Package ‘qra’

October 29, 2021

Type Package

Title Quantal Response Analysis for Dose-Mortality Data

Date 2021-10-29

Version 0.2.7

Maintainer John Maindonald <john@statsresearch.co.nz>

Description Functions are provided that implement the use of the Fieller's formula methodology, for calculating a confidence interval for a ratio of (commonly, correlated) means. See Fieller (1954) <doi:10.1111/j.2517-6161.1954.tb00159.x>. Here, the application of primary interest is to studies of insect mortality response to increasing doses of a fumigant, or, e.g., to time in coolstorage. The formula is used to calculate a confidence interval for the dose or time required to achieve a specified mortality proportion, commonly 0.5 or 0.99. Vignettes demonstrate link functions that may be considered, checks on fitted models, and alternative choices of error family. Note in particular the betabinomial error family. See also Maindonald, Waddell, and Petry (2001) <doi:10.1016/S0925-5214(01)00082-5>.

Encoding UTF-8

License GPL-3

Depends R (>= 4.1.0), lattice, latticeExtra, knitr, rmarkdown

Imports lme4, splines, ggplot2

Suggests fitODBOD, VGAM, glmTMB (>= 1.1.2), gamlss, prettydoc, DHARMa, kableExtra (>= 1.2), plotrix, dfoptim, optimx, bookdown

URL https://github.com/jhmaindonald/qra

BugReports https://github.com/jhmaindonald/qra/issues

VignetteBuilder knitr, rmarkdown, bookdown, prettydoc
Description

The values returned are those used for plot(x.lm, which=3), where x.lm is a linear model or a generalized linear model. Plot the object returned to assess how successful the weights, determined using the function `scaleLocAdjust`, have been in adjusting for heterogenous variances.

Usage

```r
checkDisp(x, span = 0.75)
```

Arguments

- `x`: Model fitted using `lm()` or `glm()`
- `span`: span parameter for use in smoothing the square root of standardized deviance residuals.
**Value**

A data frame, with:

- `linpred` Predicted values, on the scale of the linear predictor
- `absrSmooth` Smoothed values of the square roots of absolute values of standardised deviance residuals.

**Examples**

```r
royal <- subset(qra::codling1988, Cultivar=="ROYAL")
royal(glm <- glm(cbind(dead,total-dead)~ct, data=royal,
              family=quasibinomial(link='cloglog'))
royalFix <- qra::scaleLocAdjust(royal glm, lambda=2)
## Check range of indicated prior weights
range(royalFix[[2]])
## Range of updated dispersion estimates
range(summary(royalFix[[1]])[['dispersion']]/royalFix[[2]])
xy <- qra::checkDisp(royalFix[[1]])
plot(xy)
```

**Description**

Data are from trials that studied the mortality response of codling moth to fumigation with methyl bromide, for the year 1988 only.

**Usage**

```r
data(codling1988)
data(codling1989)
```

**Format**

A data frame with 77 observations (codling1988), and with 40 observations (codling1989), on the following 8 variables.

- `dose` Injected dose of methyl bromide, in gm per cubic meter
- `ct` Concentration-time sum
- `total` Number of insects in chamber
- `dead` Number of insects dying
- `PropDead` Proportion dying
- `Cultivar` a factor with 1988 levels BRAEBURN FUJI GRANNY Red Delicious and ROYAL; and with 1989 levels Gala, Red Delicious and Splendour
- `rep` replicate number, within Cultivar
- `cultRep` Cultivar/replicate combination
Details

The research that generated these data was in part funded by New Zealand pipfruit growers. The published analysis was funded by New Zealand pipfruit growers. See also DAAG::sorption.

