Relative Importance for Linear Regression in **R**: The Package relaimpo

Ulrike Grömping
TFH Berlin – University of Applied Sciences

Abstract

Relative importance is a topic that has seen a lot of interest in recent years, particularly in applied work. The **R** package **relaimpo** implements six different metrics for assessing relative importance of regressors in the linear model, two of which are recommended - averaging over orderings of regressors and a newly proposed metric (Feldman 2005) called `pmvd`. Apart from delivering the metrics themselves, **relaimpo** also provides (exploratory) bootstrap confidence intervals. This paper offers a brief tutorial introduction to the package. The methods and **relaimpo**'s functionality are illustrated using the data set `swiss` that is generally available in **R**. The paper targets readers who have a basic understanding of multiple linear regression. For the background of more advanced aspects, references are provided.

Keywords: relative importance, hierarchical partitioning, linear model, **relaimpo**, **hier.part**, variance decomposition, R^2.

1. Introduction

"Relative importance" refers to the quantification of an individual regressor's contribution to a multiple regression model. Assessment of relative importance in linear models is simple, as long as all regressors are uncorrelated: Each regressor's contribution is just the R^2 from univariate regression, and all univariate R^2-values add up to the full model R^2. In sciences with predominance of observational data, regressors are typically correlated, so that it is no longer straightforward to break down model R^2 into shares from the individual regressors. Various methods have been proposed in the literature. Darlington (1968) gives an overview of the older methods, Lindeman, Merenda, and Gold (1980, p. 119 ff.) propose averaging sequential sums of squares over all orderings of regressors, Pratt (1987) yields a justification for an earlier proposal by Hoffman (1960) that had already been rejected by Darlington (1968) and others, and Feldman (2005) makes an interesting new proposal. The **R** (R Development

Core Team 2006) package **relaimpo** (Grömping 2006) implements six different methods for assessing relative importance in linear regression. Among these, the averaging over orderings proposed by Lindeman, Merenda and Gold (`lmg`) and the newly proposed method by Feldman (`pmvd`) are the most computer-intensive and are also the recommended methods. **relaimpo** is the first R package to provide `pmvd`, while `lmg` has already been available in R package **hier.part**. **relaimpo** offers advantages over **hier.part** in terms of computation time and bootstrap confidence intervals.

Achen (1982) has introduced a distinction between "dispersion importance", i.e., importance relating to the amount of explained variance, "level importance", i.e., importance of each regressor for the response's mean, or "theoretical importance" i.e., change in the response for a given change in the regressor. The focus in **relaimpo** is on metrics for "dispersion importance". Also focussing on dispersion importance, Johnson and Lebreton (2004) define relative importance as "the proportionate contribution each predictor makes to R^2, considering both its direct effect (i.e., its correlation with the criterion) and its effect when combined with the other variables in the regression equation". This definition still leaves room for debate on which metrics are useful. However, it rules out some simple metrics immediately.

Section 2 of this paper briefly introduces the example data set `swiss` that is used for illustrating the methods and features of **relaimpo**. Section 3 presents the linear model and the relative importance metrics covered in **relaimpo**, and explains how calculations can be done based on the empirical covariance matrix of the data. Formulae are supported by example calculations for the data set `swiss`. Section 4 briefly discusses how to bootstrap regression models. The most important features of **relaimpo** are covered in Section 5. Section 6 compares **relaimpo** to other R packages concerned with relative importance, and computation times of the computer-intensive metrics are discussed in Section 7. The "Final remarks" section summarizes the most important elements of the paper and gives an outlook on intended future developments.

2. The example data set

When working with R, the dataset `swiss` is in the search path per default. A description of the variables in this dataset can be obtained by typing `?swiss` into the R console, and `plot(swiss)` provides a quick overview of the data. The dataset has 47 observations (French-speaking swiss provinces) on 6 variables, the response is a fertility index (`Fertility`), and the following regressors are available:

  `Agriculture` is the percentage of males working in agriculture,

  `Examination` is the percentage of draftees getting highest mark on an army exam,

  `Education` is the percentage of draftees having more than primary school education,

  `Catholic` is the percentage of catholics in the population (as opposed to protestant christians),

  `Infant.Mortality` is the percentage of live births who die within the first year.

This dataset is useful for demonstrating differences between relative importance approaches, since different perspectives yield different assessments of relative importance, in particular

with respect to the influence of the variable `Examination`. This phenomenon is the consequence of multicollinearity:

```
> cor(swiss)
```

```
                 Fertility Agriculture Examination   Education   Catholic Infant.Mortality
Fertility        1.0000000   0.35307918  -0.6458827 -0.66378886  0.4636847       0.41655603
Agriculture      0.3530792   1.00000000  -0.6865422 -0.63952252  0.4010951      -0.06085861
Examination     -0.6458827  -0.68654221   1.0000000  0.69841530 -0.5727418      -0.11402160
Education       -0.6637889  -0.63952252   0.6984153  1.00000000 -0.1538589      -0.09932185
Catholic         0.4636847   0.40109505  -0.5727418 -0.15385892  1.0000000       0.17549591
Infant.Mortality 0.4165560  -0.06085861  -0.1140216 -0.09932185  0.1754959       1.00000000
```

`Examination` has a relatively high positive correlation with `Education`, and both these variables have a relatively high negative correlation with `Agriculture`, `Examination` is also negatively correlated with `Catholic`. This structure leads to the strong dependence of allocation of relative importance on the way of looking at the matter.

3. The linear model and the relative importance metrics

A linear model with an intercept can be written as

$$y_i = \beta_0 + x_{i1}\beta_1 + \cdots + x_{ip}\beta_p + e_i. \tag{1}$$

The response of object i is modelled as a linear function of regressor values $x_{i1}, \ldots, x_{ip}$, with unknown coefficients $\beta_1, \ldots, \beta_p$, and e_i represents the unexplained part. For the example, $p = 5$ regressors are available for explaining the response `Fertility`.

In linear regression, the coefficients $\beta_k, k = 0, \ldots, p$, are estimated by minimizing the sum of squared unexplained parts. If we denote the estimated coefficients as $\hat{\beta}_k$ and the fitted response values as $\hat{y}_i = \hat{\beta}_0 + x_{i1}\hat{\beta}_1 + \cdots + x_{ip}\hat{\beta}_p$, the coefficient of determination R^2 can be written as

$$R^2 = \frac{\text{Model SS}}{\text{Total SS}} = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}. \tag{2}$$

R^2 measures the proportion of variation in y that is explained by the p regressors in the model.

The following output shows the linear regression results for the example data:

```
> linmod <- lm(Fertility ~ ., data = swiss)
> summary(linmod)

Call:
lm(formula = Fertility ~ ., data = swiss)

Residuals:
     Min       1Q   Median       3Q      Max
-15.2743  -5.2617   0.5032   4.1198  15.3213

Coefficients:
```

```
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      66.91518   10.70604   6.250 1.91e-07 ***
Agriculture      -0.17211    0.07030  -2.448  0.01873 *
Examination      -0.25801    0.25388  -1.016  0.31546
Education        -0.87094    0.18303  -4.758 2.43e-05 ***
Catholic          0.10412    0.03526   2.953  0.00519 **
Infant.Mortality  1.07705    0.38172   2.822  0.00734 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 7.165 on 41 degrees of freedom
Multiple R-Squared: 0.7067,        Adjusted R-squared: 0.671
F-statistic: 19.76 on 5 and 41 DF,  p-value: 5.594e-10
```

We see that R^2 is 70.67% and that all regressors except `Examination` are significant in this model, with `Fertility` increasing for higher `Infant.Mortality` and higher proportion of `Catholics` and `Fertility` decreasing for higher values for `Agriculture`, `Education` and `Examination`. This is somewhat in line with expectations, though the direction of the effect of `Agriculture` might come as a surprise.

Note that the stored linear model object `linmod` will be used in many subsequent calculations.

