Package 'LW1949'

November 12, 2014

Title An Automated Approach to Evaluating Dose-Effect Experiments Following Litchfield and Wilcoxon (1949)

Version 2014-11

Date 2014-11

Author Jean V. Adams [aut, cre]

Maintainer Jean V. Adams <jvadams@usgs.gov>

Depends R (>= 3.1.0)

Imports mgcv, MASS

Description LW1949 takes the manual approach to evaluating dose-effect experiments (Litchfield and Wilcoxon 1949) and automates the steps so that the computer can do the work.

License GPL

LazyData TRUE

URL https://github.com/JVAdams/LW1949

R topics documented:

`assessfit` *Assess Fit of Dose Response Curve*

Description

Assess the fit of a dose response curve using the chi-squared statistic. The curve is described by the intercept and slope of a straight line in the log dose vs. probit effect scale.

Usage

```
assessfit(params, DEdata, fit, simple = TRUE)
```

Arguments

`params`	A numeric vector of length two, with the estimated intercept and slope of the dose-effect relation on the log10 and probit scale. These parameters define the dose response curve.
`DEdata`	A data frame of dose-effect data (typically, the output from `dataprep` containing at least these four variables: dose, ntot, pfx, fxcateg.
`fit`	A model object that can be used to predict the corrected values (as proportions) from `distexpprop5`, the distance from the expected values (as proportions) and 0.5. Typically the output from `gamtable1()`.
`simple`	A logical scalar indicating if the output should be restricted just the P value, default TRUE.

Details

This function is used as part of a routine that attempts to find the dose response curve that minimizes the chi-squared statistic measuring the distance between the observed and expected values of the proportion affected. Following Litchfield and Wilcoxon (1949, steps B1 and B2), records for any 0% or 100% dose with expected values < 0.01% or > 99.99% are deleted, and expected values are corrected using the `correctval` function.

Value

If `simple=FALSE`, a list of length two. The first element, `chi`, is a numeric vector of length three: chistat, the chi-squared statistic; df, the degrees of freedom of `chistat`; and pval, their associated P value. The second element, `stepB`, is a matrix of three numeric vectors the same length as `obsn`, exp, expected effects; expcorr, expected effects corrected; and contrib, contributions to the chi-squared. If `simple=TRUE`, a numeric scalar, the chi-squared statistic (see details).

References

J. T. Litchfield, Jr. and F. Wilcoxon. 1949. A simplified method of evaluating dose-effect experiments[1]. Journal of Pharmacology and Experimental Therapeutics 99(2):99-113.

[1] `http://jpet.aspetjournals.org/content/96/2/99.short`

See Also

chi2 and chisq.test.

Examples

```
conc <- c(0.0625, 0.125, 0.25, 0.5, 1)
numtested <- rep(8, 5)
nalive <- c(1, 4, 4, 7, 8)
mydat <- dataprep(dose=conc, ntot=numtested, nfx=nalive)
gamfit <- gamtable1()
assessfit(log10(c(0.125, 0.5)), mydat, gamfit, simple=FALSE)
```

---

chi2 *Chi-Squared Statistic*

Description

Calculate the chi-squared statistic from observed and expected counts.

Usage

```
chi2(obsn, expn)
```

Arguments

obsn	A numeric vector of observed counts.
expn	A numeric vector of expected counts.

Value

A list of length two. The first element is a numeric vector of length three: chistat, the chi-squared statistic; df, the degrees of freedom of chistat; and pval, their associated P value. The second element is a numeric vector the same length as obsn, contributions to the chi-squared.

See Also

chisq.test.

Examples

```
chi2(c(10, 8, 3), c(7, 7, 7))
```

coefprobit *Calculate the Coefficients of a Probit Regression Fit*

Description

Calculate the coefficients from a fitted probit regression model.

Usage

```
coefprobit(pfit)
```

Arguments

pfit	An object of class glm representing a probit regression fit to dose-effect data, typically the result of a call to fitprobit.

Value

A numeric vector of length three, the intercept and slope of the dose-response curve, each with 95% confidence limits.

