

# Package ‘relevance’

May 25, 2021

**Type** Package

**Title** Calculate Relevance

**Version** 1.2

**Date** 2021-05-24

**Author** Werner A. Stahel

**Maintainer** Werner A. Stahel <stahel@stat.math.ethz.ch>

**Depends** R (>= 3.5.0)

**Imports** stats, utils

**Suggests** MASS, survival, knitr

**VignetteBuilder** knitr

**Description** Calculates relevance and significance values for simple models and for many types of regression models. These are introduced in 'Stahel, Werner A.' (2021) ``New relevance and significance measures to replace p-values.'' <<https://stat.ethz.ch/~stahel/relevance/stahel-relevance2103.pdf>>.

**License** GPL-2

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2021-05-25 20:40:03 UTC

## R topics documented:

relevance-package . . . . .	2
confintF . . . . .	4
d.blast . . . . .	5
drop1Wald . . . . .	6
getcoeftable . . . . .	7
inference . . . . .	8
ovarian . . . . .	9
plot.inference . . . . .	10

print.inference . . . . .	11
relevance.options . . . . .	13
termeffects . . . . .	13
termtable . . . . .	14
twosamples . . . . .	16

## Index 19

relevance-package      *Calculate Relevance*

### Description

Calculates relevance and significance values for simple models and for many types of regression models. These are introduced in 'Stahel, Werner A.' (2021) "New relevance and significance measures to replace p-values." <<https://stat.ethz.ch/~stahel/relevance/stahel-relevance2103.pdf>>.

### Details

The DESCRIPTION file:

```
Package:      relevance
Type:        Package
Title:       Calculate Relevance
Version:     1.2
Date:       2021-05-24
Author:      Werner A. Stahel
Maintainer:  Werner A. Stahel <stahel@stat.math.ethz.ch>
Depends:    R (>= 3.5.0)
Imports:    stats, utils
Suggests:   MASS, survival, knitr
VignetteBuilder: knitr
Description: Calculates relevance and significance values for simple models and for many types of regression models. T
License:    GPL-2
```

Index of help topics:

```
confintF      Confidence Interval for the Non-Central F and
              Chisquare Distribution
d.blast       Blasting for a tunnel
drop1Wald     Drop Single Terms of a Model and Calculate
              Respective Wald Tests
getcoeftable  Extract Components of a Fit
inference     Calculate Confidence Intervals and Relevance
              and Significance Values
ovarian       ovarian
plot.inference Plot Inference Results
```

<code>print.inference</code>	Print Tables with Inference Measures
<code>relevance-package</code>	Calculate Relevance
<code>relevance.options</code>	Options for the relevance Package
<code>termeffects</code>	All Coefficients of a Model Fit
<code>termtable</code>	Statistics for Linear Models, Including Relevance Statistics
<code>twosamples</code>	Relevance and Significance for One or Two Samples

Further information is available in the following vignettes:

`relevance-descr` 'Package relevance for calculating Relevance and Significance Measures' (source, pdf)

Relevance is a measure that expresses the (scientific) relevance of an effect. The simplest case is a single sample of supposedly normally distributed observations, where interest lies in the expectation, estimated by the mean of the observations. There is a threshold for the expectation, below which an effect is judged too small to be of interest.

The estimated relevance '*Rle*' is then simply the estimated effect divided by the threshold. If it is larger than 1, the effect is thus judged relevant. The two other values that characterize the relevance are the limits of the confidence interval for the true value of the relevance, called the secured relevance '*Rls*' and the potential relevance '*Rlp*'.

If  $Rle > 1$ , then one might say that the effect is "significantly relevant".

Another useful measure, meant to replace the p-value, is the "significance" '*Sg0*'. In the simple case, it divides the estimated effect by the critical value of the (t-) test statistic. Thus, the statistical test of the null hypothesis of zero expectation is significant if '*Sg0*' is larger than one,  $Sg0 > 1$ .

These measures are also calculated for the comparison of two groups, for proportions, and most importantly for regression models. For models with linear predictors, relevances are obtained for standardized coefficients as well as for the effect of dropping terms and the effect on prediction.