Source


extractLT

Obtain complete set of LT or LD estimates

Description

When supplied with a model object that has fitted dose-response lines for each of several levels of a factor, extractLT calls the function fieller to calculate lethal time

Usage

```r
extractLT(
    obj,
    a = 1:3,
    b = 4:6,
    link = NULL,
    logscale = FALSE,
    p = 0.99,
    eps = 0,
    offset = 0,
    df.t = NULL
)
```

```r
extractLTpwr(
    obj,
    a = 1:3,
    b = 1:3,
    link = "fpower",
    logscale = FALSE,
    p = 0.99,
    lambda = 0,
    eps = 0.015,
    offset = 0,
    df.t = NULL
)
```
Arguments

- `obj`: merMod object, created using `lmer()` or glmerMod object, created using `glmer()`.
- `a`: Subscripts for intercepts.
- `b`: Subscripts for corresponding slopes.
- `link`: Link function, for use with objects where no link was specified in the function call, but it is required to back-transform a transformation that was performed prior to the function call. Otherwise leave as `link=NULL`, and the link function will be extracted as `family(obj)[["link"]]. For a folded power function, with `extractLTPwr()`, the only available link is `fpower`, and the exponent `lambda` must be specified.
- `logscale`: Logical. Specify `TRUE`, if LT values are to be back-transformed from a logarithmic scale.
- `p`: Target response proportion.
- `eps`: Replace `prob` by `prob+eps` before transformation.
- `offset`: Use to undo scaling of time or dose variable. This is passed to the `fieller` function that `extractLT` calls.
- `df.t`: Degrees of freedom for a t-distribution approximation for ‘t’ or ‘z’ statistics. If `NULL`, a conservative (low) value will be used. For linear (but not generalized linear) models and mixed models, approximations are implemented in the `afex` package. See vignette('introduction-mixed-models',package="afex"), page 19.
- `lambda`: Power for power function.

Details

Fixed coefficients from `obj` must be for intercepts and for slopes. Starting the model formula with `0+` will commonly do what is required. The coefficients `fixef(obj)[a]` are assumed to specify line intercepts, while `fixef(obj)[b]` specify the corresponding slopes. These replace the arguments `nEsts` (subscripts for intercepts were `1:nEsts`) and `slopeAdd` (subscripts for slopes were `(nEsts+1):(nEsts+slopeAdd)`).

Value

Matrix holding LD or LD estimates.

Examples

```r
pcheck <- suppressWarnings(requireNamespace("glmmTMB", quietly = TRUE))
if(pcheck) pcheck & packageVersion("glmmTMB") >= "1.1.2"
if(pcheck){
  form <- cbind(Dead,Live)~0+trtGp/TrtTime+(1|trtGpRep)
  HawMed <- droplevels(subset(HawCon, CN=="MedFly"&LifestageTrt!="Egg"))
  HawMed <- within(HawMed,
  {trtGp <- factor(paste0(CN,LifestageTrt, sep=":"))
   trtGpRep <- paste0(CN,LifestageTrt,"",RepNumber)
   scTime <- scale(TrtTime) })
  HawMedbb.cll <- glmmTMB::glmmTMB(form, dispformula=~trtGp+splines::ns(scTime,2),
```

```r
```
Confidence Limits for Lethal Dose Estimate From Dose-response Line

Description

This uses Fieller's formula to calculate a confidence interval for a specified mortality proportion, commonly 0.50, or 0.90, or 0.99. Here "dose" is a generic term for any measure of intensity of a treatment that is designed to induce insect death.

Usage

fieller(
  phat,
  b,
  vv,
  df.t = Inf,
  offset = 0,
  logscale = FALSE,
  link = "logit",
  eps = 0,
  type = c("Fieller", "Delta"),
  maxg = 0.99
)

fieller2(
  phat,
  b,
  vv,
  df.t = Inf,
  offset = 0,
  logscale = FALSE,
  link = "fpower",
  lambda = 0,
  eps = 0,
  type = c("Fieller", "Delta"),
  maxg = 0.99
)
fieller

