Package `relevance` for calculating Relevance and Significance Measures

Werner A. Stahel, ETH Zurich

September 29, 2022

Abstract

Relevance and significance measures are characteristics of statistical results that lead to an informative inference. The relevance measure is based on the specification of a threshold of relevance and indicates whether a result is to be called (scientifically) relevant, negligible, or ambiguous.

The package `relevance` calculates these measures for a simple comparison of two samples as well as for many regression models and provides a suitable prinitng method.

1 Introduction to the relevance measure

This package implements the concepts of relevance and significance as introduced by Stahel (2021). They allow for meaningful statistical inference beyond the questionable common practice of Null Hypothesis Significance Testing that is in turn often reduced to citing a p-value.

The problem. Consider the problem of estimating an *effect*, for example, a mean (an expected value), a difference of means between two samples, or a regression coefficient.

The Zero Hypothesis Testing Paradox. In common practice, statistical inference is reduced to testing whether the effect might be zero, and the respective p-value is provided as the result. This has been widely criticized as being too simple an answer. In fact, it relates to a question that is not scientifically meaningful as seen by the "Zero Hypothesis Testing Paradox": When a study is undertaken to find a difference between samples or some influence between variables, the *true* effect—e.g., the difference between the expected values of two samples—will never be precisely zero. Therefore, the strawman hypothesis of zero true effect could in almost all reasonable applications be rejected if one had the patience and resources to obtain enough observations. Thus, the question that is answered mutates to: "Did we produce sufficiently many observations to prove the (alternative) hypothesis that was true on an apriori basis?" This does not seem to be a fascinating task.

Relevance. The scientifically meaningful question is whether the effect is *relevant*, and this needs the specification of a *relevance threshold* ζ. The *relevance measure* is defined as the ratio of the effect $\widehat{\vartheta}$ and the threshold,

$$\mathrm{Rl} = \vartheta/\zeta \; .$$

It is thus a parameter of the model. It is estimated by plugging in the estimated effect, $\widehat{\vartheta}$,

$$\mathrm{Rle} = \widehat{\vartheta}/\zeta \ ,$$

and a confidence interval is obtained in the same manner from the confidence interval for the effect parameter. Its limits are called

Rls, "secured relevance": the lower end;
Rlp, "potential relevance": the upper end.

Significance. Let us return to the problem of testing a null hypothesis, and even to the case of testing $\vartheta = 0$. The common way to express the result is to provide the p-value. However, this measure is more difficult to interpret than needed. We have been trained to compare it to the "level" of 5% and celebrate if it is *below.* It is thus a measure of lack of significance, and the desired range is just $0 \leq p \leq 0.05$. We also developed the skill of judging the values in this range as to "how significant" the result is.

In "ancient" times, before the computer produced p-values readily, statisticians examined the test statistics and then compared them to corresponding "critical values." In the widespread case that the t test was concerned, they used the t statistic as an informal quantitative measure of significance of an effect by comparing it to the number 2, which is approximately the critical value for moderate to large numbers of degrees of freedom.

The significance measure Sig0 picks up this idea, but standardizes with the actual critical value,

$$\mathrm{Sig0} = \widehat{\vartheta}\big/(q\ \mathrm{se}) \ ,$$

where se is the standard error of $\widehat{\vartheta}$ and q is the appropriate quantile. Then, the test rejects the null hypothesis $\vartheta = 0$ whenever $|\mathrm{Sig0}| > 1$, and Sig0 is proportional to the estimated effect. It is thus interpretable in a quantitative way as a measure of significance without special training.

Regression models. In regression, there are different ways to characterize the relevance of the individual terms of the model. Firstly, for scalar predictors, the coefficient is the obvious effect to examine. An alternative is the effect of dropping the predictor from the model, which also reflects its collinearity with the other predictors and generalizes to the case where the predictor is a factor (or another term with more than one degree of freedom), thus also encompassing *analysis of variance.* A third aspect is the relevance of the term for prediction of the target variable. For details, see Stahel (2021).

