Package ‘rspa’

June 19, 2019

Maintainer  Mark van der Loo <mark.vanderloo@gmail.com>
License  GPL-3
Title  Adapt Numerical Records to Fit (in)Equality Restrictions
Type  Package
LazyLoad  yes
Author  Mark van der Loo
Description  Minimally adjust the values of numerical records in a data.frame, such that each record satisfies a predefined set of equality and/or inequality constraints. The constraints can be defined using the ‘validate’ package. The core algorithms have recently been moved to the ‘lintools’ package, refer to ‘lintools’ for a more basic interface and access to a version of the algorithm that works with sparse matrices.

Version  0.2.5
Depends  R (&gt;= 2.13.0)
Imports  graphics, stats, validate, lintools
Suggests  editrules, tinytest
URL  https://github.com/markvanderloo/rspa
BugReports  https://github.com/markvanderloo/rspa/issues
RoxygenNote  6.1.1
Encoding  UTF-8
NeedsCompilation  yes
Repository  CRAN
Date/Publication  2019-06-19 15:40:03 UTC

R topics documented:

  rspla-package ........................................... 2
  match_restrictions ...................................... 2
  remove_tag ........................................... 4
  tagged_values ......................................... 4
  tag_missing ........................................... 5
**match_restrictions**

Alter numeric data records to match linear (in)equality constraints.

**Description**

Apply the successive projection algorithm to adjust each record in dat to satisfy a set of linear (in)equality constraints.

**Usage**

```r
match_restrictions(dat, restrictions, adjust, weight = rep(1, ncol(dat)), remove_tag = TRUE, ...)
```

**Arguments**

- `dat` A data.frame
- `restrictions` An object of class `validator`
- `adjust` (optional) A logical matrix of dimensions `dim(dat)` where TRUE indicates that a value may be adjusted. When missing, the `tagged_values` are used. If no tagging was applied, adjust will default to an all TRUE matrix with dimensions equal to `dim(dat)`.
- `weight` A weight vector of length `ncol(dat)` or a matrix of dimensions `dim(dat)`. 
- `remove_tag` if a value position indicator is present, remove it?
- `...` arguments passed to `project`.

**Overview**

Given a vector $x^0$, and a set linear restrictions of the form $a_i \cdot x_i = b_i$ and/or $a_i \cdot x_i \leq b_i$ with $i = 1, 2, \ldots, m$. This package finds the nearest vector to $x^0$ (in the (weighted) euclidean sense) that satisfies all restrictions.

Much of this package’s functionality, including algorithms for working with large, sparse problems has been moved to the `lintools` package. This package will serve as a front-end for application of the successive projection algorithm for data stored in data.frame like objects.
Value
dat, with values adapted.

Note on inequality restrictions
All inequality restrictions of the form $a \cdot x < b$ are treated as $a \cdot x \leq b$. The idea is to project the original record $x$ onto the boundary defined by the (in)equations. Projection on a boundary defined by a strict inequation is illdefined since the value $b$ in the restriction $a \cdot x < b$ is strictly outside the valid region.

See Also
tag_missing

Examples

# a very simple adjustment example
v <- validate::validator(
x + y >= 10,
x > 0,
y > 0
)

# x and y will be adjusted by the same amount
match_restrictions(data.frame(x=4,y=5), v)

# One of the inequalities violated
match_restrictions(data.frame(x=-1,y=5), v)

# Weighted distances: 'heavy' variables change less
match_restrictions(data.frame(x=4,y=5), v, weight=c(100,1))

# if w=1/x0, the ratio between coefficients of x0 stay the same (to first order)
x0 <- data.frame(x=4,y=5)
x1 <- match_restrictions(x0, v, weight=1/as.matrix(x0))

x0[,1]/x0[,2]
x1[,1] / x1[2]

# example of tag usage
v <- validate::validator(x + y >= 1, x>0,y>0)
d <- data.frame(x=NA,y=0.5)
d <- tag_missing(d)
# impute
d[1,1] <- 1

# only the tagged values will be altered. The tag is
# removed afterwards.
mATCH_restrictions(d,v)
remove_tag  

**Description**  
Remove cell position tags

**Usage**  
`remove_tag(dat, ...)`

**Arguments**  
- `dat` [data.frame]  
- `...` Currently not used

**Value**  
`dat` with tag removed

**See Also**  
Other tagging: `tag_missing`, `tagged_values`

---

tagged_values  

**Description**  
Retrieve tagged cell positions

**Usage**  
tagged_values(dat, ...)

**Arguments**  
- `dat` [data.frame]  
- `...` Currently not used

**Value**  
A logical matrix, or NULL

**See Also**  
Other tagging: `remove_tag`, `tag_missing`
Tag currently missing elements of a data.frame

Description
Attach an attribute that marks which cells are empty (NA).

Usage
tag_missing(dat, ...)

Arguments
dat [data.frame] to be tagged
... Currently not used.

Value
dat, tagged for missing values.

See Also
Other tagging: remove_tag, tagged_values
Index

match_restrictions, 2
project, 2
remove_tag, 4, 4, 5
rspa-package, 2
tag_missing, 3, 4, 5
tagged_values, 2, 4, 4, 5
validator, 2