Relative importance for linear regression in R: A vignette for relaimpo

Ulrike Grömping

June 16, 2006

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 several different ways of assessing relative importance of regressors in the linear model, two of which are recommended and briefly explained in this paper. Apart from delivering the metrics themselves, `relaimpo` also provides (exploratory) bootstrap confidence intervals. This vignette 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`. `R` and the package `relaimpo` are open-source software projects and can be freely downloaded from CRAN: http://cran.r-project.org.

1 Introduction

Assessment of relative importance in linear models is simple, as long as all regressors are uncorrelated. In sciences with predominance of observational data, regressors are typically correlated, so that it is not straightforward to break down model R^2 into contributions 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.119ff.) 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-Package `relaimpo` 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. In this paper, application of the R-package `relaimpo` is illustrated using the dataset `swiss` that is available with the base `R` installation. Before showing the application of `relaimpo` in Section 4, the dataset is subjected to a standard linear model approach (Section 2), and a few key properties of the recommended metrics are discussed (Section 3). `relaimpo` is also discussed in comparison to the R-package `hier.part` in Section 5, and computation times of the computer-intensive metrics are discussed in Section 6.

2 The example data analysed with lm

The dataset `swiss` is available with the base `R` installation and is already in the search path. A description of the variables in these data can be obtained by typing `? swiss` into the `R` console. The dataset has 47 observations (French-speaking swiss provinces) on 6 variables, the response is a fertility index, the regressors are

`Agriculture` percentage of males working in agriculture,

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

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

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

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

The most natural approach starts with a standard regression analysis:

```
> summary(linmod <- lm(swiss))

Call:
lm(formula = 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 agricultural effect might come as a surprise. (Note that the linear model that is saved under the name linmod will be used in many further calculations.)

If we are interested in sums of squares explained by each regressor, we can run the command

```
> 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
```

This command calculates sequential sums of squares, i.e. it enters the regressors into the model in the order they are listed. Here, for example, `Agriculture` is entered first, followed by `Examination`, `Education`, `Catholic`, and `Infant.Mortality`. `Examination` now gets a substantial share of the model variance, although it previously was not statistically significant. Would we enter the variables in different order, the result would be quite different, e.g.:

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

```
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
```

Obviously, standard sequential sums of squares are thus not appropriate for judging relative importance. The metric `lmg` in R-package `relaimpo` calculates the contribution for each regressor as the average sequential contribution to R^2 over all different orders among regressors (cf. also next section).

We can also examine all regressors as last influences in the model (SAS users know this as type III SS) by using

```
> 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
```

Here, again `Examination` comes out particularly low. Note that these contributions do not add up to the total model sum of squares and are basically equivalent to the t-values from the original significance tests obtained using `summary(lm)`. The metric `last` in `relaimpo` calculates this type of contribution.

For looking at individual contributions of each variable alone, we can e.g. calculate R^2 values from simple correlations. This metric is also included in `relaimpo` (`first`).

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

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

Obviously, different perspectives yield different assessments of relative importance in this example, in particular with respect to the influence of the variable `Examination`. This phenonmen 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 Metrics available in relaimpo

The following relative importance metrics are available in `relaimpo`:

`lmg` is the averaged sequential contribution over all orderings (Lindeman, Merenda and Gold, 1980, p.119ff).

`pmvd` is the proportional marginal variance decomposition as proposed by Feldman (2005).

`last` is the regressor's contribution when included last (cf. `drop1`-analysis in the previous section).

`first` is the regressor's contribution when included first (=squared correlation of regressor with response).

`betasq` is the squared standardized coefficient of the regressor.

`pratt` is the product of the standardized coefficient with the correlation.

The recommended metrics are `lmg` and `pmvd`. Their properties will be discussed in the next paragraph. `last` and `first` are often useful additional information. The other two are provided for users who might be used to them. Darlington (1968) discusses (among other things) usage of `last`, `first`, `betasq` and `pratt`. Note that `pratt` had originally been proposed by Hoffman (1960) and is as such discussed (and rejected) by Darlington. The metric is called `pratt` in `relaimpo`, because Pratt (1987) provided a rationale for using it. Nevertheless, the author - like Darlington - does not recommend its use.

Three of the metrics offered in `relaimpo` do naturally sum to the full model R^2. One of these, `pratt`, has the disadvantage of sometimes assigning negative contributions to one or more regressors and is therefore not recommended. The other two, `lmg` and `pmvd`, yield always non-negative contributions that sum to the full model R^2 and can therefore be recommended. As mentioned before, the `lmg` contribution of a regressor is the averaged sequential contribution over all orderings of regressors (proposed by Lindeman, Merenda and Gold, 1980, p.119ff). 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. This is a consequence of `lmg`'s property to equalize the contributions of regressors in case of stronger correlations. The equalization can be seen as a natural behavior, since substantial correlation implies that there is a lack of information regarding which contribution 'belongs' to which regressor, so that a cautious assessment may be appropriate. On the other hand, Feldman (2005) postulates that contributions of a regressor with coefficient 0 should always be 0, a property called 'exclusion'. Exclusion is in fact also a desirable behavior for a relative importance metric for some situations. `pmvd` has been constructed to guarantee exclusion, i.e. `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. In addition, like `lmg` contributions, `pmvd` contributions also sum to the full model R^2 and are always non-negative. `pmvd` allocations are different from `lmg` allocations in that they typically show larger differences between the stronger and the weaker regressors. Nevertheless, since they also tend to be more variable than `lmg` allocations, differences between regressors are no more often statistically significant than the smaller differences of `lmg` allocations. With `pmvd`, one has to be careful not to overinterpret differences, since even relatively large differences can occur by chance in case of moderate correlations between regressors; thus, the role of bootstrap confidence intervals is particularly important here.

Note that all metrics come in two different versions: for `rela=FALSE`, the scale of each metric is percentage of the response's variance. In this case, the sum over all regressors of `lmg`, `pmvd` and `pratt` respectively is just R^2. For `rela=TRUE` (default), all metrics are rescaled to sum to 100%.

4 The example data analysed with R-package relaimpo

In the following, we assume that the R-package `relaimpo` has been installed, either in the global version from CRAN or in the enhanced non-US version from http://www.tfh-berlin.de/ groemp/. The CRAN version does not contain `pmvd` that is not globally available because of a potential US patent issue. All programs with this file will use the non-US version. For the global version, you simply have to omit any usage of `pmvd`. The most basic analysis available is the calculation of the available relative importance metrics from the covariance matrix of the dataset, where the response variable is the first variable (like in the function `lm`). We now look once - for demonstration purposes - at all available metrics:

