.

- none: This method constructs a prior distribution that incorporates negligible information about the possible values of the misstatement.
- arm: This method constructs a prior distribution by translating the risks of material misstatement (inherent risk and internal control risk) from the audit risk model to an implicit sample. The method requires specification of the ir (inherent risk) and cr (internal control risk) arguments.
- bram: This method constructs a prior distribution using the Bayesian audit risk assessment model (BRAM) in which the expected most likely error and expected upper bound of the misstatement must be specified. The method requires specification of the ub argument.
- median: This method constructs a prior distribution so that the prior probability of tolerable misstatement ($\theta <$ materiality) is equal to the prior probability of intolerable misstatement ($\theta >$ materiality).
- hypotheses: This method constructs a prior distribution with custom prior probabilities for the hypotheses of tolerable misstatement ($\theta <$ materiality) and intolerable misstatement ($\theta >$ materiality). This method requires specification of the pHmin and pHplus arguments.
- sample: This method constructs a prior distribution on the basis of an earlier sample. This method requires specification of the sampleN and sampleK arguments.
- factor: This method constructs a prior distribution on the basis of an earlier sample in combination with a weighting factor. This method requires specification of the sampleN, sampleK, and factor arguments.

This section elaborates on the available likelihoods and corresponding prior distributions for the likelihood argument.

- binomial: The binomial likelihood is often used as a likelihood for attributes sampling *with* replacement. The likelihood function is defined as:

$$p(x) = \binom{n}{k} p^k (1-p)^{n-k}$$

  The conjugate *beta*(α, β) prior has probability density function:

$$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

- poisson: The Poisson likelihood is often used as a likelihood for monetary unit sampling (MUS). The likelihood function is defined as:

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

The conjugate *gamma(α, β)* prior has probability density function:

$$f(x; \alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}$$

- hypergeometric: The hypergeometric likelihood is used as a likelihood for sampling *without* replacement. The likelihood function is defined as:

$$p(x = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

The conjugate *beta-binomial(α, β)* prior (Dyer and Pierce, 1993) has probability density function:

$$f(k|n, \alpha, \beta) = \binom{n}{k} \frac{B(k + \alpha, n - k + \beta)}{B(\alpha, \beta)}$$

Value

An object of class jfaPrior containing:

confidence	a numeric value between 0 and 1 indicating the confidence level used.
materiality	if materiality is specified, a numeric value between 0 and 1 indicating the materiality used to construct the prior distribution.
expectedError	a numeric value larger than, or equal to, 0 indicating the input for the number of expected errors.
method	a character indicating the method by which the prior distribution is constructed.
likelihood	a character indicating the assumed likelihood.
N	if N is specified, an integer larger than 0 indicating the population size.
description	a list containing a description of the prior distribution, including the parameters of the prior distribution and the implicit sample on which the prior distribution is based.
statistics	a list containing statistics of the prior distribution, including the mean, mode, median, and upper bound of the prior distribution.
specifics	a list containing specifics of the prior distribution that vary depending on the method.
hypotheses	if materiality is specified, a list containing information about the hypotheses, including prior probabilities and odds for the hypothesis of tolerable misstatement (H-) and the hypothesis of intolerable misstatement (H+).

Author(s)

Koen Derks, <k.derks@nyenrode.nl>

References

Derks, K., de Swart, J., Wagenmakers, E.-J., Wille, J., & Wetzels, R. (2019). JASP for audit: Bayesian tools for the auditing practice.

Derks, K., de Swart, J., van Batenburg, P., Wagenmakers, E.-J., & Wetzels, R. (2021). Priors in a Bayesian audit: How integration of existing information into the prior distribution can improve audit transparency and efficiency. *International Journal of Auditing*, 1-16.