Choice of Relevance Thresholds. As noted above, the new relevance measure presupposes the choice of a relevance threshold. Ideally, this threshold is determined for each scientific question on the basis of specific knowledge about the phenomenon that is modeled. Since this is a severe burden, Stahel (2021) proposes some conventions for most common statistical models that may be used as a standard, like the testing level of 5% is for classical null hypothesis testing. (Note that the latter choice also affects the relevance measures Rls and Rlp.)

The convention includes, as a first step, to determine an appropriate "effect scale" for the model at hand, and then setting a relevance threshold for it. Table 1, taken from Stahel (2021) collects the proposed effect sizes and thresholds. The symbol $\%\ell$ indicates that the threshold refers to a log scale. For small effects on the log scale, these transform to the respective percentages in the original scale.

Table 1: Models, recommended effect scales and relevance thresholds

Problem	Rl tyoe	Basic model	Effect $\vartheta = g(\theta)$	Rel. thresh. ζ
One, or two paired samples	stand	$\mathcal{N}(\mu, \sigma^2)$	μ/σ	$10\,\%$
Two independent samples	stand	$\mathcal{N}(\mu_k, \sigma^2)$	$d = (\mu_1 - \mu_0)/\sigma$ $\vartheta = d/2$	$20\,\%$ $10\,\%$
Regression		$Y_i = \alpha + \underline{x}_i^\top \underline{\beta} + \varepsilon_i$ $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$		
coefficient effect	coef		$\beta_j \delta_j / \sigma$	$10\,\%$
drop effect	drop		η_J	$10\,\%$
prediction effect	pred		$-\frac{1}{2}\log(1 - R^2)$	$0.5\,\%\ell$ or $5\,\%\ell$
Relative Difference	rel	$\log(Y) \sim \mathcal{N}(\mu_k, \sigma^2)$	$\log(\mu_1/\mu_0)$	$10\,\%\ell$
Proportion	prop	$\mathcal{B}(n, p)$	$\log(p/(1-p))$	$10\,\%\ell$
Logistic regression	prop	$\text{logit}(P(Y_i = 1)) = \alpha + \underline{x}_i^\top \underline{\beta}$	$\underline{\beta}_j s_j$	$10\,\%\ell$
Correlation	corr	$\underline{Y} \sim \mathcal{N}_2(\underline{\mu}, \mathbf{\Sigma})$ $\rho = \mathbf{\Sigma}_{12}/\sqrt{\mathbf{\Sigma}_{11}\mathbf{\Sigma}_{22}}$	$\frac{1}{2}\log\left(\frac{1+\rho}{1-\rho}\right)$	$10\,\%\ell$

In the package, the thresholds used by default are given by

```
> getOption("rlv.threshold")

stand   rel  prop  corr  coef  drop  pred
 0.10  0.10  0.10  0.10  0.10  0.10  0.05
```

and can be modified by setting these options again, see below.

A classification of results. Based on the relevance measure and its confidence interval – or on the confidence interval for the original parameter and its position relative to the threshold – the result can be classified into the following cases:

- `"Rlv"` , if the effect is statistically proven to be larger than the threshold, that is, if the confidence interval for the parameter lies above the threshold, implying $\widehat{\vartheta} > \zeta$ or Rls > 1

- `"Amb"` if the confidence interval contains the threshold and thus Rls $< 1 \leq$ Rlp,
- `"Ngl"` if the interval only covers values lower than the threshold, but contains `0`, $0 \leq$ Rlp < 1, and
- `"Ctr"` if the interval only contains negative values, Rlp < 0.

2 Replication

Replication is a basic concept of science. Any empirical result should be tested by repeating the study in an independent situation and examining if the results lead to the same conclusion again. The independent repetition is called `replication` and provides new data that can be similary to the data of the `original study` or quite different.

The section describes how to applied the relevance measure to the problem of comparing the replication with the original. It can be skipped if this problem is not in the focus.

The conclusion that the replication has been successful, contradicting the original, or the result is ambugious is based on two aspects:

A. The conclusions from the replication results may be the same as those obtained in the original study.

B. The data from the replication may be "compatible" with those of the original study.

The first criterion relies on a statistical inference from the replication data, whereas the second is based on an inference about the difference between the two studies. The concept of relevance is essential for both of these aspects. Stahel (2022) describes the approach in detail.