```
> metrics <- calc.relimp(swiss, type = c("lmg", "pmvd", "last", "first",
+     "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
```

We see that all metrics agree on the importance of `Education`. `lmg` and `first` have `Examination` not far behind, while the other methods see a clear difference here. All other regressors look relatively unimportant. Note that `pratt` in this example shows its inappropriate behavior of sometimes assigning a negative contribution to one or more of the regressors.

We can chart the results for a graphical look. Figure 1 (page 7) shows barplots of the calculated metrics and can be created by the `R` statement

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

It is also interesting to see whether the observed differences in relative importance are small or large relative to variability of the estimates. For this purpose, we look at results from bootstrap resampling for a reasonable collection of metrics. Bootstrapping in `relaimpo` is done by resampling the complete observations (cf. e.g. Fox, 2002) using the R-package `boot`. A call to the function `boot.relimp` requests the bootstrap runs the results of which are stored in an

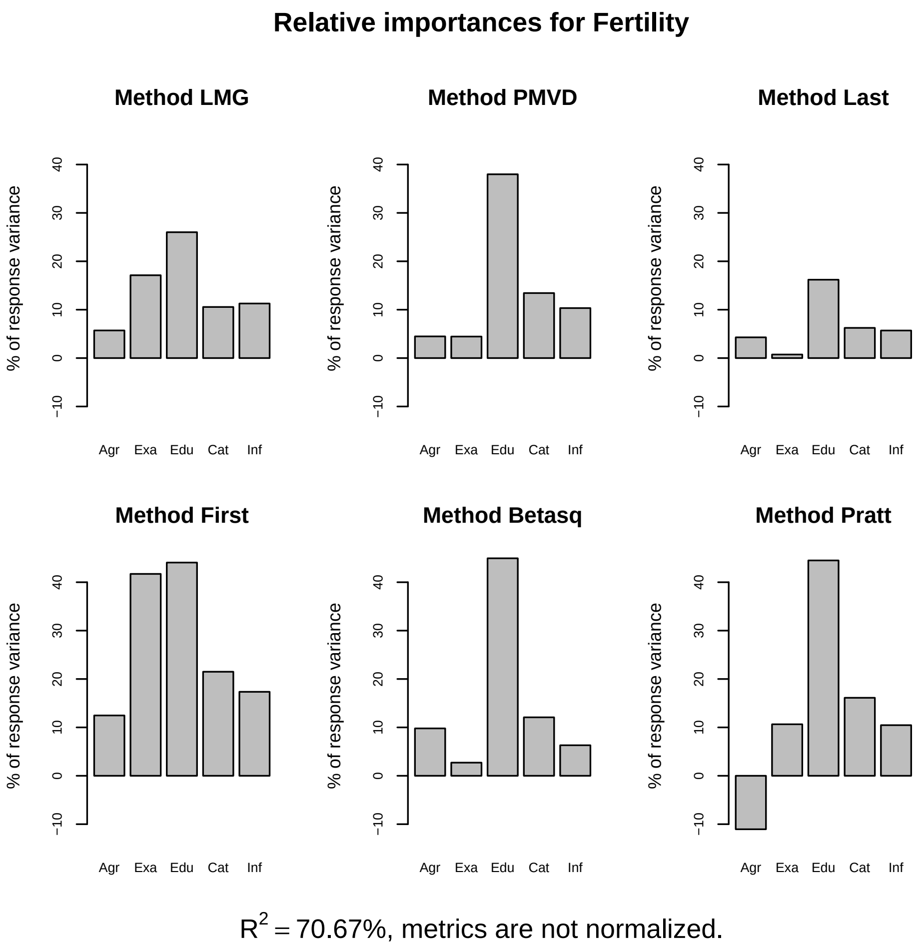

Figure 1: Bar plots of all calculated relative importance metrics.

object of class `relimplmboot`. (Warning: If you try out this code 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`.). Afterwards, the result object can be (repeatedly) evaluated with the function `booteval.relimp`. Inputs to `boot.relimp` are the response vector, the matrix (or data frame) of regressors, the number of bootstrap runs and the requested metrics. `booteval.relimp` works on the output from `boot.relimp` and allows selection of a subset of the metrics, selection of one or several confidence levels and a few further options (see the manual).