See Also

`planning` `selection` `evaluation` `report` `auditBF`

Examples

```
# Translate inherent risk (ir) and control risk (cr) to a prior distribution
auditPrior(method = 'arm', likelihood = 'binomial', expectedError = 0.025,
           materiality = 0.05, ir = 1, cr = 0.6)
```

BuildIt — *BuildIt Construction financial statements*

Description

Fictional data from a construction company in the United States, containing 3500 observations identification numbers, book values, and audit values. The audit values are added for illustrative purposes, as these would need to be assessed by the auditor in the execution stage of the audit.

Usage

```
data(BuildIt)
```

Format

A data frame with 3500 rows and 3 variables.

ID unique record identification number.

bookValue book value in US dollars ($14.47–$2,224.40).

auditValue true value in US dollars ($14.47–$2,224.40).

References

Derks, K., de Swart, J., Wagenmakers, E.-J., Wille, J., & Wetzels, R. (2019). JASP for audit: Bayesian tools for the auditing practice.

Examples

```
data(BuildIt)
```

evaluation *Evaluate a statistical audit sample*

Description

This function takes a data frame (using `sample`, `bookValues`, and `auditValues`) or summary statistics (using `nSumstats` and `kSumstats`) and performs inference on the misstatement in the sample. The function returns an object of class `jfaEvaluation` which can be used with associated `summary()` and `plot()` methods.

For more details on how to use this function, see the package vignette: `vignette('jfa',package = 'jfa')`

Usage

```
evaluation(materiality = NULL, minPrecision = NULL, method = 'binomial',
         confidence = 0.95, sample = NULL, bookValues = NULL, auditValues = NULL,
           counts = NULL, nSumstats = NULL, kSumstats = NULL,
           N = NULL, populationBookValue = NULL,
           prior = FALSE, nPrior = 0, kPrior = 0,
           rohrbachDelta = 2.7, momentPoptype = 'accounts',
           csA = 1, csB = 3, csMu = 0.5)
```

Arguments

`materiality`	a numeric value between 0 and 1 specifying the performance materiality (maximum tolerable error) as a fraction of the total size of the population. If specified, the function also returns the conclusion of the analysis with respect to the performance materiality. The value is discarded when `direct`, `difference`, `quotient`, or `regression` method is chosen.
`minPrecision`	a numeric value between 0 and 1 specifying the required minimum precision (upper bound minus most likely error) as a fraction of the total size of the population. If specified, the function also returns the conclusion of the analysis with respect to the required minimum precision.
`method`	a character specifying the method to be used in the evaluation. Possible options are `poisson`, `binomial` (default), `hypergeometric`, `mpu`, `stringer`, `stringer-meikle`, `stringer-lta`, `stringer-pvz`, `rohrbach`, `moment`, `direct`, `difference`, `quotient`, or `regression`. See the details section for more information.
`confidence`	a numeric value between 0 and 1 specifying the confidence level used in the evaluation. Defaults to 0.95 for 95% confidence.
`sample`	a data frame containing the sample to be evaluated. The sample must at least contain a column of book values and a column of audit (true) values.
`bookValues`	a character specifying the column name for the book values in the `sample`.
`auditValues`	a character specifying the column name for the audit values in the `sample`.
`counts`	a integer vector specifying the number of times each item in the sample should be counted in the evaluation (due to it being selected multiple times for the sample).

nSumstats
: an integer larger than 0 specifying the number of items in the sample. If specified, overrides the sample, bookValues and auditValues arguments and assumes that the data come from summary statistics specified by both nSumstats and kSumstats.

kSumstats
: a numeric value larger than 0 specifying the sum of errors found in the sample. If specified, overrides the sample, bookValues and auditValues arguments and assumes that the data come from summary statistics specified by both kSumstats and nSumstats.