Examples

```
conc <- c(0.0625, 0.125, 0.25, 0.5, 1)
numtested <- rep(8, 5)
nalive <- c(1, 4, 4, 7, 8)
mydat <- dataprep(dose=conc, ntot=numtested, nfx=nalive)
myfit <- fitprobit(mydat)
coefprobit(myfit)
```

constrain *Constrain Data to a Specified Range*

Description

Constrain data to a specified range, assigning values from the specified range to those outside the range, typically for graphing purposes.

Usage

```
constrain(x, xrange)
```

Arguments

x	A numeric vector of values to constrain.
xrange	A numeric vector of length two specifying the constraints, the minimum and maximum value for x.

Value

A numeric vector, the same length as `x`, in which the minimum constraint is assigned to values of `x` less than the minimum, and the maximum constraint is assigned to values of `x` greater than the maximum.

Examples

```
constrain(1:20, c(3, 19))
```

`correctval` *Predict the corrected proportion using a model fit of Table 1 of Litchfield and Wilcoxon (1949)*

Description

Given an expected proportion, calculate the corrected proportion using a model fit of Table 1 of Litchfield and Wilcoxon (1949).

Usage

```
correctval(val, fit)
```

Arguments

`val`	A numeric vector of expected values (as proportions).
`fit`	A model object that can be used to predict the corrected values (as proportions) from `distexpprop5`, the distance from the expected values (as proportions) and 0.5. Typically the output from `gamtable1()`.

Value

A numeric vector of corrected values (as proportions), the same length as `val`.

Examples

```
gamfit <- gamtable1()
correctval(c(0.37, 0.5, 0.63), gamfit)
```

`dataprep` *Prepare Data*

Description

Prepare dose-effect data for evaluation.

Usage

```
dataprep(dose, ntot, nfx)
```

Arguments

`dose`	A numeric vector of chemical concentrations.
`ntot`	A numeric vector of the number of individuals that were tested at each dose.
`nfx`	A numeric vector of the number of individuals that were affected at each dose.

Value

A data frame with eight columns (ordered by dose and proportion affected), seven numeric vectors and one logical vector: dose - chemical concentrations. ntot - the number of individuals that were tested at each dose. nfx - the number of individuals that were affected at each dose. rec - the record number corresponding to the input vectors `dose`, `ntot`, `nfx`. pfx - the proportion of individuals that were affected at each dose. log10dose - log transformed dose, `log10(dose)`. bitpfx - probit transformed proportional affected, `probit(pfx)`. fxcateg - effects category: 0 for none affected, 100 for all affected, and 50 for other proportions affected. LWkeep - logical vector identifying records to keep for Litchfield and Wilcoxon (1949, step A1) method.

References

Litchfield, JT Jr. and F Wilcoxon. 1949. A simplified method of evaluating dose-effect experiments[2]. Journal of Pharmacology and Experimental Therapeutics 99(2):99-113.

Examples

```
conc <- c(0.0625, 0.125, 0.25, 0.5, 1)
numtested <- rep(8, 5)
nalive <- c(1, 4, 4, 7, 8)
dataprep(dose=conc, ntot=numtested, nfx=nalive)
```

[2] `http://jpet.aspetjournals.org/content/96/2/99.abstract`

estimable — *Determine if a Dose-Effect Relation is Estimable*

Description

Determine if a dose-effect relation is estimable based on available data.

Usage

```
estimable(DEdata)
```

Arguments

DEdata	A data frame of dose-effect data (typically, the output from dataprep containing at least two variables: dose, a numeric vector of chemical concentrations, and pfx, a numeric vector of proportional effects at each dose.

Details

A dose-effect relation is defined to be estimable (with error) if and only if there are at least three test records and there is some (non-zero) variability in both the doses and the proportional effects.

Value

A logical scalar indicating if a dose-effect relation is estimable. If FALSE, a warning is generated.

Examples

```
conc <- c(0.0625, 0.125, 0.25, 0.5, 1)
numtested <- rep(8, 5)
nalive <- c(1, 4, 4, 7, 8)
mydat <- dataprep(dose=conc, ntot=numtested, nfx=nalive)
estimable(mydat)
nalive2 <- rep(4, 5)
mydat2 <- dataprep(dose=conc, ntot=numtested, nfx=nalive2)
estimable(mydat2)
```

fill — *Fill in Missing Values*

Description

Fill in missing values in a vector, using the last recorded value.

