Package `relevance` for calculating Relevance and Significance Measures

Werner A. Stahel, ETH Zurich

March 31, 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 estimated effect $\widehat{\vartheta}$ and the threshold,

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

The confidence interval for the effect translates to the interval for the relevance with limits

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,

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

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 $|\text{Sig0}| > 1$, 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 standardized effect or "effect size" 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	Basic model	Effect $\vartheta = g(\theta)$	Rel. thresh. ζ
One, or two paired samples	$\mathcal{N}(\mu, \sigma^2)$	μ/σ	10 %
Two independent samples	$\mathcal{N}(\mu_k, \sigma^2)$	$d = (\mu_1 - \mu_0)/\sigma$ $\vartheta = (\mu_1 - \mu_0)\sqrt{\nu_0\nu_1}/\sigma$	20 % 10 %
Regression coefficients prediction error	$Y_i = \alpha + \underline{x}_i^\top \underline{\beta} + \varepsilon_i$ $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$	 $\underline{\beta}_j \sqrt{\mathrm{MSX}^{(j)}}/\sigma$ $-\frac{1}{2}\log(1 - R^2)$	 10 % 0.5 %ℓ or 5 %ℓ
Logistic regression	$g(Y_i = 1) = \alpha + \underline{x}_i^\top \underline{\beta}$	$\underline{\beta}_j 0.6\sqrt{\mathrm{MSX}^{(j)}}/\sqrt{\phi}$	10 %ℓ
Relative Difference	$\log(Y) \sim \mathcal{N}(\mu_k, \sigma^2)$	$\log(\mu_1/\mu_0)$	10 %ℓ
Proportion	$\mathcal{B}(n, p)$	$\log(p/(1-p))$	33 %ℓ or 10 %ℓ
Correlation	$\underline{Y} \sim \mathcal{N}_2(\underline{\mu}, \boldsymbol{\Sigma})$ $\rho = \boldsymbol{\Sigma}_{12}/\sqrt{\boldsymbol{\Sigma}_{11}\boldsymbol{\Sigma}_{22}}$	$\frac{1}{2}\log((1+\rho)/(1-\rho))$	10 %ℓ

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

```
getOption("rlv.threshold")

## stand   rel  prop  coef  drop  pred
##  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`. 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 = -1.8608, df = 17.776, p-value = 0.07939
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -3.3654832  0.2054832
## 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
##
## difference of means:  1.58 ;  confidence int.: [ -0.203874,  3.363874 ]
## Rle:  4.161 ;  Rlp:  8.859 ;  Rls:  -0.537
```

The output shows the estimated effect and its confidence interval together with the relevance measures. The estimated relevance Rle compares the standardized effect $\overline{X}/S$ =1.58/3.6023881, 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
##
## difference of means:  1.58 ;  confidence int.: [ -0.203874,  3.363874 ]
## Test:     hypothesis: effect =  0
##   statistic:  1.861 ;  p value:  0.0792 .
```

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

Function `termtable`. For regression models with a linear predictor, the basic function is `termtable`. 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")`.

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

```
## lm :  Drop-term effects
## data:  swiss ;  target variable:  Fertility
##
##                        coef df   R2x coefRlp coefRls dropRls dRsy predRle
## (Intercept)      66.9151817  1    NA      NA      NA      NA           NA
## Agriculture      -0.1721140  1 0.562    9.96    0.96    0.50 .       1.12
## Examination      -0.2580082  1 0.728    8.58   -2.84    0.00         0.01
## Education        -0.8709401  1 0.640   16.65    6.73    3.76 ++      4.16
## Catholic          0.1041153  1 0.484   10.20    1.92    1.30 +       1.69
## Infant.Mortality  1.0770481  1 0.097    7.51    1.24    1.11 +       1.53
##
## Relevance codes:     -Inf ' ' 0 '.' 1 '+' 2 '++' 5 '+++' Inf
```

