

# Package ‘miscset’

September 4, 2014

**Type** Package

**Title** Miscellaneous Tools Set

**Version** 0.4

**Date** 2014-09-04

**Author** Sven E. Templer

**Maintainer** Sven E. Templer <sven.templer@gmail.com>

**Description** A set of miscellaneous R tools to simplify the work with several data types and formats.

**License** GPL-3

**Imports** parallel, data.table, gridExtra, Rcpp (>= 0.11.1), xtable

**LinkingTo** Rcpp

**Suggests** knitr

**VignetteBuilder** knitr

**URL** <https://github.com/setempler/miscset>

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2014-09-04 07:00:28

## R topics documented:

miscset-package . . . . .	2
enpaire . . . . .	2
gapply . . . . .	3
gghcl . . . . .	5
ggplotlist . . . . .	6
gregexprind . . . . .	7
leading0 . . . . .	8

lload . . . . .	8
lsall . . . . .	9
mgrepl . . . . .	10
ntri . . . . .	11
rmail . . . . .	12
scale0 . . . . .	13
squarematrix . . . . .	13
strextr . . . . .	14
strpart . . . . .	15
strrev . . . . .	16
textable . . . . .	17

## Index 19

---

miscset-package	<i>Miscellaneous R Tools</i>
-----------------	------------------------------

---

### Description

Set of miscellaneous R tools to simplify the work with several data types and formats.

### Details

The package vignette provides a comprehensive overview and examples for the usage of all available functions in the package. Open with `vignette("miscset")`.

### Author(s)

Sven E. Templer <[sven.templer@gmail.com](mailto:sven.templer@gmail.com)>

---

enpaire	<i>Create a Pairwise List from a Matrix</i>
---------	---

---

### Description

Transform a `matrix` or `dist` object to a pairwise list.

### Usage

```
enpaire(x, ...)

## Default S3 method:
enpaire(x, ...)

## S3 method for class 'dist'
enpaire(x, upper = T, lower = T, ...)

## S3 method for class 'matrix'
enpaire(x, upper = T, lower = T, ...)
```

**Arguments**

x	Object of class <code>matrix</code> .
...	Arguments passed to methods.
upper	Logical, return values from upper triangle.
lower	Logical, return values from lower triangle.

**Value**

Returns a `data.frame`. The first and second column represent the dimension names for a value in `x`. The following columns contain the values for the upper or lower triangle.

**Author(s)**

Sven E. Templer (<sven.templer@gmail.com>)

**See Also**

[squarematrix](#)

**Examples**

```
#  
  
m <- matrix(letters[1:9], 3, 3, dimnames = list(1:3,1:3))  
enpaire(m)  
enpaire(m, lower = FALSE)  
  
#
```

---

gapply	<i>Apply a Function by a Grid on a List</i>
--------	---

---

**Description**

`gapply` applies a function on a `data.frame` by using named list entries as grid. Support of multicore processing via `mclapply`.  
`levels` is a method to retrieve the grid of a `gapply` object.  
`as.data.frame` is a simplified `rbind` version which adds the grid to the output.

**Usage**

```
gapply(d, by, fun, ..., drop = TRUE, cores = 1)  
  
## S3 method for class 'gapply'  
levels(x)  
  
## S3 method for class 'gapply'
```

```
as.data.frame(x, ...)  
  
## S3 method for class 'gapply'  
as.data.table(x, keyv = NULL, ...)
```

### Arguments

d	A data.frame or data.table object.
by	A character vector with colnames for the grid to subset.
fun	A function to apply on d.
...	Additional arguments to fun or to as.data.frame method.
drop	Drop grid columns for fun.
cores	Number of multicores as mc.cores in mclapply.
x	Object of class gapply.
keyv	Character vector for data.table keys.

### Details

The output of `gapply` is a list inheriting from the S3 class `gapply` which stores the grid of the original data. Support for subsetting grid entries is not yet supported. The function `fun` receives the subset of the data frame as first argument.