The most important functions are

`twosamples()`: calculate the measures for two paired or unpaired samples or a simple mean. This function calls

`inference()`: calculates the confidence interval and significance based on an estimate and a standard error, and adds relevance for a standardized effect.

`termtable()`: deals with fits of regression models with a linear predictor. It calculates confidence intervals and significances for the coefficients of terms with a single degree of freedom. It includes the effect of dropping each term (based on the `drop1` function) and the respective significance and relevance measures.

`termeffects()`: calculates the relevances for the coefficients related to each term. These differ from the entries of `termtable` only for terms with more than one degree of freedom.

### Author(s)

Werner A. Stahel

Maintainer: Werner A. Stahel <stahel@stat.math.ethz.ch>

## References

Stahel, Werner A. (2021). New relevance and significance measures to replace p-values. To appear in PLoS ONE

## See Also

Package **regrr**, available from <https://regdevelop.r-forge.r-project.org>

## Examples

```
data(swiss)
rr <- lm(Fertility ~ . , data = swiss)
termtable(rr)
```

---

confintF

*Confidence Interval for the Non-Central F and Chisquare Distribution*

---

## Description

Confidence Interval for the Non-Central F and Chisquare Distribution

## Usage

```
confintF(f, df1, df2, testlevel = 0.05)
```

## Arguments

f	observed F value(s)
df1	degrees of freedom for the numerator of the F distribution
df2	degrees of freedom for the denominator of the F distribution
testlevel	level of the (two-sided) test that determines the confidence interval, 1 - confidence level

## Details

The confidence interval is calculated by solving the two implicit equations  $qf(f, df1, df2, x) = testlevel/2$  and  $\dots = 1 - testlevel/2$ . For  $f > 100$ , the usual  $f \pm$  standard error interval is used as a rather crude approximation.

A confidence interval for the non-centrality of the Chisquare distribution is obtained by setting  $df2$  to  $Inf$  (the default) and  $f = x2/df1$  if  $x2$  is the observed Chisquare value.

## Value

vector of lower and upper limit of the confidence interval, or, if any of the arguments has length  $> 1$ , matrix containing the intervals as rows.

**Author(s)**

Werner A. Stahel

**See Also**

[qf](#)

**Examples**

```
confintF(5, 3, 200)
## [1] 2.107 31.95
confintF(1:5, 5, 20) ## lower limit is 0 for the first 3 f values
```

---

d.blast

*Blasting for a tunnel*

---

**Description**

Blasting causes tremor in buildings, which can lead to damages. This dataset shows the relation between tremor and distance and charge of blasting.

**Usage**

```
data("d.blast")
```

**Format**

A data frame with 388 observations on the following 7 variables.

no Identification of the date and time

datetime Date and time in the format '%d.%m. %H:%M'

device Number of measuring device, 1 to 4

charge Charge of blast

distance Distance between blasting and location of measurement

tremor Tremor energy (target variable)

location Code for location of the building, loc1 to loc8

**date** date in Date format

**Details**

The charge of the blasting should be controlled in order to avoid tremors that exceed a threshold. This dataset can be used to establish the suitable rule: For a given distance, how large can charge be in order to avoid exceedance of the threshold?

**Source**

Basler and Hoffmann AG, Zurich

**Examples**

```
data(d.blast)

summary(lm(log10(tremor)~location+log10(distance)+log10(charge),
           data=d.blast))
```

---

 drop1Wald

---

*Drop Single Terms of a Model and Calculate Respective Wald Tests*


---

**Description**

drop1Wald calculates tests for single term deletions based on the covariance matrix of estimated coefficients instead of re-fitting a reduced model. This helps in cases where re-fitting is not feasible, inappropriate or costly.