In essence, the conclusion of the comparison, based on both aspects, can again be expressed by a classification. Let IEff_1 be the confidence interval in the replication and IEDS, the confidence interval for the difference. Then, the result, assuming a positive original effect, is a

(Cnf) *Confirmation,* if IEff_1 only contains relevant values (case Rlv), and the negative standardized effect difference EDS is small (cases Ngl or Amb); if IEff_1 is only significant (Amb.Sig) and the estimate $\widehat{\vartheta}_1$ is larger than the relevance threshold, we call it a *weak confirmation* (CnfW),

(Att) *Attenuation,* if IEff_1 lies on the same side of 0 as in the original study (Rlv or Amb.Sig) and IEDS is relevant (Rlv),

(Enh) *Enhancement,* if the replication suggests a clearly stronger effect, that is, case (Rlv) for IEff_1 and significantly positive EDS (Ctr); this will be rare, (Amb) *Ambiguous,* if IEff$_1$ covers the relevance threshold and it also covers zero (Amb) or the estimate $\widehat{\vartheta}_1$ is below the reference threshold,

(Anh) *Annihilation,* if IEff$_1$ covers only irrelevant values (Ngl),

(Ctr) *Contradiction,* if all values of IEff$_1$ have the opposite sign (Ctr),

(Drp) *Dropout,* if the replication failed to mimik the experimental or observational setup.

These cases are collected in Table ??.

Effect estimate	Effect Difference (standardized), IEDS		
IEff$_1$ in replication	relevant, Rlv	Amb or Ngl	contradicting, Ctr
relevant, Rlv	attenuation, Att	confirmation, Cnf	enhancement, Enh
significant, Sig	attenuation, Att	weak conf., CnfW*	—
ambiguous, Amb	ambiguous, Amb	ambiguous, Amb	—
negligible, Ngl	annihilation, Anh	annihilation, Anh**	—
contradicting, Ctr	contradiction, Ctr	—	—

Table 2: Classification of results of a replication of a relevant effect, based on the classificaion of the confidence interval IEff$_1$ for the effect in the replication and the confidence interval IEDS of the EDS. It is assumed that the original effect was relevant or at least significant. Then, the cases marked — cannot occur. * This conclusion also requires Rle $\geq$ 1; otherwise, it counts as ambiguous. ** This cannot occur if the original effect was relevant.

3 Functions

3.1 Function `twosamples`

(`onesample` is a synonym.) This function provides inference for the comparison of two samples, paired or unpaired, and also for a single sample. Its call mimics `t.test`.

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

        Welch Two Sample t-test

data:  sleep[sleep$group == 1, "extra"] and sleep[sleep$group == 2, "extra"]
t = -2, df = 18, p-value = 0.08
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.365  0.205
sample estimates:
mean of x mean of y
     0.75      2.33
```

```
> ( r.sleep <-
+     twosamples(sleep[sleep$group == 1, "extra"], sleep[sleep$group == 2, "extra"])
+ )

Two Sample t inference, equal variances assumed

difference of means:  1.58 ;  confidence int.: [ -0.204,  3.364 ]
Rle:  4.161 ;  Rlp:  8.859 ;  Rls:  -0.537   ;  Rlv.class:  Amb

Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
 Relevance threshold:  stand = 0.1
```

The output shows the estimated effect and its confidence interval together with the relevance measures. The estimated relevance Rle compares the standardized effect $\overline{X}/S$ =1.58/3.60238809680467, where S is the estimated standard deviation of the observations, to its relevance threshold 0.1. The classical results, t test statistic, standard error and p value are also calculated, but not shown with the default printing options. They can also be obtained, as well as the significance `Sig0`, by changing options (for details, see Section 4),

```
> t.oldopt <- options(show.inference = "classical")
> r.sleep

Two Sample t inference, equal variances assumed

difference of means:  1.58 ;  confidence int.: [ -0.204,  3.364 ]
Test:     hypothesis: effect =  0
  teststatistic:  1.861 ;  p value:  0.0792 .

