# Package ‘fitPS’

**June 10, 2023**

**Type** Package  
**Title** Fit Zeta Distributions to Forensic Data  
**Version** 0.2-6  
**Date** 2023-06-07  
**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  
**LazyData** true  
**Depends** foreach, R (>= 4.0.0)  
**Imports** doParallel, dplyr, Hmisc, iterators, knitr, ks, methods, pbapply, Rdpack, readxl, VGAM  
**RdMacros** Rdpack  
**RoxygenNote** 7.2.3  
**URL** [https://github.com/jmcurran/fitPS](https://github.com/jmcurran/fitPS)  
**BugReports** [https://github.com/jmcurran/fitPS/issues](https://github.com/jmcurran/fitPS/issues)  
**Suggests** rmarkdown, sp  
**VignetteBuilder** knitr  
**NeedsCompilation** no  
**Author** James Curran [aut, cre]  
**Maintainer** James Curran <j.curran@auckland.ac.nz>  
**Repository** CRAN  
**Date/Publication** 2023-06-10 11:50:02 UTC
R topics documented:

==.psData .................................................. 2
as.data.frame.psData ...................................... 3
bootCI ......................................................... 4
confint.psFit ................................................ 6
fitDist ......................................................... 7
fitted.psData ................................................ 9
fitZIlist ......................................................... 9
makePSData .................................................. 11
mean.psData .................................................. 12
plot.psFit ...................................................... 13
predict.psFit ................................................ 14
print.psData .................................................. 15
print.psFit ...................................................... 15
probFun ......................................................... 16
Psurveys ......................................................... 16
readData ....................................................... 17
rZIlist ........................................................ 18
Ssurveys ......................................................... 19
var .............................................................. 20
var.psData ...................................................... 21

Index 22

==.psData

S3 method for objects of class psData

Description

Tests to see if two objects of class psData are equal. That is their type is the same, and the data contained in data is the same. See readData for a description of the psData class.

Usage

```r
## S3 method for class 'psData'
1hs == rhs
```

Arguments

- `1hs` an object of class psData.
- `rhs` an object of class psData.

Details

NOTE: the notes member variable is ignored in this function as it is unlikely that a user would want to see if the notes are the same.
as.data.frame.psData

Value

TRUE if the two objects are equal

Examples

```r
p = readData(system.file("extdata", "p.xlsx", package = "fitPS"))
p1 = makePSData(n = 0:2, count = c(98, 1, 1), type = "P")
p2 = makePSData(n = 0:2, count = c(97, 2, 1), type = "P")
p == p1 ## TRUE
p == p2 ## FALSE
p1 == p2 ## FALSE
```

as.data.frame.psData  Converts an object of class psData to a data.frame

Description

Converts an object of class psData—see readData—to a data.frame that can be used with in functions in other packages such as vglm to fit more complicated models.

Usage

```r
## S3 method for class 'psData'
as.data.frame(x, ...)
```

Arguments

x  an object of class psData—see readData for more details.

... any other arguments passed to data.frame.

Details

If x is a psData object of type "P", i.e. it relates to numbers of groups of glass, then a data.frame with a single variable count will be return where count = rep(x$data$n + 1, x$data$rn). The counts have one added to them because the Zeta distribution requires that the counts are greater than or equal to one. If x is a psData object of type "P", i.e. it relates to group sizes, then a data.frame with a single variable count will be return where count = rep(x$data$n, x$data$rn).

Value

a data.frame with a single variable count. The number of rows in the data.frame is equal to sum(x$data$rn).

Examples

```r
p = readData(system.file("extdata", "p.xlsx", package = "fitPS"))
p.df = as.data.frame(p)
table(p.df$count)
p$data
```
Bootstrap confidence intervals or regions

Description

Use bootstrapping to generate confidence intervals, or confidence regions in the case of the zero-inflated model.