**Usage**

```
drop1Wald(object, scope=NULL, scale = NULL, test = NULL, k = 2, ...)
```

**Arguments**

object	a fitted model.
scope	a formula giving the terms to be considered for dropping. If 'NULL', 'drop.scope(object)' is obtained
scale	an estimate of the residual mean square to be used in computing Cp. Ignored if '0' or 'NULL'.
test	see <a href="#">drop1</a>
k	the penalty constant in AIC / Cp.
...	further arguments, ignored

**Details**

The test statistics and Cp and AIC values are calculated on the basis of the estimated coefficients and their (unscaled) covariance matrix as provided by the fit object. The function may be used for all model fitting objects that contain these two components as \$coefficients and \$cov.unscaled.

**Value**

An object of class 'anova' summarizing the differences in fit between the models.

**Note**

drop1Wald is used for models of class 'lm' or 'lmrob' for preparing a termtable.

**Author(s)**

Werner A. Stahel

**See Also**[drop1](#)**Examples**

```
data(d.blast)
r.blast <- lm(log10(tremor)~location+log10(distance)+log10(charge),
             data=d.blast)
drop1(r.blast)
drop1Wald(r.blast)

## Example from example(glm)
dd <- data.frame(treatment = gl(3,3), outcome = gl(3,1,9),
                counts = c(18,17,15,20,10,20,25,13,12))
r.glm <- glm(counts ~ outcome + treatment, data = dd, family = poisson())
drop1(r.glm, test="Chisq")
drop1Wald(r.glm)
```

---

`getcoefstable`*Extract Components of a Fit*

---

**Description**

Retrieve the table of coefficients and standard errors, or the scale parameter, or the factors needed for standardizing coefficients from diverse model fitting results

**Usage**

```
getcoefstable(object)
getscalepar(object)
getcoeffactor(object)
```

**Arguments**

`object` an R object resulting from a model fitting function.

**Details**

Object `regrModelClasses` contains the names of the classes for which the result should work. For other model classes, the function is not tested and may fail.

**Value**

For `getcoefstable`: Matrix containing at least the two columns containing the estimated coefficients (first column) and the standard errors (second column).

For `getscalepar`: scale parameter.

For `getcoeffactor`: vector of factors, with attributes `scale`, `fitclass` and `family` or `dist` according to object.

**Author(s)**

Werner A. Stahel

**Examples**

```
rr <- lm(Fertility ~ . , data = swiss)
getcoeftable(rr) # identical to coef(summary(rr)) or also summary(rr)$coefficients
getscalepar(rr)

if(requireNamespace("survival", quietly=TRUE)) {
  data(ovarian) ## , package="survival"
  rs <- survival::survreg(survival::Surv(futime, fustat) ~ ecog.ps + rx,
    data = ovarian, dist = "weibull")
  getcoeftable(rs)
  getcoeffactor(rs)
}
```

---

 inference

---

*Calculate Confidence Intervals and Relevance and Significance Values*


---

**Description**

Calculates confidence intervals and relevance and significance values given estimates, standard errors and, for relevance, additional quantities.

**Usage**

```
inference(estimate, se = NULL, df = NULL, testlevel = 0.05, stcoef=TRUE, rlv=TRUE,
  rlv.threshold=0.1, object=NULL)
```

**Arguments**

estimate	vector of estimates or matrix containing estimates and their standard errors. The latter is needed if se is not given.
se	vector of standard errors of the estimates
df	degrees of freedom of the residuals
testlevel	test level
stcoef	standardized coefficients, suitable for If NULL, these will be calculated from object.
rlv	logical: Should relevances be calculated?
rlv.threshold	Relevance threshold
object	model fit, needed for calculating stcoef.

**Details**

The coefficients divided by standard errors are assumed to be t-distributed with df degrees of freedom. For df==Inf, this is the standard normal distribution.