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>phat</td>
<td>Mortality proportion</td>
</tr>
<tr>
<td>b</td>
<td>Length 2 vector of intercept and slope</td>
</tr>
<tr>
<td>vv</td>
<td>Variance-covariance matrix for intercept and slope</td>
</tr>
<tr>
<td>df.t</td>
<td>Degrees of freedom for variance-covariance matrix</td>
</tr>
<tr>
<td>offset</td>
<td>Offset to be added to intercept. This can be of length 2, in order to return values on the original scale, in the case where b and vv are for values that have been scaled by subtracting offset[1] and dividing by offset[2].</td>
</tr>
<tr>
<td>logscale</td>
<td>Should confidence limits be back transformed from log scale?</td>
</tr>
<tr>
<td>link</td>
<td>Link function that transforms expected mortalities to the scale of the linear predictor</td>
</tr>
<tr>
<td>eps</td>
<td>If eps&gt;0 phat is replaced by ( \frac{p+c}{1+2c} ) before applying the transformation.</td>
</tr>
<tr>
<td>type</td>
<td>The default is to use Fieller's formula. The Delta (type=&quot;Delta&quot;) method, which relies on a first order Taylor series approximation to the variance, is provided so that it can be used for comparative purposes. It can be reliably used only in cases where the interval has been shown to be essentially the same as given by type=&quot;Fieller&quot;!</td>
</tr>
<tr>
<td>maxg</td>
<td>Maximum value of g for which a confidence interval will be calculated. Must be &lt; 1.</td>
</tr>
<tr>
<td>lambda</td>
<td>The power ( \lambda ), when using the link=&quot;fpower&quot;. (This applies to fieller2 only.)</td>
</tr>
</tbody>
</table>

Details

See the internal code for details of the value g. The calculation gives increasing wide confidence intervals as g approaches 1. If \( g \geq 1 \), there are no limits. The default value for df.t is a rough guess at what might be reasonable. For models fitted using lme4::lmer(), abilities in the lmerTest package can be used to determine a suitable degrees of freedom approximation — this does not extend to use with glmer() or glmmTMB.

Value

A vector, with elements

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>est</td>
<td>Estimate</td>
</tr>
<tr>
<td>var</td>
<td>Variance, calculated using the Delta method</td>
</tr>
<tr>
<td>lwr</td>
<td>Lower bound of confidence interval</td>
</tr>
<tr>
<td>upr</td>
<td>upper bound of confidence interval</td>
</tr>
<tr>
<td>g</td>
<td>If g is close to 0 (perhaps ( g &lt; 0.05 )), confidence intervals will be similar to those calculated using the Delta method, and the variance can reasonably be used for normal theory inference.</td>
</tr>
</tbody>
</table>
References


See Also

varRatio

Examples

redDel <- subset(qra::codling1988, Cultivar=="Red Delicious")
redDel.glm <- glm(cbind(dead,total-dead)~ct, data=redDel,
  family=quasibinomial(link='cloglog'))
vv <- summary(redDel.glm)$cov.scaled
fieller(0.99, b=coef(redDel.glm), vv=vv, link='cloglog')

foldp

Title Function to calculate ratio of p+eps to 1-p+eps.

Description

This is a convenience function that returns \( \frac{p+\epsilon}{1-p+\epsilon} \). It calculates the argument that is supplied to the log function in Tukey’s ‘flog’.

Usage

foldp(p, eps)

Arguments

p Proportion
eps Offset. The choice eps=0.01 has the same effect as replacing \( \frac{r}{n-r} \) by \( \frac{r+0.5}{n-r+0.5} \) when \( n = 50 \), or by \( \frac{r+1}{n-r+1} \) when \( n = 100 \)

Value

\( \frac{(p+\epsilon)}{(1-p+\epsilon)} \)

Examples

foldp(c(0.2,0.75), 0)
fpower

Folded Power Transformation

Description

The name “folded Power Transformation” is used because this does for power transformations what Tukey’s folded logarithm does for the logarithmic transformation. The function calculates

\[ f(p, \lambda, \epsilon) = \frac{p + \epsilon}{1 - p + \epsilon}^\lambda \]

where \( \lambda \) is the power and \( \epsilon \) is a positive offset that ensures that \( \frac{p + \epsilon}{1 - p + \epsilon} \) is greater than 0 and finite.