Usage

```r
bootCI(x, ...)

## Default S3 method:
bootCI(
  x,
  level = 0.95,
  B = 2000,
  model = c("zeta", "zi.zeta"),
  returnBootValues = FALSE,
  silent = FALSE,
  plot = FALSE,
  parallel = TRUE,
  progressBar = FALSE,
  pbopts = list(type = "txt"),
  ...
)

## S3 method for class 'psData'
bootCI(x, ...)

## S3 method for class 'psFit'
bootCI(x, model = ifelse(!x$zeroInflated, "zeta", "zi.zeta"), ...)
```

Arguments

- `x` a object of class psFit—see `readData` for more details.
- `...` other arguments.
- `level` the confidence level required—restricted to \([0.75, 1)\). This may be a vector, in which case multiple intervals, or confidence regions will be returned.
- `B` the number of bootstrap samples to take.
- `model` which model to fit to the data, either "zeta" or "zi.zeta". Maybe abbreviated to "z" and "zi". Default is "zeta".
- `returnBootValues` if TRUE then the vector (or data.frame) of bootstrapped values is returned. This can be useful for debugging or understanding the results. Default is FALSE.
silent if TRUE, then no output will be displayed whilst the bootstrapping is being undertaken. plot if TRUE then the contours for the confidence region will be plotted. This only works if model = "zi.zeta". It is ignored otherwise. parallel if TRUE then the bootstrapping is performed in parallel.

plot if TRUE and model == "zi.zeta", then a plot of the bootstrapped values will be produced and confidence contour lines will be drawn for each value in level.

parallel if TRUE, then the package will attempt to use multiple cores to speed up computation.

progressBar if TRUE, then progress bars will be displayed to show progress on the bootstrapping.

pbopts a list of arguments for the pboptions function that affect the progress bars. Ignored if progressBar = FALSE.

Details

This function uses bootstrapping to compute a confidence interval for the shape parameter in the case of the zeta model and a confidence region in the case of the zero-inflated zeta model. A smoothed bootstrap approach is taken rather than a simple percentile method. The kernel density estimation is performed by the ks package using a smoothed cross-validated bandwidth selection procedure.

Value

If returnBootVals == TRUE then the results are returned in a list with elements named ci and bootVals for the zeta model and confRegion and bootVals for the zero-inflated zeta model. The structure of ci and confregion is described below. If model == "zeta", then either a vector or a data.frame with elements/columns named "lower" and "upper" representing the lower and upper bounds of the confidence interval(s). Multiple bounds are returned in a data.frame when level has more than one value. If model == "zi.zeta", then a list with length equal to the length of level is returned. The name of each element in the list is the level with list has a single element named "95%". It is possible for there to be multiple contours for the confidence region for a given level. If there is only one contour for each value of level, then each element of the list consists of a list with elements named pi and shape which specify the coordinates of the contour(s) for that level. There is a third element named level which gives the height of the kernel density estimate at that contour. If there are multiple contours for a given value of level then each list element is a list of lists with the structure given above (level, pi, and shape). NOTE: it is quite possible that there are multiple contours for a given height. If you want a way of thinking about this consider a mountain range with two mountains of equal height. If you draw the contours for (almost) any elevation, then you would expect to capture a region from each mountain.

Methods (by class)

• bootCI(default): Bootstrap confidence intervals or regions
• bootCI(psData): Bootstrap confidence intervals or regions
• bootCI(psFit): Bootstrap confidence intervals or regions
Examples

```r
## Not run:
data(Psurveys)
roux = Psurveys$roux
confRegion = bootCI(roux, model = "zi.zeta", parallel = FALSE, plot = TRUE)

## This will not work unless you have the sp package installed
## Count how many of the points lie within the 95% confidence region
lapply(confRegion, function(cr){
  table(sp::point.in.polygon(fit$pi, fit$shape, cr$pi, cr$shape))
})
## End(Not run)
```

confint.psFit    

S3 confint method for objects of class psFit

Description

S3 confint method for objects of class psFit

Usage

```r
## S3 method for class 'psFit'
confint(object, parm, level = 0.95, ...)
```

Arguments

- `object`: an object of class psFit—see fitDist for more details
- `parm`: added for compatibility. Should be left empty as it is ignored.
- `level`: the confidence level required—restricted to [0.75, 1)
- `...`: in theory other parameters to be passed to confint, but in reality passed as extra parameters to the internal function plZIZ.