```
  names(rt)
```

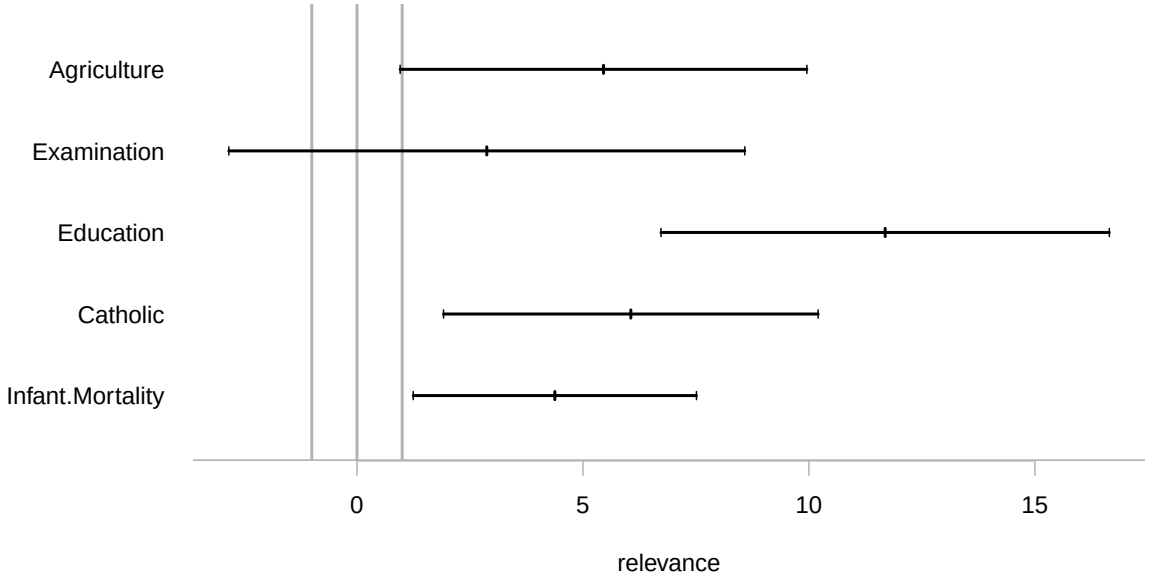
```
##  [1] "coef"      "df"        "se"        "statistic" "p.value"   "Sig0"
##  [7] "ciLow"     "ciUp"      "stcoef"    "stciLow"   "stciUp"    "testst"
## [13] "R2x"       "coefRle"   "coefRls"   "coefRlp"   "dropRle"   "dropRls"
## [19] "dropRlp"   "predRle"   "predRls"   "predRlp"
```

```
  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'. It produces an extensive output.
  }
```

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.

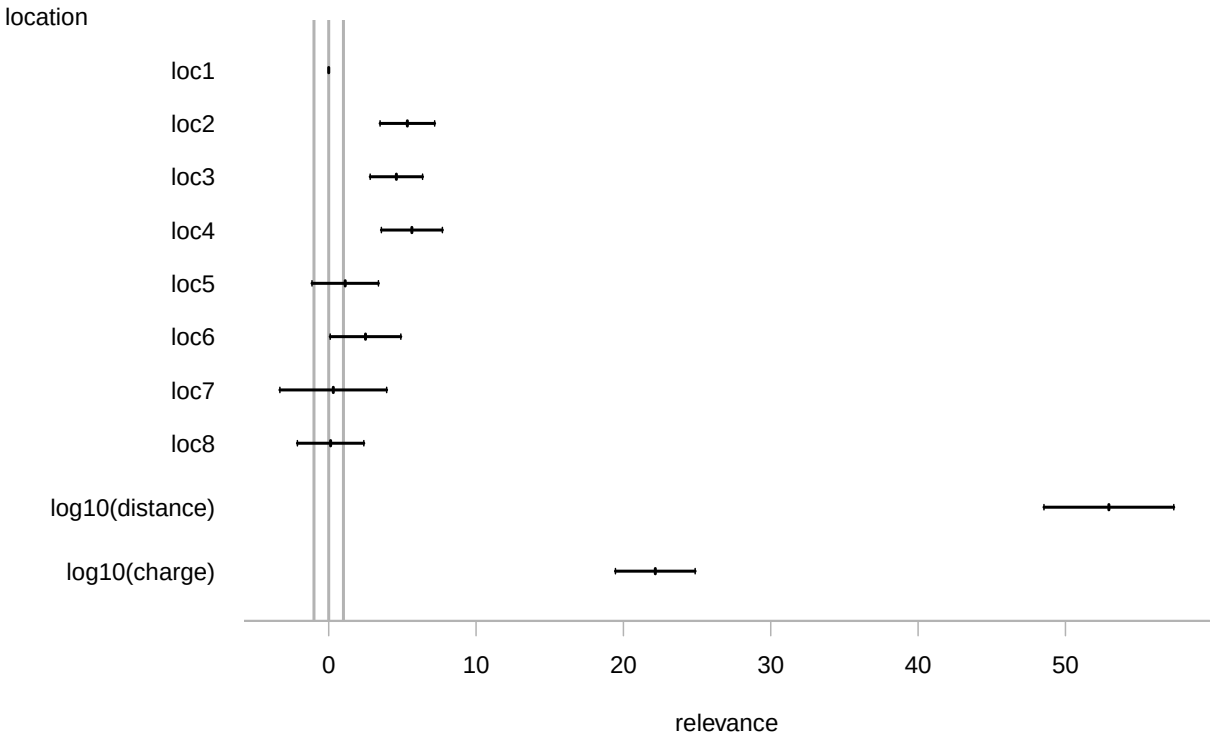
```
data(d.blast)
r.blast <-
  lm(log10(tremor)~location+log10(distance)+log10(charge), data=d.blast)
(rte <- termeffects(r.blast))

