Package `relevance` for calculating Relevance and Significance Measures

Werner A. Stahel, ETH Zurich

July 29, 2021

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

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.

2 Functions

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
##
## estimates:    mean   se
##               0.750  0.566
##               2.330  0.633
##
## difference of means:  1.58 ;  confidence int.: [ -0.204,  3.364 ]
## Rle:  4.161 ;  Rlp:  8.859 ;  Rls:  -0.537
##
##
## 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 $\overline{X}/S$ =1.58/3.602, 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 3),

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

## Two Sample t inference, equal variances assumed
##
## estimates:    mean   se
##               0.750  0.566
##               2.330  0.633
##
## difference of means:  1.58 ;  confidence int.: [ -0.204,  3.364 ]
## Test:     hypothesis: effect =  0
##   statistic:  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 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.

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

## Warning in names(rr)[lcl] <- paste(names(lx), "..sy ", sep = ""):  number of items to
replace is not a multiple of replacement length
```

```
## lm :  Drop-term inference
## data:  swiss ;  target variable:  Fertility
##
##                    coef df   R2x coefRlp coefRls   coef..sy  predRle
## (Intercept)      66.915  1  .       .       .    66.915           .
## Agriculture      -0.172  1 0.562    9.96    0.96 -0.172 .       1.12
## Examination      -0.258  1 0.728    8.58   -2.84 -0.258         0.01
## Education        -0.871  1 0.640   16.65    6.73 -0.871 ++      4.16
## Catholic          0.104  1 0.484   10.20    1.92  0.104 +       1.69
## Infant.Mortality  1.077  1 0.097    7.51    1.24  1.077 +       1.53
##
## Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
## Relevance thresholds:  coef = 0.1, drop = 0.1, pred = 0.05
```

```
  names(rt)
```

```
##  [1] "coef"      "df"        "se"        "statistic" "p.value"   "Sig0"
##  [7] "ciLow"     "ciUp"      "stcoef"    "stciLow"   "stciUp"    "testst"
## [13] "R2x"       "coefRle"   "coefRls"   "coefRlp"   "dropRle"   "dropRls"
## [19] "dropRlp"   "predRle"   "predRls"   "predRlp"
```

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

Again, other results can be selected using options.

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

```
## Warning in names(rr)[lcl] <- paste(names(lx), "..sy ", sep = ""):  number of items to
replace is not a multiple of replacement length
```

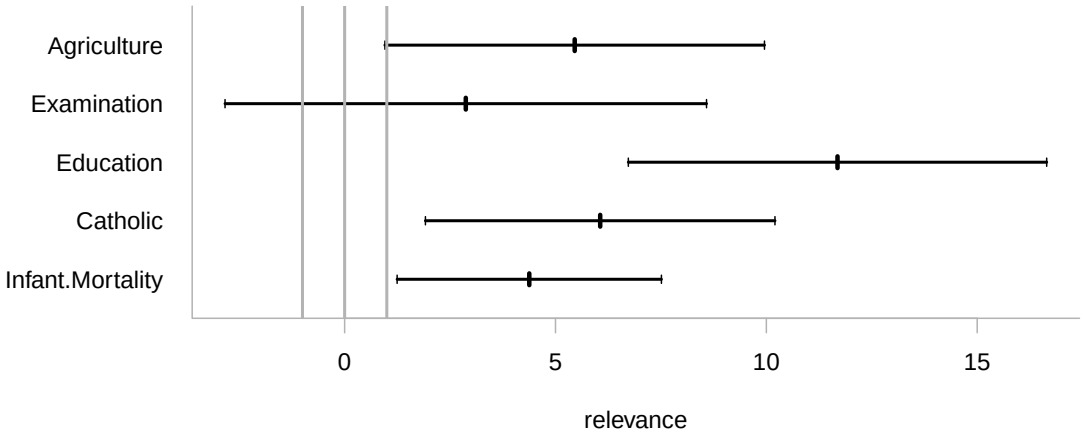
```
## lm :  Drop-term inference
## data:  swiss ;  target variable:  Fertility
##
```

```
##                    coef df      se statistic   coef..sy    ciLow    ciUp   R2x
## (Intercept)      66.915  1 10.7060      6.25 66.915      45.2939 88.5365  .
## Agriculture      -0.172  1  0.0703     -2.45 -0.172 *    -0.3141 -0.0301 0.562
## Examination      -0.258  1  0.2539     -1.02 -0.258      -0.7707  0.2547 0.728
## Education        -0.871  1  0.1830     -4.76 -0.871 ***  -1.2406 -0.5013 0.640
## Catholic          0.104  1  0.0353      2.95  0.104 **    0.0329  0.1753 0.484
## Infant.Mortality  1.077  1  0.3817      2.82  1.077 **    0.3061  1.8479 0.097
##                   Sig0
## (Intercept)       .
## Agriculture      -1.21
## Examination      -0.50
## Education        -2.36
## Catholic          1.46
## Infant.Mortality  1.40
##
## 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.