### Author(s)

Sven E. Templer (<[sven.templer@gmail.com](mailto:sven.templer@gmail.com)>)

### See Also

[mclapply](#), [setkeyv](#)

### Examples

```
#  
  
f <- function (x) c(conc.diff = diff(range(x$conc)), uptake.sum=sum(x$uptake))  
d <- gapply(CO2, c('Type', 'Treatment'), f)  
d  
levels(d)  
as.data.frame(d)  
  
#
```

---

`gghcl`*HTML Colours Like ggplot2*

---

**Description**

Calculate HTML colour code from a palette like ggplot2 uses.

**Usage**

```
gghcl(n, sub = 1:n, h = c(0, 360) + 15, c = 100, l = 65, ...)
```

**Arguments**

<code>n</code>	Numeric value to determine size of palette.
<code>sub</code>	Numeric vector with values within range from 1 to n to subset palette.
<code>h</code>	Hue of the colour. Within range of a circle's degrees.
<code>c</code>	Chroma of the colour.
<code>l</code>	Luminance of the colour. Within range from 1 to 100.
<code>...</code>	Further arguments passed to function <code>hcl</code> .

**Details**

See `?hcl` for explanation of `h`, `c` and `l`.

**Value**

Returns a character vector containing HTML colour code of the standard ggplot colour palette.

**Author(s)**

Sven E. Templer (<sven.templer@gmail.com>)

**See Also**

[hcl](#)

**Examples**

```
#  
  
# Plot some palettes:  
par(mfrow = c(3,1), mai = c(.1,.1,1,.1))  
p <- matrix(1:10, 10)  
image(p, col = gghcl(5), axes = FALSE, main = "gghcl(5)")  
image(p, col = gghcl(10), axes = FALSE, main = "gghcl(10)")  
image(p, col = gghcl(10, 1:5), axes = FALSE, main = "gghcl(10, 1:5)")  
# dev.off() # to reset \code{par}  
  
#
```

---

`ggplotlist`*Arrange a List of ggplots*

---

**Description**

Arrange a list of ggplots with `grid.arrange` and output on local graphic device or as pdf/png when a path is supplied.

**Usage**

```
ggplotlist(x, ncol = 1, path, width = 11, height = 8)
```

**Arguments**

<code>x</code>	A list containing at least one ggplot object of class gg.
<code>ncol</code>	Number of columns in the grid.
<code>path</code>	Optional. A character string that gives the path to export the plot to a file, ending with 'pdf' or 'png' (case insensitive). If missing, then the grid is returned to the current graphic device.
<code>height,width</code>	A numeric value in inches for the size of the output file in case path is provided. Else ignored.

**Author(s)**

Sven E. Templer <sven.templer@gmail.com>

**Examples**

```
#  
  
## Not run:  
library(ggplot2)  
x <- list(  
  ggplot(data.frame(a=1:5,b=1:5), aes(x=a,y=b,col=b)) + geom_line(),  
  ggplot(data.frame(a=1:5,b=1:5), aes(x=a,y=b,shape=factor(b))) + geom_point())  
ggplotlist(x, 2)  
## End(Not run)  
  
#
```

**Description**

Function to extract a certain index from `gregexpr()`.

**Usage**

```
gregexprind(pattern, text, n, ...)
```

**Arguments**

<code>pattern</code>	Character string containing a regular expression to be searched in <code>text</code> .
<code>text</code>	Character vector where the search is performed.
<code>n</code>	Numeric value or character string "last" to extract nth or last position of <code>pattern</code> in each value of <code>text</code> .
<code>...</code>	Arguments passed to function <code>gregexpr()</code> .

**Details**

See `?gregexpr` for further information on arguments. See `?regular expression` for the use of regular expressions.

**Value**

Numeric vector of length `length(text)`.

**Author(s)**

Sven E. Templer (<[sven.templer@gmail.com](mailto:sven.templer@gmail.com)>)

**See Also**

[gregexpr](#), regular expression ([regex](#))

**Examples**

```
#
gregexprind(c("a"),c("abab","ab","xyz",NA), 1)
gregexprind(c("a"),c("abab","ab","xyz",NA), 2)
gregexprind(c("a"),c("abab","ab","xyz",NA), "last")
#
```

leading0

*Numeric to Character with Leading Zero(s)*

---

**Description**

Transform numeric values to character string prepending leading zero(s).