Initially, all six metrics available in **relaimpo** are calculated:

```
> metrics <- calc.relimp(linmod, type = c("lmg", "pmvd", "first", "last",
+      "betasq", "pratt"))
> metrics

Response variable: Fertility
Total response variance: 156.0425
Analysis based on 47 observations

5 Regressors: Agriculture Examination Education Catholic Infant.Mortality
Proportion of variance explained by model: 70.67%
Metrics are not normalized (rela=FALSE).

Relative importance metrics:

                      lmg        pmvd        last     first     betasq       pratt
Agriculture      0.05709122 0.04478517 0.042869607 0.1246649 0.09791973 -0.1104860
Examination      0.17117303 0.04446868 0.007387419 0.4171645 0.02715186  0.1064274
Education        0.26013468 0.37981877 0.161962693 0.4406156 0.44943721  0.4450046
Catholic         0.10557015 0.13433174 0.062372626 0.2150035 0.12082578  0.1611768
Infant.Mortality 0.11276592 0.10333064 0.056945259 0.1735189 0.06306928  0.1046122
```

The rationale of these metrics is explained in the following two sub sections.

3.1. Simple relative importance metrics

The metric `first`

One way of looking at relative importance is to compare, what each regressor alone is able

to explain, i.e., to compare the R^2-values from p regression models with one regressor only. These univariate R^2-values are identical to the squared correlations of the regressors with the response. They are available in **relaimpo** under the name `first`. If regressors are correlated, the sum of these individual contributions is often far higher than the overall R^2 of the model with all regressors together, i.e., the overall model explains less than the sum of all individual models. This is also the case for the example data:

```
> cor(swiss[, 1], swiss[, 2:6])^2

     Agriculture Examination Education  Catholic Infant.Mortality
[1,]   0.1246649   0.4171645 0.4406156 0.2150035        0.1735189

> metrics$first

     Agriculture Examination Education  Catholic Infant.Mortality
       0.1246649   0.4171645 0.4406156 0.2150035        0.1735189
```

Note that `Examination`, the only insignificant variable in the linear model, looks second most important, when using this metric. `first` is the only metric that is completely ignorant of the other regressors in the model, no adjustment takes place. Thus, this metric does not comply with the definition of relative importance by Johnson and Lebreton (2004), in that it only uses the direct effect. Also, it is not suitable for decomposing R^2 into contributions from individual regressors, since the contributions do not naturally add up to R^2.

The metric `last`

Another way of looking at relative importance is to compare, what each regressor is able to explain in addition to all other regressors that are available. Here, we ascribe to each regressor the increase in R^2 when including this regressor as the last of the p regressors. This approach is implemented in **relaimpo** under the name `last`. If regressors are correlated, these contributions again do not add up to the overall R^2, but typically add up to far less than the overall R^2. A direct calculation from a linear model in R can be obtained using the function `drop1`. (The sums of squares calculated by this function are known to SAS users as type III SS.)

```
> drop1(linmod)

Single term deletions

Model:
Fertility ~ Agriculture + Examination + Education + Catholic +
    Infant.Mortality
                 Df Sum of Sq    RSS    AIC
<none>                        2105.0  190.7
Agriculture       1     307.7 2412.8  195.1
Examination       1      53.0 2158.1  189.9
Education         1    1162.6 3267.6  209.4
Catholic          1     447.7 2552.8  197.8
Infant.Mortality  1     408.8 2513.8  197.0
```

Note that the output of the function `calc.relimp` is identical to the type III SS divided by
the total sum of squares $\sum_{i=1}^{n}(y_i - \bar{y})^2$:

```
> TotalSS <- (nrow(swiss) - 1) * var(swiss$Fertility)
> drop1(linmod)$"Sum of Sq"[2:6]/TotalSS

[1] 0.042869607 0.007387419 0.161962693 0.062372626 0.056945259

> metrics$last

    Agriculture  Examination    Education     Catholic Infant.Mortality
    0.042869607  0.007387419  0.161962693  0.062372626      0.056945259
```

For the metric `last`, like in significance testing, `Examination` comes out particularly low.
This is not surprising, since the relation between regressors in terms of `last` is identical to
the relation between regressors in terms of t-test statistics (cf. e.g., Bring 1996). The metric
`last` has also been called "usefulness" by Darlington (1968), who also notes that basing relative
importance considerations on this metric is close to using siginificance testing for assessing
importance. Again, the contributions do not naturally decompose R^2. Also, the metric does
again not comply with the definition of relative importance by Johnson and Lebreton (2004),
in that it does not at all use the direct effect.

The metric betasq

A further approach to assessing individual contributions consists in considering standardized
coefficients. The regressors in data set `swiss` are all percentages scaled from 0 to 100 (though
most of them do not use the full scale). Imagine that one regressor is rescaled from the
0-100 scale to a 0-1 scale. Obviously, if nothing else changes, the respective coefficient has to
be multiplied by the factor 100 in order to adjust for the change. Likewise, if the response
`Fertility` were rescaled from the scale 0 to 100 to the scale 0 to 1, all β's would have to be
divided by the factor 100 in order to adjust for this rescaling. Standardized coefficients are
introduced as scale-invariant versions of the coefficients, by adjusting with estimated standard
deviations:

$$\hat{\beta}_{k,standardized} = \hat{\beta}_k \frac{\sqrt{s_{kk}}}{\sqrt{s_{yy}}}, \tag{3}$$

where s_{kk} and s_{yy} denote the empirical variances of regressor x_k and the response y respec-
tively. (As long as one only compares regressors within models for the same response y,
division by $\sqrt{(s_{yy})}$ is irrelevant.) The squared standardized coefficient has been proposed as
a metric for relative importance. It is available in **relaimpo** as `betasq`.

```
> sx <- as.numeric(lapply(swiss[, 2:6], "sd"))
> (linmod$coefficients[2:6] * sx/sd(swiss$Fertility))^2

    Agriculture Examination    Education     Catholic Infant.Mortality
     0.09791973  0.02715186   0.44943721   0.12082578       0.06306928

> metrics$betasq
```

```
   Agriculture Examination   Education   Catholic Infant.Mortality
    0.09791973  0.02715186  0.44943721 0.12082578       0.06306928
```

Bring (1996) cites an ongoing debate about whether or not this metric is appropriate for relative importance considerations. Johnson and Lebreton (2004) criticize standardized coefficients, because they take too little acount of the direct effect. Whatever stand one takes in the debate, if interest is in decomposing R^2, `betasq` is not an appropriate metric, since it does not provide a natural decomposition of R^2.

The metric `pratt`

Hoffman (1960) proposed to multiply the standardized coefficient with the marginal correlation. Since the sum of these products over all regressors is just the overall R^2, he considered this metric a natural decomposition of R^2. This proposal has been criticized and even ridiculed, but has found a powerful defender in Pratt (1987) and is therefore included in **relaimpo** under the name `pratt`.

```
> sx <- as.numeric(lapply(swiss[, 2:6], "sd"))
> (linmod$coefficients[2:6] * sx/sd(swiss$Fertility)) * cor(swiss$Fertility,
+     swiss[, 2:6])

     Agriculture Examination Education  Catholic Infant.Mortality
[1,]   -0.110486   0.1064274 0.4450046 0.1611768       0.1046122

> metrics$pratt

     Agriculture Examination Education  Catholic Infant.Mortality
      -0.1104860   0.1064274 0.4450046 0.1611768       0.1046122
```

Although Pratt (1987) found some interesting properties of the metric (cf. also Johnson and Lebreton 2004), several researchers have brought forward quite counterintuitive examples (e.g., Bring 1996), and the main criticism in the context of decomposing R^2 is that the metric can allocate negative contributions (as for `Agriculture` in the example data). Even Pratt says that the metric is not usable in this case, so that this metric is only applicable in some but not all situations. Because of this property, the present author does not recommend this metric.

Thus, the chapter of simple metrics has not offered anything that yields a useful (=at least non-negative) natural decomposition of R^2, although some of the metrics - especially combinations of several metrics - can provide insights regarding the contributions of regressors.

3.2. Computer-intensive relative importance metrics

The following metrics, `lmg` and `pmvd`, require far more computational effort. Both these metrics decompose R^2 into non-negative contributions that automatically sum to the total R^2. This is an advantage they have over all simple metrics.