N
: an integer larger than 0 specifying the total number of items in the population.

populationBookValue
: if method is one of direct, difference, quotient, or regression, a numeric value specifying the total value of the items in the population. This argument is optional otherwise.

prior
: a logical specifying if a prior distribution must be used, or an object of class jfaPrior or jfaPosterior containing the prior distribution. Defaults to FALSE for frequentist planning. If TRUE, a negligible prior distribution is chosen by default, but can be adjusted using the 'kPrior' and 'nPrior' arguments. Chooses a conjugate gamma distribution for the Poisson likelihood, a conjugate beta distribution for the binomial likelihood, and a conjugate beta-binomial distribution for the hypergeometric likelihood.

nPrior
: if prior = TRUE, a numeric value larger than, or equal to, 0 specifying the sample size of the sample equivalent to the prior information.

kPrior
: if prior = TRUE, a numeric value larger than, or equal to, 0 specifying the sum of errors in the sample equivalent to the prior information.

rohrbachDelta
: if method = 'rohrbach', a numeric value specifying Δ in Rohrbach's augmented variance bound (Rohrbach, 1993).

momentPoptype
: if method = 'moment', a character specifying the type of population (Dworin and Grimlund, 1984). Possible options are accounts and inventory. This argument affects the calculation of the central moments in the bound.

csA
: if method = "coxsnell", a numeric value specifying the α parameter of the prior distribution on the mean taint. Defaults to 1 as recommended by Cox and Snell (1979).

csB
: if method = "coxsnell", a numeric value specifying the β parameter of the prior distribution on the mean taint. Defaults to 3 as recommended by Cox and Snell (1979).

csMu
: if method = "coxsnell", a numeric value between 0 and 1 specifying the mean of the prior distribution on the mean taint. Defaults to 0.5 as recommended by Cox and Snell (1979).

Details

This section lists the available options for the methods argument.

- poisson: Evaluates the sample with the Poisson distribution. If combined with prior = TRUE, performs Bayesian evaluation using a *gamma* prior and posterior.
- binomial: Evaluates the sample with the binomial distribution. If combined with prior = TRUE, performs Bayesian evaluation using a *beta* prior and posterior.
- hypergeometric: Evaluates the sample with the hypergeometric distribution. If combined with prior = TRUE, performs Bayesian evaluation using a *beta-binomial* prior and posterior.

- `mpu`: Evaluates the sample with the mean-per-unit estimator.
- `stringer`: Evaluates the sample with the Stringer bound (Stringer, 1963).
- `stringer-meikle`: Evaluates the sample with the Stringer bound with Meikle's correction for understatements (Meikle, 1972).
- `stringer-lta`: Evaluates the sample with the Stringer bound with LTA correction for understatements (Leslie, Teitlebaum, and Anderson, 1979).
- `stringer-pvz`: Evaluates the sample with the Stringer bound with Pap and van Zuijlen's correction for understatements (Pap and van Zuijlen, 1996).
- `rohrbach`: Evaluates the sample with Rohrbach's augmented variance bound (Rohrbach, 1993).
- `moment`: Evaluates the sample with the modified moment bound (Dworin and Grimlund, 1984).
- `coxsnell`: Evaluates the sample with the Cox and Snell bound (Cox and Snell, 1979).
- `direct`: Evaluates the sample with the direct estimator (Touw and Hoogduin, 2011).
- `difference`: Evaluates the sample with the difference estimator (Touw and Hoogduin, 2011).
- `quotient`: Evaluates the sample with the quotient estimator (Touw and Hoogduin, 2011).
- `regression`: Evaluates the sample with the regression estimator (Touw and Hoogduin, 2011).

Value

An object of class `jfaEvaluation` containing:

`confidence`	a numeric value between 0 and 1 indicating the confidence level.
`materiality`	if `materiality` is specified, a numeric value between 0 and 1 indicating the performance materiality as a fraction of the total population size.
`minPrecision`	if `minPrecision` is specified, a numeric value between 0 and 1 indicating the minimum required precision as a fraction of the total population size.
`method`	a character indicating the evaluation method.
`N`	if `N` is specified, in integer larger than 0 indicating the population size.
`n`	an integer larger than 0 indicating the sample size.
`k`	an integer larger than, or equal to, 0 indicating the number of items in the sample that contained an error.
`t`	a value larger than, or equal to, 0, indicating the sum of observed taints.
`mle`	a numeric value between 0 and 1 indicating the most likely error in the population as a fraction of its total size.
`precision`	a numeric value between 0 and 1 indicating the difference between the most likely error and the upper bound in the population as a fraction of the total population size.
`popBookvalue`	if `populationBookValue` is specified, a numeric value larger than 0 indicating the total value of the population.
`pointEstimate`	if `method` is one of `direct`, `difference`, `quotient`, or `regression`, a numeric value indicating the point estimate of the population misstatement as a fraction the total population size.
`lowerBound`	if method is one of `direct`, `difference`, `quotient`, or `regression`, a numeric value indicating the lower bound of the interval around the population misstatement as a fraction the total population size.

