Package 'fitPS'

January 20, 2023

Type Package

Title Fit Zeta Distributions to Forensic Data

Version 0.2-2

Date 2023-01-19

Description Fits zeta distributions (discrete power laws) to data that arises from forensic surveys of clothing on the presence of glass and paint in various populations. The general method is described to some extent in Coulson, S.A., Buckleton, J.S., Gummer, A.B., and Triggs, C.M. (2001) <doi:10.1016/S1355-0306(01)71847-3>, although the implementation differs.

License GPL (>= 2)

Encoding UTF-8

Depends dplyr, Hmisc, VGAM, methods, readxl

RoxygenNote 7.2.3

URL https://github.com/jmcurran/fitPS

BugReports https://github.com/jmcurran/fitPS/issues

NeedsCompilation no

Author James Curran [aut, cre]

Maintainer James Curran <j.curran@auckland.ac.nz>

Repository CRAN

Date/Publication 2023-01-20 21:30:05 UTC

R topics documented:

`fitDist` *Fit a Zeta Distribution to Forensic Data*

Description

This function uses maximum likelihood estimation (MLE) to estimate the shape parameter of a zeta distribution from a set of observed counts for either the number of groups/sources of forensically interesting material (mostly glass or paint) recovered from clothing, or the number of fragments/particles in each group. This, in turn, allows the estimation of the P and S probabilities, as described by Evett and Buckleton (1990), which used in computing the likelihood ratio (LR) for activity level propositions. The data itself arises from clothing surveys. The general method is described in Coulson et al. (2001), although poor typesetting, and a lack of definition of terms makes it hard to see. This package improves on the estimation in that linear interpolation is not required, and standard numerical optimisation is used instead. The zeta distribution has probability mass function

$$p(k) = \frac{k^{-s}}{\zeta(s)}$$

where $\zeta(s)$ is the Reimann zeta function. Coulson et al. (2001) did not have an easy way to rapidly compute this quantity, hence their use of linear interpolation.

Usage

```
fitDist(x, nterms = 10, start = 1 + runif(1), ...)
```

Arguments

`x`	an object of type `psData`, usually obtained from `readData`.
`nterms`	the number of terms to compute the probability distribution for
`start`	a starting value for the optimiser
`...`	other parameters - not currently used.

Details

The function returns an object of class `psFit` which is a `list` contains four elements:

- `psData` – an object of class `psData`–see `readData`,
- `fit` – the fitted object from `nlminb`,
- `shape` – the maximum likelihood estimate of the shape parameter,
- `varshape` - the maximum likelihood estimate of the shape parameter,
- `fitted` - a named `vector` containing the first `nterms` of the `fitted distribution`.

. The output can be used in a variety of ways. If the interest is just in the shape parameter estimate, then the `shape` member of the `psFit` object contains this information. It is also displayed along with a number of fitted probabilities by the `print.psFit` method. The fitted object can also be plotted using the plot method `plot.psFit`, and to create a probability function with `probfun`.

Value

an object of class psFit–see Details.

References

Coulson, S. A., Buckleton, J. S., Gummer, A. B., and Triggs, C.M., "Glass on clothing and shoes of members of the general population and people suspected of breaking crimes", Science & Justice 2001: 41(1): 39–48.

Evett, I. W. and Buckleton, J. S., "The interpretation of glass evidence. A practical approach", Journal of the Forensic Science Society 1990: 30(4): 215–223.

See Also

plot.psFit, print.psFit, probfun.

Examples

```
p = readData(system.file("extdata", "p.xlsx", package = "fitPS"))
fit = fitDist(p)
fit
```

---

plot.psFit *S3 plot method for an object of class* psFit

Description

S3 plot method for an object of class psFit

Usage

```
## S3 method for class 'psFit'
plot(x, ylim = c(0, 1), conf = FALSE, conf.level = 0.95, ...)
```

Arguments

x	an object of class psFit, usually from fitDist
ylim	the limits of the y-axis
conf	if TRUE, then confidence intervals (based on the standard error of the shape parameter) are drawn on the plot
conf.level	the confidence level for the confidence intervals. Must be between 0.75 and 0.99.
...	other arguments passed to plot

Value

No return value, called for side effects

Examples

```
p = readData(system.file("extdata", "p.xlsx", package = "fitPS"))
fit = fitDist(p)
plot(fit)
```

`print.psFit` *S3 print method for an object of class* `psFit`

Description

S3 print method for an object of class `psFit`

Usage

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

Arguments

`x`	an object of class `psFit`, usually from `fitDist`
`...`	other arguments passed to `print`

Value

No return value, called for side effects

`probfun` *Probability Functions*

Description

Creates a probability function that allows the computation of any P or S term.

Usage

```
probfun(psFitobj)
```

Arguments

`psFitobj`	an object of class `psFit`–see `fitDist`.

Value

a function that can be used to calculate any P or S term.

Examples

```
p = readData(system.file("extdata", "p.xlsx", package = "fitPS"))
fit = fitDist(p)
P = probfun(fit)
P(0:5)
```

`readData` *Read count data from file*

Description

Reads observed counts of either the number of groups or the size of the groups. The file must have only two columns. One of the columns must be labelled P or S and the other count. It does not matter if the column names are in upper case or not. The P column can have labels 0, 1, 2, ... representing the observation of 0, 1, 2, or more groups. The corresponding count column should contain a positive (non-zero) count for each number of groups. Similarly, if the file contains S counts, then the S column can contain labels 1, 2, ... representing the observation of 1, 2, ... fragments in a group. Note that zeros are neither allowed, or useful, in the file as they both simply result in log-likelihood terms of zero, and therefore make no difference.

Usage

```
readData(fileName, ...)
```

Arguments

`fileName`	the name of the file to be read. Must be either a modern (xlsx) Excel file or a csv file.
`...`	any additional parameters which will be passed to either `read_excel` or `read.csv` depending on the extension of your input file.

Value

an object of class `psData` which is a list containing member variable `type` which is either `P` or `S`, and a variable `data` that is a `data.frame` which contains columns `n` and `rn`, representing the number of groups/fragments, and the number of times that was seen, respectively.

Examples

```
p = readData(system.file("extdata", "p.xlsx", package = "fitPS"))
p
s = readData(system.file("extdata", "s.xlsx", package = "fitPS"))
s
```

Index