The difficulty in decomposing R^2 for regression models with correlated regressors lies in the fact that each order of regressors yields a different decomposition of the model sum of squares, which is illustrated below:

```
> anova(linmod)

Analysis of Variance Table

Response: Fertility
                Df  Sum Sq Mean Sq F value     Pr(>F)
Agriculture      1  894.84  894.84 17.4288 0.0001515 ***
Examination      1 2210.38 2210.38 43.0516 6.885e-08 ***
Education        1  891.81  891.81 17.3699 0.0001549 ***
Catholic         1  667.13  667.13 12.9937 0.0008387 ***
Infant.Mortality 1  408.75  408.75  7.9612 0.0073357 **
Residuals       41 2105.04   51.34
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> reverse <- anova(lm(Fertility ~ Infant.Mortality + Catholic + Education +
+      Examination + Agriculture, data = swiss))
> reverse

Analysis of Variance Table

Response: Fertility
                Df  Sum Sq Mean Sq F value     Pr(>F)
Infant.Mortality 1 1245.51 1245.51 24.2589 1.426e-05 ***
Catholic         1 1129.82 1129.82 22.0055 3.013e-05 ***
Education        1 2380.38 2380.38 46.3628 3.068e-08 ***
Examination      1    9.49    9.49  0.1848   0.66956
Agriculture      1  307.72  307.72  5.9934   0.01873 *
Residuals       41 2105.04   51.34
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

In R, the command `anova` provides *sequential* sums of squares. "Sequential" means that the regressors are entered into the model in the order they are listed. The sequential sums of squares of all regressors do sum to the model sum of squares (in this example 5072.91 or 70.67% of the total response sum of squares of $\sum_{i=1}^{n}(y_i - \bar{y})^2 = 7177.96$). Division of sequential sums of squares by the total response sum of squares yields sequential R^2 contributions. The example above shows that the order of regressors can have a very strong impact on the relative importance assessment: for example, `Agriculture` and `Infant.Mortality` receive about three times the share when entered first in comparison to when entered last into the model. The discrepancy between the two orders is particularly striking for `Examination` which receives the largest share for one of the orders and the smallest share for the other one. Obviously, unless there is a meaningful prior knowledge that prompts a natural order among variables, an approach based on the sequential R^2 of one fixed order of regressors is not appropriate for judging relative importance. Sometimes, applied researchers apply stepwise regression and decompose R^2 based on the order obtained by this automatic approach. Even though this approach relieves the researcher from the burden to decide on an arbitrary ordering, the

automatic decision is not so much less arbitrary and is therefore not recommended (cf. also Bring 1996, and references therein).

The approach taken by the metrics `lmg` and `pmvd` is based on sequential R^2s, but takes care of the dependence on orderings by averaging over orderings, either using simple unweighted averages (`lmg`) or weighted averages with data-dependent weights (`pmvd`). First, the approaches are formally described giving the formulae for their calculation, then the functioning of both approaches is illustrated using the `swiss` data.

For describing the metrics, the following notations are useful: The R^2 for a model with regressors in set S is given as

$$R^2(S) = \frac{\text{Model SS(model with regressors in } S)}{\text{Total SS}}. \tag{4}$$

The additional R^2 when adding the regressors in set M to a model with the regressors in set S is given as

$$seqR^2(M|S) = R^2(M \cup S) - R^2(S). \tag{5}$$

The order of the regressors in any model is a permutation of the available regressors $x_1, \ldots, x_p$ and is denoted by the tuple of indices $r = (r_1, \ldots, r_p)$. Let $S_k(r)$ denote the set of regressors entered into the model before regressor x_k in the order r. Then the portion of R^2 allocated to regressor x_k in the order r can be written as

$$seqR^2\left(\{x_k\}|S_k(r)\right) = R^2\left(\{x_k\} \cup S_k(r)\right) - R^2\left(S_k(r)\right). \tag{6}$$

The metric `lmg`

With (6), the metric `lmg` (in formulae denoted as LMG) can be written as

$$\text{LMG}(x_k) = \frac{1}{p!} \sum_{r \text{ permutation}} seqR^2(\{x_k\}|r). \tag{7}$$

Orders with the same $S_k(r) = S$ can be summarized into one summand, which simplifies the formula into

$$\text{LMG}(x_k) = \frac{1}{p!} \sum_{S \subseteq \{x_1, \ldots, x_p\} \setminus \{x_k\}} n(S)!(p - n(S) - 1)! seqR^2(\{x_k\}|S).$$

Christensen (1992) has shown that this metric can also be written in a different way that some people consider to be more intuitive:

$$\text{LMG}(x_k) = \frac{1}{p} \sum_{j=0}^{p-1} \left(\sum_{\substack{S \subseteq \{x_1, \ldots, x_p\} \setminus \{x_k\} \\ n(S)=j}} \frac{seqR^2(\{x_k\}|S)}{\binom{p-1}{i}} \right).$$

This formula shows `lmg` as the average over average contributions in models of different sizes. From a heuristic and ad-hoc perspective, unweighted averaging is a natural thing to do, so that

`lmg` has been adopted or reinvented by many researchers. The very popular paper by Chevan and Sutherland (1991) has coined the expression "hierarchical partitioning" and has extended the idea to other regression models, and the R package **hier.part** implements this proposal (cf. also Section 6). The metric `lmg` has nevertheless not made it into the mainstream of applied statistical analysis (with possible exceptions in some special fields of application), which may be due to both computational burden and lack of understanding what is really calculated. Johnson and Lebreton (2004) recommend `lmg`, since it clearly uses both direct effects (orders with x_k first) and effects adjusted for other regressors in the model (orders with x_k last). Given the current choices of metrics for relative importance, the present author also recommends `lmg`, particularly if interest is in decomposing R^2.

The metric `pmvd`[1]

It is known and not surprising (cf. e.g., Feldman 2005) that the `lmg` contribution of a regressor with coefficient 0 can be positive, if this regressor is correlated with one or more strong contributors. Feldman considers this as a disadvantage of the `lmg` allocations. He introduces the metric PMVD (denoted throughout the text of this paper with its **relaimpo** name `pmvd`, in formulae as PMVD) in order to cure this disadvantage: Per construction, `pmvd` guarantees that a regressor with 0 estimated coefficient is assigned a relative importance of 0, and - more important - that the allocated contribution asymptotically approaches 0 if the true coefficient is 0. As mentioned before, `pmvd` can be seen as an average over orderings as well, but with data-dependent weights for each order:

$$\text{PMVD}(x_k) = \frac{1}{p!} \sum_{r \text{ permutation}} p(r) seqR^2(\{x_k\}|r), \tag{8}$$

where $p(r)$ denotes the data-dependent `pmvd` weights that will be discussed in the following. Note that formula (8) is not used for computation; there is a more efficient computation approach that exploits a game-theoretic background of `pmvd` (cf. Feldman 2005). Regarding the weights in formula (8), Feldman's (2005) idea was to use the sequences

$$seqR^2(\{x_{r_{k+1}}, \ldots, x_{r_p}\}|\{x_{r_1}, \ldots, x_{r_k}\}), \quad k = 1, \ldots, p-1$$

for determining the weights of all orders r. The weights have been derived from a set of axioms the most important of which is the axiom of proper exclusion that requires that a regressor with actual coefficient 0 should also be allocated an R^2 share of 0. The weights obtained are proportional to

$$L(r) = \prod_{i=1}^{p-1} seqR^2(\{x_{r_{i+1}}, \ldots, x_{r_p}\}|\{x_{r_1}, \ldots, x_{r_i}\})^{-1},$$

i.e., the weights are

$$p(r) = \frac{L(r)}{\sum_{r \text{ permutation}} L(r)}.$$

A careful look at the $L(r)$ reveals that the factors increase with increasing i. $L(r)$ will be particularly large, if the first regressor already captures a very substantial amount of the

---

[1]In the US, `pmvd` is potentially protected by US patent 6,640,204. It is therefore available for non-US users only from the author's website (cf. also Appendix A).

variance, or if the last regressor captures only a very small amount of the variance. (If the last regressor(s) do(es) not capture any variance (estimated coefficient(s) 0), $L(r)$ is infinite, but the situation is nevertheless well-behaved: Feldman shows via limiting considerations that the weights are only positive for orders with all 0 coefficient regressors last, which leads to a share of 0 for these regressors, as desired.)

Note that `pmvd`'s property of assigning a zero share to regressors with coefficient 0 comes at the price of a large variability in allocated contributions for many applications (exemplified e.g. in Figure 2). This is due to basing the weights on the potentially very variable order-dependent empirical sequential R^2 allocations. `pmvd` weights can be seen as a compromise between the equal weights of `lmg` and the (inappropriate) approach of basing relative importance assessments on one order determined by an automatic variable selection procedure (data-dependent weight 1 on one order). Given the currently available choices, `pmvd` is one of the two metrics that come closest to Johnson and Lebreton's (2004) reasonable definition of relative importance so that its use can be recommended, perhaps in comparison with `lmg` (but see also the subsection on the exclusion property on page 11).

Comparison of `lmg` *and* `pmvd` *on two examples*