`upperBound`	if method is one of `direct`, `difference`, `quotient`, or `regression`, a numeric value indicating the upper bound of the interval around the population misstatement as a fraction the total population size.
`confBound`	a numeric value indicating the upper bound on the population misstatement as a fraction the total population size.
`conclusion`	if `materiality` is specified, a character indicating the conclusion about whether to approve or not approve the population with respect to the performance materiality.
`populationK`	if `method = 'hypergeometric'`, an integer indicating the assumed total errors in the population.
`prior`	an object of class 'jfaPrior' that contains the prior distribution.
`posterior`	an object of class 'jfaPosterior' that contains the posterior distribution.
`data`	a data frame containing the relevant columns from the `sample`.

Author(s)

Koen Derks, <k.derks@nyenrode.nl>

References

Cox, D. and Snell, E. (1979). On sampling and the estimation of rare errors. *Biometrika*, 66(1), 125-132.

Derks, K., de Swart, J., van Batenburg, P., Wagenmakers, E.-J., & Wetzels, R. (2021). Priors in a Bayesian audit: How integration of existing information into the prior distribution can improve audit transparency and efficiency. *International Journal of Auditing*, 1-16.

Dworin, L. D. and Grimlund, R. A. (1984). Dollar-unit sampling for accounts receivable and inventory. *The Accounting Review*, 59(2), 218–241

Leslie, D. A., Teitlebaum, A. D., & Anderson, R. J. (1979). *Dollar-unit Sampling: A Practical Guide for Auditors*. Copp Clark Pitman; Belmont, Calif.: distributed by Fearon-Pitman.

Meikle, G. R. (1972). *Statistical Sampling in an Audit Context: An Audit Technique*. Canadian Institute of Chartered Accountants.

Pap, G., and van Zuijlen, M. C. (1996). On the asymptotic behavior of the Stringer bound. *Statistica Neerlandica*, 50(3), 367-389.

Rohrbach, K. J. (1993). Variance augmentation to achieve nominal coverage probability in sampling from audit populations. *Auditing*, 12(2), 79.

Stringer, K. W. (1963). Practical aspects of statistical sampling in auditing. *In Proceedings of the Business and Economic Statistics Section* (pp. 405-411). American Statistical Association.

Touw, P., and Hoogduin, L. (2011). *Statistiek voor Audit en Controlling*. Boom uitgevers Amsterdam.

See Also

`auditPrior` `planning` `selection` `report` `auditBF`

Examples

```
data('BuildIt')

# Draw a sample of 100 monetary units from the population using
# fixed interval monetary unit sampling
sample <- selection(population = BuildIt, sampleSize = 100,
          algorithm = 'interval', units = 'mus', bookValues = 'bookValue')$sample

# Evaluate using the Stringer bound
evaluation(materiality = 0.05, method = 'stringer', confidence = 0.95,
           sample = sample, bookValues = 'bookValue', auditValues = 'auditValue')
```

---

jfa-methods *Methods for jfa objects*

Description

Methods defined for objects returned from the `auditPrior`, `planning`, `selection`, and `evaluation` functions.

Usage

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

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

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

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

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

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

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

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

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

## S3 method for class 'summary.jfaEvaluation'
```

```
print(x, ...)