**Value**

A data.frame with the variables

estimate, se	coefficients and standard errors
ciLow, ciUp	lower and upper limit of the confidence interval
testst	t-test statistic
Sig0	significance value, i.e., test statistic divided by critical value, which in turn is the $1 - \text{testLevel}/2$ -quantile of the t-distribution.
p.value	p value
p.symbol	the conventional symbol corresponding to the p value

If rlv is TRUE,

stcoef	standardized coefficient
st.Low, st.Up	confidence interval for stcoef
Rle	estimated relevance of coef
Rls	secured relevance, lower end of confidence interval for the relevance of coef
Rlp	potential relevance, upper end of confidence interval ...
Rls.symbol	symbols for the secured relevance

**Author(s)**

Werner A. Stahel

**Examples**

```
data(d.blast)
rr <-
  lm(log10(tremor)~location+log10(distance)+log10(charge), data=d.blast)
inference(rr) ## same as
inference(summary(rr)$coef)
```

---

ovarian

*ovarian*

---

**Description**

copy of ovarian from package 'survival'. Will disappear

**Usage**

```
data("ovarian")
```

**Format**

A data frame with 26 observations on the following 6 variables.

futime a numeric vector  
 fustat a numeric vector  
 age a numeric vector  
 resid.ds a numeric vector  
 rx a numeric vector  
 ecog.ps a numeric vector

**Details**

This copy is here since the package was rejected because the checking procedure did not find it in the package

**Examples**

```
data(ovarian)
summary(ovarian)
```

---

plot.inference	<i>Plot Inference Results</i>
----------------	-------------------------------

---

**Description**

Plot the relevance interval(s) for one or several items

**Usage**

```
## S3 method for class 'inference'
plot(x, pos = NULL,
     plpars=list(lwd=c(2,1,2), endmarks=1, extend=NA, framecol="gray70"),
     xlab = "relevance", ...)
## S3 method for class 'termeffects'
plot(x, pos = NULL, single=FALSE,
     plpars=list(lwd=c(2,1,2), endmarks=1, extend=NA, framecol="gray70",
                 termeffects.gap = 0.2), xlab="relevance", ...)
```

**Arguments**

x	a vector or matrix of class inference
pos	positions of the bars in vertical direction
plpars	graphical options for this function, see Details
xlab	label for horizontal axis
single	logical: should terms with a single degree of freedom be plotted?
...	further arguments to the call of plot

**Details**

The graphical options are:

- `lwdline` width for the intervals
- `endmarks` determines the length of the end marks of the intervals as a multiplier of the default length
- `extend` extension of the vertical axis beyond the range
- `framecol` color to be used for the framing lines: axis and vertical lines at relevances 0 and 1

**Value**

none

**Author(s)**

Werner A. Stahel

**Examples**

```
data(swiss)
rr <- lm(Fertility ~ . , data = swiss)
rt <- termtable(rr)
plot(rt)
```

```
data(d.blast)
rlm <- lm(log10(tremor)~location+log10(distance)+log10(charge), data=d.blast)
rte <- termeffects(rlm)
plot(rte, single=TRUE)
```

---

print.inference

*Print Tables with Inference Measures*

---

**Description**

Print methods for objects of class "inference", "termeffects", or "printInference".

**Usage**

```
## S3 method for class 'inference'
print(x, show = getOption("show.inference"), print=TRUE,
      digits = getOption("digits.reduced"), transpose.ok = TRUE,
      legend = NULL, na.print = " ", ...)

## S3 method for class 'termeffects'
print(x, show = getOption("show.inference"),
      transpose.ok = TRUE, single = FALSE, print = TRUE, warn = TRUE, ...)

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

**Arguments**

<code>x</code>	object to be printed
<code>show</code>	determines items (columns) to be shown
<code>digits</code>	number of significant digits to be printed
<code>transpose.ok</code>	logical: May a single column be shown as a row?
<code>single</code>	logical: Should components with a single coefficient be printed?
<code>legend</code>	logical: should the legend(s) for the symbols characterizing p-values and relevances be printed? Defaults to <code>regoptions("show.symbolLegend")</code> .
<code>na.print</code>	string by which NAs are shown
<code>print</code>	logical: if FALSE, no printing will occur, used to edit the result before printing it.
<code>warn</code>	logical: Should the warning be issued if <code>termeffects</code> has nothing to print since there are no terms with more than one degree of freedom
<code>...</code>	further arguments, passed to <code>print.data.frame()</code> .

**Details**

The result, say `rr`, can be printed by using `print.printInference` writing just `print(rr)`, which is just what happens internally unless `print=FALSE` is used. This allows for editing the result before printing it, see Examples

**Value**

A kind of formatted version of `x`, with class `printInference`. For `print.inference`, it will be a character vector or a `data.frame` with attributes `head` and `tail` if applicable. For `print.termeffects`, it will be a list of such elements, with its own `head` and `tail`. It is invisibly returned.

**Author(s)**

Werner A. Stahel

**See Also**

[twosamples](#), [termtable](#), [termeffects](#), [inference](#).

**Examples**