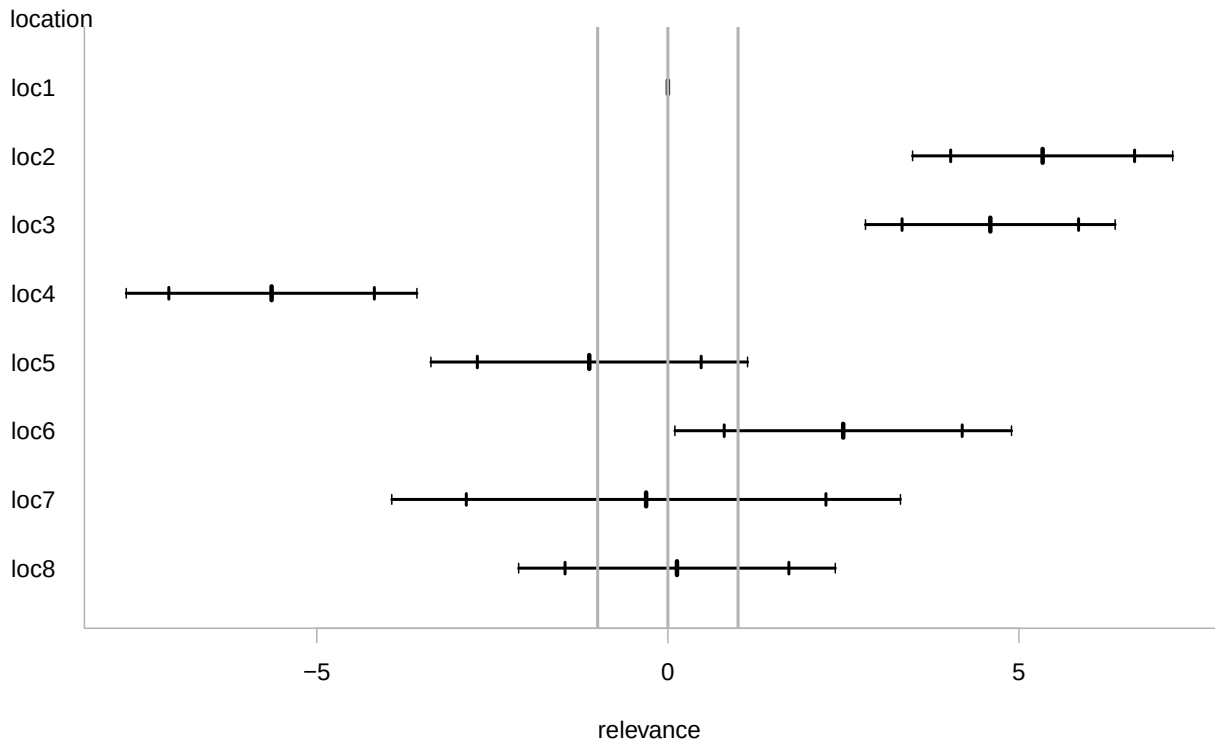
```
plot(rt)
```



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
##   0.00000      0.15306 ++   0.13169 ++  -0.16185 ++  -0.03211      0.07161 .   -0.00889
##
## Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf

print(rte, show=c("classical","coefRls","coefRls.symbol"), single=TRUE)

## Warning in names(rr)[lcl] <- paste(names(lx), "..sy ", sep = ""):  number of items to
replace is not a multiple of replacement length
## Warning in names(rr)[lcl] <- paste(names(lx), "..sy ", sep = ""):  number of items to
replace is not a multiple of replacement length
## Warning in names(rr)[lcl] <- paste(names(lx), "..sy ", sep = ""):  number of items to
replace is not a multiple of replacement length
## Warning in names(rr)[lcl] <- paste(names(lx), "..sy ", sep = ""):  number of items to
replace is not a multiple of replacement length
## Warning in names(rr)[lcl] <- paste(names(lx), "..sy ", sep = ""):  number of items to
replace is not a multiple of replacement length
## Warning in names(rr)[lcl] <- paste(names(lx), "..sy ", sep = ""):  number of items to
replace is not a multiple of replacement length