Significance codes for p.value:   0  ***  0.001  **  0.01  *  0.05  .  0.1     1

> options(t.oldopt)  ##  restore the old options
```

The function also calculates inference about the mean of a single sample. It accepts the formula version of arguments: The statement `twosamples(extra group, data=sleep)` yields the same results as the more complicated call above. It also compares two samples of binary data, resulting in inference based on Fisher's test. A single sample leads to binomial inference. See the Examples section below.

3.2 Function `correlation`

Inference about a correlation coefficient is produced by the function `correlation`. It is based on `cor.test` from the `stats` package and thus allows for choosing Spearman's nonparametric correlation.

```
> correlation(iris[1:50,1:2], method="spearman")
```

```
Correlation -- Spearman's rank c.
estimates:
0.755
 ;  confidence int.: [ 0.699, 1.271 ]
Rle:  9.853 ;  Rlp:  12.712 ;  Rls:  6.994 +++ ;  Rlv.class:

Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
 Relevance threshold:  corr = 0.1
```

3.3 Function `termtable`

For regression models with a linear predictor, the basic function is `termtable`, which is applied to a model fit object. For each term reflecting a scalar predictor, its result contains the ordinary and standardized coefficient, their confidence intervals, significance against 0, p-value, and relevances. For all types of terms, with one or more degrees of freedom, it adds the relevances for dropping the term and for its contribution to prediction.

Since this leads to 22 columns, the print method selects columns according to `getOption("show.inference")`.

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

```
lm :  Drop-term inference
data:  swiss ;  target variable:  Fertility
                   coef df   R2x coefRlp coefRls dropRls..sy  predRle
(Intercept)      66.915  1    .    12.76    6.53       .            .
Agriculture      -0.172  1 0.562    6.89    0.66    0.50 .       1.12
Examination      -0.258  1 0.728    4.68   -1.55    0.00         0.01
Education        -0.871  1 0.640   10.46    4.23    3.76 ++      4.16
Catholic          0.104  1 0.484    7.67    1.44    1.30 +       1.69
Infant.Mortality  1.077  1 0.097    7.47    1.24    1.11 +       1.53
Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
 Relevance thresholds:  coef = 0.1, drop = 0.1, pred = 0.05
```

```
>   names(rt) ## The result of termtable has 22 columns
```

```
 [1] "coef"          "df"            "se"            "teststatistic"
 [5] "p.value"       "Sig0"          "ciLow"         "ciUp"
 [9] "stcoef"        "stciLow"       "stciUp"        "R2x"
[13] "coefRle"       "coefRls"       "coefRlp"       "dropRle"
```

```
[17] "dropRls"       "dropRlp"       "predRle"       "predRls"
[21] "predRlp"       "effect"        "effLow"        "effUp"
```

```
>   if(interactive()) { ## too much avoidable output for the vignette
+     str(rt)
+     data.frame(rt) ## or  print(rt, show="all")
+     ## This avoids selection and preparation of columns by 'print.inference'.
+   }
```