Table 1 shows a comparison of `pmvd` and `lmg` for two different four-regressor models for the `swiss` data: One model omits the regressor `Agriculture`, the other the regressor `Examination`. Hence, there are 24 orders of the four regressors in each of the models, so that it is feasible to show all the allocated sequential R^2s and the `pmvd` weights. In the example with `Agriculture` omitted, the `pmvd` weights are very concentrated on a few orders: the orders with `Examination` last are the only ones that receive a weight of more than 1% (bold-face weights). This is the reason that the `pmvd` share for `Examination` is so much lower than the `lmg` share. In the example with `Examination` omitted the weights are far more balanced, orders with `Agriculture` or `Infant.Mortality` last are the ones that tend to receive the higher weights, orders with low allocations for `Education` receive a low weight.

Is `pmvd`*'s property "exclusion" desirable?*

Feldman (2005) specifically constructs `pmvd` to allocate a share of zero to any regressor with coefficient zero (called "exclusion"). However, `lmg`'s property of allowing correlated regressors to benefit from each other's shares does have a plausible background, at least for some purposes. Consider the linear model

$$y_i = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + e_i \tag{9}$$

based on the causal chain (assuming linear relations)

$$x_1 \longrightarrow x_2 \longrightarrow y. \tag{10}$$

In this linear model, β_1 is zero. For prediction purposes, x_1 does not contribute anything, once x_2 is known. Thus, the request for a zero allocated share for x_1 seems reasonable. However, if one is interested in relative importance from a causal perspective, x_1's contribution should not be zero in spite of its zero coefficient. The linear model equation (9) is compatible with different types of underlying causal structures, among them the structure (10), for which a 0 coefficient for x_1 does not imply no causal influence of x_1. Model uncertainty increases with

| | | Agriculture omitted Allocated R^2 (Total 66.39%) | | | | | Examination omitted Allocated R^2 (Total 69.93%) | | | |
| | pmvd | 1 | 2 | 3 | 4 | pmvd | 1 | 2 | 3 | 4 |
Order	weight	Exam	Educ	Cath	Infa	weight	Agri	Educ	Cath	Infa
1234	0.0024	41.72	8.83	6.94	8.90	**0.0255**	12.47	32.46	19.30	5.71
2134	0.0026	6.49	44.06	6.94	8.90	**0.0567**	0.86	44.06	19.30	5.71
2314	0.0046	0.03	44.06	13.39	8.90	**0.1136**	6.77	44.06	13.39	5.71
1324	0.0016	41.72	14.46	1.31	8.90	**0.0142**	12.47	39.40	12.36	5.71
3124	0.0009	21.52	14.46	21.50	8.90	**0.0168**	3.33	39.40	21.50	5.71
3214	0.0023	0.03	35.95	21.50	8.90	**0.0607**	6.77	35.95	21.50	5.71
2341	**0.3097**	0.13	44.06	13.39	8.80	**0.1763**	3.68	44.06	13.39	8.80
1342	0.0012	41.72	12.29	1.31	11.07	0.0026	12.47	31.35	12.36	13.76
3142	0.0006	21.52	12.29	21.50	11.07	0.0031	3.33	31.35	21.50	13.76
3241	**0.1540**	0.13	35.95	21.50	8.80	**0.0942**	3.68	35.95	21.50	8.80
1243	0.0047	41.72	8.83	4.48	11.36	**0.0109**	12.47	32.46	13.33	11.68
2143	0.0052	6.49	44.06	4.48	11.36	**0.0243**	0.86	44.06	13.33	11.68
3412	0.0004	21.00	12.29	21.50	11.59	0.0038	5.50	31.35	21.50	11.59
3421	**0.0414**	0.13	33.16	21.50	11.59	**0.0319**	3.68	33.16	21.50	11.59
1423	0.0058	41.72	8.28	4.48	11.91	0.0072	12.47	24.88	13.33	19.26
2413	0.0082	5.43	44.06	4.48	12.42	**0.0452**	0.13	44.06	13.33	12.42
2431	**0.2793**	0.13	44.06	9.78	12.42	**0.1635**	3.68	44.06	9.78	12.42
1432	0.0021	41.72	12.29	0.47	11.91	0.0030	12.47	31.35	6.86	19.26
4123	0.0029	36.28	8.28	4.48	17.35	0.0078	14.37	24.88	13.33	17.35
4213	0.0038	5.43	39.13	4.48	17.35	**0.0222**	0.13	39.13	13.33	17.35
4231	**0.1272**	0.13	39.13	9.78	17.35	**0.0805**	3.68	39.13	9.78	17.35
4132	0.0011	36.28	12.29	0.47	17.35	0.0033	14.37	31.35	6.86	17.35
4312	0.0004	21.00	12.29	15.74	17.35	0.0035	5.50	31.35	15.74	17.35
4321	**0.0378**	0.13	33.16	15.74	17.35	**0.0294**	3.68	33.16	15.74	17.35
	pmvd	1.17	40.47	13.25	11.49	pmvd	4.43	40.53	14.71	10.26
	lmg	18.03	25.74	10.46	12.16	lmg	6.62	35.93	15.16	12.22

Table 1: Orders, `pmvd` weights and R^2s for two reduced examples.

increasing correlation among regressors, so that `lmg`'s property of allowing correlated regressors to benefit from each other's shares can be seen as a way to take care about uncertainty of information regarding the true underlying structure. Hence, particularly when the focus of the research is more on causal than on predictive importance, `lmg`'s behaviour appears to be more appropriate than `pmvd`'s.

3.3. Calculating relative importances based on the covariance matrix

In the previous section, computer-intensive methods have been discussed. If computing time is an issue, it can be helpful to reconsider calculation methods developed when computers were not widely available. Old textbooks on linear regression provide methods to calculate most regression results from the $(p + 1) \times (p + 1)$ covariance matrix **S** of the response y and the p regressors that are in this section denoted by a $1 \times p$-row vector **x**. Following e.g., Lindeman, Merenda, and Gold (1980, p. 103 ff), the relevant formulae are outlined below.

This sub section is meant for readers who want to dive into the calculations done in **relaimpo**. Defining variances and covariances as

$$s_{yy} = \frac{1}{n-1} \sum_{i=1}^{n} (y_i - \bar{y})^2,$$

$$s_{jk} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \bar{x}_{.j})(x_{ik} - \bar{x}_{.k})$$

and

$$s_{yk} = s_{ky} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ik} - \bar{x}_{.k})(y_i - \bar{y}),$$

where $\bar{x}_{.j}$ denotes the arithmetic mean of the j-th regressor, the corresponding variance-covariance matrices can be defined as

$$\mathbf{S_{xx}} = (s_{jk})_{\substack{j=1,\ldots,p \\ k=1,\ldots,p}},$$

$$\mathbf{S_{xy}} = (s_{ky})_{k=1,\ldots,p}, \qquad \mathbf{S_{yx}} = \mathbf{S_{xy}}^T,$$

and can be combined into the overall variance-covariance matrix as

$$\mathbf{S} = \left[\begin{array}{c|c} s_{yy} & \mathbf{S_{yx}} \\ \hline \mathbf{S_{xy}} & \mathbf{S_{xx}} \end{array} \right].$$

The estimated coefficients for regressors $x_1, \ldots, x_p$ in model (1) can be calculated as

$$\hat{\beta}_{1,\ldots,p} = \mathbf{S_{xx}}^{-1} \mathbf{S_{xy}}$$

(the coefficient β_0 is irrelevant for relative importance), and the unexplained variance of the response is obtained as

$$s_{yy} - \mathbf{S_{yx}} \mathbf{S_{xx}}^{-1} \mathbf{S_{xy}}. \tag{11}$$

The coefficient of determination R^2 from regressing y on $\mathbf{x}$ can thus be obtained as

$$R^2 = \frac{\mathbf{S_{yx}} \mathbf{S_{xx}}^{-1} \mathbf{S_{xy}}}{s_{yy}}. \tag{12}$$

If the regressors in $\mathbf{x}$ are partitioned into two groups $\mathbf{x}_1$ and $\mathbf{x}_2$, i.e., the variance covariance matrix is partitioned as

$$\mathbf{S} = \left[\begin{array}{cc|c} s_{yy} & \mathbf{S_{yx_2}} & \mathbf{S_{yx_1}} \\ \mathbf{S_{x_2 y}} & \mathbf{S_{x_2 x_2}} & \mathbf{S_{x_2 x_1}} \\ \hline \mathbf{S_{x_1 y}} & \mathbf{S_{x_1 x_2}} & \mathbf{S_{x_1 x_1}} \end{array} \right], \tag{13}$$

the unexplained variance-covariance matrix of y and $\mathbf{x}_2$ after adjusting for $\mathbf{x}_1$ can be calculated as

$$\begin{pmatrix} s_{yy} & \mathbf{S_{yx_2}} \\ \mathbf{S_{x_2 y}} & \mathbf{S_{x_2 x_2}} \end{pmatrix} - \begin{pmatrix} \mathbf{S_{yx_1}} \\ \mathbf{S_{x_2 x_1}} \end{pmatrix} \mathbf{S_{x_1 x_1}}^{-1} \begin{pmatrix} \mathbf{S_{x_1 y}} \\ \mathbf{S_{x_1 x_2}} \end{pmatrix}^T. \tag{14}$$

Note that (14) is the inverse of the appropriately-sized top left block of $\mathbf{S}^{-1}$ with $\mathbf{S}$ partitioned as shown in (13). The top left element of (14) is the unexplained variance of the response y after regressing y on $\mathbf{x}_1$. Hence, the difference between the top left element of (14) and the unexplained variance (11) from the full model divided by s_{yy} is the sequential R^2 when adding the regressors in $\mathbf{x}_2$ to the model in addition to the regressors in $\mathbf{x}_1$ ($seqR^2(\mathbf{x}_2|\mathbf{x}_1)$ when identifying $\mathbf{x}_1$ and $\mathbf{x}_2$ with the sets of regressors contained in them, cf. (5)). Note that all relevant calculations for sequential linear models can be based on the variance-covariance matrix and its conditional forms according to the formulae given in this section, i.e., after calculation of the variance-covariance matrix, no further calculations with all observations are needed. Thus, `calc.relimp` can also be applied to a variance-covariance matrix directly, as documented with package **relaimpo**.

4. Bootstrapping regression models

So far, point estimates for R^2 decompositions have been obtained. In order to assess, which regressors are clearly different and which are similar in terms of relative importance, variability of the estimates needs to be estimated. In particular the computer-intensive metrics `lmg` and `pmvd` are not easily tackled by proper distributional results (even if there is a distributional result on `lmg` for multivariate normal data, cf. e.g., Budescu (1993)). Therefore, **relaimpo** provides a bootstrapping facility for assessing the variability of all metrics.

When bootstrapping regression models, there are two principally different reasonable approaches (cf. e.g., Davison and Hinkley 1997; Fox 2002): The regressors can be considered fixed in some situations, e.g., for experimental data. In our data example, where the fixed set of all French-speaking swiss provinces is investigated, it can also be justified to work with fixed regressors. If regressors are considered fixed, only the error terms are random. Contrary, in most relative importance investigations, it makes far more sense to consider also the regressors as random, since the observations are a random sample from a larger population. These two scenarii prompt two different approaches for bootstrapping.

For fixed regressors, bootstrapping is based on repeated sampling from the residuals of the regression model. For random regressors, the complete observation rows - consisting of regressors and response - are resampled. Typically, bootstrapping for fixed regressors yields somewhat narrower confidence intervals than bootstrapping for random regressors. For example, consider the 95%-bootstrap confidence intervals for `lmg` shares in the swiss dataset:

