Audit sampling with jfa: : **CHEAT SHEET**

Basics

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

```
auditPrior(method = 'none',
           likelihood = 'poisson', ...)
```

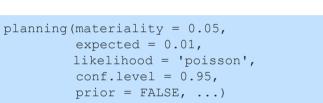## Calculate the minimum sample size

`jfa::planning()`

Given a performance materiality or a minimum precision, this function calculates the minimum sample size to achieve these objectives based on the binomial, Poisson, or hypergeometric `likelihood`. A `prior` can be specified to perform Bayesian planning.

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

```
planning(materiality = 0.05,
         expected = 0.01,
         likelihood = 'poisson',
         conf.level = 0.95,
         prior = FALSE, ...)
```

Select the required items from the population

`jfa::selection()`

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.

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

Evaluate the misstatement in the population

`jfa::evaluation()`

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.

```
evaluation(materiality = 0.05,
           method = 'poisson',
           conf.level = 0.95,
           x = 0,
           n = 93,
           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', ...)
```