```
Classes ‘termtable’ and 'data.frame':        6 obs. of  24 variables:
 $ coef         : num  66.915 -0.172 -0.258 -0.871 0.104 ...
 $ df           : num  1 1 1 1 1 1
 $ se           : num  10.706 0.0703 0.2539 0.183 0.0353 ...
 $ teststatistic: num  6.25 -2.45 -1.02 -4.76 2.95 ...
 $ p.value      : num  NA 1.87e-02 3.15e-01 2.43e-05 5.19e-03 ...
 $ Sig0         : num  NA -1.212 -0.503 -2.356 1.462 ...
 $ ciLow        : num  45.2939 -0.3141 -0.7707 -1.2406 0.0329 ...
 $ ciUp         : num  88.5365 -0.0301 0.2547 -0.5013 0.1753 ...
 $ stcoef       : logi  NA NA NA NA NA NA
 $ stciLow      : logi  NA NA NA NA NA NA
 $ stciUp       : logi  NA NA NA NA NA NA
 $ R2x          : num  NA 0.562 0.728 0.64 0.484 ...
 $ coefRle      : num  9.64 3.78 1.57 7.34 4.56 ...
 $ coefRls      : num  6.528 0.661 -1.548 4.226 1.44 ...
 $ coefRlp      : num  12.76 6.89 4.68 10.46 7.67 ...
 $ dropRle      : num  NA 3.65 1.51 7.09 4.4 ...
 $ dropRls      : num  NA 0.497 0 3.761 1.302 ...
 $ dropRlp      : num  NA 6.65 4.44 10.36 7.45 ...
 $ predRle      : num  NA 1.12338 0.00781 4.15624 1.68738 ...
 $ predRls      : num  NA 0 0 1.2 0 ...
 $ predRlp      : num  NA 3.72 1.72 7.54 4.52 ...
 $ effect       : num  0.964 -0.378 -0.157 -0.734 0.456 ...
 $ effLow       : num  0.653 -0.689 -0.468 -1.046 0.144 ...
 $ effUp        : num  1.2761 -0.0661 0.1548 -0.4226 0.7673 ...
 - attr(*, "type")= chr "terms"
 - attr(*, "testtype")= chr "F"
 - attr(*, "method")= chr "lm :  Drop-term inference"
 - attr(*, "fitclass")= chr "lm"
 - attr(*, "formula")=Class 'formula'  language Fertility ~ Agriculture + Examination + Educat
  .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
 - attr(*, "data.name")= chr "swiss"
 - attr(*, "rlv.threshold")= Named num [1:3] 0.1 0.1 0.05
  ..- attr(*, "names")= chr [1:3] "coef" "drop" "pred"
                   coef df      se teststatistic  p.value   Sig0   ciLow
(Intercept)      66.915  1 10.7060          6.25       NA     NA 45.2939
Agriculture      -0.172  1  0.0703         -2.45 1.87e-02 -1.212 -0.3141
Examination      -0.258  1  0.2539         -1.02 3.15e-01 -0.503 -0.7707
Education        -0.871  1  0.1830         -4.76 2.43e-05 -2.356 -1.2406
Catholic          0.104  1  0.0353          2.95 5.19e-03  1.462  0.0329
Infant.Mortality  1.077  1  0.3817          2.82 7.34e-03  1.397  0.3061
                    ciUp stcoef stciLow stciUp    R2x coefRle coefRls coefRlp
(Intercept)      88.5365     NA      NA     NA     NA    9.64   6.528   12.76
Agriculture      -0.0301     NA      NA     NA 0.5622    3.78   0.661    6.89
Examination       0.2547     NA      NA     NA 0.7279    1.57  -1.548    4.68
Education        -0.5013     NA      NA     NA 0.6396    7.34   4.226   10.46
Catholic          0.1753     NA      NA     NA 0.4838    4.56   1.440    7.67
Infant.Mortality  1.8479     NA      NA     NA 0.0971    4.35   1.238    7.47
                 dropRle dropRls dropRlp predRle predRls predRlp effect effLow
(Intercept)           NA      NA      NA      NA      NA      NA  0.964  0.653
Agriculture         3.65   0.497    6.65 1.12338     0.0    3.72 -0.378 -0.689
Examination         1.51   0.000    4.44 0.00781     0.0    1.72 -0.157 -0.468
Education           7.09   3.761   10.36 4.15624     1.2    7.54 -0.734 -1.046
Catholic            4.40   1.302    7.45 1.68738     0.0    4.52  0.456  0.144
Infant.Mortality    4.21   1.114    7.24 1.53360     0.0    4.31  0.435  0.124
                   effUp
(Intercept)       1.2761
Agriculture      -0.0661
Examination       0.1548
Education        -0.4226
Catholic          0.7673
Infant.Mortality  0.7470
```

Again, other results can be selected using options.

