Linear edit manipulation and error localization with the `editrules` package

Edwin de Jonge and Mark van der Loo

May 2, 2011

Abstract

This vignette is far from finished. Version 1.0 fo the package will have the full vignette. `editrules` is a package to define, parse, manipulate and check linear and other data restrictions in R. Verbose restrictions can be entered at the commandline or stored as a (text) file or a database. This vignette is under construction.

Contents

1 Introduction

The value domain of real numerical data records with n variables is often restricted to a subdomain of $\mathbb{R}^n$ due to linear equality and inequality relations which the variables in the record have to obey. Examples include equality restrictions imposed by financial balance accounts, positivity demands on certain variables or limits on the ratio of variables.

Any such restriction is of the form

$$\mathbf{a} \cdot \mathbf{x} \odot b \text{ with } \odot \in \{<, \leq, =\}, \tag{1}$$

where $\mathbf{x}$ is a numerical data record, $\mathbf{a}$, $\mathbf{x} \in \mathbb{R}^n$ and $b \in \mathbb{R}$. In data editing literature, data restriction rules refered to as *edits*, or *edit rules*. We will call edits, written in the form (1), edits in *normal form*.

Large complex surveys are often endowed with dozens or even hundreds of edit rules. For example, the Dutch Structural Business Survey, which aims to report on the financial structure of companies in the Netherlands, contains on the order of 100 variables, endowed with a similar number of linear equality and inequality restrictions.

Defining and manipulating large edit sets can be a daunting task without convenient tools. Also, in practice on will frequently encounter records not obeying all edit rules, which gives rise to the error localization problem: which variables contain the errors that cause a record to violate certain edits?

The `editrules` package for the R statistical computing environment (R Development Core Team, 2011) aims to provide an environment to conveniently define, parse and check linear (in)equality restrictions, perform common edit manipulations and offer error localization functionality based on the (generalized) paradigm of Fellegi and Holt (1976). This paradigm is based on the assumption that errors are distributed randomly over the variables, and there is no detectable cause of error. The paradigm also decouples the detection from correction of corrupt variables. Certain causes of error, such as sign flips, typing errors or rounding errors can be detected and are closely related to their resolution. The reader is referred to the `deducorrect` package (van der Loo et al., 2011; Scholtus, 2008, 2009) for treating such errors.

The following chapters demonstrates the functionality of the `editrules` package with coded examples as well a description of of the underlying theory and algorithms. For a detailed per-function description the reader is referred to the reference manual accompanying the package. Unless mentioned otherwise, all code shown in this paper can be executed from the R commandline after loading the `editrules` package.

2 Defining and testing numerical restrictions

2.1 The `editmatrix` object

For computational processing, a set of edits of the form in Eq. (1) is most conveniently represented in matrix form. Defining such matrices in code is tedious and prone to error. The `editmatrix` function can be used to verbosely define a set of linear relations in any of the forms

$$\mathbf{a} \cdot \mathbf{x} \odot b \text{ with } \odot \in \{<, \leq, =, \geq, >\}. \tag{2}$$

For example, suppose we have record set with the variables

turnover	t
personell cost	c_p
housing cost	c_h
total cost	c_t
profit	p,

subject to the rules

$$
\begin{aligned}
t &= c_t + p && (3)\\
c_t &= c_h + c_p && (4)\\
p &\leq 0.6t && (5)\\
c_t &\leq 0.3t && (6)\\
c_p &\leq 0.3t && (7)\\
t &> 0 && (8)\\
c_h &> 0 && (9)\\
c_p &> 0 && (10)\\
c_t &> 0. && (11)
\end{aligned}
$$

Here, the equality restrictions correspond to balance accounts, the 3rd, 4th and 5th can be considered sanity checks and the last four edits demand positivity of certain variables. Such a set of edits is stored as an `editmatrix` object. It can be declared as follows:

```
> (E <- editmatrix(c(
+ "t == ct + p",
+ "ct == ch + cp",
+ "p <= 0.6*t",
+ "cp <= 0.3*t",
+ "ch <= 0.3*t",
+ "t >0",
+ "ch > 0",
+ "cp > 0",
+ "ct > 0"), normalize=TRUE))

Edit matrix:
   ct  p    t ch cp Ops CONSTANT
e1 -1 -1  1.0  0  0  ==        0
e2  1  0  0.0 -1 -1  ==        0
e3  0  1 -0.6  0  0  <=        0
e4  0  0 -0.3  0  1  <=        0
e5  0  0 -0.3  1  0  <=        0
e6  0  0 -1.0  0  0   <        0
e7  0  0  0.0 -1  0   <        0
e8  0  0  0.0  0 -1   <        0
e9 -1  0  0.0  0  0   <        0

Edit rules:
e1 : t == ct + p
e2 : ct == ch + cp
e3 : p <= 0.6*t
```

```
e4 : cp <= 0.3*t
e5 : ch <= 0.3*t
e6 : 0 < t
e7 : 0 < ch
e8 : 0 < cp
e9 : 0 < ct
```