```
> bootresult <- boot.relimp(linmod, b = 1000, type = c("lmg", "pmvd", "last",
+     "first"))
> eval <- booteval.relimp(bootresult, typesel = c("lmg", "pmvd"), level = c(0.8,
+     0.9))
> 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= bca ):
Relative Contributions with confidence intervals:

                                          Lower         Upper
                      percentage 0.8   0.9   0.8    0.9    0.8    0.9
Agriculture.lmg       0.0570     ___DE __CDE 0.0305 0.0283 0.0761 0.0842
Examination.lmg       0.1711     ABC__ ABCD_ 0.1047 0.0906 0.2365 0.2547
Education.lmg         0.2601     ABC__ ABCD_ 0.1538 0.1174 0.3498 0.3851
Catholic.lmg          0.1055     _BCDE ABCDE 0.0513 0.0389 0.1864 0.2147
Infant.Mortality.lmg  0.1127     _BCDE ABCDE 0.0448 0.0329 0.1889 0.2088

Agriculture.pmvd      0.0447     __CDE __CDE 0.0140 0.0059 0.0763 0.0849
Examination.pmvd      0.0444     ABCDE ABCDE 0.0006 0.0001 0.2343 0.2919
Education.pmvd        0.3798     AB___ ABC__ 0.1995 0.1532 0.5742 0.6167
Catholic.pmvd         0.1343     _BCD_ ABCDE 0.0612 0.0438 0.3020 0.3541
Infant.Mortality.pmvd 0.1033     _BCDE _BCDE 0.0210 0.0119 0.2097 0.2407

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.


 Differences between Relative Contributions:

                                                   Lower         Upper
                                   difference 0.8 0.9 0.8    0.9    0.8    0.9
Agriculture-Examination.lmg        -0.114      *   *   -0.182 -0.206 -0.061 -0.047
Agriculture-Education.lmg          -0.203      *   *   -0.307 -0.343 -0.124 -0.092
Agriculture-Catholic.lmg           -0.048              -0.151 -0.185 0.0163 0.0409
Agriculture-Infant.Mortality.lmg   -0.055              -0.150 -0.179 0.0177 0.0376
Examination-Education.lmg          -0.088              -0.231 -0.277 0.0173 0.0496
Examination-Catholic.lmg           0.0656              -0.025 -0.059 0.1643 0.1934
Examination-Infant.Mortality.lmg   0.0584              -0.075 -0.108 0.1760 0.2034
Education-Catholic.lmg             0.1545              -0.008 -0.073 0.2919 0.3243
Education-Infant.Mortality.lmg     0.1473      *       0.0070 -0.035 0.2709 0.3025
Catholic-Infant.Mortality.lmg      -0.007              -0.098 -0.124 0.0927 0.1313