```
> fixedlmg <- booteval.relimp(boot.relimp(linmod, b = 1000, fixed = TRUE),
+       bty = "perc", level = 0.95)
> randomlmg <- booteval.relimp(boot.relimp(linmod, b = 1000), bty = "perc",
+       level = 0.95)
> output <- rbind(fixedlmg$lmg.lower, fixedlmg$lmg.upper,
+       randomlmg$lmg.lower, randomlmg$lmg.upper)
> output <- as.matrix(t(output))
> colnames(output) <- c("fixed.lower", "fixed.upper", "random.lower",
+       "random.upper")
> rownames(output) <- c(fixedlmg$namen[2:6])
> output
```

```
                fixed.lower fixed.upper random.lower random.upper
Agriculture      0.04687450  0.08840079   0.03167901    0.1138819
Examination      0.11227899  0.26490901   0.08973842    0.2889251
Education        0.16903735  0.37788122   0.06980345    0.3844111
Catholic         0.04593381  0.19673048   0.03704190    0.2396492
Infant.Mortality 0.03533646  0.21817709   0.02973998    0.2380548
```

If results are to apply to an overall population of which the present data are a random sample, bootstrapping for random regressors is the appropriate choice. Also note that bootstrapping for fixed regressors is more vulnerable to model misspecification and requires the error variances to be constant (cf. e.g., Davison and Hinkley 1997).

Bootstrapping in **relaimpo** is done using the functions `boot` and `boot.ci` from R package **boot**. R package **boot** is based on the book by Davison and Hinkley (1997). Percentile intervals, BCa intervals, normal intervals and basic intervals are supported (default: BCa intervals, except for ranks for which intervals are always percentile intervals).

5. Features of the package relaimpo

5.1. Forced to percentages: `rela=TRUE`

All metrics come in two different versions: so far, we have discussed the natural scale of each metric. **relaimpo** allows to force metrics to sum to 100% (instead of R^2 or no meaningful sum). This is achieved by the option `rela=TRUE`. The default is `rela=FALSE`.

```
> calc.relimp(linmod, type = c("lmg", "pmvd"), rela = TRUE)

Response variable: Fertility
Total response variance: 156.0425
Analysis based on 47 observations

5 Regressors: Agriculture Examination Education Catholic Infant.Mortality
Proportion of variance explained by model: 70.67%
Metrics are normalized to sum to 100% (rela=TRUE).


Relative importance metrics:


                       lmg        pmvd
Agriculture      0.08078165  0.06336911
Examination      0.24220256  0.06292130
Education        0.36807952  0.53742742
Catholic         0.14937728  0.19007370
Infant.Mortality 0.15955899  0.14620846
```

5.2. Adjustment: Some regressors always stay in the model

In many applications, some influences have to be included into the model and are not meant to

compete with the other influences, e.g., gender in epidemiological or marketing applications. If one wants to include these influences first and only decompose the remaining R^2 among the remaining regressors, this can be achieved by the option `always`. In the `swiss` example, there is no natural variable one would want to treat like this. Purely for demonstration purposes, `Catholic` is now forced to be always first:

```
> calc.relimp(linmod, type = c("lmg", "pmvd"), always = "Catholic")

Response variable: Fertility
Total response variance: 156.0425
Analysis based on 47 observations

5 Regressors:
Proportion of variance explained: 70.67%

One Regressor always included in model:
 Catholic
21.5 % of variance explained by this regressor

Relative importance of 4 regressors assessed:
 Agriculture Examination Education Infant.Mortality
49.17 % of variance decomposed among these

Metrics are not normalized (rela=FALSE).

Relative importance metrics:

                      lmg        pmvd
Agriculture        0.04021013 0.04643858
Examination        0.10365086 0.01535452
Education          0.25622307 0.34780386
Infant.Mortality   0.09164744 0.08213453
```

5.3. Treatment of missing values

If functions `calc.relimp` or `boot.relimp` are applied to a formula object, the option `na.action` is available and will work like usual in R. If nothing is specified in the function call, the overall setting of na.action (that can be viewed by `options("na.action")`) will be applied. For objects other than formula objects, **relaimpo** bases all analyses on complete cases only. Whenever observations are deleted because of missing values, a warning is printed.

5.4. Bootstrapping

A call to the function `boot.relimp`[2] requests bootstrap runs the results of which are stored

---

[2]Warning: If you try out the code of this sub section yourself, note that `b=1000` requires a little patience. For simple code-checking, you may want to choose a smaller number for `b`. It is a good idea to always set `b` explicitly, default is `b=1000`.

in an object of class `relimplmboot`. Afterwards, the result object can be (repeatedly) evaluated with the function `booteval.relimp`. `booteval.relimp` works on the output from `boot.relimp` and allows among other things selection of a subset of the metrics, selection of one or several confidence levels and suppression of confidence intervals for differences or ranks.