**Usage**

```
leading0(num, digits = 2)
```

**Arguments**

num                    Numeric vector (character also possible) to transform.  
digits                 Numeric value of minimum length of output strings.

**Value**

Character vector with same length of strings of each value. Original "string" is prepended by zero(s). String length is at least `max(nchar(as.character(num)))`.

**Author(s)**

Sven E. Templer <svен.templer@gmail.com>

**Examples**

```
#  
  
# use with paste to generate strings of equal size:  
paste0("observation", leading0(1:10, 3))  
  
#
```

---

lload*Load RData Objects to a List*

---

**Description**

Load multiple .RData files and return a (simplified) list.

**Usage**

```
lload(path = ".", pattern = ".RData", recursive = FALSE,  
      simplify = TRUE, verbose = TRUE)
```



**Arguments**

path	Character string with the path, as used in <a href="#">list.files</a> .
pattern	A regular expression for file name patterns, as used in <a href="#">list.files</a> .
recursive	Logical. Search the path recursive.
simplify	Logical, unlist when there are only unique object names.
verbose	Logical. Print information on screen about loading process.

**Value**

Returns a list of length n, when there are n data files loaded. All objects are stored in sublists. Names are according to files, and names of sublists to objects per file. If simplified, the list is of length m, when there are m objects in total loaded.

**Author(s)**

Sven E. Templer (<[sven.templer@gmail.com](mailto:sven.templer@gmail.com)>)

**See Also**

[load](#)

**Examples**

```
#
```

---

```
lsall
```

*List Object Details*

---

**Description**

Return a data.frame with a list of all objects of a specified environmet.

**Usage**

```
lsall(envir = .GlobalEnv, ...)
```

**Arguments**

envir	An environment where to look for objects.
...	Arguments forwarded to ls.

**Value**

Returns a data.frame with object names, lengths, classes, modes and sizes or NULL if the environment is empty.

**Author(s)**

Sven E. Templer <sven.templer@gmail.com>

**See Also**

[ls](#)

**Examples**

```
#  
  
lsall()  
obj1 <- 1:3  
obj2 <- data.frame(1:3)  
obj3 <- list(1:3)  
lsall()  
  
#
```

---

mgrepl

*Multiple Pattern Matching and Replacement*

---

**Description**

mgrepl searches for any or all patterns and returns logical values. Combination of the results is done via the logic functions any or all. Multicore feature is made available by parallel::mclapply. msub and mgsub replace multiple patterns in a vector by the order of

**Usage**

```
mgrepl(patterns, text, log.fun = any, cores = 1, ...)
```

```
msub(patterns, replacement, text, ...)
```

```
mgsub(patterns, replacement, text, ...)
```

**Arguments**

patterns	A character vector containing a regular expression ( <a href="#">regex</a> ) to be searched in text.
text	Character vector where the search and replace is performed.
log.fun	For mgrepl. Logical function (any or all) to evaluate occurrence of each pattern in patterns in each value of text. Can also be custom. See examples.
cores	Numeric value for how many cores to use for computation using mclapply.
replacement	Character vector with the replacement pattern.
...	Further arguments passed to functions grepl(), sub() and gsub().

**Value**

msub and mgsub:

Same as text but with all patterns replaced.

mgrepl:

Logical vector of sam length as text where TRUE means either any or all patterns in patternlist are matched in text depending on log.fun.

**Author(s)**

Sven E. Templer (<sven.templer@gmail.com>)

**See Also**

[grep](#), [mclapply](#)

**Examples**

```
#

# Compare different "log.fun" parameters:
mgrepl(c("a","b"), c("ab","ac","bc"), any)
mgrepl(c("a","b"), c("ab","ac","bc"), all)
mgrepl(letters[1:3], c("ax","xab","xbc"), function (x) sum(x)>1)

# Replace several patterns (globally):
msub(letters[1:3], "<replaced>", letters[1:5])
mgsub(letters[1:2], "<replaced>", c("ab","ba","acb",NA))

#
```

---

ntri

*Return Triangular Numbers*


---

**Description**

Return the series of triangular (/triangle) numbers up to a number of n rows of a triangle. The series has the entry number "A000217" at <https://oeis.org/A000217> and starts like this: 0, 1, 3, 6, 10, ...