Usage

```
fill(x)
```

Arguments

x	A vector, can be character, numeric, or logical.

Details

Similar to `na.locf` in the `zoo` package, but works for "" in character vectors as well.

Value

A vector the same length as `x`, with all NAs or ""s replace by the last value for the vector. Note that and missing values at the beginning of the vector will not be replaced.

Examples

```
numvec <- c(NA, 1:5, NA, NA, NA, 10:12, NA)
fill(numvec)

charvec <- c("", letters[1:5], "", "", "", letters[10:12], "")
fill(charvec)
```

`fitlinear` *Determine Linear Regression Coefficients from Dose-Effect Data*

Description

Determine coefficients (intercept and slope) from dose-effect data using simple linear regression on the log10 dose vs. probit effect scale.

Usage

```
fitlinear(DEdata, fit, constr = c(0.0001, 0.9999))
```

Arguments

`DEdata`	A data frame of dose-effect data (typically, the output from `dataprep` containing eight variables: dose, ntot, nfx, pfx, log10dose, bitpfx, fxcateg, and LWkeep.
`fit`	A model object that can be used to predict the corrected values (as proportions) from `distexpprop5`, the distance from the expected values (as proportions) and 0.5. Typically the output from `gamtable1()`.
`constr`	A numeric vector of length two, indicating the constraints (see `constrain`) applied to the proportional effects, default c(0.0001, 0.9999).

Value

A numeric vector of length two, the estimated intercept and slope.

Examples

```
conc <- c(0.0625, 0.125, 0.25, 0.5, 1)
numtested <- rep(8, 5)
nalive <- c(1, 4, 4, 7, 8)
mydat <- dataprep(dose=conc, ntot=numtested, nfx=nalive)
gamfit <- gamtable1()
fitlinear(mydat, gamfit)
```

fitLW — *Apply Litchfield and Wilcoxon Evaluation of Dose-Effect Experiments*

Description

Automatically apply Litchfield and Wilcoxon's (1949) evaluation of dose-effect experiments.

Usage

```
fitLW(DEdata)
```

Arguments

DEdata	A data frame of dose-effect data (typically, the output from dataprep containing at least eight variables: dose, ntot, nfx, pfx, log10dose, bitpfx, fxcateg, and LWkeep.

Value

A list of length three: chi = the chi-squared statistic with associated P value and degrees of freedom, params = the estimated intercept and slope of the dose-response curve on the log10 probit scale, LWest = the Litchfield Wilcoxon estimates of ED50 with 95% confidence intervals and other metrics used in their step-by-step approach (ED16, ED84, S, and slope).

Examples

```
dose <- c(0.0625, 0.125, 0.25, 0.5, 1)
numtested <- rep(8, 5)
numalive <- c(1, 4, 4, 7, 8)
mydat <- dataprep(dose=dose, ntot=numtested, nfx=numalive)
mydat
fitLW(mydat)
```

fitprobit — *Fit a Probit Regression to Dose-Effect Data*

Description

Fit a probit regression to dose-effect data, using the log10 of the dose, the binomial family, and the probit link.

Usage

```
fitprobit(DEdata)
```

Arguments

DEdata	A data frame of dose-effect data (typically, the output from dataprep containing eight variables: dose, ntot, nfx, pfx, log10dose, bitpfx, fxcateg, and LWkeep.

Value

A an object of class `glm`.

Examples

```
conc <- c(0.0625, 0.125, 0.25, 0.5, 1)
numtested <- rep(8, 5)
nalive <- c(1, 4, 4, 7, 8)
mydat <- dataprep(dose=conc, ntot=numtested, nfx=nalive)
fitprobit(mydat)
```

---

fxcat *Define Effect Category*

Description

Define three effect categories, 0 for none affected, 100 for all affected, and 50 for other proportions affected.

Usage

```
fxcat(dat)
```

Arguments

`dat`	A data frame of raw toxicity data, including these three variables: dose (the concentration of the applied chemical), ntot (the number of individuals tested), and nfx (the number of affected individuals).