```
> bootresult <- boot.relimp(linmod, b = 1000, type = c("lmg", "pmvd", "last",
+     "first"), fixed = FALSE)
```

Bootstrapping makes computation times a real issue, if many bootstrap runs are required (cf. also Section 7). The recommended BCa bootstrap intervals require a substantial number of bootstrap runs (default `b=1000` should not be decreased, cf. e.g., Hesterberg, Moore, Monaghan, Clipson, and Epstein 2005) and are themselves slow to calculate. It may be an alternative to work with percentile confidence intervals (always used for ranks) or normal distribution based confidence intervals in order to get at least an indication of variability. Coverage probabilities for percentile confidence intervals with `b=1000` and normal confidence intervals with `b=200` have been investigated in some simulations and have proven to be somewhat liberal (non-coverage up to twice nominal level). For this reason, the output contains a warning that bootstrap confidence intervals can be somewhat liberal. Performance of BCa intervals has not been simulated (since they take so much longer); they might well perform better, if calculated from a large enough number of bootstrap resamples.

5.5. Bootstrap confidence intervals for ranks and their visualization

In many applications, the main interest is a ranking of regressors. Rankings of regressors can obviously be obtained by comparing the allocated shares. In order to assess stability of the ranking, bootstrap confidence intervals for ranks are provided (always percentile intervals). Rank bootstrap intervals are visualized by a lettering system.

```
> booteval.relimp(bootresult, typesel = c("lmg", "pmvd"), level = 0.9,
+     bty = "perc", nodiff = TRUE)

Response variable: Fertility
Total response variance: 156.0425
Analysis based on 47 observations

5 Regressors: Agriculture Examination Education Catholic Infant.Mortality
Proportion of variance explained by model: 70.67%
Metrics are not normalized (rela=FALSE).

Relative importance metrics:

                    lmg        pmvd
Agriculture      0.05709122 0.04478517
Examination      0.17117303 0.04446868
Education         0.26013468 0.37981877
Catholic          0.10557015 0.13433174
```

```
Infant.Mortality 0.11276592 0.10333064
```

```
 Confidence interval information ( 1000 bootstrap replicates, bty= perc ):
Relative Contributions with confidence intervals:

                                  Lower  Upper
                       percentage 0.9    0.9    0.9
Agriculture.lmg        0.0570     __CDE  0.0359 0.1034
Examination.lmg        0.1711     ABCD_  0.1042 0.2625
Education.lmg          0.2601     ABCD_  0.0915 0.3724
Catholic.lmg           0.1055     ABCDE  0.0388 0.2195
Infant.Mortality.lmg   0.1127     ABCDE  0.0407 0.2269

Agriculture.pmvd       0.0447     __CDE  0.0099 0.0925
Examination.pmvd       0.0444     ABCDE  0.0005 0.3438
Education.pmvd         0.3798     ABCD_  0.0561 0.5763
Catholic.pmvd          0.1343     ABCDE  0.0244 0.2956
Infant.Mortality.pmvd  0.1033     _BCDE  0.0098 0.2532
```

```
Letters indicate the ranks covered by bootstrap CIs.
(Rank bootstrap confidence intervals always obtained by percentile method)
CAUTION: Bootstrap confidence intervals can be somewhat liberal.
```

Lettering shows that `Education` cannot be in the last position, while `Agriculture` can be in the last three positions only for both metrics. Furthermore, `Examination` cannot be in the last position for `lmg` (no restriction for `pmvd`), while `Infant.Mortality` cannot be in the first position for `pmvd` (no restriction for `lmg`). Note that `lmg` confidence intervals are contained in `pmvd` confidence intervals for all variables but `Agriculture` in this example, which exemplifies the fact that `pmvd` is a more variable metric than `lmg` under many circumstances, as was already mentioned earlier.

5.6. Bootstrap confidence intervals for differences

If there is a detailed interest in the comparison of relative importances for pairs of regressors, confidence intervals for differences may be of interest.

```
> eval <- booteval.relimp(bootresult, typesel = c("lmg", "pmvd"), level = 0.9,
+     bty = "perc", norank = TRUE)
> eval
```

```
Response variable: Fertility
Total response variance: 156.0425
Analysis based on 47 observations
```

```
5 Regressors: Agriculture Examination Education Catholic Infant.Mortality
Proportion of variance explained by model: 70.67%
```

```
Metrics are not normalized (rela=FALSE).

Relative importance metrics:

                        lmg        pmvd
Agriculture       0.05709122 0.04478517
Examination       0.17117303 0.04446868
Education         0.26013468 0.37981877
Catholic          0.10557015 0.13433174
Infant.Mortality 0.11276592 0.10333064


 Confidence interval information ( 1000 bootstrap replicates, bty= perc ):
Relative Contributions with confidence intervals:

                                Lower  Upper
                    percentage  0.9    0.9
Agriculture.lmg       0.0570    0.0359 0.1034
Examination.lmg       0.1711    0.1042 0.2625
Education.lmg         0.2601    0.0915 0.3724
Catholic.lmg          0.1055    0.0388 0.2195
Infant.Mortality.lmg  0.1127    0.0407 0.2269

Agriculture.pmvd      0.0447    0.0099 0.0925
Examination.pmvd      0.0444    0.0005 0.3438
Education.pmvd        0.3798    0.0561 0.5763
Catholic.pmvd         0.1343    0.0244 0.2956
Infant.Mortality.pmvd 0.1033    0.0098 0.2532

CAUTION: Bootstrap confidence intervals can be somewhat liberal.


 Differences between Relative Contributions:

                                             Lower  Upper
                                 difference 0.9 0.9  0.9
Agriculture-Examination.lmg        -0.114    *  -0.191 -0.046
Agriculture-Education.lmg          -0.203    *  -0.303 -0.038
Agriculture-Catholic.lmg           -0.048       -0.173 0.0395
Agriculture-Infant.Mortality.lmg   -0.055       -0.173 0.0401
Examination-Education.lmg          -0.088       -0.228 0.1024
Examination-Catholic.lmg            0.0656      -0.069 0.1818
Examination-Infant.Mortality.lmg    0.0584      -0.098 0.1950
Education-Catholic.lmg              0.1545      -0.109 0.3234
Education-Infant.Mortality.lmg      0.1473      -0.074 0.2844
Catholic-Infant.Mortality.lmg      -0.007       -0.130 0.1189
```

```
Agriculture-Examination.pmvd         0.0003         -0.273 0.0703
Agriculture-Education.pmvd          -0.335      *   -0.538 -0.011
Agriculture-Catholic.pmvd           -0.089         -0.258 0.0307
Agriculture-Infant.Mortality.pmvd   -0.058         -0.217 0.0551
Examination-Education.pmvd          -0.335         -0.559 0.2137
Examination-Catholic.pmvd           -0.089         -0.256 0.2784
Examination-Infant.Mortality.pmvd   -0.058         -0.212 0.2715
Education-Catholic.pmvd              0.2454         -0.150 0.5001
Education-Infant.Mortality.pmvd      0.2764         -0.092 0.5104
Catholic-Infant.Mortality.pmvd       0.0310         -0.167 0.2174


* indicates that CI for difference does not include 0.
CAUTION: Bootstrap confidence intervals can be somewhat liberal.
```

The confidence intervals for differences show which differences in contributions can be considered statistically significant (in an exploratory sense). Stars mark those contributions for which the confidence interval does not contain the equality case. Note that the procedure generates a substantial amount of output, if confidence intervals for differences are not suppressed, since confidence intervals for the metrics themselves are also always shown. (The `norank=TRUE`-option suppresses the lettering for rank confidence intervals.)

5.7. Plotting results

Barplots of results can be obtained using the plot-methods on the output objects of functions `calc.relimp` and `booteval.relimp`. Figure 1 shows barplots of all metrics for the `swiss` example (`metrics` calculated on page 4) and can be created by the R statements

```
> par(cex.axis = 0.8)
> plot(metrics, names.abbrev = 3)
```

Since bootstrapping has taken place, barplots with variability lines can also be created (Figure 2).