**Arguments**

n Positive integer value for sequence length.

**Value**

Returns an integer vector of length n.

**Author(s)**

Sven E. Templer (<sven.templer@gmail.com>)

---

rmall

*Remove All Objects from Global Environment*

---

**Description**

Remove all objects from the global environment.

**Usage**

```
rmall(...)
```

**Arguments**

... Arguments forwarded to ls to get all objects.

**Author(s)**

Sven E. Templer <sven.templer@gmail.com>

**See Also**

[rm](#), [ls](#)

**Examples**

```
#  
  
a <- b <- letters  
ls()  
rmall()  
ls()  
  
#
```

---

scale0	<i>Scale to Range 0 to 1</i>
--------	------------------------------

---

**Description**

Scale numeric values to a range from 0 to 1.

**Usage**

```
scale0(x)
```

**Arguments**

x                    Numeric vector to transform.

**Author(s)**

Sven E. Templer <sven.templer@gmail.com>

**Examples**

```
#  
  
scale0(0:10)  
scale0(-1:3)  
scale0(2:3)  
  
#
```

---

squarematrix	<i>Create a Square Matrix</i>
--------------	-------------------------------

---

**Description**

Transform any m x n matrix to a square matrix by column/row names. Stops if no or duplicated dimnames are provided in x.

**Usage**

```
squarematrix(x)
```

**Arguments**

x                    Object of class matrix.

**Value**

Returns a matrix.

**Author(s)**

Sven E. Templer (<sven.templer@gmail.com>)

**Examples**

```
#  
  
m <- matrix(1:6, 2, dimnames=list(2:3,1:3))  
m  
squarematrix(m)  
  
#
```

---

strext

*Extract a Substring*

---

**Description**

This function extracts substring(s) which match a given pattern.

**Usage**

```
strext(x, pattern, sep = " ", mult = F, unlist = F, cores = 1)
```

**Arguments**

x	Character vector.
pattern	Regular expression.
sep	Character string which separates the fields.
mult	Logical, if multiple matching fields should be returned, or otherwise NA.
unlist	Logical, unlists multiple results.
cores	Integer for number of computational cores to use.

**Value**

A list of character vectors containing the substrings that are matching pattern and are separated by sep or NA if the pattern could not be found.

**Author(s)**

Sven E. Templer (<sven.templer@gmail.com>)

**Examples**

```
#  
  
s <- c("A1 B1 C1", "A2 B2", "AA A1", "AA", "B1 A1", "BB AB A1")  
strexp(s, "[AB][[:digit:]]$")  
strexp(s, "[AB][[:digit:]]$", mult = TRUE)  
strexp(s, "[AB][[:digit:]]$", mult = TRUE, unlist = TRUE)  
strexp(s, "[C][[:digit:]]$")  
  
#
```

---

**strpart***Split String and Return Part*

---

**Description**

Return the nth part of a splitted string.

**Usage**

```
strpart(x, split, n, ..., roll = F)
```

**Arguments**

x	Character vector.
split	Regular expression splitting strings.
n	Number of part to extract.
...	Arguments passed to <code>strsplit</code> .
roll	Logical, if to use the last when less than maximum parts.

**Value**

A character vector of the extracted parts.

**Author(s)**

Sven E. Templer (<[sven.templer@gmail.com](mailto:sven.templer@gmail.com)>)

**See Also**

[strsplit](#)

**Examples**

```
#  
  
strpart(c("abc", "abcb", "abc"), "", 4)  
strpart(c("abc", "abcb", "abc"), "", 4, roll=TRUE)  
  
#
```

---

strrev

*Reverse Text Strings*

---

**Description**

Create a reverse version of strings.

**Usage**

```
strrev(x)
```

**Arguments**

x                   vector with strings. Is coerced to character.

**Value**

Returns a character vector with reversed strings.

**Author(s)**

Sven E. Templer (<sven.templer@gmail.com>)

**See Also**

[rev](#)

**Examples**

```
#  
  
strrev(c("abc", "asdf"))  
  
#
```



---

`textable`*Table to Latex*

---

**Description**

This function enhances `xtable` as it writes the latex code to a file and adds a header/footer for the document structure to allow a direct creation of a pdf with for example `pdflatex`.