Usage

fpower(p, lambda, eps)

Arguments

- \( p \) Mortality proportion
- lambda Power lambda for the power transformation
- eps If eps>0 phat is replaced by \( \frac{p + \epsilon}{1 + \epsilon} \) before applying the power transformation.

Value

The transformed values of fpower(p).

Examples

```r
p <- (0:10)/10
ytrans <- fpower(p, lambda=0.25, eps=1/450)
```

getRho

Extract estimates of the intra-class correlation from a glmmTMB model object with beta-binomial error.

Description

The intra-class correlation is calculated as \((1 + \exp(\theta))^{-1}\), where \( \theta \) is the estimate given by the formula specified in the argument dispformula.

Usage

getRho(obj, varMult = FALSE)
**getScaleCoef**

Extract scaling coefficients from vector returned by `scale()`

**Description**

The function `scale()` replaces \( x \) by \( (x-a)/b \), where \( a \) is mean(\( x \)) and \( b \) is sd(\( x \)). The quantities \( a \) and \( b \) are available as attributes of the object that is returned.

**Usage**

```r
getScaleCoef(z)
```

**Arguments**

- \( z \) Object returned by `scale()`
Details

Use of a scaled explanatory variable can be helpful in getting a model to fit. The scaling coefficient(s) will then be needed when the fitted model is used with explanatory variable values on the original scale.

Value

A vector, whose elements are the scaling coefficients $a$ and $b$, or if `scale=FALSE` then $a$.

Examples

```r
z <- scale(1:10)
qra::getScaleCoef(z)
```

---

**gpsWithin**

*Use given vector to identify groups with specified categories*

Description

Any one-dimensional object whose values distinguish groups may be supplied.

Usage

```r
gpsWithin(x, f)
```

Arguments

- `x`: One-dimensional object whose values distinguish groups
- `f`: One-dimensional object or list of objects, the combinations of whose values specify categories within which groups are to be defined.

Value

Integer vector whose values, within each specified category, run from 1 to the number of groups

Examples

```r
repnum <- with(qra::codling1988, gpsWithin(cultRep, Cultivar))
table(codling1988$Cultivar, repnum)
```
**graphSum**

*Draw graphs of insect mortality or other exposure-response data*

**Description**

Datasets that are in mind hold, for each replicate of each combination of each of a several factors (e.g., species, lifestages, temperatures), mortalities for each of a number of values of "dose". See for example the dataset help page `codling1989`.

**Usage**

```r
graphSum(
  df,
  subSet = NULL,
  link = "cloglog",
  logScale = FALSE,
  dead = "Dead",
  tot = "Tot",
  dosevar = "logCT",
  Rep = "Rep",
  fitRep = NULL,
  fitPanel = NULL,
  byFacet = ~Species,
  layout = NULL,
  maint = "Codling Moth, MeBr",
  ptSize = 2,
  xzeroOffsetFrac = 0.08,
  yzeroOneOffsets = c(-0.08, 0.08),
  yEps = 0.005,
  xlab = expression(bold("CT ") * "(gm.h." * m^{ -3 } * ")")
)
```

