Audit sampling with jfa:: CHEAT SHEET



Basics

ifa is an R package that facilitates statistical planning, selection, and evaluation of audit samples.

The package provides five functions that allow users to easily apply Bayesian or classical probability theory in the standard audit sampling workflow.

Installation

Installing the package can be done via: install.packages('jfa')

Loading the package can be done via: library(jfa)

Example

the intended workflow.

The data for this example can be loaded via: data('BuildIt')

Construct a prior probability distribution (optional)

jfa::auditPrior()

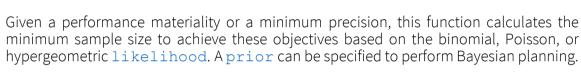
This function creates a prior distribution for the misstatement in the population based on audit evidence specified via the method argument. The prior distribution can be used as input for the prior argument in other functions to perform Bayesian inference.

• likelihood: Specifies the family of the prior probability distribution.



Calculate the minimum sample size

jfa::planning()



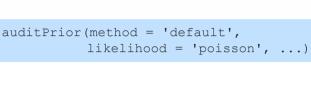
• expected: A fraction or an integer specifying the expected errors in the sample.

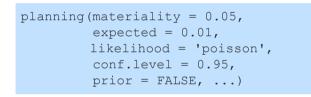
This function takes a data frame and performs sampling according to one of three

popular method's: random sampling, cell sampling, or fixed interval sampling. Sampling

is done in combination with one of two sampling units: items or monetary units.

Select the required items from the population





The blue code blocks next to the function descriptions provide a working example of







Evaluate the misstatement in the population

jfa::evaluation()

jfa::selection()



This function takes a data sample (using data, values, and values.audit) or summary statistics from a sample (using x and n) and performs statistical evaluation on the misstatement in the population according to the specified method. A prior can be specified to perform Bayesian evaluation.

• prior: An object returned by auditPrior () that specifies the prior distribution.

```
selection(data = BuildIt,
  size = 93,
  units = 'rows',
  method = 'interval', ...)
```

method = 'poisson',

conf.level = 0.95,

prior = FALSE, ...)

Create a report of the statistical results

jfa::report()



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

```
report(object = evaluationResult,
file = 'report.html', ...)
```

evaluation (materiality = 0.05,

x = 0n = 93,