**Usage**

```
textable(d, file, caption = NULL, rownames = FALSE, landscape = FALSE,  
        pt.size = 10, margin = 2, digits = NULL, align = NULL, label = NULL)
```

**Arguments**

<code>d</code>	Object to transform to a latex table.
<code>file</code>	Character string with output file name. If missing or "", the output is printed to the screen.
<code>caption</code>	Character vector with title of table.
<code>rownames</code>	Logical, include row names of <code>d</code> .
<code>landscape</code>	Logical, use a landscape format for wider tables.
<code>pt.size</code>	Integer from 10 to 13 for the size of the characters.
<code>margin</code>	Margin between table and page border in cm.
<code>digits</code>	Number of digits to print from numeric values.
<code>align</code>	Character vector with 'l', 'c', 'r' for aligning the columns left, centered or right. Length is either one or 1 (for <code>rownames</code> column) + number of columns in <code>d</code> (even if <code>rownames = FALSE</code> )
<code>label</code>	Character vector with the latex label or HTML anchor.

**Value**

Creates a file at the path given in `file` containing the latex code with the table and document structure for an A4 latex article.

**Author(s)**

Sven E. Templer (<[sven.templer@gmail.com](mailto:sven.templer@gmail.com)>)

**See Also**

[xtable](#)

**Examples**

```
#  
textable(head(trees), rownames=TRUE, digits=4, caption='R "trees" dataset.')
```

```
#
```

# Index

- \*Topic **all**
    - lsall, 9
    - rmall, 12
  - \*Topic **colour**
    - gghcl, 5
  - \*Topic **expression**
    - gregexprind, 7
  - \*Topic **extract**
    - strextr, 14
    - strpart, 15
  - \*Topic **ggplot**
    - gghcl, 5
  - \*Topic **gregexpr**
    - gregexprind, 7
  - \*Topic **grep**
    - mgrepl, 10
  - \*Topic **gsub**
    - mgrepl, 10
  - \*Topic **index**
    - gregexprind, 7
  - \*Topic **inverse**
    - strrev, 16
  - \*Topic **list**
    - lsall, 9
  - \*Topic **matrix**
    - enpaire, 2
    - squarematrix, 13
  - \*Topic **miscellaneous**
    - miscset-package, 2
  - \*Topic **misc**
    - miscset-package, 2
  - \*Topic **multiple**
    - mgrepl, 10
  - \*Topic **number**
    - ntri, 11
  - \*Topic **pairwise**
    - enpaire, 2
  - \*Topic **palette**
    - gghcl, 5
  - \*Topic **part**
    - strpart, 15
  - \*Topic **regular**
    - gregexprind, 7
  - \*Topic **remove**
    - rmall, 12
  - \*Topic **reverse**
    - strrev, 16
  - \*Topic **scale**
    - scale0, 13
  - \*Topic **square**
    - squarematrix, 13
  - \*Topic **string**
    - strextr, 14
    - strpart, 15
  - \*Topic **sub**
    - mgrepl, 10
  - \*Topic **tools**
    - miscset-package, 2
  - \*Topic **triangle**
    - ntri, 11
  - \*Topic **triangular**
    - ntri, 11
  - \*Topic **zero**
    - leading0, 8
- as.data.frame.gapply (gapply), 3  
as.data.table.gapply (gapply), 3
- enpaire, 2
- gapply, 3  
gghcl, 5  
ggplotlist, 6  
gregexpr, 7  
gregexprind, 7  
grep, 11  
grid.arrange, 6
- hcl, 5

leading0, 8  
levels.gapply (gapply), 3  
list.files, 9  
lload, 8  
load, 9  
ls, 10, 12  
lsall, 9  
  
mclapply, 4, 11  
mgrepl, 10  
mgsub (mgrepl), 10  
miscset (miscset-package), 2  
miscset-package, 2  
msub (mgrepl), 10  
  
ntri, 11  
  
regex, 7, 10  
rev, 16  
rm, 12  
rmall, 12  
  
scale0, 13  
setkeyv, 4  
squarematrix, 3, 13  
strextr, 14  
strpart, 15  
strrev, 16  
strsplit, 15  
  
textable, 17  
  
xtable, 17