```
data(d.blast)
r.blast <-
  lm(log10(tremor)~location+log10(distance)+log10(charge), data=d.blast)
rt <- termtable(r.blast)
## print() : first default, then "classical" :
rt
print(rt, show="classical")

class(te <- termeffects(r.blast)) # "termeffects"
rr <- print(te, print=FALSE)
```

```
attr(rr, "head") <- sub("lm", "Linear Regression", attr(rr, "head"))
class(rr) # "printInference"
rr # <=> print(rr)
```

relevance.options      *Options for the relevance Package*

### Description

List of options used in the relevance package to select items and formats for printing inference elements

### Usage

```
relevance.options
rlv.symbols
p.symbols
```

### Format

The format is: List of 22 \$ digits.reduced : 3 \$ testlevel : 0.05 \$ show.confint : TRUE \$ termtable : TRUE \$ vif : TRUE \$ rlvThres : stand=0.1, rel=0.1, prop=0.1, coef=0.1, drop=0.1, pred=0.05 \$ show.ifc : "relevance" \$ show.ifc.relevance : "Rle" "Rlp" "Rls" "Rls.symbol" \$ show.ifc.test : "Sig0" "p.symbol" \$ show.ifc.classical : "statistic" "p.value" "p.symbol" \$ show.term.relevance : "df" "R2.x" "coefRlp" "coefRls" ... \$ show.term.test : "df" "ciLow" "ciUp" "R2.x" ... \$ show.term.classical : "statistic" "df" "ciLow" "ciUp" ... \$ show.term.eff.relevance : "coef" "coefRls.symbol" \$ show.term.eff.test : "coef" "p.symbol" \$ show.term.eff.classical : "coef" "p.symbol" \$ show.symbol.legend : TRUE \$ na.print : "." \$ p.symbols : List, see below \$ rlv.symbols : List, see below  
rlv.symbols List \$ symbol : " " "." "+" "++" "+++" \$ cutpoint: -Inf 0 1 2 5 Inf  
p.symbols List \$ symbol : "\*\*\*" "\*" "\*" "\*" "." " " \$ cutpoint: 0 0.001 0.01 0.05 0.1 1

### Examples

```
relevance.options
options(relevance.options) ## restores the package's default options
```

termeffects      *All Coefficients of a Model Fit*

### Description

A list of all coefficients of a model fit, possibly with respective statistics

### Usage

```
termeffects(object, se = 2, df = df.residual(object), rlv = TRUE)
```

**Arguments**

object	a model fit, produced, e.g., by a call to <code>lm</code> or <code>regr</code> .
se	logical: Should inference statistics be generated?
df	degrees of freedom for t-test
rlv	logical: Should relevances be calculated?

**Value**

a `list` with a component for each term in the model formula. Each component is a `termtable` for the coefficients corresponding to the term.

**Author(s)**

Werner A. Stahel

**See Also**

[dummy.coef](#), [termtable](#)

**Examples**

```
data(d.blast)
r.blast <-
  lm(log10(tremor)~location+log10(distance)+log10(charge), data=d.blast)
termeffects(r.blast)
```

---

termtable

*Statistics for Linear Models, Including Relevance Statistics*

---

**Description**

Calculate a table of statistics for (multiple) regression models with a linear predictor

**Usage**