Details

NOTE: the method for ZIZ model is a little computationally intensive and possibly (almost certainly) unstable.

Value

if the Zeta model is used (i.e. object comes from a call to fitDist), then a list with two items: `wald` and `prof` containing the Wald and profile likelihood confidence intervals respectively for the shape parameter of the fitted Zeta distribution is returned. In general these should be relatively close to each other. **NOTE** These values are for the VGAM parameterisation of the Zeta distribution which uses $s' = s - 1$. This means they can be used without alteration in dzeta. If a Zero-Inflated Zeta model is used (i.e. object comes from a call to fitZIDist) then list of a confidence regions is returned with an element for each value of level. The confidence regions are data.frames with variables `pi` and `shape` which can be used with lines or polygon to draw the confidence region.
Examples

```r
data(Psurveys)
roux = Psurveys$roux
fit = fitDist(roux)
confint(fit)

## Not run:
fit.zi = fitZIDist(roux)
cr = confint(fit.zi, level = c(0.80, 0.95))
plot(cr[["0.95"]], type = "l")
polygon(cr[["0.8"]])
## End(Not run)
```

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

```r
fitDist(x, nterms = 10, start = 1, ...)
fitdist(x, nterms = 10, start = 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 `optim`,
- `shape` – the maximum likelihood estimate of the shape parameter,
- `var.shape` - the maximum likelihood estimate of the shape parameter,
- `fitted` - a named vector containing the first n terms of the fitted distribution.
- `zeroInflated` - set to `FALSE` for this model

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`. **NOTE** The value of the shape parameter that is printed (if you print the fitted object) is different from that value that is stored in shape. The stored value is for the VGAM parameterisation of the Zeta distribution which uses \( s' = s - 1 \). Therefore the printed value is \( s = s' + 1 \). If you intend to use the fitted value with `dzeta`, then you should use the stored value \( s' \).

If `start` is not specified, then it is chosen randomly from (0.5, 1). The reason the lower value is not zero is that small starting values seem to cause instability in the likelihood. If you specify your own starting value, it would be sensible to keep it above 0.5.

Value

an object of class `psFit` – see Details.

Functions

- `fitdist()`: Fit a Zeta Distribution to Forensic Data export

References


See Also

`plot.psFit`, `print.psFit`, `probfun`.

Examples

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

S3 fitted method for an object of class psFit

Description

S3 fitted method for an object of class psFit

Usage

## S3 method for class 'psData'
fitted(object, ...)

Arguments

object an object of class psFit, usually from fitDist or fitZIDist.
... other arguments passed to fitted—not used.

Value

a named vector of fitted probabilities

fitZIDist

Fit a Zero-Inflated Zeta Distribution to Forensic Data

Description

This function uses maximum likelihood estimation (MLE) to estimate mixing parameter and the shape parameter of a Zero-Inflated 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 Zero-Inflated Zeta distribution has probability mass function

\[ p(k) = \begin{cases} 
\pi + \frac{(1-\pi)}{\zeta(s)}, & k = 0, \\
\frac{(1-\pi)^{k-s}}{\zeta(s)}, & k = 1, 2, \ldots
\end{cases} \]

where \( \zeta(s) \) is the Reimann Zeta function.

Usage

fitZIDist(x, nterms = 10, start = c(0.5, 1), ...)

fitZIDist(x, nterms = 10, start = c(0.5, 1), ...)

fitzidist(x, nterms = 10, start = c(0.5, 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 seven elements:

- **psData** – an object of class psData—see `readData`,
- **fit** – the fitted object from `optim`,
- **pi** - the maximum likelihood estimate of the mixing parameter,
- **shape** – the maximum likelihood estimate of the shape parameter,
- **var.cov** – the estimated variance-covariance matrix for the parameters,
- **fitted** – a named vector containing the first `nterms` of the fitted distribution.
- **zeroInflated** – set to TRUE for this model.

The output can be used in a variety of ways. If the interest is just in the mixing and shape parameter estimates, then the `pi` and `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`. **NOTE** The value of the shape parameter that is printed (if you print the fitted object) is different from that value that is stored in `shape`. The stored value is for the VGAM parameterisation of the Zeta distribution which uses $s' = s - 1$. Therefore the printed value is $s = s' + 1$. If you intend to use the fitted value with `dzeta`, then you should use the stored value $s'$.

If `start` is not specified, then it is set to (0.5, 1). The reason the starting values are not zero is that small starting values seem to cause instability in the likelihood. If you specify your own starting value, it would be sensible to keep both above 0.5.

Value

an object of class `psFit`—see Details.

References


See Also

`plot.psFit`, `print.psFit`, `probfun`. 
**makePSData**

Create a survey data set manually

**Description**

Create a survey data set from the command line rather than reading data in from a file. This function is likely to be only useful where there are a very small number of group sizes, or sizes of groups of glass.

**Usage**