## lm :  Term effects
##
## $ (Intercept)
##     2.96
## $ location
```

```
##
## c("   .", " 3.49", " 2.81", " 3.57", "-1.13", " 0.10", "-3.31", "-2.13")      c("   .", " 3
##
## c("   .", " 3.49", " 2.81", " 3.57", "-1.13", " 0.10", "-3.31", "-2.13") ++   c("   .", " 3
##
## c("   .", " 3.49", " 2.81", " 3.57", "-1.13", " 0.10", "-3.31", "-2.13") ++   c("   .", " 3
##
## c("   .", " 3.49", " 2.81", " 3.57", "-1.13", " 0.10", "-3.31", "-2.13") ++   c("   .", " 3
##
## c("   .", " 3.49", " 2.81", " 3.57", "-1.13", " 0.10", "-3.31", "-2.13")      c("   .", " 3
##
## c("   .", " 3.49", " 2.81", " 3.57", "-1.13", " 0.10", "-3.31", "-2.13") .    c("   .", " 3
##
## c("   .", " 3.49", " 2.81", " 3.57", "-1.13", " 0.10", "-3.31", "-2.13")      c("   .", " 3
##
## c("   .", " 3.49", " 2.81", " 3.57", "-1.13", " 0.10", "-3.31", "-2.13")      c("   .", " 3
## $ log10(distance)
##     48.5 +++  48.5 ***
## $ log10(charge)
##     19.5 +++  19.5 ***
##
## Significance codes for p.value:   0  ***  0.001  **  0.01  *  0.05  .  0.1     1
## Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf

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



Function `inference.` This function generates statistics describing relevance and significance for many 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) )

## Warning in names(rr)[lcl] <- paste(names(lx), "..sy ", sep = ""):  number of items to
replace is not a multiple of replacement length

## lm
## data:  d.blast ;  target variable:  log10(tremor)
##
## $ termtable
##                  coef df   R2x coefRlp coefRls   coef..sy  predRle
## (Intercept)     2.964  1  .        .       .     2.964          .
```

```
## location          .      7 0.102     .      .          . +++     3.65
## log10(distance) -1.518  1 0.477    18.6   15.77 -1.518 +++     9.48
## log10(charge)    0.636  1 0.102    10.1    7.86  0.636 +++     5.50
##
##
## $ termeffects
## $location
## [1]  0.00000       0.15306 ++    0.13169 ++   -0.16185 ++   -0.03211
## [6]  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
##
## St.dev.error:  0.143  on  352  degrees of freedom
## 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 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
  - `stand`: an effect standardized by a standard deviation, like Cohen's d for two samples,
  - `rel`: a relative effect, that is, a change in a prameter expressed as a percentage of the parameter,
  - `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, test`: selects `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)`

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

## Warning in names(rr)[lcl] <- paste(names(lx), "..sy ", sep = ""): number of items to
replace is not a multiple of replacement length

## lm : Drop-term inference
## data: swiss ; target variable: Fertility
##
##      (Intercept)           Agriculture           Examination
##    66.915 66.915         -0.172 -0.172 .       -0.258 -0.258
##        Education              Catholic      Infant.Mortality
##    -0.871 -0.871 ++       0.104  0.104 +        1.077  1.077 +
##
## 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
```

Function `print.` These options are used when calling the `print` methods on the objects produced by the functions in Section 2. These objects are either of class `inference` or `termeffects`. The methods `print.inference` and `print.termeffects` convert such objects 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:

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

## Linear Regression :  Term effects
##
## $ location
##  0.00000      0.15306 ++   0.13169 ++  -0.16185 ++  -0.03211      0.07161 .   -0.00889
##
## Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
```

4 Examples

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

sleep data.