```
>   t.oldopt <- options(show.inference = "classical")
>   rt
```

```
lm :  Drop-term inference
data:  swiss ;  target variable:  Fertility
                   coef df      se teststatistic p.value..sy
(Intercept)      66.915  1 10.7060          6.25           .
```

```
Agriculture      -0.172  1  0.0703         -2.45   0.019 *
Examination      -0.258  1  0.2539         -1.02   0.315
Education        -0.871  1  0.1830         -4.76   0.000 ***
Catholic          0.104  1  0.0353          2.95   0.005 **
Infant.Mortality  1.077  1  0.3817          2.82   0.007 **
Significance codes for p.value:   0  ***  0.001  **  0.01  *  0.05  .  0.1     1
```

```
>   options(t.oldopt)  ##  restore the old options
```

Plot. `inference` objects relate to a specific plotting method that shows the confidence interval(s) on the relevance scale. Here is the example.

```
> plot(rt)
```

3.4 Function `termeffects`

For terms with more than one degree of freedom, notably for factors with more than two levels, the function `termeffects` calculates effects of levels and respective inference measures. As seen here, there are `print` and `plot` methods for the resulting objects.

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

```
lm :  Term effects


$ location
       1             2             3             4             5
 0.00000       0.15306 +     0.13169 +    -0.16185 +    -0.03211
       6             7             8
 0.07161 .    -0.00889       0.00372
Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
```

```
>   plot(termeffects(r.blast))  ## plot effects for terms with >1 df
```

3.5 Function `inference`

This function generates statistics describing relevance and significance for several situations, mainly for regression models. When it is applied to a model fit object, it calls `termtable` and `termeffects` and stores the `summary` of the object. The corresponding printing method includes a final part that describes the global aspects of the model as shown here.

```
>   ( rr <- inference(r.blast) )

lm
data:  d.blast ;  target variable:  log10(tremor)
$ termtable
                  coef df   R2x coefRlp coefRls dropRls..sy  predRle
(Intercept)      2.964  1     .   15.27   13.18        .           .
location             .  7 0.102       .       .     5.49 +++     3.65
log10(distance) -1.518  1 0.477   13.63   11.53   11.42 +++     9.48
log10(charge)    0.636  1 0.102    9.62    7.53    7.46 +++     5.50

$ termeffects
$location
       1             2             3            4             5
 0.00000       0.15306 +     0.13169 +    -0.16185 +    -0.03211
       6             7             8
 0.07161 .    -0.00889       0.00372

Relevance codes:     -Inf      0  .  1  +  2  ++  5  +++  Inf
 Relevance thresholds:  coef = 0.1, drop = 0.1, pred = 0.05


 Multiple R^2:  0.795;  Adjusted R^2:  0.79
 F-statistic:   152   on 9 and 352 d.f.;  p.value:  1.82e-115
```

`inference` also applies to other situations where an estimate, its standard error and the number of observations is available.

3.6 Function `replication`

This function generates inference for a replication study based on its inference results for the original and replication studies and the inference on the difference between the two, as described in Section 2.

The function produces an object of class `inference` and `replication`. There are `print` and `plot` methods for this class.

Here is a series of examples taken from the classical set of replications reported by the “Open Science Collaboration” in 2015: Open Science Collaboration (2015). “Estimating the reproducibility of psychological science.” Science 349, 943-952. The 10 studies implying one sample or paired samples test were selected.

```
> data(d.osc15Onesample)
> tx <- structure(d.osc15Onesample[,c("effecto","teststatistico","no")],
+       names=c("effect","teststatistic","n"))
> ty <- structure(d.osc15Onesample[,c("effectr","teststatisticr","nr")],
+       names=c("effect","teststatistic","n"))
> ( rr <- replication(tx, ty, rlv.threshold=0.1) )
```

```
replication

     Rle   Rlp    Rls
1  -1.29  2.28  -4.85
2  -7.52 -2.55 -12.49
3   1.25  2.39   0.10
4  -2.91  3.05  -8.86
5  -2.19  0.52  -4.90
6  -0.79  1.86  -3.43
7  -0.75  1.90  -3.40
8  -0.51  1.57  -2.59
9  -0.32  1.11  -1.76
10  0.14  1.33  -1.04
Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
 Relevance threshold:   = 0.1
```

```
> plot(rr)
```

3.7 Generally useful functions

The package includes several functions that are not directly related to relevance or significance, see their help pages for details and examples.

showd allows for inspecting a data.frame or vector in a brief informative way.