```r
makePSData(n, count = NULL, type = c("P", "S"), notes = NULL)
makeData(n, count = NULL, type = c("P", "S"), notes = NULL)
createPSData(n, count = NULL, type = c("P", "S"), notes = NULL)
```

**Arguments**

- `n` Either the number of groups of glass or the size of different groups of glass, or a vector of observed groups of glass, or group sizes. See details for a longer explanation.
- `count` Either the number of people in the survey sample who had `n` groups of glass on their clothing, or the number of people who had a group of glass of size `n`.
- `type` either "P" or "S"
- `notes` a `bibentry` or a character string which allows extra information about the data to be stored, such as the source, or reference. NULL by default.

**Details**

If `count` is NULL, then it is assumed that `n` consists of actual observed group sizes or numbers of groups of glass found on a survey of `N` individuals. That is, one could provide `n = rep(0:1, 98, 1)` or `n = 0:1, count = c(98, 1)`. The former is more useful when performing simulation studies.

**Value**

A `psData` object—see `readData` for more details.

**See Also**

`readData`
Examples

```r
## recreate the data read in the readData example
p1 = makePSData(n = c(0, 1, 2), count = c(98, 1, 1), type = "P")
s1 = makePSData(n = 1:3, count = c(1, 1, 1), type = "S")
p1
s1
```

---

**mean.psData**

An S3 method for computing the mean of clothing survey for the number of groups or size of groups

---

**Description**

An S3 method for computing the mean of clothing survey for the number of groups or size of groups

**Usage**

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

**Arguments**

- `x` an object of class `psData`—`readData` for more details.
- `...` other arguments which are passed to `sum`

**Value**

the mean of the data. If there are \( r_i \) observations of the value \( n_i \) then the mean is given by

\[
\sum_i \frac{r_i \times n_i}{\sum_i r_i}
\]

**Examples**