```
> par(cex.axis = 0.9)
> plot(booteval.relimp(bootresult, typesel = c("lmg", "pmvd"), level = 0.9),
+       names.abbrev = 2, bty = "perc")
```

6. Comparison to related R packages

R has two further packages that are devoted to relative importance. Package **relimp** by Firth (2006) implements a method of comparing the contributions of two groups of regressors that has been proposed by Silber, Rosenbaum, and Ross (1995). This method does something quite special that is not very closely related to the scope of **relaimpo**. Hence, it is not further considered here. Package **hier.part** by Walsh and Mac Nally (2005) is more closely related and is therefore discussed in this section in comparison to **relaimpo**. (For some of the code in this section, **hier.part** needs to be installed.)

Relative importances for Fertility
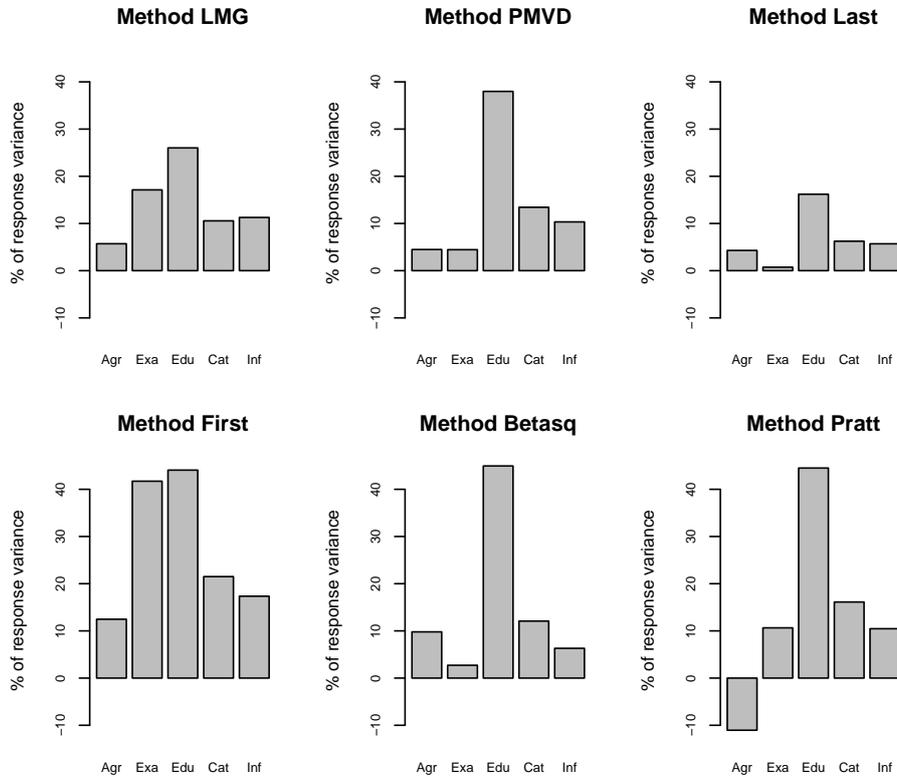
$R^2 = 70.67\%$, metrics are not normalized.

Figure 1: Bar plots of all calculated relative importance metrics.

Note that **hier.part** is more general than **relaimpo** in that it covers more general regression models and more goodness-of-fit statistics, while **relaimpo** is restricted to the linear model with goodness-of-fit statistic R^2. On the other hand, **relaimpo** is more general for the linear model in that it covers more metrics for relative importance. Furthermore, **relaimpo** provides appropriate bootstrap confidence intervals and makes use of the specifics of linear models for being faster in computing and thus allowing more regressors.
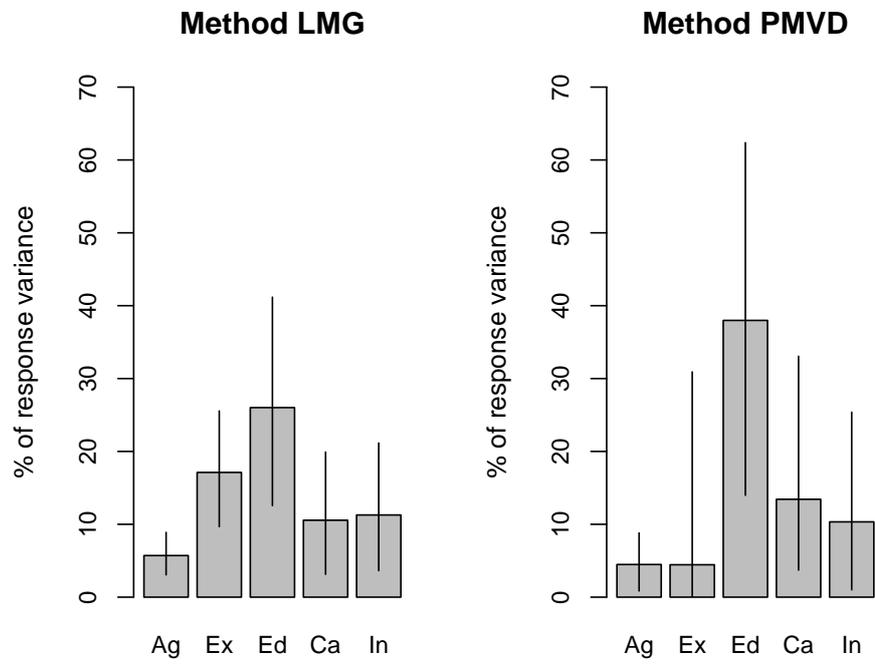
For comparing **hier.part** to **relaimpo**, the goodness-of-fit choice for the former has to be chosen as R^2 (gof=Rsqu). Then the standard output of **hier.part** for the example data (barplot turned off, since almost identical to that in **relaimpo**) is shown below:

```
> hier.part(swiss[, 1], swiss[, 2:6], gof = "Rsqu", barplot = F)


$gfs
 [1] 0.0000000 0.1246649 0.4171645 0.4406156 0.2150035 0.1735189 0.4326045
 [8] 0.4492484 0.2482782 0.3172607 0.5054845 0.4302471 0.5363016 0.5745071
[15] 0.5647800 0.3309201 0.5568480 0.4460681 0.5397679 0.6422541 0.5660833
```

Relative importances for Fertility
with 90% bootstrap confidence intervals

Figure 2: Bar plots of lmg and pmvd with confidence intervals.

```
[22] 0.3858919 0.5748498 0.6190960 0.5409672 0.6625438 0.6497897 0.6443624
[29] 0.5447723 0.6993476 0.6638654 0.7067350


$IJ
                          I          J      Total
Agriculture      0.05709122 0.06757369 0.1246649
Examination      0.17117303 0.24599144 0.4171645
Education        0.26013468 0.18048097 0.4406156
Catholic         0.10557015 0.10943335 0.2150035
Infant.Mortality 0.11276592 0.06075300 0.1735189


$I.perc
                         I
Agriculture       8.078165
Examination      24.220256
Education        36.807952
```

```
Catholic         14.937728
Infant.Mortality 15.955899
```

The first bit of output (`gfs`) lists the R^2 values for all sub models. Then, `IJ` shows the individual and joint contributions of each regressor, and `I.perc` shows a percentage rescaling of the individual contributions. In fact, `I.perc` from **hier.part** coincides with **relaimpo**'s `lmg` for `rela=TRUE`, `I` from **hier.part** coincides with **relaimpo**'s `lmg` for `rela=FALSE`, and `J` from **hier.part** is the difference between `first` and `lmg` for `rela=FALSE`. The following program illustrates how the relevant portion of the output from **hier.part** can be reproduced using **relaimpo** (`metrics` calculated on page 4):

```
> IJ <- cbind(I = metrics$lmg, J = metrics$first - metrics$lmg,
+     Total = metrics$first)
> I.perc <- as.matrix(100 * calc.relimp(linmod, type = "lmg", rela = T)$lmg)
> colnames(I.perc) = ""
> list(IJ = IJ, I.perc = I.perc)

$IJ
                          I          J     Total
Agriculture      0.05709122 0.06757369 0.1246649
Examination      0.17117303 0.24599144 0.4171645
Education        0.26013468 0.18048097 0.4406156
Catholic         0.10557015 0.10943335 0.2150035
Infant.Mortality 0.11276592 0.06075300 0.1735189

$I.perc

Agriculture       8.078165
Examination      24.220256
Education        36.807952
Catholic         14.937728
Infant.Mortality 15.955899
```

Note that `I.perc` could have been obtained by the much simpler command