```
>   showd(d.blast)
```

```
dim:  388 6
          date location device distance charge tremor
1   2002-07-19     loc5      1       44  0.952   2.93
98  2002-07-19     loc1      2       69  0.952   1.76
195 2002-07-19     loc6      3      108  0.952   0.62
...
219 2002-09-11     loc6      3       94  7.072   3.77
340 2002-10-17     loc3      4       49  5.824   6.59
364 2002-11-15     loc3      4       44  4.644   8.27
388 2002-02-06     loc3      4       77  3.952   3.95
```

`logst` is a version of a ”started log” that copes with zeros and even with negative values in a suitable way, and `asinp` implements an appropriate transformation for percentages.

Functions that apply to data with missing values (`NA` s) are `sumNA, dropNA, replaceNA` and `formatNA` .

4 Options

The package works with some specific options, see `?relevance.options` . The more important ones are the following.

- `rlv.threshold` : vector of relevance thresholds for
  - `rel` : a relative effect, that is, a change in a prameter expressed as a percentage of the parameter,
  - `stand` : an effect standardized by a standard deviation, like Cohen’s d for two samples,
  - `prop` : a proportion, expressed in logit units,
  - `corr` : a correlation coefficient,
  - `coef` : a coefficient in the linear predictor of a regression model,
  - `drop` : the effect of dropping a term from a regression model,
  - `pred` : the effect of a term on the prediction accuracy.
- `show.inference` : selects the inference items to be presented by the `print` methods. Currently, three styles are implemented:
  - `relevance` : selects the columns determined by `getOption("show.simple.relevance")` , `getOption("show.terms.relevance")` and `getOption("show.termeffects.relevance")` , for the three print methods (see below), repectively; these are the important columns for inference based on relevance;
  - `classical` and `test` : these select `getOption("show.?.classical")` , in the same manner, suitable for inference based on p values or significance, respectively.

  The choice of any elements of the vector resulting from a call of `towsamples` or any columns of a `termtable` object is achieved by typing, e.g.,
  `options(show.inference=c("classical","Sig0","Rls"))` .
- `rlv.symbols` and `p.symbols` : symbols to be used for characterizing `Rls` or p-values, respectively,
- `digits.reduced` : digits used for relevance and significance measures and test statistics. These numbers are rounded to `digits.reduced` decimals, `p-values` to one more.

- `na.print` : symbol to print `NA` values.

The package's defaults can always be restored by typing `options(relevance.options)`

Here is an example of choosing more detailed ouput for termeffects.

```
>   t.opt <- options(show.terms.relevance=c("coef", "dropRls", "dropRls.symbol"))
>   rt
```

```
lm :  Drop-term inference
data:  swiss ;  target variable:  Fertility
                   coef dropRls..sy
(Intercept)      66.915       .
Agriculture      -0.172    0.50 .
Examination      -0.258    0.00
Education        -0.871    3.76 ++
Catholic          0.104    1.30 +
Infant.Mortality  1.077    1.11 +
Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
 Relevance thresholds:  coef = 0.1, drop = 0.1, pred = 0.05
```

```
> ## restore the old options
>   options(t.opt) ## the former options
>   options(relevance.options) ## restore the package's defaults
```

4.1 Function `print`

These options are used when calling the `print` methods on the objects produced by the functions in Section 3. These objects are either of class `inference` or `termeffects`. The methods `print.inference` and `print.termeffects` accept an argument `show` that acts as if the corresponding printing options had been changed. Thus,
`print(rt, show=c("coef", "dropRls", "dropRls.symbol"))` leads to the output shown above.

The printing methods convert their first argument into printable form by producing an object of class `printInference`. They terminate by calling the method `print.printInference`, which in turn produces the output—unless `print=FALSE` is set. This two-step procedure allows for editing the output in the following manner:

```
> rpr <- print(termeffects(r.blast), print=FALSE)
> attr(rpr, "head") <- sub("lm", "Linear Regression", attr(rpr, "head"))
> rpr
```