```r
data(Psurveys)
mean(Psurveys$roux)
```
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,
  ci.type = c("wald", "prof"),
  log.scale = FALSE,
  ...
)

Arguments

x

an object of class psFit, usually from fitDist or fitZIDist.

ylim

the limits of the y-axis.

conf

if TRUE, and the model is the the Zeta model (as opposed to the Zero-Inflated Zeta (ZIZ), then confidence intervals (based on the standard error of the shape parameter) are drawn on the plot. If the ZIZ model has been used, then this is ignored.

conf.level

the confidence level for the confidence intervals. Must be between 0.75 and 0.99.

ci.type

Specifies the type of confidence interval. If conf == TRUE, then then ci.type can be either "wald" "prof" (or an abbreviation), depending on whether the Wald interval or the profile likelihood interval should be used. Note that these are intervals on the shape parameter and not the density heights. Therefore the intervals around the probabilities should not really be thought of as confidence intervals but rather something more similar to a "sensitivity" interval.

log.scale

if TRUE the y-axis is changed to a logarithmic (base 10) axis.

...

other arguments passed to plot.

Value

No return value, called for side effects
### Examples

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

## An example with Wald generated intervals
plot(fit, conf = TRUE)
plot(fit, conf = TRUE, ci.type = "p")
```

### predict.psFit

S3 predict method for an object of class `psFit`

#### Description

S3 predict method for an object of class `psFit`

#### Usage

```r
## S3 method for class 'psFit'
predict(
  object, n = 1, newdata, interval, level = 0.95, 
  ...)
```

#### Arguments

- **object**: an object of class `psFit`, usually from `fitDist` or `fitZIDist`.
- **newdata**: an optional vector of integers at which to calculate \( \Pr(X = x) \).
- **interval**: either "none", "prof", or "wald" and can be abbreviated. If "prof" or "wald" AND the Zeta model has been used then an interval, based on the bounds of a 100 * level confidence interval for the shape parameter, is given for each predicted probability. The interval is provided based on either a Profile Likelihood, or a Wald, confidence interval for the shape, and therefore cannot really be regarded as a confidence interval for the probabilities. The intervals might be more sensibly regarded as a measure of how sensitive the probabilities are to the choice of shape parameter. NOTE: this parameter is ignored if the Zero-inflated (ZIZ) model has been used.
- **level**: the level of a confidence interval. Ignored if interval == "none".
- **...**: other arguments passed to `predict`—not used

#### Value

either a named vector of fitted probabilities, or a data.frame with columns predicted, lower, and upper and the row names set to show what terms are being calculated
Examples

data(Psurveys)
roux = Psurveys$roux
fit = fitDist(roux)
predict(fit, interval = "prof")

print.psData  

S3 print method for an object of class psData

Description
S3 print method for an object of class psData

Usage

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

Arguments

x an object of class psData, usually from readData or makePSData
...
other arguments passed to print

Value
No return value, called for side effects

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 or fitZIDist.
...
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**

```r
probfun(psFitobj)
```

**Arguments**

- `psFitobj` an object of class `psFit`—see `fitDist` and `fitZIDist`.

**Value**

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

**Examples**

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

---

**Psurveys**  
*Number of Groups of Glass Data*

**Description**

Count data from five different surveys looking at the number of sources/groups of glass found on the upper surfaces of clothing taken from the general public.

**Usage**

```r
data(Psurveys)
```

**Format**

A list with six objects of class `psData`—see `readData` for more details. The elements of the list are named: `coulson`, `jackson`, `lau`, `pettard`, `ross`, and `roux`, corresponding to the lead author in each of the references given below. `lau`, `pettard`, and `ross` were taken from Coulson et al. (2001) rather than the original source.
Source


References


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, notes = NULL, ...)
Arguments

fileName  
the name of the file to be read. Must be either a modern (xlsx) Excel file or a csv file.

notes  
any additional information about the data, such as the source or a reference.

...  
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 variables:

• type – either “P” or “S”
• data – a data.frame which contains columns n and rn, representing the number of groups/fragments, and the number of times that was seen, respectively.
• notes — either a bibentry or a character string which allows extra information about the data to be stored, such as the source, or reference.