The `editmatrix` object stores the linear relations as an augmented matrix $[\mathbf{A}, \mathbf{b}]$ and a character vector containing the operators. Most functions of the `editrules` package expect `E` to be in normal form. This is achieved by the optional argument `normalize=TRUE` in the previous statement. Alternatively, an `editmatrix` can be normalized by with the `normalize` function. The function `isNormalized` can be used to test whether an `editmatrix` is normalized.

As can be seen, the editmatrix object is presented as a matrix, as well as a set of textual edit rules. The `editrules` package is capable of coercing a set of R expressions to an `editmatrix` and *vice versa*. Internally, the `editmatrix` function processes the parsetree of the textual R expressions as provided by the R internal `parse` function.

In the example, the edits were automatically named `e1`, `e2`, ..., `e9`. It is possible to name and comment edits by reading them from a `data.frame`.

```
> # generate a csv text string
> E.csv <-
+ 'name , edit       , description
+ "b1"  ,    t == ct + p    ,  "total balance"
+ "b2"  ,    ct == ch + cp  ,  "cost balance"
+ "s1"  ,    p <= 0.6*t     ,  "profit sanity"
+ "s2"  ,    cp <= 0.3*t    ,  "personell cost sanity"
+ "s3"  ,    ch <= 0.3*t    ,  "housing cost sanity"
+ "p1"  ,    t >0           ,  "turnover positivity"
+ "p2"  ,    ch > 0         ,  "housing cost positivity"
+ "p3"  ,    cp > 0         ,  "personel cost positivity"
+ "p4"  ,    ct > 0         ,  "total cost positivity"'
> # read into a data.frame
> E.df <- read.csv(textConnection(E.csv))
> # transform to an editmatrix
> editmatrix(E.df)

Edit matrix:
     ct  p    t ch cp Ops CONSTANT
b1   -1 -1  1.0  0  0  ==        0
b2    1  0  0.0 -1 -1  ==        0
s1    0  1 -0.6  0  0  <=        0
s2    0  0 -0.3  0  1  <=        0
s3    0  0 -0.3  1  0  <=        0
p1    0  0  1.0  0  0   >        0
p2    0  0  0.0  1  0   >        0
p3    0  0  0.0  0  1   >        0
p4    1  0  0.0  0  0   >        0

Edit rules:
b1   : t == ct + p [   total balance ]
b2   : ct == ch + cp [   cost balance ]
s1   : p <= 0.6*t [   profit sanity ]
```

```
s2   : cp <= 0.3*t [   personell cost sanity ]
s3   : ch <= 0.3*t [   housing cost sanity ]
p1   : t > 0 [   turnover positivity ]
p2   : ch > 0 [   housing cost positivity ]
p3   : cp > 0 [   personel cost positivity ]
p4   : ct > 0 [   total cost positivity ]
```

The ability to read edit sets from a `data.frame` facilitates defining and maintaining the rules outside of the R environment, by storing them in a user-filled database for example. Note that manipulating and combining edits, for example through elimination methods will cause `editrules` to drop or change the names and drop the comments, as they become meaningless after certain manipulations.

2.2 Some simple manipulations

Edit checking, retrieving the matrix, operators and coefficients...

3 Manipulation of linear restrictions

3.1 Value substitution

3.2 Gaussian elimination

3.3 Fourier-Motzkin elimination

4 Error localization for numerical data

4.1 The generalized Fellegi-Holt paradigm

4.2 General binary search with the `choicepoint` algorithm

4.3 Error localization with `cp.editmatrix`

5 Conclusions

References

Fellegi, I. P. and D. Holt (1976). A systematic approach to automatic edit and imputation. *Journal of the Americal Statistical Association 71*, 17–35.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Scholtus, S. (2008). Algorithms for correcting some obvious inconsistencies and rounding errors in business survey data. Technical Report 08015, Statistics Netherlands, Den Haag. The papers are available in the inst/doc directory of the R package or via the website of Statistics Netherlands.

Scholtus, S. (2009). Automatic correction of simple typing error in numerical data with balance edits. Technical Report 09046, Statistics Netherlands, Den Haag. The papers are available in the inst/doc directory of the R package or via the website of Statistics Netherlands.

van der Loo, M., E. de Jonge, , and S. Scholtus (2011). *deducorrect: Deductive correction of simple rounding, typing and sign errors*. R package version 0.9-2.