```
termtable(object, summary = summary(object), testtype = NULL,
  r2x = TRUE, rlv = TRUE, rlv.threshold = getOption("rlv.threshold"),
  testlevel = getOption("testlevel"))
```

relevance.modelclasses

**Arguments**

object	result of a model fitting function like <code>lm</code>
summary	result of <code>summary(object)</code> . If NULL, the summary will be called.
testtype	type of test to be applied for dropping each term in turn. If NULL, it is selected according to the class of the object, see Details.
r2x	logical: should the collinearity measures “R2.x” (see below) for the terms be calculated?
rlv	logical: Should relevances be calculated?
rlv.threshold	Relevance thresholds, vector containing the elements <code>rel</code> : threshold for relative effects, <code>coef</code> : for standardized coefficients, <code>drop</code> : for drop effects, <code>pred</code> : for prediction intervals.
testlevel	1 - confidence level

**Details**

`relevance.modelclasses` collects the names of classes of model fitting results that can be handled by `termtable`.

If `testtype` is not specified, it is determined by the class of `object` and its attribute family as follows:

- For `t` for objects of class `lm`, `lmrob` and `glm` with families `quasibinomial` and `quasipoisson`,
- `Chi-squared` for other `glms` and `survreg`

**Value**

`data.frame` with columns

`coef`: coefficients for terms with a single degree of freedom

`df`: degrees of freedom

`se`: standard error of `coef`

`statistic`: value of the test statistic

`p.value`, `p.symbol`: p value and symbol for it

`Sig0`: significance value for the test of `coef==0`

`ciLow`, `ciUp`: confidence interval for `coef`

`stcoef`: standardized coefficient (standardized using the standard deviation of the 'error' term, `sigma`, instead of the response's standard deviation)

`st.Low`, `st.Up`: confidence interval for `stcoef`

`R2.x`: collinearity measure ( $= 1 - 1/vif$ , where `vif` is the variance inflation factor)

`coefRle`: estimated relevance of `coef`

`coefRls`: secured relevance, lower end of confidence interval for the relevance of `coef`

coefRlp: potential relevance, the upper end of the confidence interval.  
 dropRle, dropRls, dropRlp: analogous values for drop effect  
 predRle, predRls, predRlp: analogous values for prediction effect

In addition, it has attributes

testtype: as determined by the argument testtype or the class and attributes of object.  
 fitclass: class and attributes of object.  
 family, dist: more specifications if applicable

### Author(s)

Werner A. Stahel

### References

Werner A. Stahel (2020). Measuring Significance and Relevance instead of p-values. Submitted

### See Also

[getcoeftable](#); for printing options, [print.inference](#)

### Examples

```
data(swiss)
rr <- lm(Fertility ~ . , data = swiss)
rt <- termtable(rr)
rt
```

---

twosamples

*Relevance and Significance for One or Two Samples*

---

### Description

Inference for a difference between two independent samples or for a single sample: Collect quantities for inference, including Relevance and Significance measures

### Usage

```
twosamples(x, ...)
onesample(x, ...)

## Default S3 method:
twosamples(x, y = NULL, paired = FALSE,
  hypothesis = 0, var.equal = TRUE,
  testlevel=getOption("testlevel"), rlv.threshold=getOption("rlv.threshold"), ...)
## S3 method for class 'formula'
twosamples(formula, data, subset, na.action, ...)
```



**Arguments**

<code>x</code>	a formula or the data for the first or the single sample
<code>y</code>	data for the second sample
<code>paired</code>	logical: In case <code>x</code> and <code>y</code> are given. are their values paired?
<code>hypothesis</code>	the null effect to be tested, and anchor for the relevance
<code>var.equal</code>	logical: In case of two samples, should the variances be assumed equal? Only applies for quantitative data.
<code>testlevel</code>	level for the test, also determining the confidence level
<code>rlv.threshold</code>	Relevance threshold, or a vector of thresholds from which the element <code>stand</code> is taken for quantitative data and the element <code>prop</code> , for binary data. For the formula method:
<code>formula, data</code>	formula and data for the formula method
<code>subset, na.action</code>	subset and <code>na.action</code> to be applied to data
<code>...</code>	further arguments. For the formula method, this includes the arguments <code>var.equal</code> , <code>testlevel</code> and <code>coderlv.threshold</code> of the default method.

**Value**

an object of class 'effecttable', a `data.frame` with columns (components)

`effect`: for quantitative data: estimated difference between expectations of the two samples, or mean in case of a single sample.

For binary data: log odds (for one sample or paired samples) or log odds ratio (for two samples)

`ciLow, ciUp`: confidence interval for the effect

`Rle, Rls, Rlp`: relevance measures: estimated, secured, potential

`Sig0`: significance measure for test or 0 effect

`Sigth`: significance measure for test of effect == relevance threshold

`p.value`: p value for test against 0

In addition to the columns/components, it has `attributes`

`method`: problem and inference method

`effectname`: label for the effect

`hypothesis`: the null effect

`n`: number(s) of observations

`means`: in the case of 2 independent samples: their means

`statistic`: test statistic

`V`: single observation variance

`df`: degrees of freedom for the t distribution

`rlv.threshold`: relevance threshold

**Note**

onesample and twosamples are identical.

**Author(s)**

Werner A. Stahel

**References**

see those in [relevance-package](#).

**See Also**

[t.test](#), [binom.test](#)

**Examples**