**Arguments**

- `df` Data frame from which data will be taken
- `subSet` NULL, or an expression, such as for example `expression(LifeStage=="Eggs")` that evaluates to a logical that specifies the required data subset. If not NULL then the subsetting information is pasted on after the main title
- `link` Link function. If character, obtain from `make.link`. Alternatively, a function may be supplied as argument.
- `logScale` Logical, indicating whether the dose ($x$-variable) is on a log scale.
- `dead` Character; name of column holding number dead
- `tot` Character; column holding total number
**HawCon**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dosevar</td>
<td>Character; column holding &quot;dose&quot; values</td>
</tr>
<tr>
<td>Rep</td>
<td>Character; NULL, or column holding replicate number, within panel</td>
</tr>
<tr>
<td>fitRep</td>
<td>Character; NULL, or column holding replicate fitted values</td>
</tr>
<tr>
<td>fitPanel</td>
<td>Character; NULL, or column holding panel fitted values</td>
</tr>
<tr>
<td>byFacet</td>
<td>Graphics formula specifying factor combination that determines panel layout</td>
</tr>
<tr>
<td>layout</td>
<td>Graphics formula that can be supplied to <code>grid_facet</code></td>
</tr>
<tr>
<td>maint</td>
<td>Main title</td>
</tr>
<tr>
<td>ptSize</td>
<td>Pointsize, by default 2</td>
</tr>
<tr>
<td>xzeroOffsetFrac</td>
<td>$x$-axis zero offset fraction, required when scale is logarithmic</td>
</tr>
<tr>
<td>yzeroOneOffsets</td>
<td>Length two vector, giving 0 100 mortalities, on the scale of the link function.</td>
</tr>
<tr>
<td>yEps</td>
<td>Fractional increase at bottom and top of $y$ user range to accommodate points for mortalities of 0 and 1.</td>
</tr>
<tr>
<td>xlab</td>
<td>Expression specifying x-axis label</td>
</tr>
<tr>
<td>ylabel</td>
<td>If not NULL, $y$-axis label</td>
</tr>
<tr>
<td>ytklab</td>
<td>Place $y$ axis tiks and labels at these mortalities</td>
</tr>
</tbody>
</table>

**Value**

No return value, called for side effects

---

### Description

The counts of live/dead were derived by injecting a known number of individuals of the target life stage into citrus fruits, subjecting them to treatment and then counting the number of individuals emerging.

### Usage

```r
data("HawCon")
```

### Format

A data frame with 106 observations on the following 10 variables.

**Species**  Species of fruitfly

**CN** Common name, in abbreviated form. MedFly is ‘Mediterranean Fruit Fly’. MelonFly is ‘Melon Fly’

**LifestageTrt**  Lifestage treated
RepNumber Replicate number
PropDead Fraction dead
TrtTime Treatment time (days)
Dead a numeric vector
Live a numeric vector
Total a numeric vector

Details

The help page for HawCon in the ColdData has further details.

Source

Dr Peter Follett

References

A paper is in the course of preparation.

Examples

data(HawCon)
str(HawCon)

kerrich

Kerrich Coin Toss Trial Outcomes

Description

A data set containing 2,000 trials of coin flips from statistician John Edmund Kerrich’s 1940s experiments while imprisoned by the Nazis during World War Two.

Usage

data("kerrich")

Format

The format is: List of 1 $ : chr [1:2000] "0" "0" "0" "1" ...

Source


References

malesINfirst12

Examples

```r
data(kerrich)
```

---

**malesINfirst12**  Number of males among first 12 in families of 13 children

**Description**

The number of male children among the first 12 children of family size 13 in 6115 families taken from the hospital records in the nineteenth century Saxony (Lindsey (1995), p.59). The thirteenth child is ignored to assuage the effect of families non-randomly stopping when a desired gender is reached.

**Usage**

```r
data("malesINfirst12")
```

**Format**

A data frame with 13 observations on the following 2 variables.

- **No_of_Males**  a numeric vector
- **freq**  a numeric vector

**Details**

Data are available in the fitODBOD package.

**Source**

fitODBOD package

**References**


**Examples**

```r
data(malesINfirst12)
boxplot(freq ~ No_of_Males, data=malesINfirst12)
```
rayBlight  

*Incidence of ray blight disease of pyrethrum*

**Description**

An assessment of the incidence of ray blight disease of pyrethrum in 62 sampling units, containing 6 plants each.

**Usage**

```r
data("rayBlight")
```

**Format**

The format is: `int [1:62] 4 6 6 6 6 6 6 6 6 4 6 ...`

**Source**

`epiphy` package.