## S3 method for class 'jfaPrior'
summary(object, digits = 3, ...)

## S3 method for class 'jfaPosterior'
summary(object, digits = 3, ...)

## S3 method for class 'jfaPlanning'
summary(object, digits = 3, ...)

## S3 method for class 'jfaSelection'
summary(object, digits = 3, ...)

## S3 method for class 'jfaEvaluation'
summary(object, digits = 3, ...)

## S3 method for class 'jfaPrior'
plot(x, xmax = 0.5, ...)

## S3 method for class 'jfaPosterior'
plot(x, xmax = 0.5, ...)

## S3 method for class 'jfaPlanning'
plot(x, xmax = 0.5, ...)

## S3 method for class 'jfaSelection'
plot(x, ...)

## S3 method for class 'jfaEvaluation'
plot(x, xmax = 0.5, ...)
```

Arguments

`...`	further arguments, currently ignored.
`object, x`	an object of class `jfaPrior`, `jfaPosterior`, `jfaPlanning`, `jfaSelection`, or `jfaEvaluation`.
`digits`	an integer specifying the number of digits to which output should be rounded. Used in `summary`.
`xmax`	a number between 0 and 1 specifying the x-axis limits of the plot. Used in `plot`.

Value

The `summary` methods return a `data.frame` which contains the input and output.

The `print` methods simply print and return nothing.

`planning` *Plan a statistical audit sample*

Description

This function calculates the minimum sample size for a statistical audit sample based on the binomial, Poisson, or hypergeometric likelihood. The function returns an object of class `jfaPlanning` which can be used with associated `summary()` and `plot()` methods.

For more details on how to use this function, see the package vignette: `vignette('jfa',package = 'jfa')`

Usage

