Package 'LW1949'

February 23, 2015

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

Version 2015-01

Date 2015-01

Author Jean V. Adams [aut, cre]

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

Depends R (>= 3.1.0)

Imports mgcv, tcltk, 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 to find the dose-response curve that minimizes the chi-squared statistic measuring the distance between the observed and expected values of the response (the proportion affected). Following Litchfield and Wilcoxon (1949, steps B1 and B2), records with expected effects < 0.01% or > 99.99% are deleted, and other expected effects 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`, chi-squared statistic; `df`, degrees of freedom; and `pval`, 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

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

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

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, chi-squared statistic; df, degrees of freedom; and pval, 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`.

References

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

Examples

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

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

`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. Journal of Pharmacology and Experimental Therapeutics 99(2):99-113. [link][3].

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)
```

[3] 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 if and only if there are at least two 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, resetwhen = rep(FALSE, length(x)))
```

Arguments

x	A vector, can be character, numeric, or logical.
resetwhen	A logical vector, the same length as x.

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)
newgroup <- c(1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0)
fill(numvec)
fill(numvec, newgroup)

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(1e-04, 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 the number of records with partial mortalities (npartmort) as well as other metrics used in their step-by-step approach (ED16, ED84, S with 95% confidence intervals, N', and fED50).

References

Litchfield, JT Jr. and F Wilcoxon. 1949. A simplified method of evaluating dose-effect experiments. Journal of Pharmacology and Experimental Therapeutics 99(2):99-113. [link][4].

Examples

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

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

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

References

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

Examples

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

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

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(x, extremes = c(0, 100), nconsec = 2)
```

Arguments

x	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)]
# the original vector need not be ordered
vec <- c(100, 4, 100, 4, 0, 100, 0, 4, 0, 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

An example of how to use the functions in **LW1949** is given in this vignette [link][6]. Use `dataprep` to create a data frame with the results of a dose-effect experiment. Use `fitLW` and `fitprobit` to fit dose-effect relations. And use `plotDE` to plot the results.

U.S. Geological Survey (USGS) Computer Program **LW1949** version 2015-01. Written by Jean V. Adams, USGS - Great Lakes Science Center glsc.usgs.gov[7], 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, github.com/JVAdams/LW1949[8], *jvadams (at) usgs (dot) 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. Journal of Pharmacology and Experimental Therapeutics 99(2):99-113. [link][9].

[6] `https://github.com/JVAdams/LW1949/blob/master/Vignette.md`
[7] `http://www.glsc.usgs.gov/`
[8] `https://github.com/JVAdams/LW1949`
[9] `http://jpet.aspetjournals.org/content/96/2/99.abstract`

LWP *User Friendly Evaluation of Dose-Effect Experiments using Litchfield-Wilcoxon and Probit Methods*

Description

User friendly evaluation of dose-effect experiments using automated Litchfield Wilcoxon (1949) and probit estimation methods. This function has been tailored for non-R users with input data set up in a particular way (see Details).

Usage

```
LWP(rawfile = NULL, descrcolz = 4, saveplots = TRUE, showplots = FALSE,
  saveresults = TRUE, showresults = TRUE, returnresults = FALSE)
```

Arguments

`rawfile`	A character scalar specifying the path of the input data as a csv file. If NULL, default, the user will be prompted to browse to a file using a menu.
`descrcolz`	A numeric scalar, the number of columns to use as the description of the test, from 1 to `descrcolz`, default 4.
`saveplots`	A logical scalar indicating if plots should be saved to a pdf file, default TRUE.
`showplots`	A logical scalar indicating if plots should be shown on screen, default FALSE.
`saveresults`	A logical scalar indicating if results should be saved to a csv file, default TRUE. The csv file is given the same name (plus the suffix "Smry") and is placed in the same directory as the input file.
`showresults`	A logical scalar indicating if results should be printed to the console, default TRUE. These results include the chi-squared statistic, degrees of freedom, and p-value for the Litchfield Wilcoxon method.
`returnresults`	A logical scalar indicating if results should be returned by the function, default FALSE.

Details

The input data must include at least these seven columns, with these names in the header row:

- `Test ID` = A character or numeric vector, the unique identifier for each test
- `Source` = A character vector, the source of the chemical
- `Batch` = A character or numeric vector, the batch of the chemical
- `Species` = A character vector, the species tested
- `TFM Conc. (mg/L)` = A numeric vector, the concentration of TFM in mg/L
- `No. Tested` = A numeric vector, the number of animals tested
- `No. Dead` = A numeric vector, the number of animals dead

Value

If `returnresults=TRUE`, a data frame with 11 rows per test and 2 more columns than the input data. Three columns from the input data are not included (`TFM Conc. (mg/L)`, `No. Tested`, and `No. Dead`). Five columns are added: the parameter (`param`), the method used (`method`), the estimate (`estimate`), and the 95% confidence interval of the estimate (`lower95ci` and `upper95ci`)

References

Litchfield, JT Jr. and F Wilcoxon. 1949. A simplified method of evaluating dose-effect experiments. Journal of Pharmacology and Experimental Therapeutics 99(2):99-113. [link][10].

Examples

```
## Not run:
LWP()

## End(Not run)
```

---

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 character scalar, the title for the dose (x) axis, default "Dose".
`ylab`	A character 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`.

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

Examples

```
dose <- c(0.0625, 0.125, 0.25, 0.5, 1)
ntested <- rep(8, 5)
nalive <- c(1, 4, 4, 7, 8)
mydat <- dataprep(dose=dose, ntot=ntested, nfx=nalive)
# 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, LWmod, simple = FALSE)
```

Arguments

pct	A numeric vector of effects (in percents) for which to estimate the effective dose(s).
LWmod	If simple=TRUE, a numeric vector of length two giving the intercept and slope of the linear relation between the dose (x, the concentration of the applied chemical on the log10 scale), and the proportion of affected individuals (y, on the probit scale, with 0s converted to 0.1% and 1s converted to 99.9%). If simple=FALSE, a list with the results of fitting a Litchfield and Wilcoxon model to dose-effect data, the output from fitLW.
simple	A logical scalar indicating whether to carry out a simple estimation of effective doses from the intercept and slope (TRUE), or an estimation of effective doses with confidence intervals from the Litchfield and Wilcoxon model (default, FALSE).

Details

Follows methods outlined in Litchfield and Wilcoxon (1949). Specifically, for the 95% confidence intervals, see page 105, and equation 13 in the Appendix (corresponding to Nomograph 4).

Value

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

If `simple=TRUE`, a numeric vector the same length as `pct` with the estimated effective doses. If `simple=FALSE`, an n*4 numeric matrix with the given effects (`pct`), the effective doses (`ED`), and Litchfield and Wilcoxon's (1949) 95% confidence intervals for the effective doses (`lower` and `upper`). The number of rows of the matrix, n, is the length of `pct`.

References

Litchfield, JT Jr. and F Wilcoxon. 1949. A simplified method of evaluating dose-effect experiments. Journal of Pharmacology and Experimental Therapeutics 99(2):99-113. [link][11].

Examples

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

dose <- c(0.0625, 0.125, 0.25, 0.5, 1)
ntested <- rep(8, 5)
nalive <- c(1, 4, 4, 7, 8)
mydat <- dataprep(dose=dose, ntot=ntested, nfx=nalive)
fLW <- fitLW(mydat)
predlinear(c(25, 50, 99.9), fLW)
```

`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.

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

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 digits 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))
```