```
> cbind(I.perc = 100 * IJ[,"I"]/sum(IJ[,"I"]))
```

Since the example serves the purpose of underscoring the connection between results from **hier.part** and **relaimpo**, the more complicated second call to function `calc.relimp` has been used.

7. Computation times

The metrics `lmg` and `pmvd` require a lot of computation in case of many regressors. If one wants to apply these for many regressors and potentially even in connection with a bootstrap analysis, it is helpful to know in advance how much computing time will be needed. Table 2

shows computing times for 3 to 12 regressors for both `lmg` and `pmvd` and for comparison also for **hier.part** (`barplot` turned off). All times are averages over 100 runs on a Windows XP Professional system, AMD Athlon XP 1700+, 1.47GHz, 256MB RAM. We see that **relaimpo**'s CPU times are virtually unaffected by the change in sample size, while **hier.part** times do change significantly. This is due to the fact that calculation of metrics in **relaimpo** is based on the covariance matrix (as explained in Section 3.3) which is only calculated once while **hier.part** calculates $2^p - 1$ regression models using all observations.

	100 observations			1000 observations		
p	hier.part	lmg	pmvd	hier.part	lmg	pmvd
3	0.13	0.02	0.02	0.27	0.02	0.02
4	0.26	0.03	0.03	0.60	0.03	0.03
5	0.53	0.06	0.05	1.24	0.06	0.05
6	1.09	0.10	0.09	2.61	0.10	0.09
7	2.23	0.18	0.18	5.49	0.19	0.18
8	4.61	0.33	0.37	11.46	0.33	0.37
9	9.49	0.64	0.78	23.90	0.64	0.78
10	19.50	1.25	1.74	49.84	1.23	1.72
11	40.02	2.46	4.22	104.09	2.44	4.22
12	82.42	4.93	11.64	218.84	4.92	11.64

Table 2: CPU times in seconds (average of 100 runs each) for p equi-correlated regressors with variances 1 and pairwise correlations 0.5.

For **relaimpo**, we see that `pmvd` takes longer than `lmg` for large numbers of regressors p. This is due to the fact that formula (8) cannot be simplified as much as formula (7). (Nevertheless, as mentioned before, formula (8) is not directly used for computations.) In fact, the time for `lmg` roughly doubles when adding a regressor, while the growth factor for times for `pmvd` increases with increasing number of regressors, so that the time difference between the two methods increases quite dramatically with increasing numbers of regressors (for 15 regressors, for example, `pmvd` needs about 525 seconds CPU, while `lmg` needs about 43 seconds).

8. Final remarks

The functionality of R package **relaimpo** has been explained and illustrated in this paper, using the data set `swiss` that is available in R. This dataset has a complicated correlation structure among regressors which makes assessment of relative importances somewhat ambiguous. R package **relaimpo** broadens R's possibilities of assessing relative importances in linear models by offering a choice of six metrics two of which - `lmg` and `pmvd` - are recommended if interest is in decomposing R^2. `pmvd` is newly provided, and for `lmg` computing times are substantially improved vs. the package **hier.part**. For all metrics, **relaimpo** offers bootstrap confidence intervals for the estimated relative importances themselves as well as for pairwise differences of relative contributions and for regressors' ranks in terms of relative importance. These help preventing the analyst from over-interpreting differences.

The availability of six metrics - which are already a selection from even more candidates - emphasizes the fact that there is no unique unchallenged metric for relative importance in case of correlated regressors. The recommended metrics `lmg` and `pmvd` are also not

entirely satisfactory, but are so far the metrics that come closest to Johnson and Lebreton's (2004) definition of relative importance, as cited in the introduction. There is a substantial amount of relatively recent literature on relative importance. A (certainly incomplete) list of further references on relative importance can be found on the author's website http://www.tfh-berlin.de/~groemp/.

Regarding future developments, it is intended to allow treating regressors in groups in order to support applications with large numbers of regressors that are not unusual e.g., in marketing applications. If there are e.g., 30 regressors, a complete analysis including bootstrap confidence intervals will be prohibitive even on modern computers. Often, regressors can be grouped, and an analysis of twelve groups, say, will again be feasible. Being able to handle grouped regressors will also make it possible to support factors with more than two levels. As a further step, factors with interaction terms will require the procedure to be able to respect a pre-defined hierarchy, i.e., some orders of regressors (interaction before main effect) must be excluded from the possibilities. This would also be a desirable extension of **relaimpo**. Furthermore, it would be useful to be able to accomodate weights in the analysis, since often survey data come with observation weights that reflect sampling probabilities. Further suggestions from users are welcome.

Acknowledgements

References

Achen CH (1982). *Interpreting and Using Regression*. Sage, Thousand Oaks, CA.

Bring J (1996). "A Geometric Approach to Compare Variables in a Regression Model." *The American Statistician*, **50**, 57–62.

Budescu DV (1993). "Dominance Analysis: A New Approach to the Problem of Relative Importance of Predictors in Multiple Regression." *Psychological Bulletin*, **114**, 542–551.

Chevan A, Sutherland M (1991). "Hierarchical Partitioning." *The American Statistician*, **45**, 90–96.

Christensen R (1992). "Comment on Chevan and Sutherland." *The American Statistician*, **46**, 74.

Darlington RB (1968). "Multiple Regression in Psychological Research and Practice." *Psychological Bulletin*, **69**, 161–182.

Davison A, Hinkley DV (1997). *Bootstrap Methods and Their Application*. Cambridge University Press, Cambridge.

Feldman B (2005). "Relative Importance and Value." Manuscript version 1.1, 2005-03-19, URL http://www.prismanalytics.com/docs/RelativeImportance.pdf.

Firth D (2006). **relimp**: *Relative Contribution of Effects in a Regression Model.* R package version 0.9-6.

Fox J (2002). "Bootstrapping Regression Models." In "An R and S-PLUS Companion to Applied Regression: A Web Appendix to the Book.", Sage, Thousand Oaks, CA. URL http://CRAN.R-project.org/doc/contrib/Fox-Companion/appendix-bootstrapping.pdf.

Grömping U (2006). **relaimpo**: *Relative Importance of Regressors in Linear Models.* R package version 1.1-1.

Hesterberg T, Moore DS, Monaghan S, Clipson A, Epstein R (2005). "Bootstrap Methods and Permutation Tests." In DS Moore, GP McCabe (eds.), "Introduction to the Practice of Statistics," Freeman, New York.

Hoffman PJ (1960). "The Paramorphic Representation of Clinical Judgment." *Psychological Bulletin*, pp. 116–131.

Johnson JW, Lebreton JM (2004). "History and Use of Relative Importance Indices in Organizational Research." *Organizational Research Methods*, **7**, 238–257.

Lindeman RH, Merenda PF, Gold RZ (1980). *Introduction to Bivariate and Multivariate Analysis.* Scott, Foresman, Glenview, IL.

Pratt JW (1987). "Dividing the Indivisible: Using Simple Symmetry to Partition Variance Explained." In T Pukkila, S Puntanen (eds.), "Proceedings of Second Tampere Conference in Statistics," pp. 245–260. University of Tampere.

R Development Core Team (2006). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Silber JH, Rosenbaum PR, Ross RN (1995). "Comparing the Contributions of Groups of Predictors: Which Outcomes Vary with Hospital Rather than Patient Characteristics?" *Journal of the American Statistical Association*, **90**, 7–18.

Walsh C, Mac Nally R (2005). **hier.part**: *Hierarchical Partitioning.* R package version 1.0-1.

A. Global and non-US version

There are two versions of the R package **relaimpo**. The version on CRAN (`http://CRAN.R-project.org/`) is globally licensed under GPL version 2 (or later). There is an extended version which includes the metric `pmvd` (cf. (8)) that is licensed according to GPL version 2 under the geographical restriction "outside of the US" because of potential issues with US patent 6,640,204. This version can be obtained from the author's website at `http://www.tfh-berlin.de/~groemp/`. Whenever you load the package, a display tells you, which version you are loading.

Affiliation:

Ulrike Grömping
Department II – Mathematics, Physics, Chemistry
TFH Berlin – University of Applied Sciences
D-13353 Berlin, Germany
E-mail: `groemping@tfh-berlin.de`
URL: `http://www.tfh-berlin.de/~groemp/`