Agriculture-Examination.pmvd       0.0003              -0.148 -0.213 0.0668 0.0804
Agriculture-Education.pmvd         -0.335      *   *   -0.547 -0.591 -0.160 -0.108
Agriculture-Catholic.pmvd          -0.089      *   *   -0.291 -0.350 -0.021 -0.003
Agriculture-Infant.Mortality.pmvd -0.058               -0.178 -0.215 0.0334 0.0524
Examination-Education.pmvd         -0.335      *       -0.568 -0.607 -0.055 0.0559
Examination-Catholic.pmvd          -0.089              -0.293 -0.340 0.0218 0.1180
Examination-Infant.Mortality.pmvd -0.058               -0.217 -0.266 0.1193 0.2018
Education-Catholic.pmvd            0.2454      *       0.0182 -0.060 0.4813 0.5446
Education-Infant.Mortality.pmvd    0.2764      *   *   0.0653 0.0133 0.5210 0.5767
Catholic-Infant.Mortality.pmvd     0.0310              -0.085 -0.120 0.2142 0.2771

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

The bootstrapping functions have generated a substantial amount of output that is discussed now: First, all bootstrapped metrics are simply listed next to each other for reference. A second block of output shows bootstrapping results for individual relative importances for all requested metrics: Here (since ranks have been bootstrapped, and `norank` has not been set), apart from the confidence limits themselves, we find an indication which ranks are compatible with the bootstrap results. Looking at the 90% confidence level for `lmg`, for example, the output tells us, that `Education` and `Examination` are not last, while `Agriculture` is neither first nor second. 90% confidence intervals for `pmvd` agree with `lmg` that `Agriculture` is neither first nor second and that `Education` is not last. In addition, `Infant.Mortality` is assessed to be not first, and `Examination` can be in any position.

The next block of output gives exploratory bootstrap-based confidence intervals of pairwise differences of contributions. Here, according to `lmg`, the only significant differences obtained from 90% confidence are those between `Agriculture` (as the weakest regressor) and `Education` and `Examination` (as the strongest regressors). For `pmvd`, the differences between `Education` and all other regressors except `Catholic` is considered significant, when referring to the 90% confidence intervals.

80% confidence intervals find additional significant differences. Note, however, that bootstrap confidence intervals can be somewhat liberal so that too much reliance especially on intervals with low confidence levels is not recommended. (More research on the behavior of the bootstrap intervals is needed.) Let us now look graphically at the bootstrap output: So far, the only available graphic is a barplot with confidence indication, which for this example can be created by the code

```
> par(cex.axis = 0.7, cex.sub = 0.8)
> plot(booteval.relimp(bootresult, typesel = c("lmg", "pmvd"), level = 0.9),
+     names.abbrev = 3)
```

The resulting Figure 2 (page 11) supports the perception that there is qualitative agreement on dominance of `Education` and low importance of `Agriculture`, together with medium importance of `Catholic` and `Infant.Mortality`, plus severe disagreement on `Examination`. Knowing that `pmvd` is a metric that assigns a (close to) zero share to regressors with coefficient 0 (cf. Section 3), this picture might indicate that `Examination` is a consequence of `Education` only and is not important in its own right. However, the `pmvd` contribution estimate for `Examination` is extremely variable, so that any interpretation is tentative only.

5 Comparison to hier.part

Now, let us turn to comparison of the possibilities of packages `relaimpo` and `hier.part`. For some of the code in this section, you need to install `hier.part` if you want to run it. Package `hier.part` implements hierarchical partitioning, a term introduced by Chevan and Sutherland (1991) for (unweighted) averaging over orderings in a variety of models. `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.

Let us now look at the standard output of `hier.part` for the example data (barplot turned off, since it looks almost the same as barplot in `relaimpo`).

```
> 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
[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
```