```
planning(materiality = NULL, minPrecision = NULL, expectedError = 0,
         likelihood = 'binomial', confidence = 0.95, N = NULL,
         prior = FALSE, nPrior = 0, kPrior = 0,
         increase = 1, maxSize = 5000)
```

Arguments

`materiality`	a numeric value between 0 and 1 specifying the performance materiality (i.e., maximum upper limit) as a fraction of the total population size. Can be `NULL`, but `minPrecision` should be specified in that case.
`minPrecision`	a numeric value between 0 and 1 specifying the minimum precision (i.e., upper bound minus most likely error) as a fraction of the total population size. Can be `NULL`, but `materiality` should be specified in that case.
`expectedError`	a numeric value between 0 and 1 specifying the expected errors in the sample relative to the total sample size, or a number (>= 1) that represents the number of expected errors in the sample. It is advised to set this value conservatively to minimize the probability of the observed errors exceeding the expected errors, which would imply that insufficient work has been done in the end.
`likelihood`	a character specifying the likelihood assumed in the calculation. This can be either `binomial` for the binomial likelihood, `poisson` for the Poisson likelihood, or `hypergeometric` for the hypergeometric likelihood. See the details section for more information about the available likelihoods.
`confidence`	a numeric value between 0 and 1 specifying the confidence level used in the planning. Defaults to 0.95 for 95% confidence.
`N`	an integer larger than 0 specifying the total population size. Only required when `likelihood = 'hypergeometric'`.
`prior`	a logical specifying whether to use a prior distribution when planning, or an object of class `jfaPrior` or `jfaPosterior` containing the prior distribution. Defaults to `FALSE` for frequentist planning. If `TRUE`, a negligible prior distribution is chosen by default, but can be adjusted using the `kPrior` and `nPrior` arguments. Chooses a conjugate gamma distribution for the Poisson likelihood, a conjugate beta distribution for the binomial likelihood, and a conjugate beta-binomial distribution for the hypergeometric likelihood.
`nPrior`	if `prior = TRUE`, a numeric value larger than, or equal to, 0 specifying the sample size of the sample equivalent to the prior information.

`kPrior`	if `prior = TRUE`, a numeric value larger than, or equal to, 0 specifying the sum of errors in the sample equivalent to the prior information.
`increase`	an integer larger than 0 specifying the desired increase step for the sample size calculation.
`maxSize`	an integer larger than 0 specifying the maximum sample size that is considered in the calculation. Defaults to 5000 for efficiency. Increase this value if the sample size cannot be found due to it being too large (e.g., for a low materiality).

Details

This section elaborates on the available likelihoods and corresponding prior distributions for the `likelihood` argument.

- `binomial`: The binomial likelihood is often used as a likelihood for attributes sampling *with* replacement. The likelihood function is defined as:

  $$p(x) = \binom{n}{k} p^k (1-p)^{n-k}$$

  The conjugate *beta*(α, β) prior has probability density function:

  $$f(x; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1} (1-x)^{\beta-1}$$

- `poisson`: The Poisson likelihood is often used as a likelihood for monetary unit sampling (MUS). The likelihood function is defined as:

  $$p(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

  The conjugate *gamma*(α, β) prior has probability density function:

  $$f(x; \alpha, \beta) = \frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}$$

- `hypergeometric`: The hypergeometric likelihood is used as a likelihood for sampling *without* replacement. The likelihood function is defined as:

  $$p(x = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

  The conjugate *beta-binomial*(α, β) prior (Dyer and Pierce, 1993) has probability density function:

  $$f(k|n, \alpha, \beta) = \binom{n}{k} \frac{B(k+\alpha, n-k+\beta)}{B(\alpha, \beta)}$$

Value

An object of class `jfaPlanning` containing:

`confidence`	a numeric value between 0 and 1 indicating the confidence level.
`materiality`	a numeric value between 0 and 1 indicating the specified materiality. Can be `NULL`.
`minPrecision`	a numeric value between 0 and 1 indicating the minimum precision to be obtained. Can be `NULL`.

`expectedError`	a numeric value larger than, or equal to, 0 indicating the expected errors input.
`likelihood`	a character indicating the specified likelihood.
`N`	an integer larger than 0 indicating the population size (only returned if `N` is specified).
`sampleSize`	an integer larger than 0 indicating the required sample size.
`errorType`	a character indicating whether the expected errors where specified as a percentage or as an integer.
`expectedSampleError`	a numeric value larger than, or equal to, 0 indicating the number of errors that are allowed in the sample.
`expectedBound`	a numeric value between 0 and 1 indicating the expected upper bound if the sample goes according to plan.
`expectedPrecision`	a numeric value between 0 and 1 indicating the expected precision if the sample goes according to plan.
`populationK`	if `likelihood = 'hypergeometric'`, an integer larger than 0 indicating the assumed population errors.
`prior`	if a prior distribution is specified, an object of class `jfaPrior` that contains information about the prior distribution.
`expectedPosterior`	if a prior distribution is specified, an object of class `jfaPosterior` that contains information about the expected posterior distribution.

Author(s)

Koen Derks, <k.derks@nyenrode.nl>

References

Derks, K., de Swart, J., van Batenburg, P., Wagenmakers, E.-J., & Wetzels, R. (2021). Priors in a Bayesian audit: How integration of existing information into the prior distribution can improve audit transparency and efficiency. *International Journal of Auditing*, 1-16.

Dyer, D. and Pierce, R.L. (1993). On the choice of the prior distribution in hypergeometric sampling. *Communications in Statistics - Theory and Methods*, 22(8), 2125 - 2146.

See Also

`auditPrior` `selection` `evaluation` `report` `auditBF`

Examples

```
# Frequentist planning using a binomial likelihood
planning(materiality = 0.05, expectedError = 0.025, likelihood = 'binomial')

# Bayesian planning using a negligible beta prior
planning(materiality = 0.05, expectedError = 0.025, likelihood = 'binomial',
         prior = TRUE)

# Bayesian planning using an informed beta prior
planning(materiality = 0.05, expectedError = 0.025, likelihood = 'binomial',
         prior = auditPrior(method = 'median', materiality = 0.05))
```

`report` *Create a statistical audit sampling report*

Description

This function takes an object of class `jfaEvaluation` as returned by the `evaluation()` function automatically generates a `html` or `pdf` report containing the analysis results and their interpretation.

For more details on how to use this function, see the package vignette: `vignette('jfa',package = 'jfa')`

Usage