**References**


**Examples**

```r
data(rayBlight)
barplot(table(rayBlight))
```

---

scaleLocAdjust  

*Estimate dispersion as a function of predicted values*

**Description**

A loess smooth is applied to the square roots of the standardized deviance residuals. The inverses of values from the smooth, raised to the power of `lambda`, are then used as prior weights to update the model. A value of `lambda` that is a little more than 2.0 has often worked well.

**Usage**

```r
scaleLocAdjust(x, lambda = 2, span = 0.75)
```
Arguments

- **x**: Model fitted using `glm` or, possibly `lm`
- **lambda**: Power of smooth of square roots of absolute values of residuals, to try for values whose inverses will be used as weights
- **span**: span parameter for use in smoothing the square root of standardized deviance residuals.

Details

This function is primarily for experimental use, in investigating possible ways to model a dispersion factor that varies with the fitted value.

Value

A list, with elements

- **model**: Model updated to use the newly calculated weights
- **estDisp**: Estimated dispersions

Note

The dispersion estimates that correspond to the updated model are obtained by dividing the dispersion value given by `summary()` for the updated model by the (prior) weights supplied when the model was updated. The approach for obtaining varying dispersion estimates is used because, empirically, it has been found to work well for at least some sets of data. In particular, there seems no obvious theoretical basis for the choice of `lambda`. In the example given, used because the data is publicly available, the method has limited success.

See Also

- `checkDisp`

Examples

```r
ROYAL <- subset(qra::codling1988, Cultivar=="ROYAL")
ROYAL(glm(cbind(dead,total-dead)~ct, data=ROYAL,
       family=quasibinomial(link='cloglog')))
ROYALFix <- qra::scaleLocAdjust(ROYAL glm)
## Check range of indicated prior weights
range(ROYALFix[[2]])
## Range of updated dispersion estimates
range(summary(ROYALFix[[1]])[['dispersion']] / ROYALFix[[2]])
```
**varRatio**

First order approximation to variance of y-ordinate to slope ratio

**Description**

In contexts where an LD99 estimate will be used as a data value in a further analysis step, the inverse of the variance may be used as a weight. The y-ordinate is for the link function transformed value of a specified mortality proportion, commonly 0.50, or 0.90, or 0.99

**Usage**

`varRatio(phat = 0.99, b, vv, link = "cloglog")`

**Arguments**

- **phat** : Mortality proportion
- **b** : Length 2 vector of intercept and slope
- **vv** : Variance-covariance matrix for intercept and slope
- **link** : Link function that transforms expected mortalities to the scale of the linear predictor

**Details**

This function should only be used, in order to speed up calculations that use the function `fieller` (call `fieller` with `type="Delta"`), in a context where it is to be used many times, and where a check has been made that its use leads to confidence intervals that are a close approximation to those given with the default argument (`type="Fieller"`).

**Value**

A vector, with elements

- **xhat** : Estimate
- **var** : Variance, calculated using the Delta method. See the help page for `fieller` for further details and references.

**Examples**

```r
redDel <- subset(qra::codling1988, Cultivar=="Red Delicious")
redDel.glm <- glm(cbind(dead,total-dead)~ct, data=redDel, family=quasibinomial(link="cloglog"))
vv <- summary(redDel.glm)$cov.scaled
qra::varRatio(0.99, b=coef(redDel.glm), vv=vv, link="cloglog")
```
Index

* datasets
  - codling1988, 3
  - HawCon, 13
  - kerrich, 14
  - malesINfirst12, 15
  - rayBlight, 16

  checkDisp, 2, 17
  codling1988, 3
  codling1989, 12
  codling1989 (codling1988), 3

  extractLT, 4
  extractLTpwr (extractLT), 4

  fieller, 5, 6, 18
  fieller2 (fieller), 6
  foldp, 8
  fpower, 9

  getRho, 9
  getScaleCoef, 10
  gpsWithin, 11
  graphSum, 12

  HawCon, 13

  kerrich, 14

  make.link, 12
  malesINfirst12, 15

  rayBlight, 16

  scaleLocAdjust, 2, 16

  varRatio, 8, 18