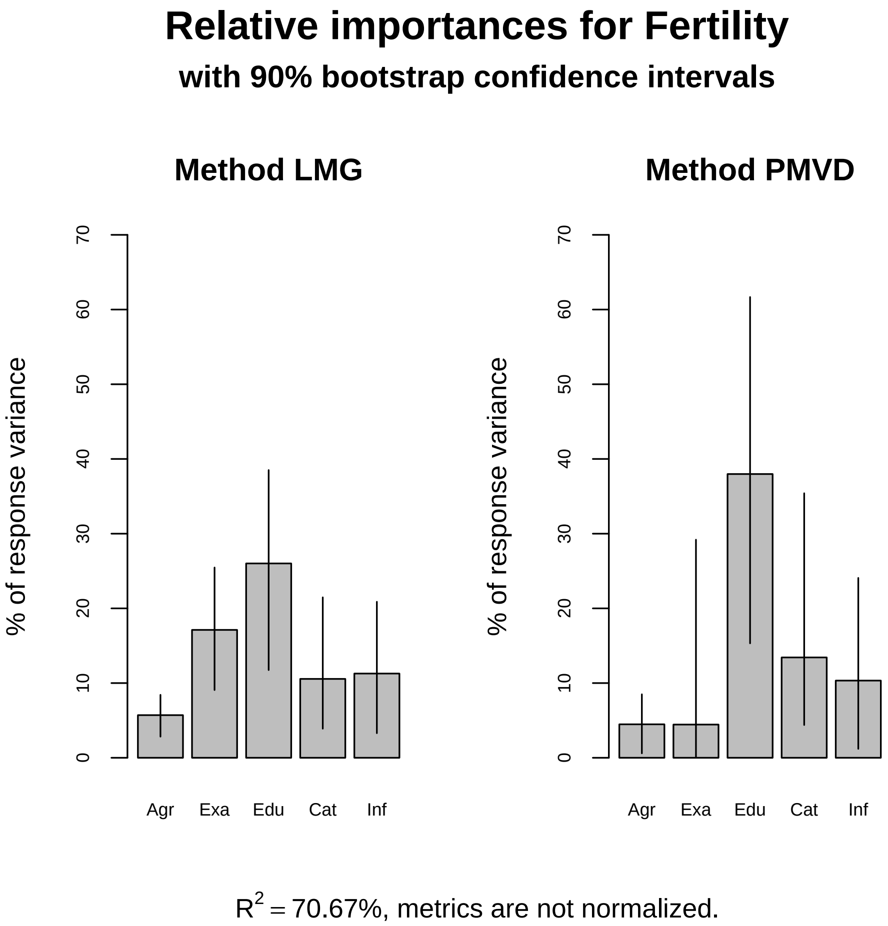

Figure 2: Bar plots of lmg and pmvd with confidence intervals.

```
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`) simply 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 little program illustrates how we can reproduce the relevant portion of the output from `hier.part` using `relaimpo`:

```
> interim <- calc.relimp(linmod, rela = F, type = c("lmg", "first"))
> mat <- cbind(I = interim$lmg, J = interim$first - interim$lmg,
+     Total = interim$first)
> mat

                          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

> interim <- calc.relimp(linmod, rela = T, type = "lmg")
> matrix(100 * interim$lmg, 5, 1, dimnames = list(interim$namen[2:6],
+     "I.perc"))

                    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*mat[,1]/sum(mat[,1]))`.

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.

Table 1: CPU times in seconds from 100 runs each for p equi-correlated regressors with variances 1 and pairwise correlations 0.5

	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

6 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 1 (page 13) 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 which is only calculated once while `hier.part` calculates $2^p - 1$ regression models using all observations. For `relaimpo`, we see that `pmvd` takes longer than `lmg` for large numbers of regressors p. 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). There may be some potential in making the calculations for `pmvd` more efficient (internal function `pmvdcalc`, suggestions welcome).

Bootstrapping obviously makes computation times a real issue, if many bootstrap runs are required. The recommended BCa bootstrap intervals (`bty="bca"` in `booteval.relimp`, default) require very large numbers of bootstrap runs (default: `b=1000`) 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 based on a smaller number of bootstrap runs. 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). This is the reason for the warning in the output. Performance of BCa intervals has not been simulated (since they take so much longer); they might well perform better.

7 Final Remarks

The functionality of R-package `relaimpo` has been explained and illustrated in this vignette, using the data set `swiss` from the `R` datasets. This dataset has a complicated correlation structure among regressors which makes assessment of relative relative importances somewhat ambiguous. R-package `relaimpo` broadens R's possibilities of assessing relative importances in linear models: It provides the additional metric `pmvd` (in the non-US version). Also, `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. Further references can be found on http://www.tfh-berlin.de/ groemp/.

REFERENCES

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

Darlington, R.B. (1968) Multiple regression in psychological research and practice. *Psychological Bulletin* **69**, 161–182.

Feldman, B. (2005) Relative Importance and Value. Manuscript (Version 1.1, March 19 2005), downloadable at http://www.prismanalytics.com/docs/RelativeImportance.pdf.

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.

Hoffman, P.J. (1960) The paramorphic representation of clinical judgment. *Psychological Bulletin* **57**, 116–131.

Lindeman, R.H., Merenda, P.F. and Gold, R.Z. (1980) *Introduction to Bivariate and Multivariate Analysis*, Glenview IL: Scott, Foresman.

Pratt, J.W. (1987) Dividing the indivisible: Using simple symmetry to partition variance explained. In: Pukkila, T. and Puntanen, S. (Eds.): *Proceedings of second Tampere conference in statistics*, University of Tampere, Finland, 245–260.

R Development Core Team (2005) *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.