```
report(object, file = 'report.html', format = 'html_document')
```

Arguments

`object`	an object of class `jfaEvaluation` as returned by the `evaluation()` function.
`file`	a character specifying the name of the report (e.g. `report.html`). By default, the report is created in your current working directory.
`format`	a character specifying the output format of the report. Possible options are `html_document` (default) and `pdf_document`, but compiling to `pdf` format requires a local version of MikTex.

Value

A `html` or `pdf` file containing a report of the evaluation.

Author(s)

Koen Derks, `<k.derks@nyenrode.nl>`

See Also

`auditPrior` `planning` `selection` `evaluation` `auditBF`

Examples

```
data('BuildIt')

# Draw a sample of 100 monetary units from the population using
# fixed interval monetary unit sampling
sample <- selection(population = BuildIt, sampleSize = 100,
          algorithm = 'interval', units = 'mus', bookValues = 'bookValue')$sample

# Evaluate using the Stringer bound
result <- evaluation(confidence = 0.95, materiality = 0.05,
                     method = 'stringer', sample = sample,
                     bookValues = 'bookValue', auditValues = 'auditValue')

## Not run:
 report(result)
```

```
## End(Not run)
```

---

selection *Select a statistical audit sample*

Description

This function takes a data frame and performs statistical sampling according to one of three algorithms: random sampling, cell sampling, and fixed interval sampling. Sampling is done on the level of two possible sampling units: records or monetary units. The function returns an object of class `jfaSelection` which can be used with associated `summary()` and a `plot()` methods.

For more details on how to use this function, see the package vignette: `vignette('jfa',package = 'jfa')`

Usage

```
selection(population, sampleSize, units = 'records', algorithm = 'random',
          bookValues = NULL, intervalStartingPoint = 1, ordered = TRUE,
          ascending = TRUE, withReplacement = FALSE, seed = 1)
```

Arguments

`population` a data frame containing the population of items the auditor wishes to sample from.

`sampleSize` an integer larger than 0 specifying the number of sampling units that need to be selected from the population. Can also be an object of class `jfaPlanning`.

`units` a character specifying the sampling units used. Possible options are `records` (default) for selection on the level of items or `mus` for selection on the level of monetary units.

`algorithm` a character specifying the sampling algorithm used. Possible options are `random` (default) for random sampling, `cell` for cell sampling, or `interval` for fixed interval sampling.

`bookValues` a character specifying the name of the column in the `population` that contains the book values of the items.

`intervalStartingPoint` if `algorithm = 'interval'`, an integer larger than 0 specifying the starting point of the algorithm.

`ordered` a logical specifying whether to first order the items in the `population` according to the value of their `bookValues`. Defaults to `TRUE`.

`ascending` if `ordered = TRUE`, a logical specifying whether to order the population `bookValues` from smallest to largest. Defaults to `TRUE`.

`withReplacement` if `algorithm = 'random'`, a logical specifying whether sampling should be performed with replacement. Defaults to `FALSE`.

`seed` if `algorithm = 'random'` or `algorithm = 'cell'`, an integer specifying a seed to reproduce results. Defaults to 1.

Details

The first part of this section elaborates on the two possible options for the `units` argument:

- `records`: In record sampling each item in the population is seen as a sampling unit. An item of $5000 is therefore equally likely to be selected as an item of $500.
- `mus`: In monetary unit sampling each monetary unit in the population is seen as a sampling unit. An item of $5000 is therefore ten times more likely to be selected as an item of $500.

The second part of this section elaborates on the three possible options for the `algorithm` argument:

- `random`: In random sampling each sampling unit in the population is drawn with equal probability.
- `cell`: In cell sampling the sampling units in the population are divided into a number (equal to the sample size) of intervals. From each interval one sampling unit is selected with equal probability.
- `interval`: In fixed interval sampling the sampling units in the population are divided into a number (equal to the sample size) of intervals. From each interval one sampling unit is selected according to a fixed starting point (specified by `intervalStartingPoint`).

Value

An object of class `jfaSelection` containing:

`population`	a data frame containing the input population.
`sample`	a data frame containing the selected sample of items.
`units`	a character indicating the sampling units that were used to create the selection.
`algorithm`	a character indicating the the algorithm that was used to create the selection.
`bookValues`	if `bookValues` is specified, a character indicating the name of the book value column.

Author(s)

Koen Derks, `<k.derks@nyenrode.nl>`

References

Leslie, D. A., Teitlebaum, A. D., & Anderson, R. J. (1979). *Dollar-unit Sampling: A Practical Guide for Auditors*. Copp Clark Pitman; Belmont, Calif.: distributed by Fearon-Pitman.

Wampler, B., & McEacharn, M. (2005). Monetary-unit sampling using Microsoft Excel. *The CPA journal*, 75(5), 36.

See Also

`auditPrior` `planning` `evaluation` `report` `auditBF`

Examples