Value

A numeric vector the same length as `prob` with quantiles on the probit scale.

Examples

```
test <- data.frame(
dose=c(0.0625, 0.125, 0.25, 0.5),
ntot=rep(8, 4),
nfx = c(0, 4, 6, 8))
cbind(test, fxcat(test))
```

gamtable1 *Fit a smooth GAM to Table 1 of Litchfield and Wilcoxon (1949)*

Description

Fit a smooth GAM function to replace looking up values in Table 1 of Litchfield and Wilcoxon (1949).

Usage

```
gamtable1()
```

Details

Note that for an expected value of 37 Table 1 gives a corrected value of 9.4, but for an expected value of 63 it gives a corrected value of 90.5. To ensure that both values add to 100, I used corrected values of 9.45 and 90.55. The expected and corrected values from Table 1 are then used to build a GAM model, which is used as input to the `correctval` function.

Value

A `gamObject` that can be used to predict the corrected values (as proportions) from `distexpprop5`, the distance from the expected values (as proportions) and 0.5

Examples

```
fit <- gamtable1()
summary(fit)
plot(fit)
```

invprobit *Convert Probit Scale to Proportions*

Description

Convert values on the probit scale to their proportions on the 0 to 1 scale.

Usage

```
invprobit(quan)
```

Arguments

`quan`	A numeric vector of probit quantiles.

Details

Simply calls `pnorm(quan)`.

Value

A numeric vector of proportions the same length as `quan`.

Examples

```
invprobit(c(-3, -1, 0, 1, 3))
```

---

keeponly *Eliminate Consecutive Extreme Values*

Description

Generate the index for eliminating values beyond a given maximum number of consecutive extremes allowed.

Usage

```
keeponly(orderedx, extremes = c(0, 100), nconsec = 2)
```

Arguments

`orderedx`	A numeric vector.
`extremes`	A numeric vector of length two, boundary limits of numeric vector, default c(0, 100).
`nconsec`	An integer scalar, the maximum number of consecutive extreme values allowed, default 2.

Value

A logical vector for selecting all elements of `orderedx` without exceeding `nconsec` consecutive extreme values.

Examples

```
vec <- c(0, 0, 0, 4, 4, 4, 100, 100, 100, 100)
vec[keeponly(vec)]
```

LW1949 *Automated Litchfield and Wilcoxon (1949) Evaluation of Dose-Effect Experiments*

Description

LW1949 is an automated approach to Litchfield and Wilcoxon's (1949) evaluation of dose-effect experiments. **LW1949** was first introduced by Adams et al. (*in preparation*).

Details

U.S. Geological Survey (USGS) Computer Program **LW1949** version 2014-11. Written by Jean V. Adams, USGS - Great Lakes Science Center[3], Ann Arbor, Michigan, USA. Written in programming language R (R Core Team, 2014, www.R-project.org), version 3.1.1 (2014-07-10). Run on a PC with Intel(R) Core(TM) I7-4600m CPU, 2.90 GHz processor, 16.0 GB RAM, and Microsoft Windows 7 Enterprise operating system 2009 Service Pack 1. Source code is available from Jean V. Adams on GitHub[4], <jvadams@usgs.gov>.

Disclaimer: Although this program has been used by the USGS, no warranty, expressed or implied, is made by the USGS or the United States Government as to the accuracy and functioning of the program and related program material nor shall the fact of distribution constitute any such warranty, and no responsibility is assumed by the USGS in connection therewith.

References

Adams, JV, KS Slaght, and MA Boogaard. *In preparation*. An automated approach to Litchfield and Wilcoxon's evaluation of dose-effect experiments.

Litchfield, JT Jr. and F Wilcoxon. 1949. A simplified method of evaluating dose-effect experiments[5]. Journal of Pharmacology and Experimental Therapeutics 99(2):99-113.

`plotDE` *Plot Dose-Effect Experiments*

Description

Plot the results of dose-effect experiments.