Examples

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

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

rZIzeta

Generate zero inflated zeta random variates

Description

Generate zero inflated zeta random variates

Usage

rZIzeta(n, pi = 0.5, shape = 1, offset = 0)

rzizeta(n, pi = 0.5, shape = 1, offset = 0)

rzizeta(n, pi = 0.5, shape = 1, offset = 0)

Arguments

n  
the number of observations.

pi  
the mixing parameter for the zero-inflated zeta model—must be in (0, 1).

shape  
the shape parameter for the zero-inflated zeta. Must be greater than zero.

offset  
the zeta distribution returns random variates that are greater than, or equal to one. If the offset is greater than 0, then the distribution is anchored on (has minimum value of) \(1 - \text{offset}\).
Details

Technically this function returns values from the one-inflated zeta distribution. However, if offset is greater than zero (and typically we expect it to be 1), then the minimum random variate value is \(1 - \text{offset}\). We chose the name "zero-inflated zeta" as more people are familiar with zero-inflated models.

Value

a vector of random variates from a zero-inflated zeta model

Examples

data(Psurveys)
roux = Psurveys$roux
fit.zi = fitZIDist(roux)
x = rZIzeta(n = sum(roux$data$rn), pi = fit.zi$pi, shape = fit.zi$shape)
table(x)

Ssurveys  Size of Groups of Glass Data

Description

Count data from six different surveys looking at the number of sources/groups of glass found on the upper surfaces of clothing taken from the general public.

Usage

data(Psurveys)

Format

A list with five objects of class psData—see readData for more details. The elements of the list are named: jackson, lau, pettard, ross, and roux, corresponding to the lead author in each of the references given below. lau, pettard, and ross were taken from Coulson et al. (2001) rather than the original source.

Source

References


---

table

| var | Variance generic |

**Description**

Variance generic

**Usage**

`var(x, ...)`

**Arguments**

- `x` : an object for which we want to compute the sample variance.
- `...` : Any additional arguments to be passed to `var`. 
var.psData

An S3 method for computing the variance of clothing survey for the number of groups or size of groups

Description

An S3 method for computing the variance of clothing survey for the number of groups or size of groups

Usage

## S3 method for class 'psData'
var(x, ...)

Arguments

x an object of class psData—readData for more details.

... other arguments which are passed to sum

Value

the mean of the data. If there are $r_i$ observations of the value $n_i$ then the variance is computed by $E[X^2] - E[X]^2$, where $E[X]$ is computed using

$$\sum_i r_i \times n_i$$

$\sum_i r_i$, and $E[X^2]$ is computed by

$$\sum_i r_i \times n_i^2$$

$\sum_i r_i$. We realise that the computational formula, $E[X^2] - E[X]^2$, is usually not regarded as computationally stable, but the magnitude of the numbers involved is such that, that this is not likely to cause an issue.

Examples

data(Psurveys)
var(Psurveys$roux)
Index

* datasets
  Psurveys, 16
  Ssurveys, 19
  ==.psData, 2
  as.data.frame.psData, 3
  bibentry, 11, 18
  bootCI, 4
  confint.psFit, 6
  createPSData(makePSData), 11
  dzeta, 6, 8, 10
  fitDist, 6, 7, 9, 13–16
  fitdist(fitDist), 7
  fitted.psData, 9
  fitZIDist, 6, 9, 9, 13–16
  fitZIdist(fitZIDist), 9
  fitzidist(fitZIDist), 9
  lines, 6
  makeData(makePSData), 11
  makePSData, 11, 15
  mean.psData, 12
  optim, 8, 10
  pboptions, 5
  plot.psFit, 8, 10, 13
  polygon, 6
  predict.psFit, 14
  print.psData, 15
  print.psFit, 8, 10, 15
  probfun, 8, 10, 16
  Psurveys, 16
  readData, 2–4, 7, 8, 10–12, 15, 16, 17, 19, 21
  rzizeta, 18
  rzizeta(rZIzeta), 18
  Ssurveys, 19
  sum, 12, 21
  var, 20
  var.psData, 21
  vglm, 3