```
data('BuildIt')

# Draw a sample of 100 monetary units from the population using
# fixed interval monetary unit sampling
selection(population = BuildIt, sampleSize = 100,
          algorithm = 'interval', units = 'mus', bookValues = 'bookValue')
```

Index

∗ **Bayes**
    auditBF, 4
∗ **audit**
    auditBF, 4
    auditPrior, 7
    evaluation, 11
    planning, 17
    report, 20
    selection, 21
∗ **bound**
    evaluation, 11
∗ **confidence**
    evaluation, 11
∗ **datasets**
    BuildIt, 10
∗ **distribution**
    auditBF, 4
    auditPrior, 7
∗ **evaluation**
    evaluation, 11
    report, 20
∗ **factor**
    auditBF, 4
∗ **jfa**
    jfa-package, 2
∗ **package**
    jfa-package, 2
∗ **planning**
    planning, 17
∗ **prior**
    auditBF, 4
    auditPrior, 7
∗ **report**
    report, 20
∗ **sample**
    planning, 17
    selection, 21
∗ **selection**
    selection, 21
∗ **size**
    planning, 17

auditBF, 4, *10*, *14*, *19*, *20*, *22*
auditPrior, *6*, 7, *14*, *15*, *19*, *20*, *22*

BuildIt, 10

evaluation, *6*, *10*, 11, *15*, *19*, *20*, *22*

jfa *(*jfa-package*)*, 2
jfa-methods, 15
jfa-package, 2

planning, *6*, *10*, *14*, *15*, 17, *20*, *22*
plot.jfaEvaluation *(*jfa-methods*)*, 15
plot.jfaPlanning *(*jfa-methods*)*, 15
plot.jfaPosterior *(*jfa-methods*)*, 15
plot.jfaPrior *(*jfa-methods*)*, 15
plot.jfaSelection *(*jfa-methods*)*, 15
print.jfaEvaluation *(*jfa-methods*)*, 15
print.jfaPlanning *(*jfa-methods*)*, 15
print.jfaPosterior *(*jfa-methods*)*, 15
print.jfaPrior *(*jfa-methods*)*, 15
print.jfaSelection *(*jfa-methods*)*, 15
print.summary.jfaEvaluation *(*jfa-methods*)*, 15
print.summary.jfaPlanning *(*jfa-methods*)*, 15
print.summary.jfaPosterior *(*jfa-methods*)*, 15
print.summary.jfaPrior *(*jfa-methods*)*, 15
print.summary.jfaSelection *(*jfa-methods*)*, 15

report, *6*, *10*, *14*, *19*, 20, *22*

selection, *6*, *10*, *14*, *15*, *19*, *20*, 21
summary.jfaEvaluation *(*jfa-methods*)*, 15
summary.jfaPlanning *(*jfa-methods*)*, 15
summary.jfaPosterior *(*jfa-methods*)*, 15
summary.jfaPrior *(*jfa-methods*)*, 15
summary.jfaSelection *(*jfa-methods*)*, 15