Usage

```
plotDE(DEdata, xlab = "Dose", ylab = "Affected  (%)", ylim = c(0.1,
  99.9), ...)
```

Arguments

`DEdata`	A data frame of dose-effect data (typically, the output from `dataprep` containing at least five variables: dose, pfx, log10dose, bitpfx, fxcateg.
`xlab`	A caracter scalar, the title for the dose (x) axis, default "Dose".
`ylab`	A caracter scalar, the title for the affects (y) axis, default "Affected (%)".
`ylim`	A numeric vector of length two giving the y coordinate range for affects (%), default c(0.1, 99.9). Observed effects beyond this range will be plotted at the limits of this range using an open symbol.
`...`	Additional arguments to `plot`.

[3] http://www.glsc.usgs.gov/
[4] https://github.com/JVAdams/LW1949
[5] http://jpet.aspetjournals.org/content/96/2/99.abstract

Examples

```
dose <- c(0.0625, 0.125, 0.25, 0.5, 1)
numtested <- rep(8, 5)
numalive <- c(1, 4, 4, 7, 8)
mydat <- dataprep(dose=dose, ntot=numtested, nfx=numalive)
# just plot the raw data
plotDE(mydat)
# plot the raw data and some fitted lines
fLW <- fitLW(mydat)
fp <- fitprobit(mydat)
plotDE(mydat)
abline(fp$coef, lty=2)
abline(fLW$params)
legend("topleft", c("Litchfield-Wilcoxon", "Probit"), lty=c(1, 2), bg="white")
```

---

predlinear *Determine the Effective Dose from a Linear Regression Fit*

Description

Determine the effective dose for a specified percent effect from the intercept and slope of a linear regression.

Usage

```
predlinear(pct, b0, b1)
```

Arguments

pct	A numeric vector of effects (in percents) for which to estimate the effect dose(s).
b0	A numeric vector (more commonly a scalar) giving the intercept of the dose-response curve.
b1	A numeric vector (more commonly a scalar) giving the slope of the dose-response curve. x = dose (the concentration of the applied chemical on the log10 scale), and y, the proportion of affected individuals (on the probit scale, with 0s converted to 0.1% and 1s converted to 99.9%).

Value

A numeric vector the same length as `pct` giving the estimated dose at the specified percent effect.

Examples

```
predlinear(c(16, 50, 84, 99.9), 1.700875, 2.199559)
```

predprobit *Determine the Effective Dose from a Probit Regression Fit*

Description

Determine the effective dose for a specified percent effect from a fitted probit regression model.

Usage

```
predprobit(pct, pfit)
```

Arguments

pct	A numeric scalar of the effect (as a percent) for which to estimate the effective dose.
pfit	An object of class glm representing a probit regression fit to dose-effect data, typically the result of a call to fitprobit.

Value

A numeric vector of length three, the effective dose and the lower and upper 95% confidence limits.

Examples

```
conc <- c(0.0625, 0.125, 0.25, 0.5, 1)
numtested <- rep(8, 5)
nalive <- c(1, 4, 4, 7, 8)
mydat <- dataprep(dose=conc, ntot=numtested, nfx=nalive)
myfit <- fitprobit(mydat)
predprobit(50, myfit)
```

prettylog *Pretty Breakpoints on Log Scale*

Description

Compute a sequence of "round" values which cover the range of x on the log scale.

Usage

```
prettylog(x, lead = c(1, 5), extra = 5)
```

Arguments

x	A numeric vector.
lead	An integer vector giving the desired lead digitis of pretty values on the log scale, default c(1, 5).
extra	An integer scalar giving the desired number of additional non-log scale values to include, default 5.

Value

A numeric vector of pretty values covering the range of `x` on the log scale.

Examples

```
vals <- rlnorm(100, 6)
summary(vals)
prettylog(vals, 1, 0)
prettylog(vals, 1)
prettylog(vals, c(1, 2, 5))
```

---

probit *Convert Proportions to the Probit Scale*

Description

Convert proportions to the probit scale.

Usage

```
probit(prob)
```

Arguments

`prob`	A numeric vector of proportions.

Details

Simply calls `qnorm(prob)`.

Value

A numeric vector the same length as `prob` with quantiles on the probit scale.

Examples

```
probit(c(0.001, 0.01, 0.1, 0.5, 0.9, 0.99, 0.999))
```