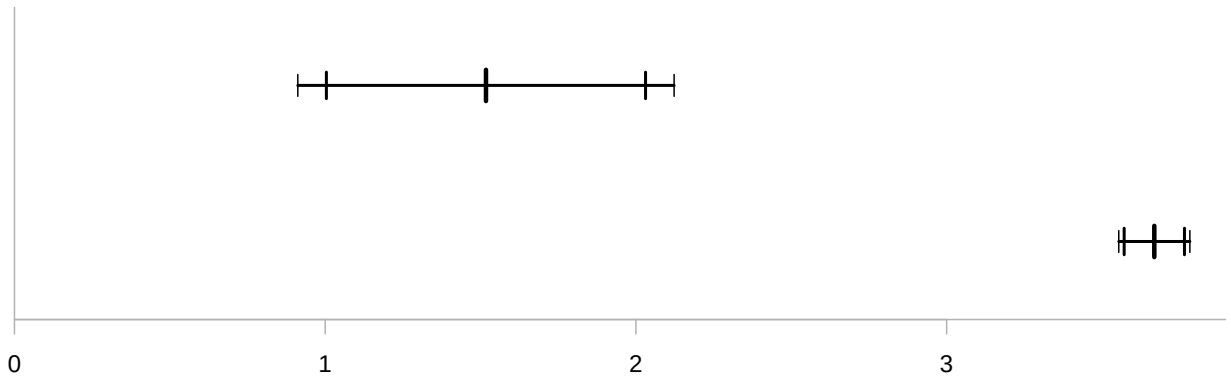
```
data(sleep)
dd <- subset(sleep, group==2)
onesample(60*dd$extra, rlv.threshold=60, standardize=FALSE)

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

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
## data:  d.everest ;  target variable:  log(y)
## estimates:    mean   se
##               1.517  0.256
##               3.668  0.052
##
## difference of means:  2.15 ;  confidence int.: [ 1.7, 2.6 ]
## Test:     hypothesis: effect =  0
##   statistic:  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
## data:  d.everest ;  target variable:  log(y)
## estimates:    mean   se
##               1.517  0.256
##               3.668  0.052
##
## difference of means:  2.15 ;  confidence int.: [ 1.7, 2.6 ]
## Rle:  22.273 ;  Rlp:  26.97 ;  Rls:  17.575 +++
##
##
## Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
## Relevance threshold:  stand = 0.1
```

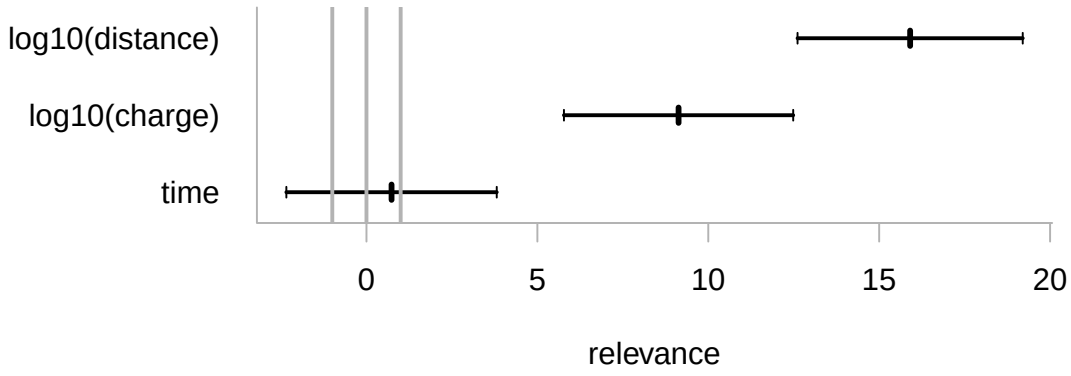
```
pltwosamples(log(y)~g, data=d.everest)
```



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) )

## Warning in names(rr)[lcl] <- paste(names(lx), "..sy ", sep = ""):  number of items to
replace is not a multiple of replacement length

## lm :  Drop-term inference
## data:  dd ;  target variable:  log10(tremor)
```

```
##
##                    coef df   R2x coefRlp coefRls    coef..sy  predRle
## (Intercept)      3.7331  1  .       .       .     3.7331         .
## location         .       2 0.232    .       .          . ++      2.41
## log10(distance) -2.0391  1 0.219   19.20   12.61 -2.0391 +++    11.42
## log10(charge)    0.7567  1 0.247   12.48    5.78  0.7567 ++      5.08
## time             0.0702  1 0.105    3.81   -2.34  0.0702         0.00
##
## Relevance codes:     -Inf     0  .  1  +  2  ++  5  +++  Inf
## Relevance thresholds:  coef = 0.1, drop = 0.1, pred = 0.05

plot(rt)
```



Reference

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