```
t.test(sleep[sleep$group == 1, "extra"], sleep[sleep$group == 2, "extra"])
twosamples(sleep[sleep$group == 1, "extra"], sleep[sleep$group == 2, "extra"])

## Two-sample test, wilcox.test example, Hollander & Wolfe (1973), 69f.
## Permeability constants of the human chorioamnion (a placental membrane)
## at term and between 12 to 26 weeks gestational age
d.permeabililty <-
  data.frame(perm = c(0.80, 0.83, 1.89, 1.04, 1.45, 1.38, 1.91, 1.64, 0.73, 1.46,
                    1.15, 0.88, 0.90, 0.74, 1.21), atterm = rep(1:0, c(10,5))
  )
t.test(perm~atterm, data=d.permeabililty)
twosamples(perm~atterm, data=d.permeabililty)

## one sample
onesample(sleep[sleep$group == 2, "extra"])
```

# Index

- \* **datasets**
    - d.blast, [5](#)
    - ovarian, [9](#)
  - \* **distribution**
    - confintF, [4](#)
  - \* **hplot**
    - plot.inference, [10](#)
  - \* **htest**
    - twosamples, [16](#)
  - \* **package**
    - relevance-package, [2](#)
  - \* **regression**
    - drop1Wald, [6](#)
    - inference, [8](#)
    - print.inference, [11](#)
    - termeffects, [13](#)
    - termtable, [14](#)
  - \* **relevance**
    - inference, [8](#)
    - plot.inference, [10](#)
    - print.inference, [11](#)
    - relevance-package, [2](#)
    - termeffects, [13](#)
    - termtable, [14](#)
    - twosamples, [16](#)
  - \* **univar**
    - confintF, [4](#)
  - \* **utilities**
    - getcoeftable, [7](#)
    - relevance.options, [13](#)
  - \* **utility**
    - print.inference, [11](#)
- attributes, [17](#)
- binom.test, [18](#)
- class, [17](#)
- confintF, [4](#)
- d.blast, [5](#)
- data.frame, [15](#), [17](#)
- drop1, [6](#), [7](#)
- drop1Wald, [6](#)
- dummy.coef, [14](#)
- getcoefficient (getcoeftable), [7](#)
- getcoeftable, [7](#), [16](#)
- getscalepar (getcoeftable), [7](#)
- inference, [8](#), [12](#)
- list, [14](#)
- lm, [15](#)
- onesample (twosamples), [16](#)
- ovarian, [9](#)
- p.symbols (relevance.options), [13](#)
- plot.inference, [10](#)
- plot.termeffects (plot.inference), [10](#)
- print.data.frame, [12](#)
- print.inference, [11](#), [16](#)
- print.printInference (print.inference), [11](#)
- print.termeffects (print.inference), [11](#)
- qf, [5](#)
- relevance (relevance-package), [2](#)
- relevance-package, [2](#)
- relevance.modelclasses (termtable), [14](#)
- relevance.options, [13](#)
- rlv.symbols (relevance.options), [13](#)
- t.test, [18](#)
- termeffects, [12](#), [13](#)
- termtable, [12](#), [14](#), [14](#)
- twosamples, [12](#), [16](#)