##
## lm :  Term effects
```

```
##
## $ location
##         loc1              loc2              loc3              loc4
##  0.000000000       0.153060685 ++    0.131693370 ++   -0.161845768 ++
##         loc5              loc6              loc7              loc8
## -0.032111116       0.071610451 .    -0.008886328       0.003717678
##
## Relevance codes:     -Inf ' ' 0 '.' 1 '+' 2 '++' 5 '+++' Inf
```

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

```
##
## lm :  Term effects
##
## $ (Intercept)
##  ;  confidence int.: [ NA, NA ]
##
## $ location
##              coef  psy coefRls cRsy
## loc1  0.000000000          NaN
## loc2  0.153060685 ***     3.49 ++
## loc3  0.131693370 ***     2.81 ++
## loc4 -0.161845768 ***     3.57 ++
## loc5 -0.032111116        -1.13
## loc6  0.071610451 *       0.10 .
## loc7 -0.008886328        -3.31
## loc8  0.003717678        -2.13
##
## $ log10(distance)
##                      coef  psy coefRls cRsy
## log10(distance) -1.518295 ***    48.54 +++
##
## $ log10(charge)
##                    coef  psy coefRls cRsy
## log10(charge) 0.6355849 ***    19.46 +++
##
## Relevance codes:     -Inf ' ' 0 '.' 1 '+' 2 '++' 5 '+++' Inf
```

```
plot(termeffects(r.blast), single=TRUE)  ## plot all effects
```



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,
  - `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.ifc=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.term.relevance=c("coef", "dropRls", "dropRls.symbol"))
rt

## lm :  Drop-term effects
## data:  swiss ;  target variable:  Fertility
##
##                        coef df   R2x coefRlp coefRls dropRls dRsy predRle
## (Intercept)      66.9151817  1    NA      NA      NA      NA           NA
## Agriculture      -0.1721140  1 0.562    9.96    0.96    0.50 .       1.12
## Examination      -0.2580082  1 0.728    8.58   -2.84    0.00         0.01
## Education        -0.8709401  1 0.640   16.65    6.73    3.76 ++      4.16
## Catholic          0.1041153  1 0.484   10.20    1.92    1.30 +       1.69
## Infant.Mortality  1.0770481  1 0.097    7.51    1.24    1.11 +       1.53
##
## Relevance codes:     -Inf ' ' 0 '.' 1 '+' 2 '++' 5 '+++' Inf

## restore the old options
options(list = t.opt)
```

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
##         loc1              loc2              loc3              loc4
##  0.000000000       0.153060685 ++    0.131693370 ++   -0.161845768 ++
##         loc5              loc6              loc7              loc8
## -0.032111116       0.071610451 .    -0.008886328       0.003717678
##
## Relevance codes:     -Inf ' ' 0 '.' 1 '+' 2 '++' 5 '+++' Inf
```

Reference

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

This is the end of the story for the time being, 31.3.2021.

Werner Stahel, `stahel at stat.math.ethz.ch`