```
Linear Regression :  Term effects
```

```
$ location
       1             2             3             4             5
 0.00000       0.15306 +     0.13169 +    -0.16185 +    -0.03211
       6             7             8
 0.07161 .    -0.00889       0.00372
Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
```

5 Examples

Here, we document the examples that appear in the basic reference.

5.1 sleep data

```
> data(sleep)
> dd <- subset(sleep, group==2)
> onesample(60*dd$extra, rlv.threshold=60, standardize=FALSE)
```

```
One Sample t inference
estimates:
mean  se  140    38
mean :  140 ;  confidence int.: [  53.9, 225.7 ]
Rle:  2.33 ;  Rlp:  3.762 ;  Rls:  0.898 . ;  Rlv.class:  Amb.Sig

Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
 Relevance threshold:   = 60
```

5.2 Anchoring

The experiment is described as follows: “Students were asked to guesstimate the height of Mount Everest. One group was ‘anchored’ by telling them that it was more than 2000 feet, the other group was told that it was less than 45,500 feet. The hypothesis was that respondents would be influenced by their ‘anchor,’ such that the first group would produce smaller numbers than the second”. The true height is 29,029 feet.

The inference about the difference between the groups and a graphical display are obtained as follows.

```
> data(d.everest)
> rr <- twosamples(log(y)~g, data=d.everest, var.equal=TRUE)
> print(rr, show="classical")
```

```
Two Sample t inference, equal variances assumed
target variable:  log(y)
difference of means:  2.15 ;  confidence int.: [ 1.7, 2.6 ]
```

```
Test:     hypothesis: effect =  0
  teststatistic:  9.961 ;  p value:  0 ***

Significance codes for p.value:   0  ***  0.001  **  0.01  *  0.05  .  0.1     1
```

```
> rr
```

```
Two Sample t inference, equal variances assumed
target variable:  log(y)
difference of means:  2.15 ;  confidence int.: [ 1.7, 2.6 ]
Rle:  21.509 ;  Rlp:  26.045 ;  Rls:  16.972 +++ ;  Rlv.class:  Rlv

Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
 Relevance threshold:  rel = 0.1
```

```
> pltwosamples(log(y)~g, data=d.everest)
```

5.3 Blasting

When digging a tunnel in a populated area, it is important to make sure that the blasting does not damage nearby buildings. To this end, the tremor caused by the blastings is measured in the basement of such houses, along with the distance and the charge used, and a model is used to predict the resulting tremor. The dataset `d.blast` contains such data for a freeway tunnel beneath a Swiss city. The logarithmic tremor is modelled as a linear function of the logarithmic distance and charge, an additive adjustment to the house where the measurements are taken (factor `location`). For the example in the paper, a subset is used, and `time`, a rescaled calendar day, is appended.

```
> dd <- d.blast[seq(1,388,3),]
> dd <- na.omit(dd[dd$location %in% paste("loc",c(1,2,4),sep=""),])
> dd$time <- as.numeric(dd$date-min(dd$date))/365
> rlm <- lm(log10(tremor)~location+log10(distance)+log10(charge)+time, data=dd,
+           contrasts=list(location="contr.sum"))
> ( rt <- termtable(rlm) )
```

```
lm :  Drop-term inference
data:  dd ;  target variable:  log10(tremor)
                   coef df   R2x coefRlp coefRls dropRls..sy  predRle
(Intercept)      3.7331  1     .   18.71   12.63        .           .
location              .  2 0.232       .       .     2.06 ++      2.41
log10(distance) -2.0391  1 0.219   17.72   11.64    10.02 +++    11.42
log10(charge)    0.7567  1 0.247   11.32    5.24     4.65 ++      5.08
time             0.0702  1 0.105    3.77   -2.32     0.00         0.00
```

```
Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
 Relevance thresholds:  coef = 0.1, drop = 0.1, pred = 0.05
```

```
> plot(rt)
```

References

Stahel, Werner A. (2021). *New relevance and significance measures to replace p-values.* PLOS ONE, June 16, 2021, doi.org/10.1371/journal.pone.0252991

Stahel, Werner A. (2022). *Replicability: Terminology, Measuring Success, and Strategy.* Available in the documentation.